Module Reference

Document authored by: Aleksander Rebane

Document version: 1.1 (05/2011)

Special thanks to the Beta Test Team, who were invaluable not just in tracking down bugs, but in making this a better product.

Germany
Native Instruments GmbH
Schlesische Str. 28
D-10997 Berlin
Germany
www.native-instruments.de

USA
Native Instruments North America, Inc.
5631 Hollywood Boulevard
Los Angeles, CA 90028
USA
www.native-instruments.com

Table of Contents

1 Panel

Panel Modules provide onscreen controls for various REAKTOR processes. They can be independently set to appear on the A and B Panel Views and they can be arranged differently on those panels. Some Panel Modules are for display purposes only. Those include Lamp Modules, Meter Modules, and Scope Modules for displaying REAKTOR processes as well as graphics and text Modules for ornamentation. The remaining Modules generate or route data for REAKTOR processing. Those include Fader, Knob, Button, Switch, and Menu Modules, and an XY controller (XY Module) which can be used for both display and control.

1.1 Fader/Knob

Fig. 1.1 Knob and Fader Modules

1.1.1 Overview

The Fader and Knob Modules are Panel Elements which let you control the values that are sent from the Module output port in the Structure. The two are essentially the same Module, with the type of control (knob or fader) being chosen in the Module's View page under the Style radio button selector. Custom control styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.
With the fader or knob control in the panel you adjust the value which is output (as Event and Audio) by the corresponding Module in the structure. The signal is Mono unless the output is connected to a Polyphonic input, in which case all Voices receive the same value. The Fader and Knob Modules are discussed in more detail in the Application Reference in subsection 8.2.1. Please refer to chapter 8 as a whole for more information on the Instrument Panel and Modules that have Panel representations.

Although the fader and the knob are essentially the same Module, there are separate menu entries for them in the *Built-In Module > Panel* menu.

Application

The Knob and Fader Modules have the obvious application of providing you controls with which you can tweak your effect or instrument from the Instrument Panel.

1.1.2 Ports

- **(Out)** "Out" is the Event output port for the value that is set from the Module's Panel representation.

1.1.3 Example: Sampler with Trigger Button

The simplest sampler Instrument can be built using the Sampler Module (↑7.1, Sampler) which receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its trigger signal from a Button Module (↑1.2, Button). Its amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Ampl", as shown in the Structure below. Note that you can create a Knob Module (or sometimes a Button or Fader Module) for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry. Make sure that the Button at the "Trig" (trigger) input port has been set to Trigger Mode in its Function page.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into the Sampler Module's Sample Map.

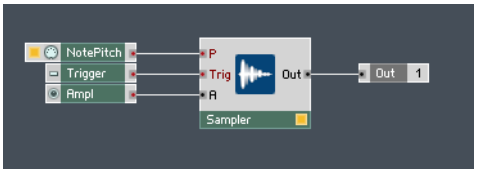


Fig. 1.2 The Structure for a simple sampler Instrument.

1.2 Button

Fig. 1.3 A Button Module

1.2.1 Overview

The Button Module is a Panel element which lets you easily send single Events to your Structure or toggle between two Structure states (also via Events). The Button Module has three different operation modes which can be set in the Function page with the Mode radio button selector: Toggle, Trigger, and Gate.

- **Toggle**: the button has two states, “On” and “Off”. When you turn the button on, an Event with the value specified by the On Value edit field is sent. When you turn the button off, an Event with the value specified by the Off Value edit field is sent. The Toggle Mode is useful for controlling signal flow. A “bypass” button that turns an effect on and off is most likely in Toggle Mode.
- **Trigger**: when you click the button, a trigger signal in the form of an Event is sent. The Event carries the value specified by the On Value edit field. This Mode is useful for triggering samples.
- **Gate**: similar to the Toggle Mode, the button has two states, “On” and “Off”. When you press (and hold) the button, an Event with the value specified by the On Value edit field is sent. The moment that you release the button, an Event with the value specified by the Off Value edit field is sent. With a button in this Mode you could send Gate signals to envelopes.

The signal is Mono unless the output is connected to a Polyphonic input, in which case all Voices receive the same value. The Button Module is discussed in more detail in the Application Reference in subsection 8.2.2. Please refer to chapter 8 as a whole for more information on the Instrument Panel and Modules that have Panel representations. Custom control styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

Application

The Button Module supplies you with a button on your Instrument Panel. Buttons are useful for triggering Events (such as sample playback), toggling between Structure states (such as turning an effect on and off), and for Gate signals (such as triggering notes).

1.2.2 Ports

- **(Out)** "Out" is the Event output port for the trigger, toggle, or gate Event as determined by your actions on the Module's Panel representation.

1.2.3 Example: Sampler with Trigger Button

The simplest sampler Instrument can be built using the Sampler Module (↑7.1, Sampler)which receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its trigger signal from a Button Module (↑1.2, Button). Its amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Ampl", as shown in the Structure below. Note that you can create a Knob Module (or sometimes a Button or Fader Module) for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry. Make sure that the Button at the "Trig" (trigger) input port has been set to Trigger Mode in its Function page.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into the Sampler Module's Sample Map.

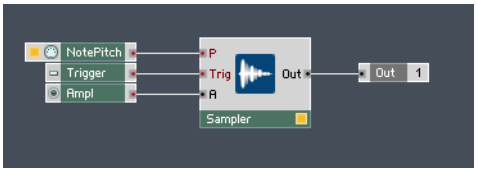


Fig. 1.4 The Structure for a simple sampler Instrument.

1.3 List

Fig. 1.5 List Module

1.3.1 Overview

The List Module lets you construct lists on your Instrument Panel. Selecting an entry from the List Module's Panel representation causes an Event with the corresponding value to be sent from its output port. Which values correspond to which list entries, can be configured in the Module's Function page. The List Module has four different Panel representation styles: Buttons, Menu, Spin, and Text Panel. These styles are shown in that order in the figure above. Please refer to subsection 8.2.4 in the Application Reference for more de-

tailed information on editing the List Module's entries in the Function page, among other Properties. Custom control styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

Please refer to chapter 8 as a whole for more information on the Instrument Panel and Modules that have Panel representations.

Application

The List Module enables you to switch between a number of Structure states via a suitable Panel representation. This can be useful, for example, when you want to have access from the Panel when choosing if a signal is routed through two filter sections in a parallel or in a serial manner. You can combine the List Module with a Selector Module to choose from a number of signals that are to be routed to another part of the Structure. Such an example is shown below. Please refer to the header "Switch vs. Router vs. Selector" in the subsection 8.2.3 in the Application Reference for more detailed information on the differences when applying the List and Selector Module's versus just using a Switch Module.

1.3.2 Ports

- **(Out)** "Out" is the Event output port for the value that corresponds to the list entry that has been selected on the Module's Panel representation.

1.3.3 Properties: View Page

Choosing the List Module's Panel Representation

You can choose between four styles for the List Module's Panel representation. This is done by choosing the desired menu entry from the Style drop-down menu in the Module's View page, as shown in the figure below. The following styles are available:

- **Button**: Each list entry in the Module's Function page creates a button. All buttons will be arrayed vertically in the Instrument Panel. The currently activated button will be displayed in the Indicator color of the Instrument (please refer to section 8.3 in the Application Reference for more information on changing the different colors of the Instrument). The size of the buttons can be adjusted in the Module's View page.
- **Menu**: Each list entry in the Module's Function page creates a new menu entry in this Panel representation which is a drop-down menu.

- **Text Panel**: Each list entry in the Module's Function page creates a new entry in a list which displays multiple entries at the same time. If you have created more entries than are able to fit into the text panel display specified by the Width and Height edit field in the View page, you will get scrollbars in the panel.
- **Spin**: Each list entry in the Module's Function pages creates a new entry in a list. You can switch through the list using a + and a - button to the right of the list entry in the Panel representation.

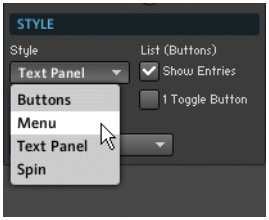
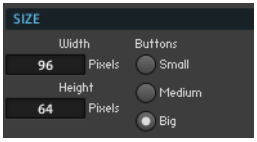


Fig. 1.6 Use the Style drop-down menu to choose the Panel representation of the List Module.

Changing the Size of the List's Panel Representation

Depending on the size of your list, you might need a small or large area for the List Module's Panel representation.

► To change the width and height of the List Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



Please refer to subsection 8.2.4 in the Application Reference for more detailed information on editing the List Module's entries in the View page, among other Properties.

1.3.4 Example: Routing Signals Using a List and Selector Module

The List Module can be used in conjunction with a Selector Module to provide functionality similar to the Switch Module (see screenshot below). The Selector Module "selects" which signal to forward to its output port depending on the value at the "Pos" (position)

input port. For example, a “Pos” value of “0” will cause the Selector Module to forward only the signal at the “0” input port to its output port. Similarly, a “Pos” value of “1” will cause the Selector Module to forward only the signal at the “1” input port to its output port. By connecting a List Module to the “Pos” input port you can use the Panel representation of the List Module to control the routing of the signal from the Instrument Panel.

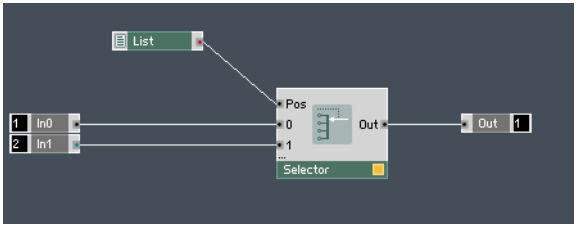


Fig. 1.7 Here you see a common Structure for routing signals. Here, a List Module is used to supply the “position” value the “Pos” input port of a Selector Module.

See below for a Panel representation of the List Module.

1.4 Switch

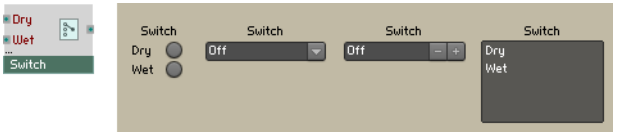


Fig. 1.8 Switch Modules

1.4.1 Overview

The Switch Module is a special type of Panel Element. It does not contain any signal processing of its own but establishes a switchable connection between other Modules. Multiple signals can be connected to the Switch Module's input ports, but which of these signals it forwards, is chosen on the Instrument Panel by its Panel representation. All the other unselected inputs are disconnected. Modules whose outputs are disconnected, through the action of a switch or otherwise, are automatically deactivated to reduce unnecessary load to the CPU. A Module’s status lamp remains dark while the Module is deactivated.

Please refer to subsection 7.4 in the Application Reference for more information on active and inactive Modules.

The Switch Module is a hybrid Module (as can be identified by its green input and output ports). Therefore it can process Event signals as well as Audio signals, depending on the type of wires that are connected to it. The three dots below the input ports indicate that the Switch module supports dynamic input port management. When a wire is dragged to an empty part of the in-port region (the left edge of the Module icon) of a dynamic-input Module while holding down the Ctrl key in Windows (Cmd key in Mac OS X), a new input port is created automatically.
The Switch Module has four different Panel representation styles: Buttons, Menu, Spin, and Text Panel. These styles are shown in that order in the figure above. Please refer to subsection 8.2.3 in the Application Reference for more detailed information on editing the Switch Module's entries in the Function page, among other Properties. Custom control styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

Please refer to chapter 8 as a whole for more information on the Instrument Panel and Modules that have Panel representations.

Application

The Switch Module enables you to switch between a number of Structure states via a suitable Panel representation. What is special about the Switch Module is that it can actually deactivate unused input signal branches. This can be useful when you want to save CPU on unused parts of the Structure. Use the Switch Module when choosing one of several effects that should be used to treat a signal or when choosing which oscillator signal to use as a starting point for synthesis.

Please refer to the header "Switch vs. Router vs. Selector" in the subsection 8.2.3 in the Application Reference for more detailed information on the differences between the different Modules that can be used for routing signals.

1.4.2 Ports

Input Ports

- **(In1)** "In1" is the hybrid input port for the first signal that can be selected on the Panel representation to be forwarded to the output.

- **(...)** "In2, In3, ..." are the dynamic "channel" input ports. Additional "channel" input ports can be created (if all existing "channel" input ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Switch Module. The maximum number of "channel" input ports for the Switch Module is 40.

Output Ports

- **(Out)** "Out" is the hybrid output port for the signal at the input port that corresponds to the chosen list entry in the Module's Panel representation.

1.4.3 Properties: View Page

Choosing the Switch Module's Panel Representation

You can choose between four styles for the Switch Module's Panel representation. This is done by choosing the desired menu entry from the Style drop-down menu in the Module's View page, as shown in the figure below. The following styles are available:

- **Button**: Each a in the Module's Function page creates a button. All buttons will be arrayed vertically in the Instrument Panel. The currently activated button will be displayed in the Indicator color of the Instrument (please refer to section 8.3 in the Application Reference for more information on changing the different colors of the Instrument). The size of the buttons can be adjusted in the Module's View page.
- **Menu**: Each input port of the Switch Module creates a new menu entry in this Panel representation which is a drop-down menu.
- **Text Panel**: Each input port of the Switch Module creates a new entry in a list which displays multiple entries at the same time. If you have created more entries than are able to fit into the text panel display specified by the Width and Height edit field in the View page, you will get scrollbars in the panel.
- **Spin**: Each input port of the Switch Module creates a new entry in a list. You can switch through the list using a + and a - button to the right of the list entry in the Panel representation.

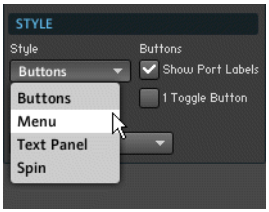
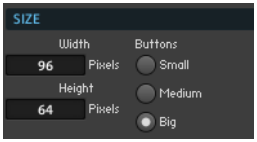


Fig. 1.9 Use the Style drop-down menu to choose the Panel representation of the Switch Module.

Changing the Size of the Switch's Panel Representation

Depending on the number of input ports, you might need a small or large area for the Switch Module's Panel representation.

► To change the width and height of the Switch Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



Please refer to subsection 8.2.3 in the Application Reference for more detailed information on editing the Switch Module's entries in the View page, among other Properties.

1.4.4 Example: Routing Signals Using a Switch

In the following example you are shown how to use the Switch Module to choose which signals are forwarded to an Output Terminal. Let's say we have a Macro that is applied to a signal as an effect. We call the processed signal the “wet” signal and the unprocessed signal the “dry” signal. See the screenshot below for an exemplary structure.

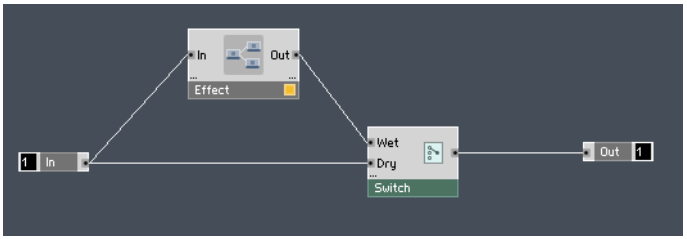


Fig. 1.10 Here a Switch Module is used to forward either the “wet” signal or the “dry” signal to the Output Terminal.

If we go to the Instrument Panel, the Switch can be set to have a Panel representation as shown in the figure below. The Switch Module lets you choose between the different signals arriving at its input ports. Here the input ports have been labeled “Wet” and “Dry” and these names also appear on the Instrument Panel. If you click on the “Wet” button, then the signal arriving at the “Wet” input port of the Switch Module is routed to its output port. If you press the “Dry” button, then the signal arriving at the “Dry” input port of the Switch Module is routed to its output port.

Fig. 1.11 A possible Panel representation of the Switch Module.

An important case is when you have Event signals connected to the input ports of the Switch Module. “Switching” to one of these signals causes the value that was sent last to that particular input port to be sent from the output port of the Switch Module in the form of an Event.

1.5 Lamp

Fig. 1.12 Lamp Modules

1.5.1 Overview

The Lamp Module is an indicator lamp for a Monophonic signal. It has two modes of operation which can be set in the Module's Function page:

- **Discrete**: In Discrete mode the lamp in the Instrument Panel lights up as long as the input signal (sampled at 25 Hz) is within the range set with the Min and Max edit fields in the Function page. That is, the lamp is lit up when the signal's value is larger than "Min" and less than or equal to "Max".
- **Continuous**: In Continuous mode, the lamp color will gradually fade in and out for values between "Min" and "Max". For values above "Max" the lamp is lit up and for values below "Min" the lamp is unlit.

Application

Lamp Modules are good for monitoring signals from the Instrument Panel. Lamps are especially useful when you only need a rough idea of the signal level and therefore only require a simple display. Another common use for the Lamp Module is to activate Modules within a Structure (such as the Audio Table Module in the example in section 11.2 in the Application Reference). This is done by engaging the Always Active checkbox in the Lamp Module's Function page and connecting the Lamp Module to an output port of the Module you wish to activate.

Although the Lamp Module only accepts Monophonic signals, connecting a Polyphonic signal to its input port will still keep the Module, from which the Polyphonic signal stems, active.

1.5.2 Ports

- **(In)** "In" is the input port for the Monophonic signal to be displayed by the lamp display in the Instrument Panel.

1.5.3 Properties: Function Page

Keeping the Lamp Module Always Active

A common use for the Lamp module is to activate other Modules within a Structure. In such a case you would connect the output port of another Module to the Lamp Module which then would need to have the Always Active feature active.

► To keep the Lamp Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Changing between Discrete and Continuous Mode

The Lamp Module has two operation modes: Discrete mode and Continuous mode. They are described in the Module Overview subsection above.

► To activate Continuous mode, go to the Lamp Module's Function page and engage the Continuous Mode checkbox. Otherwise, with the Continuous Mode checkbox disengaged, the Lamp Module operates in Discrete mode.

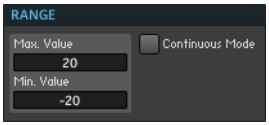


Fig. 1.13 The Lamp Module's Function page holds the edit fields to set the "Max" and "Min" values (in dB) as well as the checkbox to switch between Discrete and Continuous operation mode.

Setting the "Min" and "Max" Values

The general functionality of the Lamp Module is described in the Module Overview subsection above. Both operation modes (Discrete and Continuous) behave according to the "Min" and "Max" values.

► To set the "Min" and "Max" values (in dB), go to the Lamp Module's Function page and enter the desired values into the Min. Value and Max. Value edit fields, respectively.

1.5.4 Properties: View page

Choosing the Lamp Color

The color of the lamp can be chosen in the View page of the Module Properties. You can either choose from preset colors with the (red, green, blue, yellow or Indicator color), or you can define custom colors for the On and Off positions of the lamp using the color picker of your OS.

► To choose a preset color combination for the lamp, click on the desired color in the Default Color ratio button selector, shown in the figure below.

Please refer to section 8.3 in the Application Reference for more information on changing the Indicator color of your Instrument.

► To set the On and Off states of your lamp to custom colors, press the Set On Color and Set Off Color buttons, respectively. The color picker dialog window of your OS will let you pick a desired color. The Set On Color and Set Off Color buttons are shown in the figure below.

Turning Off the Lamp Frame

You might want to place several Lamps next to each other to create a pixilated display of some sort. In that case the nice frame, with which the Lamp Module comes, becomes superfluous. With the frame removed, you are able to place the lamps pixel accurate side by side in the Instrument Panel without any gaps between them.

► To turn off the Lamp Module's frame on the Instrument Panel, disengage the Has Frame checkbox in the View page (shown in the screenshot below).

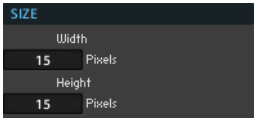
Setting Label and Lamp Size

Depending on your Instrument Panel layout, you might want a larger or smaller lamp with a larger or smaller Label and Value display.

For other visibility settings for the Panel representation (Label, Picture, and Value, please refer to subsection 8.2.1 in the Application Reference).

► To change the size of your Lamp Module's Label and Value display, choose the desired size from the Label/Value radio button selector in the View page (shown in the screenshot above).
► To choose between preset sizes for the Lamp Module's Panel representation, choose the desired size from the Set Default Size radio button selector, shown in the screenshot below.

► To exactly set the width and height of the Lamp Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



Custom lamp styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

1.5.5 Example: LFO Monitor

This example shows how to use the Lamp Module to monitor an LFO Module's (↑9.1, LFO) signal level from the Instrument Panel. The simple Structure for this is shown in the figure below.

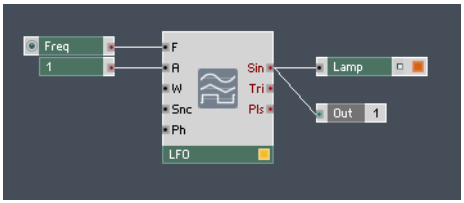


Fig. 1.14 You can use the Lamp Module to monitor the signal level of an LFO from the Instrument Panel.

The Lamp Module's Panel representation has color corresponding to the "off" state and a color corresponding to the "on" state. The values that yield these states are set in the Module's Function page with the Max. Value and Min. Value edit fields. Since the LFO in the example outputs a waveform in the range [-1 … 1], the range of the Lamp Module has been set accordingly. Also, the Continuous Mode checkbox has been enables for the Lamp Module to represent values between "-1" and "1" in a gradient that goes from the "off" color to the "on" color.

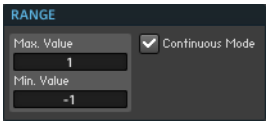


Fig. 1.15 To be able to display a smooth transition from the "off" color to the "on" color of the Lamp Module, engage the Continuous Mode checkbox.

1.6 Level Lamp

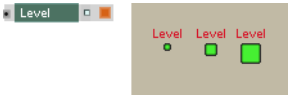


Fig. 1.16 Level Lamp Modules

1.6.1 Overview

The Level Lamp Module is an indicator lamp for a Monophonic signal with logarithmic settings. It has two modes of operation which can be set in the Module's Function page:

- **Discrete**: In Discrete mode the lamp in the Instrument Panel lights up as long as the input signal (sampled at 25 Hz) is within the range set (in dB) with the Min and Max edit fields in the Function page. That is, the lamp is lit up when the signal's value is larger than "Min" and less than or equal to "Max".
- **Continuous**: In Continuous mode, the lamp color will gradually fade in and out for values between "Min" and "Max". For values above "Max" the lamp is lit up and for values below "Min" the lamp is unlit.

Application

Level Lamp Modules are good for monitoring signals from the Instrument Panel. Level Lamps are especially useful when you only need a rough idea of the signal level and therefore only require a simple display. Another common use for the Level Lamp Module is to activate Modules within a Structure (such as the Audio Table Module in the example in section 11.2 in the Application Reference). This is done by engaging the Always Active checkbox in the Level Lamp Module's Function page and connecting the Level Lamp Module to an output port of the Module you wish to activate.

Although the Level Lamp Module only accepts Monophonic signals, connecting a Polyphonic signal to its input port will still keep the Module, from which the Polyphonic signal stems, active.

1.6.2 Ports

- **(In)** "In" is the input port for the Monophonic signal to be displayed by the lamp display in the Instrument Panel.

1.6.3 Properties: Function Page

Keeping the Level Lamp Module Always Active

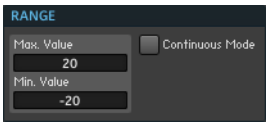
A common use for the Level Lamp module is to activate other Modules within a Structure. In such a case you would connect the output port of another Module to the Level Lamp Module which then would need to have the Always Active feature active.

► To keep the Level Lamp Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Changing between Discrete and Continuous Mode

The Level Lamp Module has two operation modes: Discrete mode and Continuous mode. They are described in the Module Overview subsection above.

► To activate Continuous mode, go to the Level Lamp Module's Function page and engage the Continuous Mode checkbox. Otherwise, with the Continuous Mode checkbox disengaged, the Level Lamp Module operates in Discrete mode.



Setting the "Min" and "Max" Values

The general functionality of the Level Lamp Module is described in the Module Overview subsection above. Both operation modes (Discrete and Continuous) behave according to the "Min" and "Max" values.

► To set the "Min" and "Max" values (in dB), go to the Level Lamp Module's Function page and enter the desired values into the Min. Value and Max. Value edit fields, respectively.

1.6.4 Properties: View page

Choosing the Level Lamp Color

The color of the lamp can be chosen in the View page of the Module Properties. You can either choose from preset colors with the (red, green, blue, yellow or Indicator color), or you can define custom colors for the On and Off positions of the lamp using the color picker of your OS.

► To choose a preset color combination for the lamp, click on the desired color in the Default Color ratio button selector, shown in the figure below.

Please refer to section 8.3 in the Application Reference for more information on changing the Indicator color of your Instrument.

► To set the On and Off states of your lamp to custom colors, press the Set On Color and Set Off Color buttons, respectively. The color picker dialog window of your OS will let you pick a desired color. The Set On Color and Set Off Color buttons are shown in the figure below.

Turning Off the Level Lamp Frame

You might want to place several Level Lamps next to each other to create a pixilated display of some sort. In that case the nice frame, with which the Level Lamp Module comes, becomes superfluous. With the frame removed, you are able to place the lamps pixel accurate side by side in the Instrument Panel without any gaps between them.

► To turn off the Level Lamp Module's frame on the Instrument Panel, disengage the Has Frame checkbox in the View page (shown in the screenshot below).

Setting Label and Level Lamp Size

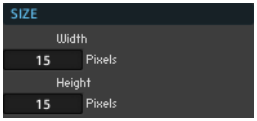
Depending on your Instrument Panel layout, you might want a larger or smaller lamp with a larger or smaller Label and Value display.

For other visibility settings for the Panel representation (Label, Picture, and Value, please refer to subsection 8.2.1 in the Application Reference).

► To change the size of your Level Lamp Module's Label and Value display, choose the desired size from the Label/Value radio button selector in the View page (shown in the screenshot above).

► To choose between preset sizes for the Level Lamp Module's Panel representation, choose the desired size from the Set Default Size radio button selector, shown in the screenshot below.

► To exactly set the width and height of the Level Lamp Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



Custom lamp styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

1.6.5 Example: Activating a Table Module

The Level Lamp Module can also be applied for its "Always Active" feature. In this example, it is connected to the output of an Event Table Module to activate it for whatever purpose you wish to use it for. For this application the Always Active checkbox in the Level Lamp Module's Function page has to be activated. However, it does not matter if the incoming signal is Polyphonic and therefore the Level Lamp Module's input port gets muted — the Modules lying upstream will be activated nevertheless.

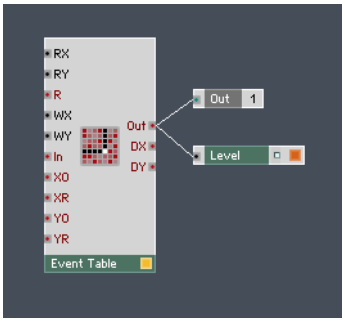


Fig. 1.17 Use the Level Lamp Module to activate Modules upstream.

1.7 RGB Lamp

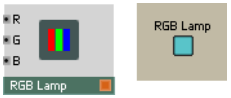


Fig. 1.18 An RGB Lamp Module

1.7.1 Overview

The RGB Lamp Module is a resizable, colored display. Its color is controlled by the values appearing at its three color input ports: "R" (red), "G" (green), and "B" (blue). The Module's horizontal and vertical size can be set in pixels in the View page.

Application

With the RGB Lamp Module you can create custom displays and controls that change their color depending on what is going on in the Instrument's Structure.

1.7.2 Ports

Input Ports

- **(R)** "R" (red) is the Audio input port for the intensity of the red component. The range of values at this input port is [0 … 1]. When disconnected, the default value, "0", is used.

- **(G)** "G" (green) is the Audio input port for the intensity of the green component. The range of values at this input port is [0 … 1]. When disconnected, the default value, "0", is used.
- **(B)** "B" (blue) is the Audio input port for the intensity of the blue component. The range of values at this input port is [0 … 1]. When disconnected, the default value, "0", is used.

1.7.3 Properties: Function Page

Keeping the RGB Lamp Module Always Active

Before the RGB Lamp Module shows the colors that have been specified by its input port, you need to activate it.

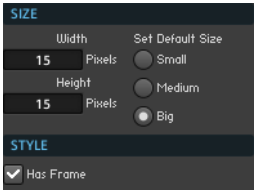
► To activate the RGB Lamp Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

1.7.4 Properties: View page

Turning Off the RGB Lamp Frame

You might want to place several RGB Lamps next to each other to create a pixilated display of some sort. In that case the nice frame, with which the RGB Lamp Module comes, becomes superfluous. With the frame removed, you are able to place the lamps pixel accurate side by side in the Instrument Panel without any gaps between them.

► To turn off the RGB Lamp Module's frame on the Instrument Panel, disengage the Has Frame checkbox in the View page (shown in the screenshot below).



Setting the RGB Lamp Size

Depending on your Instrument Panel layout, you might want a larger or smaller lamp.

For other visibility settings for the Panel representation including the Labe and Label size, please refer to subsection 8.2.1 in the Application Reference).

► To choose between preset sizes for the RGB Lamp Module's Panel representation, choose the desired size from the Set Default Size radio button selector, shown in the screenshot above.

► To exactly set the width and height of the RGB Lamp Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure above.

1.7.5 Example: Setting RGB Color from the Panel

This example shows how you can control the color of the RGB Lamp Module's Panel representation with three knobs. To create the Structure shown below, just right-click on each input port and choose the *Create Control* menu entry.

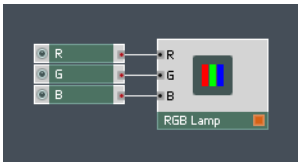


Fig. 1.19 Use the Create Control menu entry to quickly create the controls for the red, green, and blue values.

Note that you can use the RGB Lamp Module as a background for another Panel element, for example. In the figure shown below, the RGB Lamp Module's size has been changed.

Fig. 1.20 The Panel representation of the RGB Lamp Module.

1.8 Meter

Fig. 1.21 A Meter Module

1.8.1 Overview

The Meter Module is a value indicator for a Monophonic signal. The value of the incoming signal is sampled (at 25 Hz) and displayed on a linear scale in the Instrument Panel. The displayed range is set in the Function page of the Module's Properties. The color and size of the display can be set in the Module's View page.

The Meter Module is the same as the Numeric Readout Module. The only difference is that the Numeric Readout Module's default setting is to only show the value field in its Panel representation without the animated picture of the meter.

Application

The Meter Module is good for monitoring signals from the Instrument Panel. A common application is to monitor envelope levels or the output signal of an LFO. Of course, if you decide to modulate a parameter with both an LFO and an envelope then it's even more useful to monitor the resulting signal, since that might be (a little bit) more unpredictable than the regular oscillations of an LFO,
for example.

1.8.2 Ports

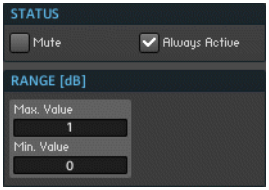
- **(In)** "In" is the input port for the signal whose value is to be displayed on the meter,

1.8.3 Properties: Function Page

Keeping the Meter Module Always Active

Before the Meter Module shows the value that has been specified by its input port, you need to activate it.

► To activate the Meter Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Setting the Range for Displayed Values

► To set the range for the values that correspond to a minimum and maximum display state of the Meter Module, go to the Module's Function page and enter the corresponding values into the Min. Value and Max. Value edit fields (shown in the screenshot above).

1.8.4 Properties: View page

Choosing the Meter Color

The color of the meter can be chosen in the View page of the Module Properties. You can either choose from preset colors with the (red, green, blue, yellow or Indicator color), or you can define custom colors for the On and Off positions of the meter using the color picker of your OS.

► To choose a preset color combination for the meter, click on the desired color in the Default Color ratio button selector, shown in the figure below.

Please refer to section 8.3 in the Application Reference for more information on changing the Indicator color of your Instrument.

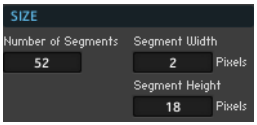
► To set the On and Off states of your meter to custom colors, press the Set On Color and Set Off Color buttons, respectively. The color picker dialog window of your OS will let you pick a desired color. The Set On Color and Set Off Color buttons are shown in the figure below.

Setting Meter and Label Size

Depending on your liking you might want a meter with a different number of segments, of different size, or with a larger or smaller Label and Value display.

For other visibility settings for the Panel representation (Label, Picture, and Value, please refer to subsection 8.2.1 in the Application Reference).

► The size of the Meter Module's Panel representation depends on the number and size of the segments it comprises. To change the width and height of the segments, enter the desired size into the Segment Width and Segment Height edit fields in the Meter Module's View page (shown in the screenshot below).
► To change the number of segments in the meter, enter the desired number into the Number of Segments edit field (shown in the screenshot below).



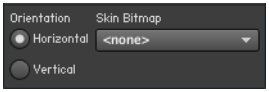
► To quickly change the size of the Meter Module's Panel representation, you can use one of the three default sizes.
► To choose a default size for the Meter Module, choose the desired size from the Set Default Size radio button selector, shown in the screenshot below.

Use the Set Default Size radio button selector in the Meter Module's View page to quickly change the size of your Meter Module's Panel representation. Use the Label/Value radio button selector to change the size of the Label and Value in the Module's Panel representation.
► To change the size of your Meter Module's Label and Value display, choose the desired size from the Label/Value radio button selector in the View page (shown in the screenshot above).

Changing the Meter Orientation

The Meter Module's Panel representation can sit either vertically or horizontally in the Instrument Panel.

► To change the Meter Module's orientation, click on the appropriate radio button in the Orientation radio button selector.



Custom meter styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

1.8.5 Example: LFO Monitor

This example shows how to use the Meter Module to monitor an LFO Module's (↑9.1, LFO) signal level from the Instrument Panel. The simple Structure for this is shown in the figure below.

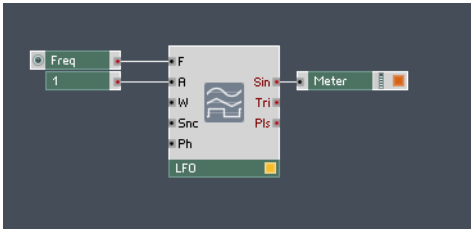


Fig. 1.22 You can use the Meter Module to monitor the signal level of an LFO from the Instrument Panel.

The Meter Module's Panel representation consists of a row of segments of which an increasing number is lit up, depending where the incoming value lies in the range [Min … Max]. This range is set in the Module's Function page with the Max. Value and Min. Value edit fields. Since the LFO in the example outputs a waveform in the range [-1 … 1], the range of the Meter Module for this example has been set accordingly, as shown in the figure below. The resulting Panel representation is shown in the second figure below. Note that since the LFO signal is dynamic (for frequency values other than 0 Hz), the meter in the screenshot is actually moving up and down.

Fig. 1.23 The incoming range of values lies in the range [-1 … 1] so the range for the Meter Module has been set accordingly.

Fig. 1.24 The Panel representation of the Meter Module.

1.9 Level Meter

Fig. 1.25 Level Meter Module

1.9.1 Overview

The Level Meter Module is a value indicator for a Monophonic signal with logarithmic settings. The value of the incoming signal is sampled (at 25 Hz) and displayed on a linear scale in the Instrument Panel. The displayed range is set in the Function page of the Module's Properties. The color and size of the display can be set in the Module's View page.

Application

The Level Meter Module is good for monitoring signals from the Instrument Panel. A common application is to monitor envelope levels or the output signal of an LFO. Of course, if you decide to modulate a parameter with both an LFO and an envelope then it's even more useful to monitor the resulting signal, since that might be (a little bit) more unpredictable than the regular oscillations of an LFO,
for example.

1.9.2 Ports

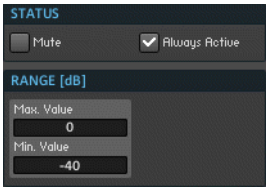
- **(In)** "In" is the input port for the signal whose value is to be displayed on the meter,

1.9.3 Properties: Function Page

Keeping the Level Meter Module Always Active

Before the Level Meter Module shows the value that has been specified by its input port, you need to activate it.

► To activate the Level Meter Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Setting the Range for Displayed Values

► To set the range for the values (in dB) that correspond to a minimum and maximum display state of the Level Meter Module, go to the Module's Function page and enter the corresponding values (in dB) into the Min. Value and Max. Value edit fields (shown in the screenshot above).

1.9.4 Properties: View page

Choosing the Level Meter Color

The color of the meter can be chosen in the View page of the Module Properties. You can either choose from preset colors with the (red, green, blue, yellow or Indicator color), or you can define custom colors for the On and Off positions of the Meter using the color picker of your OS.

► To choose a preset color combination for the meter, click on the desired color in the Default Color ratio button selector, shown in the figure below.

Please refer to section 8.3 in the Application Reference for more information on changing the Indicator color of your Instrument.

► To set the On and Off states of your meter to custom colors, press the Set On Color and Set Off Color buttons, respectively. The color picker dialog window of your OS will let you pick a desired color. The Set On Color and Set Off Color buttons are shown in the figure below.

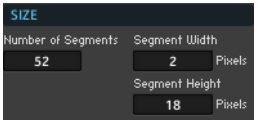
Setting Level Meter and Label Size

Depending on your liking you might want a meter with a different number of segments, of different size, or with a larger or smaller Label and Value display.

For other visibility settings for the Panel representation (Label, Picture, and Value, please refer to subsection 8.2.1 in the Application Reference).

► The size of the Level Meter Module's Panel representation depends on the number and size of the segments it comprises. To change the width and height of the segments, enter the desired size into the Segment Width and Segment Height edit fields in the Level Meter Module's View page (shown in the screenshot below).

► To change the number of segments in the meter, enter the desired number into the Number of Segments edit field (shown in the screenshot below).



► To quickly change the size of the Level Meter Module's Panel representation, you can use one of the three default sizes.

► To choose a default size for the Level Meter Module, choose the desired size from the Set Default Size radio button selector, shown in the screenshot below.

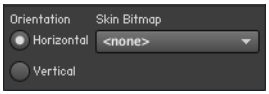
Use the Set Default Size radio button selector in the Level Meter Module's View page to quickly change the size of your Level Meter Module's Panel representation. Use the Label/Value radio button selector to change the size of the Label and Value in the Module's Panel representation.

► To change the size of your Level Meter Module's Label and Value display, choose the desired size from the Label/Value radio button selector in the View page (shown in the screenshot above).

Changing the Level Meter Orientation

The Level Meter Module's Panel representation can sit either vertically or horizontally in the Instrument Panel.

► To change the Level Meter Module's orientation, click on the appropriate radio button in the Orientation radio button selector.



Custom meter styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

1.9.5 Example: Signal Level Display

The Level Meter Module is good for using the meter Panel representation to display incoming signal levels in the logarithmic scale. This means that although values incoming to the Level Meter Module are in the range [0 … 1], for example, the display shows the level equivalent in the logarithmic scale. For example, an envelope follower Structure (see example for the Peak Detector Module, see also ↑12.14, Peak Detector) outputs an envelope level in the range [0 … 1]. The Level Meter Module then enables you to see from the Instrument Panel what control signal the envelope follower has generated from the initial incoming signal.

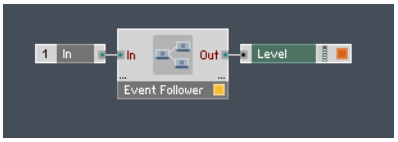


Fig. 1.26 The Level Meter Module displays the signal level from the "Event Follower" Macro, which is in the range [0 … 1], in the logarithmic scale.

1.10 Picture

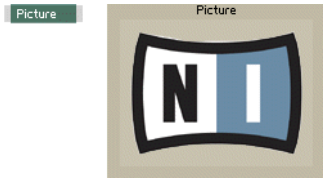


Fig. 1.27 A Picture Module

1.10.1 Overview

The Picture Module enables you to display a picture on the Instrument Panel. Pictures can be loaded from 24-bit Bitmap (*.bmp) and 32-bit uncompressed Targa (*.tga) images. The advantage of the Targa file format is that it supports an alpha channel, which can be used as a mask for the visible portion of the graphic — the unmasked portion will then be transparent. You can load pictures using the Select Picture drop-down menu in the View page of any object that can display pictures. Opening a picture automatically brings up the Picture Properties window where you specify all relevant picture settings. By default, the dis-

play size will be set to the size of the image. It is possible to modify the display size in pixels in the Module's View page and the visibility of a frame can be turned on or off. The image itself cannot be resized inside REAKTOR; for this you need to use external picture editing software. All loaded pictures are automatically shared and available to all Modules that use pictures. The Picture Module has no input or output ports.

Application

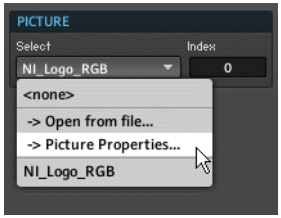
The Picture Module is usually applied for decoration for your Instrument Panel. As with the Multi Picture Module (↑1.11, Multi Picture), you can place a Mouse Area Module (↑1.18, Mouse Area) over the Picture Module to add graphics to the Mouse Area interaction field.

1.10.2 Properties: Function Page

Loading a Picture

Before the Picture Module displays a picture in the Instrument Panel, you need to load the picture file that the Module should use.

1. To load a picture file for the Picture Module, go to its Function page and open the Select Picture drop-down menu (shown in the figure below).
2. Now either choose the *-> Open from file...* menu entry to load a new picture file or choose a menu entry corresponding to a picture file that has already been loaded into the Ensemble by another Module.



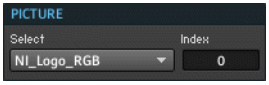
3. This opens up the Picture Properties dialog window. This window shows the picture preview. In the case of the Picture Module it usually suffices to just press the OK button to load the picture and continue with your work. Please refer to section 8.5 in the Application Reference for more information on working with the Picture Properties dialog window.

These instructions should be considered as the example for the Picture Module.

Choosing the Picture Index

If the picture you have loaded consists of a number of animation frames which you have also configured with the Picture Properties dialog window (please refer to subsection 8.5.3 in the Application Reference for more information on this), you may choose which animation frame is displayed in the Instrument Panel.

► To choose which animation frame is displayed in the Instrument Panel, go to the Module's Function page and select the index of the animation frame with the Index edit field. This edit field is shown in the figure below.



1.10.3 Properties: View Page

Turning on the Picture Frame

► To turn on the frame for the Picture Module's Panel representation, engage the Has Frame checkbox in the View page (shown in the screenshot below).

Setting the Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for your picture. This can be useful when you want to tile certain parts of your picture when the display size exceeds the size of the actual picture used.

Please refer to subsection 8.5.4 in the Application Reference for more information on resizability.

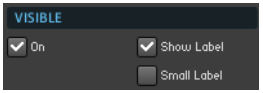
► To set the width and height of the Picture Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure above.

Choosing Label Size and Visibility Settings

► To reduce the size of your Picture Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

► To make the Picture Module's Panel representation disappear, disengage the On checkbox (shown in the figure below).



► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Picture Module's display disappear.

1.11 Multi Picture

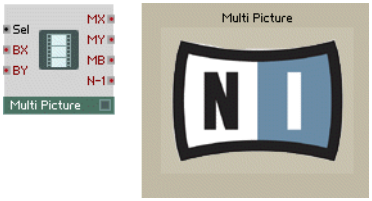


Fig. 1.28 Multi Picture Module

1.11.1 Overview

The Multi Picture Module is a user definable display and a 2-dimensional controller (like the XY module, see also ↑1.14, XY) that reports the mouse position and mouse button status at its output ports. It supports multi-frame animation so that the animation frame selected by the "Sel" (select) input port can be displayed by the Multi Picture Module's Panel representation. For this the loaded picture has to be set up in the Picture Properties window by indicating the number of animation frames and their orientation (horizontal or vertical) within the loaded picture file.

You can load pictures using the Select Picture drop-down menu in the View page of any object that can display pictures. Opening a picture automatically brings up the Picture Properties window where you specify all relevant picture settings. By default, the display size will be set to the size of the image. It is possible to modify the display size in pixels in the Module's View page and the visibility of a frame can be turned on or off. The image itself cannot be resized inside REAKTOR; for this you need to use external picture editing software. All loaded pictures are automatically shared and available to all other Modules that use pictures as well.
Pictures can be loaded from 24-bit Bitmap (*.bmp) and 32-bit uncompressed Targa (*.tga) images. The advantage of the Targa file format is that it supports an alpha channel, which can be used as a mask for the visible portion of the graphic — the unmasked portion will then be transparent. This is useful for round knobs on a square background, for example.
The Multi Picture Module has two operation modes:

- If the Incremental Mouse Mode is turned off, the absolute mouse position on the Module's Panel representation is translated to values between the "Min" and "Max" values set in the Function page. The display width and height define the range and resolution for mouse movements and the Mouse Reso edit field in the Function page has no effect.
- With the Incremental Mouse Mode turned on, the "MX" (mouse X) and "MY" (mouse Y) values are controlled by the relative movement of the mouse compared to the point where the left the mouse button was pressed. The X and Y mouse resolution is given by the Mouse Reso edit field in the Function page.

Application

The Multi Picture Module can be used to add animations to your Instrument Panel. These animations can be coupled to Modules such as the Mouse Area Module (↑1.18, Mouse Area) to create custom controls.

1.11.2 Ports

Input Ports

- **(Sel)** "Sel" (select) is the Audio input port for selecting the animation index (animation frame) by number. The range of values at this input port is [0 ... "N-1"] where "N" is the number of animation frames. The default value that is used in the case that this input port is disconnected is "0". The "Sel" input port is sampled at the display rate.

- **(BX)** "BX" (base X) is the Event input port to set a new base value from which the incremental increase in the "MX" (mouse X) value is calculated. The "BX" (base X) value is only used if Incremental Mode has been activated.
- **(BY)** "BY" (base Y) is the Event input port to set a new base value from which the incremental increase in the "MY" (mouse Y) value is calculated. The "BY" (base Y) value is only used if Incremental Mode has been activated.

Output Ports

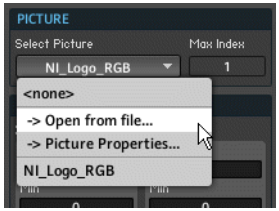
- **(MX)** "MX" (mouse X) is the Event output port for the horizontal (X) mouse position when the mouse is within the Multi Picture Module's display area in the Instrument Panel. Events with the current mouse position are only sent when the mouse button is depressed.
- **(MY)** "MY" (mouse Y) is the Event output port for the vertical (Y) mouse position when the mouse is within the Multi Picture Module's display area in the Instrument Panel. Events with the current mouse position are only sent when the mouse button is depressed.
- **(MB)** "MB" (mouse button) is the Event output port for the left mouse button status. It is "1" when the mouse button is depressed; otherwise it is "0". Events are sent from this output port only when the mouse is within the Multi Picture Module's display area in the Instrument Panel.
- **(N - 1)** "N-1" is the Event output port for the value "N - 1" where "N" denotes the number of animation frames you have entered in the Picture Properties dialog window.

1.11.3 Properties: Function Page

Loading a Picture

Before the Multi Picture Module displays a picture in the Instrument Panel, you need to load the picture file that the Module should use.

1. To load a picture file for the Multi Picture Module, go to its Function page and open the Select Picture drop-down menu (shown in the figure below). The Max Index display shows the value "N-1" where "N" denotes the number of animation frames in your loaded picture.



2. Now either choose the *-> Open from file...* menu entry to load a new picture file or choose a menu entry corresponding to a picture file that has already been loaded into the Ensemble by another Module.
3. This opens up the Picture Properties dialog window. This window shows the picture preview and lets you specify how REAKTOR should extract the animation frames from the loaded picture. Please refer to section 8.5 in the Application Reference for more information on working with the Picture Properties dialog window. Press the OK button when you are finished setting the picture properties.

Keeping the Multi Picture Module Always Active

If the Multi Picture Module is not connected to an active signal flow, then you are unable to switch the displayed pictures via the "Sel" (select) input port. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Multi Picture Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Activating Incremental Mouse Mode

The output values for the Multi Picture Module are calculated in two different ways, depending on the operation mode. These two modes are discussed in the Module's Overview, above.

► To toggle between the two Multi Picture operation modes, engage or disengage the Incremental Mouse Mode checkbox accordingly. This checkbox is shown in the screenshot of the Function page below.

Setting the Range and Resolution for the Output Values

The range and resolution of the Multi Picture Module's values at the "MX" (mouse X) and "MY" (mouse Y) output ports are set in the Output Range area of the Module's Function page, shown in the screenshot below. The edit fields contained in this area function analogously to the ones contained in a conventional Knob or Fader Module. To learn how these edit fields affect the output values, please refer to subsection 8.2.1 in the Application Reference.

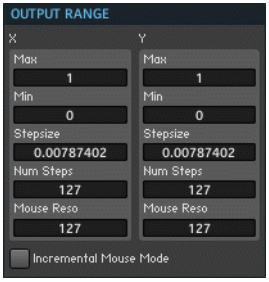


Fig. 1.29 The range and resolution of the Multi Picture Module's output values are set in the Output Range area of its Function page.

1.11.4 Properties: View Page

Turning on the Picture Frame

► To turn on the frame for the Multi Picture Module's Panel representation, engage the Has Frame checkbox in the View page (shown in the screenshot below).

Setting the Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for your picture. This can be useful when you want to tile certain parts of your picture when the display size exceeds the size of the actual picture used.

Please refer to subsection 8.5.4 in the Application Reference for more information on resizability.

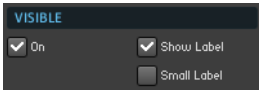
► To set the width and height of the Picture Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure above.

Choosing Label Size and Visibility Settings

► To reduce the size of your Multi Picture Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

► To make the Picture Module's Panel representation disappear, disengage the On checkbox (shown in the figure below).



► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Multi Picture Module's display disappear.

1.11.5 Example: Value Edit Field

This example shows how to build a custom value edit field that displays numbers with one decimal point precision. The edit field also enables you to change the values by clicking and dragging the mouse over the edit field and to reset the value to "0" by double clicking on the field. Start by creating a Macro with an output port labeled "Out". Make sure the Macro is set to Monophonic Mode before you start building this example Structure. First we will start with building the value display. We want to display decimal numbers from

-9.9 to 9.9, in 0.1 step increments. For this purpose you will need three Multi Picture Modules (↑1.11, Multi Picture): one for the sign (+ and -), one for the integers (0 to 9), and one for the first decimal place after zero (.0 to .9).
After inserting the three Multi Picture Modules into the Structure, designate one for the sign and go to its Function page. There, in the Select Picture menu you will find the *-> Open from file...* menu entry, click on it. This opens the Load Image File dialog box. Go to the *Tutorial Ensembles* folder and from there enter the folder labeled *Module Reference* where you will find the file "plus_minus.tga". Select the file and click the Open button. In the following Picture Properties dialog window, make sure that the Alpha Channel checkbox is activated. Next, make sure that the Animation Width and Animation Height edit fields both carry the value "32". You should now see a "+" and a "-" with transparent backgrounds next to each other in the Picture Preview display. If so, click the OK button to continue.

Fig. 1.30 The Picture Properties dialog window.

Now follow this same procedure for loading the digit animations to the other the Multi Picture Modules. Designate one of the Modules for integer digits and load the "numbers.tga" file from the same location as "plus_minus.tga" into the Multi Picture Module whilst making sure that the same settings for the Alpha Channel checkbox and Animation Height and Animation Width edit fields apply. For the third Multi Picture Module, repeat the aforementioned procedure, but this time loading the file "decimals.tga" into the Module. This last Module will be dedicated to display the first position after the decimal point. Go to the Instrument Panel and arrange the signs and digits in a way that makes sense when reading from left to right: first the sign, then the integer, and then the decimal fraction.

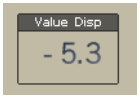


Fig. 1.31 The display part of the value edit field.

Now we need a way to split up the information that the value display should display into three parts: the sign (if the value is positive or negative), the integer part, and the decimal fraction. The first split will happen between the sign and the numbers. For that, insert a Rectify/Sign Module into the structure. The incoming value to the value display goes into the "In" input port of the Rectify/Sign Module. The "Sign" output port of that Module outputs a "1" if the incoming value is positive and "-1" if the incoming value is negative. We need to convert these values to specify the picture index of the animation loaded into the Multi Picuture Module. For that purpose, add an Add Module (↑4.2, Add, +) into the structure and use it to add "1" to the signal from the "Sign" output port of the Rectify/Sign Modue. The output of the Add Module will then be "0" if the value at the input of the Rectify/Sign Module is negative and "2" if the value is positive. After connecting the output port of the Add Module to the "Sel" input port of the Multi Picture Module, the "0" will address the first picture in the animation, the minus sign, and the value "2" will address the last picture of the animation, the plus sign. To test out the functionality of the value display, create a Knob Module at the "In" input of the Rectify/Sign Module by right-clicking on the "In" input port and clicking on the *Create Control* menu item. Set the range of the knob to [-9.9 … 9.9] by going to the Module's Function page and setting the Max Value edit field to "9.9" and the Min Value edit field to "-9.9".
On to the numbers: insert two Modulo Modules (↑4.9, Modulo) into the structure. Connect the "|x|" output port of the Rectify/Sign Module the "A" input port of one Modulo Module. Since we want to split the number into integers and fractions, it is only natural to create a

Constant with the value "1" at the "B" input port of the first Modulo Module. Now the "Div" output port of the Modulo Module carries the value that is the largest multiple of "1" and is at the same time smaller than the value received from the "|x|" output port of the Rectify/Sign Module. For example, if |x| = 5.2, then Div = 5 or if |x| = 3.8, then Div = 3. Connect the "Div" output port to the "Sel" input port of the Multi Picture Module that is meant to display the integer part of the incoming signal value. Since we want to display signals within the range -9.9 to 9.9, then the "Div" value has a range [0 ... 9] which exactly matches the indices of the pictures of the animation file "numbers.tga" loaded into that Multi Picture Module. Go to the Instrument Panel and move the knob that you previously created and watch the sign and the integer numbers change in the Panel display of the Multi Picture Modules. Next, hook up the decimal fraction part of the incoming signal, namely the value sent to the "Mod" output of the Modulo Module, to the "A" input port of the second Modulo Module. Next, create a Constant with the value "0.1" at the "B" input port of that Modulo Module. Wire the "Div" output port that same Modulo Module to the "Sel" input of the last remaining Multi Picture Module. With the "B" input port of the first Modulo Module receiving a constant value of "0.1", the "Mod" output port of the Modulo Module outputs the remainder of the incoming value, divided by "0.1". To demonstrate this with the numbers used in the previous example, if |x| = 5.2, then for the first Modulo Module, Mod = 0.2 and sending that value to the "A" input of the second Modulo Module, we get 0.2 / 0.1 = 2 at the "Div" output of the second Modulo Module. In the same way, if |x| = 3.8, then for the first Modulo Module, Mod = 0.8 and when that value is sent to the "A" input port of the second Modulo Module, the "Div" output of that Module is 0.8 / 0.1 = 8. Note that since the range of the "Mod" output of the first Modulo Module is [0 ... 0.9] and in the second Modulo Module we divide by "0.1", the range of the "Div" output of the second Modulo Module is [0 ... 9]. In this case "Mod" has a range [0 ... 9] which exactly matches the indices of the pictures of the animation file "decimals.tga" loaded into that Multi Picture Module. Go to the Instrument Panel and move the previously created knob to test out the functionality of the value display.

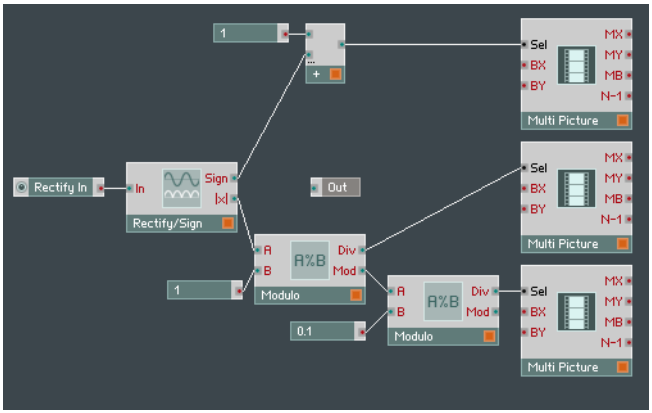


Fig. 1.32 The Structure for sorting out the sign and decimal parts of the value.

Now that you have a functioning value display, you probably want a way to manipulate the values directly through the value display itself, not with an extra knob cluttering your Panel layout. This can conveniently be achieved with the Mouse Area Module. Place it into the structure in front of the Rectify/Sign Module, leaving a bit of space for a Merge Module that you will insert later. By default, the Mouse Area Module is transparent in the Instrument Panel. To make testing the functionality easier you should make the mouse area visible. First, the Mouse Area Module needs to have a graphic element in the Instrument Panel. In the Appearance page of the Module, click on the Bar Outline Style selector to create a filled rectangle as the graphical element of the mouse area in the Instrument Panel. Now you have a 100 % transparent rectangle marking the mouse area in the Instrument Panel; for ease of manipulation you should to make the appearance more opaque. To do this, enter the value "80" into the Inactive Transparency edit field and the value "70" into the Active Transparency edit field of the Module's Appearance page. To add one more touch, click on the Left Button selector to make the active state of the mouse area appear during a left-click on the mouse area in the Instrument Panel. Now you should be able to see the mouse area in the Instrument Panel and see it become darker when you click on it. Unlock the Panel and move the mouse area over your value display. Set the dimensions of the mouse area to cover all digits of the value display using the Size X and Size Y edit fields in the Appearance page of the Mouse Area Module. After finding good settings for the dimensions, ("30" for Size X and "60" for Size Y should be good) lock the Panel again.

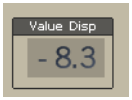


Fig. 1.33 The value edit field with the mouse area.

The Mouse Area Module can supply us with the values that we want to display with the structure that you built above. When you click and drag the mouse over the mouse area in the Instrument Panel, the "X" and "Y" output ports output events carrying the "x" and "y" coordinates, respectively, as specified in the Function page of the Module. For our purpose, we are only interested in clicking and dragging the mouse vertically to change the values in the value display since that seems the most intuitive setup for most Instrument layouts. For this reason we will leave the "X" output (and hence the settings in the Range X area of the Function page) alone and focus on the "y" coordinate. Connect the "Y" output port of the Mouse Area Module to the "In" input port of the Rectify/Sign Module. In order to be able to control the output values of the "X" and "Y" output ports of the Mouse Area Module over the mouse area in the Instrument Panel much like a fader, that is, that clicking and dragging changes the output value, engage the Incremental Mouse checkbox in the Function page of the Mouse Area Module.

The range of values that we want to get out of the Mouse Area Module is [-9.9 ... 9.9]. To specify these parameters, go to the Function page of the Mouse Area Module and enter "-9.9" into the Min Y Value edit field and "9.9" into the Max Y Value edit field in the Range Y area. Next, we want to determine the precision with which one can manipulate the output values. The relevant parameters to this are edited with the Num Steps X and Mouse Reso X edit fields. If Num Steps X and Mouse Reso X are too large, you would have to drag the mouse all the way to the top of the screen to see any change in the output value. On the other hand, if these parameters are too small, dragging the mouse over the mouse area just a small distance already effects a great change in the output value. In order to have the right value increments at the output, the Num Steps X value should be an integer multiple of the Mouse Reso X value. Try playing around with the values and seeing the effect it has on the value display, but in the end settling with a value of "1000" in the Num Steps X edit field and "500" in the Mouse Reso X edit field should be fine.

You now have a structure that displays values and lets you directly manipulate the values by clicking and dragging the mouse over the display. We would like to add one additional feature to this setup, namely, the possibility to double-click the mouse area to reset the value to a certain number. The Mouse Area Module has an output port labeled "Db" that

sends out an event with the value "1" every time the mouse area on the Panel is double-clicked. You now have two sources of events to determine the number that your value edit field is supposed to display. To combine them, place a Merge Module into the structure between the Mouse Area Module and the Rectify/Sign Module. Replace the wire to the "In" input port of the Rectify/Sign Module with a wire from the "Out" output port of the Merge Module. Next, connect the "Y" output port of the Mouse Area Module to the input port of the Merge Module and finally, while holding Ctrl, drag a wire from the "Db" output of the Mouse Area Module to the three dots on the left side of the Merge Module. The latter action creates a second port for the Merge Module. When you click and drag the mouse over the mouse area in the Instrument Panel, a value is sent to the Merge Module from the "Y" output port of the Mouse Area Module. This value is then forwarded in the form of an event (carrying the same value as was received from the "Y" output port) to the value display part of your structure which starts with the Rectify/Sign Module. If you now double click the mouse area in the Instrument Panel, the value "1" is sent to the Merge Module. Now the output port of the Merge Module sends an event with the value "1". This should be reflected in the Instrument Panel where the value display should display "+1.0". Each new event at any of the input ports of the Merge Module triggers an event at the output port with the same value as the last incoming event. During initialization, the value from the lowest input port of the Merge Module is sent last to its output port so in the current structure, if you press the Run/Stop Audio button twice (turning the audio off and then on again) to reinitialize the structure, your value display will show the initialization value coming from the "Db" output port of the Mouse Area Module, namely "1". Depending on the circumstances, this might not be what you want, so changing the port order of the wires connected to the Merge Module can be a way to get the desired reinitiation behavior. In this case switching the port orders around will yield the initialization value "0" from the "Y" port of the Mouse Area Module to be sent to the Merge Module's output last.
Regardless of the port order of the wires at the Merge Module's input, you might not want the value edit field to be reset to "+1.0" when you double click the display. Having a reset-behavior of "0" might be much more natural. This can be changed very easily with the help of only one additional Module. Insert a Value Module into the structure and connect the "Db" output port of the Mouse Area Module to the "Trig" input port. Any event at the "Trig" input port will cause an event with the value lying at the "In" input port of the Value Module to be sent from its output port. In this case we leave the "In" input port disconnected which means that an event from the "Db" output port of the Mouse Area Module causes an event with the value "0" to be sent from the output of the Value Module. This is exactly

what you want: you replaced an event carrying the value "1" with an even carrying the value "0"! Now replace the wire at the input port of the Merge Module that is connected to the "Db" output port of the Mouse Area Module with a wire to the output port of the Value Module. Now go to the Instrument Panel, make sure the value display is nonzero and double click it. Voila! The value display should now reset to the value "0".
Your value edit field is almost complete. You are probably asking, “Why almost and not fully complete??” The reason lies yet again in the initialization behavior. If you reinitialize the structure or load the Instrument anew, regardless of the value that the display was set to, it will always show "0". If you want to use this structure in an Instrument, it is desirable to have the Instrument's state upon saving fully recalled. Knob, fader, and value display settings count among the things you usually do not want to change when you reload an Instrument. This is where the Snap Value Module comes into play. Insert one into the structure and connect its input port to the output port of the Merge Module. The output port of the Snap Value Module should now be connected to the input port of the Rectify/Sign Module and the "Out" Out Port that you added to your Macro's structure in the very beginning. The Snap Value Module takes the last value that was received at its input port and stores it in memory. When the structure is reinitialized or the Instrument is reloaded, this value will be sent to the output port of the Snap Value Module (and on to the structure downstream) instead of the initialization values from the Mouse Area Module. To ensure that the Snap Value Module properly deals with the values that it is supposed to store, go to its Function page and set the Max Value edit field to "9.9" and the Min Value edit field to "-9.9". This range is the one we have been using in this example. Also, to retain the functionality that we have achieved until now, that is, that the values from the Mouse Area Module are sent to the rest of the structure (and now, to the Out Port), make sure that the Event Thru checkbox in the Function page of the Snap Value Module is engaged. If the Event Thru checkbox would be disengaged, the values arriving at the Snap Value Module's input port would not be forwarded to the output port except during initialization. Your complete structure should now look like the following picture.

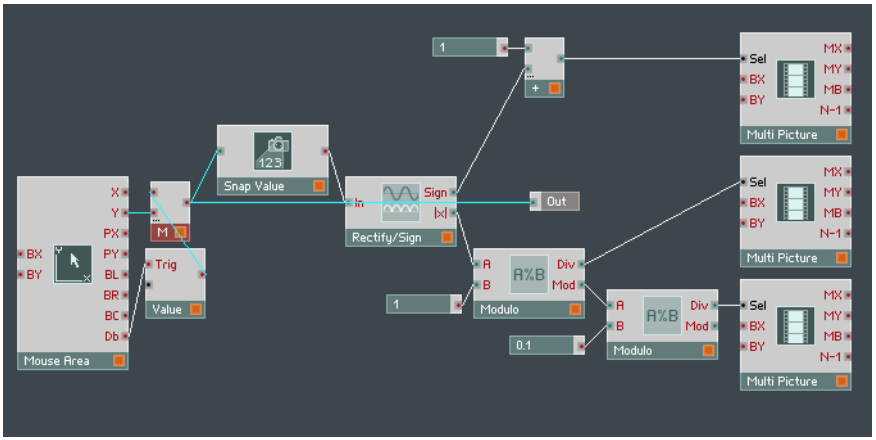


Fig. 1.34 The final Structure of the value edit field.

1.12 Text

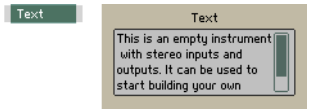


Fig. 1.35 A Text Module

1.12.1 Overview

The Text Module lets you display a written text with simple alignment options in the Instrument Panel. The displayed text should be entered in the Module's Info page. The Text Module does not process any signals.

Application

The main application for the Text Module is to add text to the Instrument Panel. However, you can also use it to add text to the Structure. For example it can be used for noting the author and creation date of an instrument and to explain how it works. Longer explanations can be entered into the Info page. With the Hints feature turned on you can display such a text as a fly-out info when hovering over the Module with your mouse cursor.

The keyboard shortcut for toggling the Hints feature on and off is Ctrl+I if you are in Windows (Cmd+I if you are in Mac OS X).

1.12.2 Properties: Info Page

Entering the Display Text

The Info page for the Text Module serves to let you enter the text that the Module should display.

- To display a text on the Instrument Panel and in the Module's fly-out info in the Structure, go to the Text Module's Info page and enter your text in the edit field below. This is shown in the screenshot below.

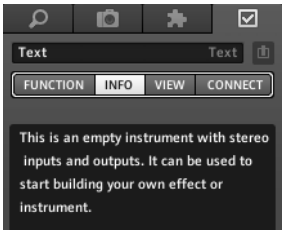


Fig. 1.36 Use the Info page to enter the text that should be displayed in the Instrument Panel and fly-out info.

This one instruction step here comprises the example for the Text Module (besides placing it into the Structure, of course).

1.12.3 Properties: View Page

Setting the Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for your text. If the display width is insufficient to have the text displayed in one line, the text will be wrapped. If the text requires more vertical space than the chosen display height, a vertical scrollbar appears on the right side of the display.

► To set the width and height of the Text Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

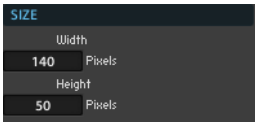


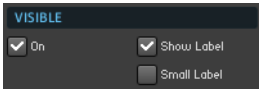
Fig. 1.37 Use the Width and Height edit fields to change the display size of the Text Module.

Choosing Label Size and Visibility Settings

► To reduce the size of your Text Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

► To make the Text Module's Panel representation disappear, disengage the On checkbox (shown in the figure below).



► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Text Module's display disappear.

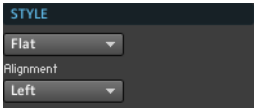
Choosing the Text Style

You can choose between three styles of Panel representations of the Text Module: Flat, Transparent, and Frame.

► To choose the Panel representation style for the Text Module, go to the Module's View page and select the appropriate menu entry from the Style drop-down menu, as shown in the figure below.

For the Panel representation you can choose between three text alignment options: left, center, and right.

► To choose the text alignment, go to the Module's View page and choose the corresponding menu entry from the Alignment drop-down menu, as shown in the figure below.



1.13 Multi Text

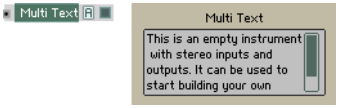


Fig. 1.38 Multi Text Module

1.13.1 Overview

The Multi Text Module provides a changeable text display for your Instrument Panel. Any number of texts can be added in the Module's Function page, where they can also be edited or deleted. Only one text is displayed at a time on the Instrument Panel. The displayed text is chosen by the input port.

Application

Use the Multi Text Module to create custom controls whose texts or labels change depending on the control state. The Multi Text Module is also useful when creating interactive tutorials in REAKTOR.

1.13.2 Ports

- **(In)** "In" is the Audio input port for the number of the text item to be displayed. The value "0" addresses the first text, the value "1" the second text, and so on.

1.13.3 Properties: Function Page

Entering the Display Texts

You can enter several texts to be displayed by the Multi Text Module in the Module's Function page.

► To create a list of texts, go to the Multi Text Module's Function page.

► To enter the first text in the list, just enter the appropriate text in the large edit field, shown in the figure below.

► To add another text at the end of the list, press the Append button.

► To switch between the different texts, click and drag the Text ID edit field or directly enter the new text ID value into that edit field.

► To insert a new text item at the position after the current text ID, press the Insert button.
► To delete the current text from the list, press the Delete button.

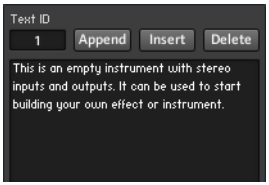


Fig. 1.39 Use the Multi Text Module's Function page to add and edit texts that should be displayed.

Keeping the Multi Text Module Always Active

If the Multi Text Module is not connected to an active signal flow, then you are unable to switch the displayed texts via the input port. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Multi Text Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

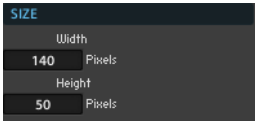
Fig. 1.40 To activate the Multi Picture Module, engage the Always Active checkbox in its Function page.

1.13.4 Properties: View Page

Setting the Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for your text. If the display width is insufficient to have the text displayed in one line, the text will be wrapped. If the text requires more vertical space than the chosen display height, a vertical scrollbar appears on the right side of the display.

► To set the width and height of the Multi Text Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

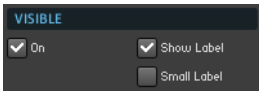


Choosing Label Size and Visibility Settings

► To reduce the size of your Multi Text Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

► To make the Multi Text Module's Panel representation disappear, disengage the On checkbox (shown in the figure below).



► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Multi Text Module's display disappear.

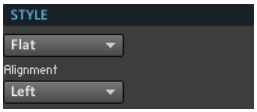
Choosing the Text Style

You can choose between three styles of Panel representations of the Multi Text Module: Flat, Transparent, and Frame.

► To choose the Panel representation style for the Text Module, go to the Module's View page and select the appropriate menu entry from the Style drop-down menu, as shown in the figure below.

► For the Panel representation you can choose between three text alignment options: left, center, and right.

► To choose the text alignment, go to the Module's View page and choose the corresponding menu entry from the Alignment drop-down menu, as shown in the figure below.



► The Style drop-down menu lets you choose the Panel representation of the Multi Text Module. Use the Alignment drop-down menu to select the text alignment.

1.13.5 Example: Instructions for Tutorial

You can use the Multi Text Module to document tutorial Instruments, for example. As shown in the Structure below, use the List Module to control which text is shown in the area of the Instrument Panel designated for the Multi Text Module.

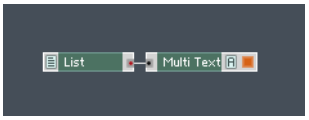


Fig. 1.41 Use the List Module which text is to be displayed by the Multi Text Module.

Next, you need to enter the desired texts into the Multi Text Module. Each text has an ID number (starting with "1"), by which the List Module (↑1.3, List) addresses it. Click the Append button (shown in the figure below) to add the first text. Type in the text and press the Append button again to add another text. Use the Text ID edit field to go back to edit or view a text. The resulting Panel representation is shown in the second figure below.

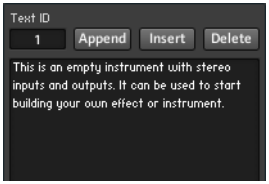


Fig. 1.42 Use the Multi Text Module's Function page to add and edit texts.

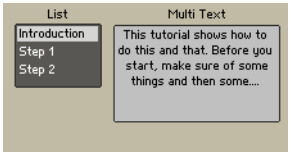


Fig. 1.43 The Panel representation of the Text Module along with the List Module .

1.14 XY

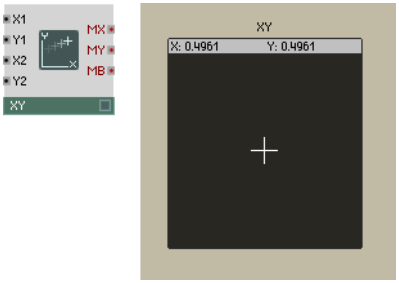


Fig. 1.44 An XY Module

1.14.1 Overview

The XY Module has two functions. It displays Audio input signals and acts as a 2-dimensional controller to be used with the mouse. Incoming data at the "X1", "Y1", "X2", and "Y2" input ports can be visualized on the XY Module's Panel representation in 9 different forms including pixels, rectangles, crosshairs, lines, and as an oscilloscope display.
Similarly, when the mouse cursor lies on the XY Module's Panel representation and the (left) mouse button is depressed, you can send Events with values relating to the cursor position on the display from the Module's output ports. The last position can be visually represented by a crosshair with a size of your choice.
The XY Module has two operation modes:

- If the Incremental Mouse Mode is turned off, the absolute mouse position on the Module's Panel representation is translated to values between the "Min" and "Max" values set in the Function page. . The values are output to the "MX" (mouse X) and "MY" (mouse Y) output ports for the horizontal and vertical position, respectively. The display width and height define the range and resolution for mouse movements and the Mouse Reso edit field in the Function page has no effect.
- With the Incremental Mouse Mode turned on, the "MX" (mouse X) and "MY" (mouse Y) values are controlled by the relative movement of the mouse compared to the point where the left the mouse button was pressed. The X and Y mouse resolution is given by the Mouse Reso edit field in the Function page.

Application

A common application for the XY Module is to create scroll bars for custom displays on your Instrument Panel. You can also use the XY Module as an oscilloscope and use the crosshairs to make measurements of the incoming signal which are then directly forwarded to the Structure for further processing. Of course, you can also use the XY Module also as a conventional XY controller.

1.14.2 Ports

Input Ports

- **(X1)** "X1" is the Audio input port for the X1 coordinate of the visual object.
- **(Y1)** "Y1" is the Audio input port for the Y1 coordinate of the visual object.
- **(X2)** "X2" is the Audio input port for the X2 coordinate of the visual object.

- **(Y2)** "Y2" is the Audio input port for the Y2 coordinate of the visual object.

Output Ports

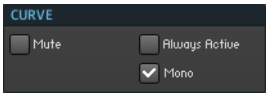
- **(MX)** "MX" (mouse X) is the Event output port for the horizontal (X) mouse position when the mouse is within the XY Module's display area in the Instrument Panel. Events with the current mouse position are only sent when the mouse button is depressed.
- **(MY)** "MY" (mouse Y) is the Event output port for the vertical (Y) mouse position when the mouse is within the XY Module's display area in the Instrument Panel. Events with the current mouse position are only sent when the mouse button is depressed.
- **(MB)** "MB" (mouse button) is the Event output port for the left mouse button status. It is "1" when the mouse button is depressed; otherwise it is "0". Events are sent from this output port only when the mouse is within the XY Module's display area in the Instrument Panel.

1.14.3 Properties: Function Page

Keeping the XY Module Always Active

If the XY Module is not connected to an active signal flow, then the incoming values at the input ports are not displayed by the Panel representation. In such a case you can activate the Module with the Always Active checkbox.

► To activate the XY Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Activating Incremental Mouse Mode

The output values for the XY Module are calculated in two different ways, depending on the operation mode. These two modes are discussed in the Module's Overview, above.

► To toggle between the two XY Module operation modes, engage or disengage the Incremental Mouse Mode checkbox accordingly. This checkbox is shown in the screenshot of the Function page below.

Setting the Range and Resolution for the Output Values

The range and resolution of the XY Module's values at the "MX" (mouse X) and "MY" (mouse Y) output ports are set in the Output Range area of the Module's Function page, shown in the screenshot below. The edit fields contained in this area function analogously to the ones contained in a conventional Knob or Fader Module. To learn how these edit fields affect the output values, please refer to subsection 8.2.1 in the Application Reference.

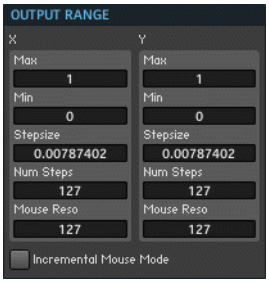


Fig. 1.45 The range and resolution of the XY Module's output values are set in the Output Range area of its Function page.

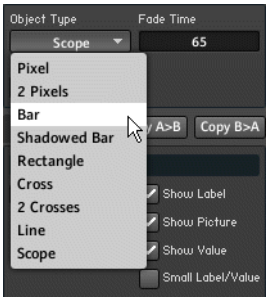
1.14.4 Properties: View Page

In the View page you can set the type of object that visualizes the incoming Audio signals. There are nine options:

- **Pixel:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of the pixel.
- **2 Pixels:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of the first pixel; "X2" and "Y2" determine the coordinates of the second pixel.
- **Bar:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of one corner of the bar; "X2" and "Y2" determine the coordinates of the second corner.
- **Shadowed Bar:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of one corner of the shadowed bar; "X2" and "Y2" determine the coordinates of the second corner.
- **Rectangle:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of one corner of the rectangle; "X2" and "Y2" determine the coordinates of the second corner.
- **Cross:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of the crosshair.

- **2 Crosses:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of the first crosshair; "X2" and "Y2" determine the coordinates of the second crosshair.
- **Line:** The input ports "X1" and "Y1" control the horizontal and vertical coordinates of the line starting point and the coordinates of the line's ending point are set with the "X2" and "Y2" input ports.
- **Scope:** The scope object displays fast moving Audio data that is determined by the "X1" and "Y1" coordinates. Note that all other display modes evaluate the input ports only at the lower graphics update rate.

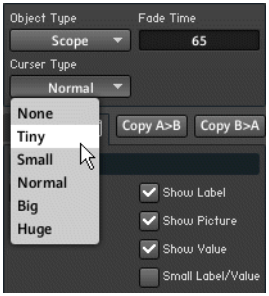
► To choose the object type, go to the XY Module's View page and select the corresponding menu entry from the Object Type drop-down menu, as shown in the figure below.



► Use the Object Type drop-down menu to choose the data display type.

The XY Module's display fades from one display (data) state to the next. You can set the fade time, in milliseconds in the range [0 … 100].

► To set the display fade time, go to the XY Module's View page and enter the desired fade time in milliseconds into the Fade Time edit field, as shown in the figure below.



► Use the Fade Time edit field to set the crosshair size and the Cursor Type drop-down menu to choose the cursor size.

► You can also set the size of the crosshair appearing at the mouse position.
► To set the size of the crosshair, choose the appropriate menu entry from the Cursor Type drop-down menu in the View page (shown in the screenshot above).

1.14.5 Example: Multiplex Sequencer

This example shows how to build a sequencer with variable length, but with a maximum of 16 steps. The first part of the Structure is shown in the figure below. There you see that the core of the sequencer is a Multiplex 16 Module (↑8.5, Multiplex 16 Sequencer) and the values of the steps are received from Knob Modules (↑1.1, Fader/Knob) at the input ports "0" to "15". The sequence length of the step at which the sequencer "folds back" to the first step, is set with the knob labeled "Length" at the "Len" input port. The actual position of the sequencer is received at the "Pos" (position) input port in the form of a regular Events with incremental values. The value of the Event determines the sequencer position where values greater than "Len" are "folded back" to the set sequencer range.

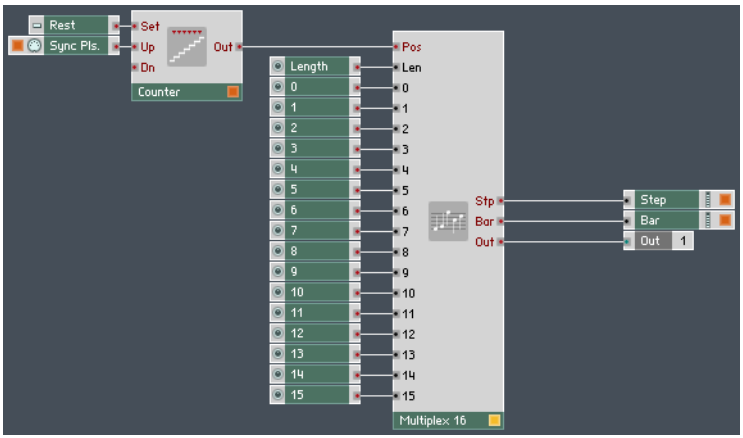


Fig. 1.46 Multiplex Sequencer

In this example, the "Pos" values are ultimately received from the Sync Clock Module (↑2.15, Sync Pulse) which sends clock Events at regular time intervals, synchronized to the MIDI Clock. The rate of outgoing Events can be set at the Sync Clock Module's (↑2.15, Sync Pulse) Function page, with the Rate drop-down menu. In order to get regular Events carrying incremental values, the Counter Module (↑13.2, Counter) has been used. Each Event from the Sync Clock Module (↑2.15, Sync Pulse) increments the output of the

Counter Module by one, which then ends up being used as the "Pos" value for the Multiplex 16 Module (↑8.5, Multiplex 16 Sequencer). At the Counter Module's "Set" input port you should have a Button Module (↑1.2, Button). In its Function page, set the Button Module to Trigger Mode and its "Max" value to zero. This enables you to reset the sequencer to its first position from the Instrument Panel.
As an additional feature, the XY Module has been used to graphically track the current sequencer position. The final Structure with this implemented is shown in the first figure at the end of the example and the final Panel representation of the sequencer is shown at the last figure at the end of the example. Please look at that figure to see where we are headed with the XY Module. The XY Module should be configured to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). This corresponds to going to its View page and choosing the *Rectangle* menu entry from the Object Type drop-down menu (shown in the figure below). Also, you don't need a cursor and therefore choose the *None* menu entry from the Cursor Type drop-down menu (also shown in the figure below). While you are already in the XY Module's View page, disengage the Show Label and Show Value checkboxes and set the Height and Width edit fields so that the XY Module's Panel representation looks similar to what is depicted in the last figure of this example.

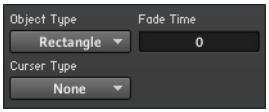


Fig. 1.47 To draw a rectangle from (X1, Y1) to (X2, Y2), choose the Rectangle menu entry from the Object Type drop-down menu.

The rectangle object type causes the XY Module's Panel representation to draw a rectangle from the coordinates (X1, Y1) to the coordinates (X2, Y2). We want to turn the XY Module into a row of 16 "imaginary" rectangles, each corresponding to one of the 16 steps in the sequencer. This is done the easiest when each box has the dimensions 1 by 1. For this reason, it is necessary to set the range of displayed values correctly in the XY Module's Function page. Set the "Min" and "Max" values for the X coordinate to "0" and "16", respectively. "Min" and "Max" for the Y coordinate should be "0" and "1", respectively. This is shown in the figure below.

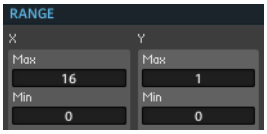


Fig. 1.48 Use the Min and Max edit fields to set the range for the X and Y coordinates to [0 ... 16] and [0 ... 1], respectively.

Now that you have configured the XY Module, you just need to feed the correct values from the Multiplex 16 Module (↑8.5, Multiplex 16 Sequencer) to the input ports for the corresponding coordinates. Remember you wish to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). As shown in the Structure depicted below, connect the "Stp" (step) output port off the Multiplex 16 Module to the "X1" input port of the XY Module, and the value Step + 1 (using an Add Module, see ↑4.2, Add, +) to the "X2" input port. Furthermore, connect the constant value "1" to the "Y2" input port. Your resulting Structure should look like the first figure below and its Instrument Panel counterpart should look similar to what is shown in the second figure. Numeric Display Modules (↑1.8, Meter) have been added at the "Stp" and "Bar" output ports to give you quantitative visual access to the step and bar positions of the sequencer.

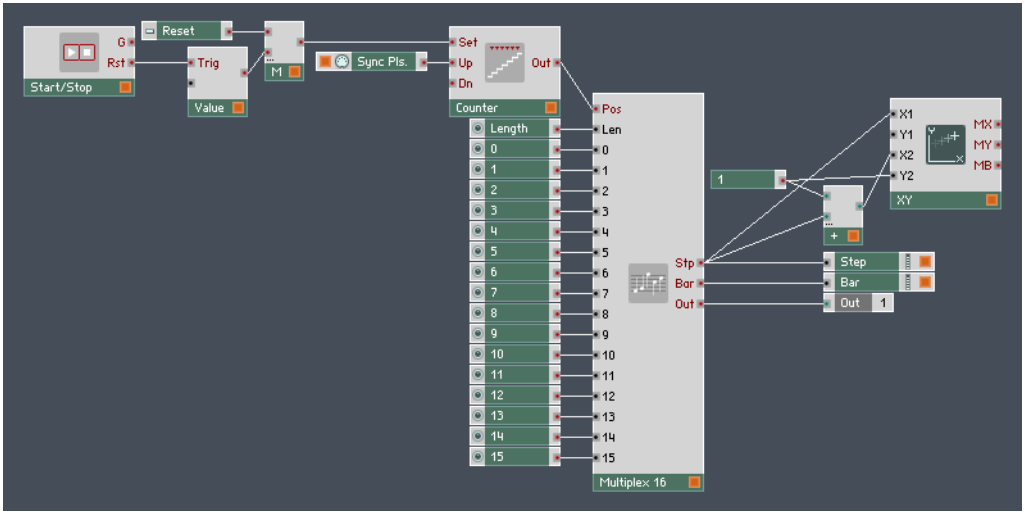


Fig. 1.49 The Structure for a simple sequencer with variable length and a simple step position display.

An additional feature to add would be that resetting the MIDI Clock also causes the sequencer position to be reset as well. Let's use the Start/Stop Module (↑2.13, Start/Stop) as that source of "reset" Events. For this, a reset operation should send the value "0" to the Counter Module's (↑13.2, Counter) "Set" input port. Since the "Rst" output port sends an Event with value "1", use a Value Module (↑13.15, Value) to change the Event's value to "0" and the a Merge Module to merge this reset signal with that received from the "Reset" button.

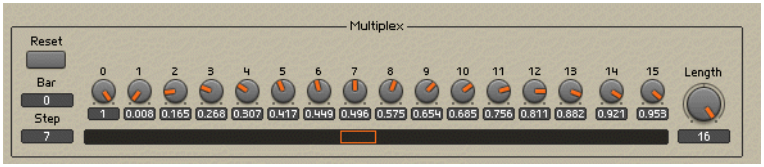


Fig. 1.50 The Instrument Panel of the simple sequencer.

1.15 Scope

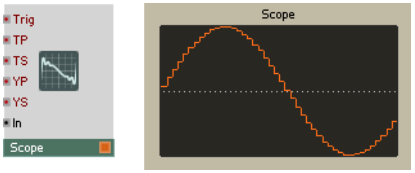


Fig. 1.51 A Scope Module

1.15.1 Overview

The Scope Module is an oscilloscope for displaying a time-varying Audio signal. Every time an Event is received at the "Trg" (trigger) input port, the Module starts recording the Audio signal at the "In" input port into its buffer and displays the content of this buffer on it's on the Panel representation. When no trigger Events are received the display continues showing the buffered signal. The displayed buffer can be shown at varying scales and positions by adjusting the values at the relevant input ports. The size of the graph in the Instrument Panel can be set in the Module's View page and the size of the signal buffer can be set in the Function page.

Application

The Scope Module is used to create a flexible oscilloscope for displaying time-varying signals. Typically you would connect an A to E Trig Module (↑14.8, A to E (Trig)) to the "Trg" (trigger) input port for triggering the Scope Module from a periodic Audio signal with zero-crossings.

The oscilloscope is the bread and butter analysis tool for analog circuit engineers and it is no different when designing DSP circuits. If you want to learn to creatively program your own DSP Structures, knowing how to monitor the signal at different points with an oscilloscope is a key skill on the way there.

1.15.2 Ports

Input Ports

- **(Trg)** "Trg" (trigger) Monophonic Event input port for the trigger signal that starts the signal trace on the display. A trigger signal is considered as a positive valued Event.
- **(TP)** "TP" (time position) Monophonic Event input port for controlling the offset in time (Time Position) of the trace in milliseconds. The trace starts at the left edge of the display "TP" milliseconds after the trigger signal (at the "Trg" (trigger) input port). The range of values at this input port should be [0 … <buffer time set in the Properties>]. When this input port is disconnected, the default value that is used is "0".
- **(TS)** "TS" (time scale) Monophonic Event input port for controlling the time scale of the displayed trace. From the left to the right edge the display shows "TS" milliseconds of the signal. The range of values at this input port should be [0 … <buffer time set in the Properties>]. When this input port is disconnected, the default value that is used is "0".
- **(YP)** "YP" (Y position) Monophonic Event input port for controlling the amplitude offset of the trace. With this input port disconnected, the default value, "0", is used. This means that the signal trace is drawn with the zero-point reference taken as the dotted line through the middle of the display. When YP = -1, the bottom edge of the display is taken as this reference point and for YP = +1 the top edge is taken as the zero-point in the amplitude scale.

- **(YS)** "YS" (Y scale) is the Monophonic Event input port for controlling the amplitude scaling of the signal trace. The difference between the signal values at the top and at the bottom of the display is 2*YS. When YP = 0, the display shows values between +YS and –YS. The range of values at this input port should be [-1 … 1]. When this input port is disconnected, the default value that is used is "0".
- **(In)** "In" Monophonic Audio input port for the signal to be displayed.

Note that the "Trg" (trigger), "TS" (time scale), "YS" (Y scale), and "In" input ports should all be connected in order for you to see a well-defined signal on the scope display.

1.15.3 Properties: Function Page

Keeping the Scope Module Always Active

If the Scope Module is not connected to an active signal flow, then the incoming values at the input ports are not displayed by the Panel representation. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Scope Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Fig. 1.52 To activate the Scope Module, engage the Always Active checkbox in its Function page.

Changing the Buffer Size

► To change the time interval of the incoming signal that is buffered in the Scope Module's internal buffer, go to the Module's Function page and enter the new time in milliseconds into the Buffer edit field, shown in the figure below.

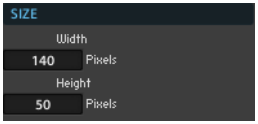
Fig. 1.53 Enter the desired buffer value in milliseconds into the Buffer edit field.

1.15.4 Properties: View Page

Setting the Oscilloscope Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for your oscilloscope.

► To set the width and height of the Scope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

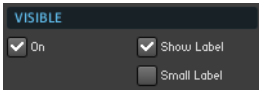


Choosing Label Size and Visibility Settings

► To reduce the size of your Scope Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

► To make the Scope's Panel representation disappear, disengage the On checkbox (shown in the figure below).



► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Scope Module's display disappear.

1.15.5 Example: Simple Scope

This example shows how to display a periodic Monophonic Audio signal on a scope. The incoming signal should have periodic (regular) zero crossings in order for the A to E Trig Module (↑14.8, A to E (Trig)) to create a trigger signal from it. Of course, you could use other mechanisms for triggering the scope trace. The Structure below shows the essential components for the scope. The A to E Trig Module creates a trigger signal from a positive zero crossing of the incoming signal. Upon sending a trigger signal to the Scope Module,

an interval of the signal incoming at the Scope Module's "In" input port is buffered that corresponds to the time interval set in the Module's Function page with the Buffer edit field. A knob with the range [0 … 1] and label "X" sends the scaling of the time axis of the scope to the "TS" input port. The scaling for the "Y" (signal level) axis is set in the logarithmic scale with the knob labeled "Y" (range [-32 … 32]), converted to linear scale with the Exp Lvl-to-A Module (↑4.15, Exp (Lvl-to-A) Module), and then sent to the "YS" input port.

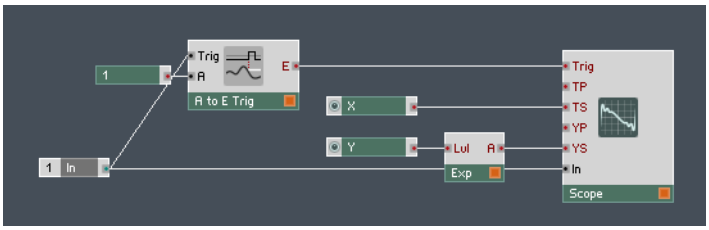


Fig. 1.54 The Structure for a simple scope that displays a Monophonic periodic Audio signal with regular zero crossings.

1.16 Multi Display

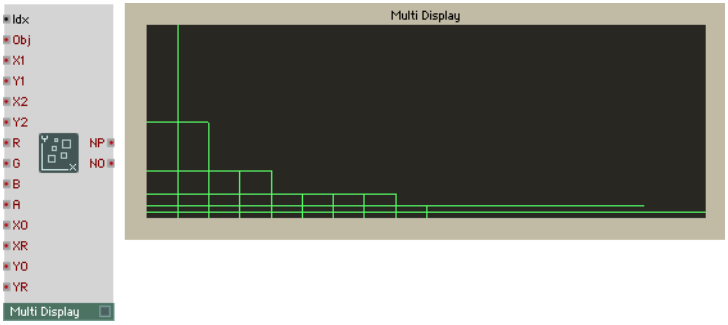


Fig. 1.55 A Multi Display Module

1.16.1 Overview

The Multi Display Module enables you to display and manipulate multiple graphical objects. Graphical objects can be chosen to be crosses, bars, rectangles, lines, and (animated) pictures. A range of parameters including an object's type, its coordinates, transparency, and color can be defined individually for each graphical object. The Properties of the

Multi Display Module include a variety of options for customization of the display, including background color and picture, as well as how the objects are addressed by the "Idx" (index) input port. All Multi Display input ports only accept Monophonic signals.
The major difference between the Multi Display and Poly Display Module is how the number of graphical objects is specified. For the Multi Display Module the number of objects is defined by the Number of Objects edit field in its Function page whereas for the Poly Display Module, that number is specified by the number Polyphonic Voices of the parent Instrument. The advantage of the Multi Display Module is that it can display any number of graphical objects, regardless of the number of Polyphonic Voices of the parent Instrument. Each object can then be addressed independently using the "Idx" (index) input port. In contrast, the Poly Display Module could be considered easier to program as it doesn't require any index-processing "code", and all objects can be addressed simultaneously by virtue of parallel Polyphonic Voice processing. However, the number of objects is limited to the number of Polyphonic Voices of the parent Instrument.

Application

When used in conjunction with the Mouse Area (↑1.18, Mouse Area) and Snap Value Array (↑14.27, Snap Value Array) Modules, the Multi Display Module allows sophisticated custom interface elements such as sequencers to be constructed. You could also create a scope using the value "-3" at the "Obj" (object) input port. With the number of objects equal to the width of the Multi Display's Panel representation in pixels, you can draw smooth lines that are updated at the display rate.
The order of Events at the various input ports is significant when "programming" the Multi Display Module. For each object, the Event addressing the object with its index value has to arrive at the "Idx" input port before the objects parameters are set with Events at the other input ports. Please refer to subsection 9.2.2 in the Application Reference for more information on manipulating the order of Events.
For more complex applications such as the ones mentioned above, it is necessary to populate the internal state of the Multi Display Module using the Iteration Module (↑13.13, Iteration). This allows for nearly instantaneous updating of the display in the Instrument Panel.

1.16.2 Ports

Input Ports

- **(Idx)** "Idx" (index) is the Audio input port for the index of the graphical element that you wish to address. The index is 1-based; i.e. the index of the first object is "1" (not "0"). Fractional values are rounded to the nearest integer. Where "Idx" (index) values are outside the range [1... N], the behavior depends on the setting of the Index Behaviour radio button selector in the Function page. The stacking order of graphical objects is determined by their "Idx" values. Objects with lower "Idx" values appear on top of objects with higher "Idx" values. It is essential to set the "Idx' value before transmitting Events to the other input ports.
- **(Obj)** "Obj" (object) is the Event input port to specify the graphical appearance of the object currently addressed at the "Idx" input port (or all objects if the Ignore Object checkbox has been engaged in the Function page). Fractional values are rounded to the nearest integer. Obj = -4 corresponds to a cross. The cross consists of a vertical and horizontal line. When the Center to X1, Y1 checkbox in the Function page is disengaged, the starting and end points for the horizontal line are given by "X1" and "X2", respectively and the starting and end points for the vertical line are given by the "Y1" and "Y2", respectively. Obj = -3 corresponds to a line from the coordinates "X1" and "Y1" to the coordinates "X1" and "Y1" of the object of the same type. Obj = -2 corresponds to a line drawn from the coordinates "X1" and "Y1" to the coordinates "X2" and "Y2" (when the Center to X1, Y1 checkbox in the Function page is disengaged). Obj = -1 corresponds to a bar whose opposite corners are given by the coordinates pairs (X1, Y1) and (X2, Y2) when the Center to X1, Y1 checkbox in the Function page is disengaged. Obj = 0 corresponds to a rectangle whose opposite corners are given by the coordinates pairs (X1, Y1) and (X2, Y2) when the Center to X1, Y1 checkbox in the Function page is disengaged. If you have loaded an animated picture with the Picture drop-down menu in the Function page you can send values in the range [1 ... NP] to the "Obj" input port where "NP" is the number of animation frames in the picture. In this case the "Obj" value specifies which animation frame should be displayed at the location specified by the coordinate pair (X1, Y1) (when the Center to X1, Y1 checkbox in the Function page is disengaged). When the Center to X1, Y1 checkbox has been engaged in the Function page, the "X1" and "Y1" coordinates define the the center of the object.
- **(X1)** "X1" is the Event input port for specifying the X1 coordinate.

- **(Y1)** "Y1" is the Event input port for specifying the Y1 coordinate.
- **(X2)** "X2" is the Event input port for specifying the X2 coordinate.
- **(Y2)** "Y2" is the Event input port for specifying the Y2 coordinate.
- **(R)** "R" (red) is the Event input port for specifying the amount of red component in the RGB color of the object addressed by the "Idx" (index) input port. The range of values at this input port is [0 … 1]. If this input port is disconnected, the default value, "1", is used. If the Ignore RGB checkbox has been engaged in the Function page, this "R" (red) value is applied to all objects.
- **(G)** "G" (green) is the Event input port for specifying the amount of green component in the RGB color of the object addressed by the "Idx" (index) input port. The range of values at this input port is [0 … 1]. If this input port is disconnected, the default value, "1", is used. If the Ignore RGB checkbox has been engaged in the Function page, this "G" (green) value is applied to all objects.
- **(B)** "B" (blue) is the Event input port for specifying the amount of blue component in the RGB color of the object addressed by the "Idx" (index) input port. The range of values at this input port is [0 … 1]. If this input port is disconnected, the default value, "1" is used. If the Ignore RGB checkbox has been engaged in the Function page, this "B" (blue) value is applied to all objects.
- **(A)** "A" (alpha) is the Event input port for the object transparency of the object addressed by the "Idx" (index) input port. The range of values at this input port is [0 … 1]. A = 0 corresponds to a fully transparent object and A = 1 corresponds to a fully opaque object. If this input port is disconnected, the default value, "1" (fully opaque), is used. If the Ignore A checkbox has been engaged in the Function page, the "A" (alpha) value is applied to all objects.
- **(XO)** "XO" (X origin) is the Event input port for the lowest visible "X" values. Connecting this input port to a control can be used for scrolling. Values incoming at this input port override the X Origin edit field in the Function page. When disconnected, the value from the X Origin edit field in the Function page is used.
- **(YO)** "YO" (Y origin) is the Event input port for the lowest visible "Y" values. Connecting this input port to a control can be used for scrolling. Values incoming at this input port override the Y Origin edit field in the Function page. When disconnected, the value from the Y Origin edit field in the Function page is used.

- **(XR)** "XR" (X range) is the Event input port for the horizontal viewing range (or zoom). Values incoming at this input port override the X Range edit field in the Function page. When disconnected, the value from the X Range edit field in the Function page is used.
- **(YR)** "YR" (Y range) is the Event input port for the vertical viewing range (or zoom). Values incoming at this input port override the Y Range edit field in the Function page. When disconnected, the value from the Y Range edit field in the Function page is used.

Output Ports

- **(NP)** "NP"(number of pictures) is the Event output port for reporting the number of animation frames in the loaded picture.
- **(NO)** "NO" (number of objects) is the Event output port for reporting the number of graphical objects set in the Number of Objects edit field in the Function page.

1.16.3 Properties: Function Page

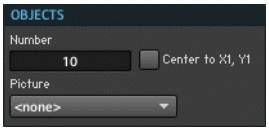
Keeping the Multi Display Module Always Active

If the Multi Display Module is not connected to an active signal flow it will not display its internal state in the Panel representation. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Multi Display Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

► To set the number of objects displayed by the Multi Display Module, go to its Function page and enter the desired number into the Number of Objects edit field, shown in the screenshot below. Note that reducing the number of objects will also delete any coordinate, transparency, and color data that was saved with the superfluous objects.

► To define the object location only by its center coordinates (for which the pair of values (X1, Y1) is used), go to the Module's Function page and engage the Center to X1, Y1 checkbox (shown in the screenshot below).



► In the Objects area of the Multi Display Module's Function page you can set the number of objects, the way the coordinates are specified, and load an (animated) picture to represent the graphical objects.
To load the (animated) picture to represent the object for "Obj" (object) values greater or equal to "1":

1. Go to the Module's Function page and open the Picture drop-down menu (shown in the figure above).
2. Now either choose the *-> Open from file...* menu entry to load a new picture file or choose a menu entry corresponding to a picture file that has already been loaded into the Ensemble by another Module.
3. This opens up the Picture Properties dialog window. This window shows the picture preview and lets you specify how REAKTOR should extract the animation frames from the loaded picture. Please refer to section 8.5 in the Application Reference for more information on working with the Picture Properties dialog window. Press the OK button when you are finished setting the picture properties.

Choosing the Origin and Range of Displayed Values

The origin and range of the X and Y coordinates in your Multi Display Module determine where (if a at all) your graphical objects, whose coordinates are set with the "X1", "Y1", "X2", and "Y2" input ports, will appear in the Multi Display. The displayed interval in the X dimension in the Multi Display Module's Panel representation is then [XO ... XO + XR] where "XO" is set with the X Origin edit field and "XR" is set with the X Range edit field (shown in the screenshot below). Similarly, the displayed interval in the Y dimension in the Multi Display Module's Panel representation is then [YO ... YO + YR] where "YO" is set with the Y Origin edit field and "YR" is set with the Y Range edit field (shown in the screenshot below).

You can also set these values directly from the corresponding input ports. In the case that these input ports are connected, the values in the corresponding edit fields are overridden.

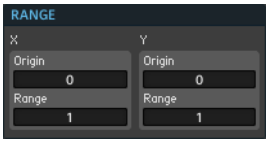


Fig. 1.56 Use the Range area of the Multi Display Module's Function page to set the range of coordinates that are displayed on the Module's Panel representation.

Setting All Graphical Objects to the Same Type, Color, and or Transparency

► To set all graphical objects to have the same object type, engage the Ignore Object checkbox in the Module's Function page.

► To set all graphical objects to have the same color (that is, the same "R", "G", and "B" values), engage the Ignore RGB checkbox in the Module's Function page.

► To set all graphical objects to have the same alpha (transparency) value, engage the Ignore A checkbox in the Module's Function page.

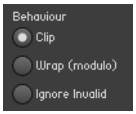
Fig. 1.57 Use the Ignore checkboxes to set the values at the corresponding input ports ("Obj", "R", G", "B", or "A") to all graphical objects, independent of their index value.

Setting the Index Behaviour

To choose the way to treat "Idx" (index) values at the "Idx" input port that exceed the number of objects ("NO" value), press the corresponding radio button from the Behaviour selector (shown in the figure below). The three choices are as follows:

- **Clip:** "Idx" (index) values that exceed the allowed range address the object with the highest or lowest index, depending on if the higher or lower bound has been exceeded.
- **Wrap (modulo):** "Idx" values that exceed the number of objects are "wrapped" around the "NO" (number of objects) value. This means that the value "NO" is subtracted from the "excessive" index value: Idx' = Idx - NO.
- **Ignore Invalid:** "Idx" (index) values that exceed the number of objects are ignored.

► Use the Behaviour radio button selector in the Multi Display Module's Function page to determine how the Module deals with "Idx" (index) values that exceed the allowed range.

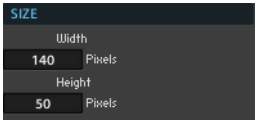


1.16.4 Properties: View Page

Setting the Multi Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for the graphical objects of your Multi Display Module.

► To set the width and height of the Multi Display Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

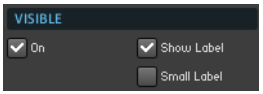


Choosing Label Size and Visibility Settings

► To reduce the size of your Multi Display Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

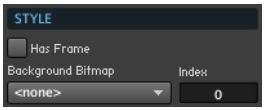
► To make the Multi Display Module's Panel representation disappear, disengage the On checkbox (shown in the figure below).



► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Multi Display Module's display disappear.

Turning on the Multi Display's Frame

- To turn on the frame for the Multi Display Module's Panel representation, engage the Has Frame checkbox in the View page (shown in the screenshot below).



The Style area of the View page lets you toggle the visibility of the frame with the Has Frame checkbox and choose a background picture with the Background Bitmap drop-down menu and the Index edit field.

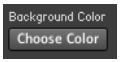
Loading a Background Picture for the Multi Display

To load a background picture for the Multi Display Module:

1. Go to the Module's View page and open the Background Bitmap drop-down menu (shown in the figure above).
2. Now either choose the *-> Open from file...* menu entry to load a new picture file or choose a menu entry corresponding to a picture file that has already been loaded into the Ensemble by another Module.
3. This opens up the Picture Properties dialog window. This window shows the picture preview and lets. If you have loaded a picture consisting of animation frames, the Picture Properties dialog window also lets you specify how REAKTOR should extract the animation frames from the loaded picture. Please refer to section 8.5 in the Application Reference for more information on working with the Picture Properties dialog window. Press the OK button when you are finished setting the picture properties.
4. If you have loaded and configured an animated picture, you can choose the picture index of the animation frame to be used as a background picture with the Index edit field (shown in the screenshot above).

Choosing a Background Color

► To choose a background color for the Multi Display Module's Panel representation using the color picker dialog window of your OS, go to the Multi Display Module's View page. Then press the Choose Color button and use the OS-specific dialog window to choose your color of choice.



Setting Borders for the Multi Display's Panel Representation

Often you have some border for your background picture where you don't necessarily want the graphical objects to appear. With the Picture Borders edit fields (see screenshot below) you can specify the border area where the graphical objects should not appear. By increasing the border either on the left or right, or top or bottom, you increase the effective area of the Multi Display Module's Panel representation such that the area on which the graphical objects are shown corresponds to the dimensions specified with the Width and Height edit fields, as shown above.

► To specify the background picture's top, bottom, left, and right border (in pixels), use the corresponding edit fields in the View page, shown in the screenshot below.

1.16.5 Example: Simple Multi Display

Usually instantaneously updating a Multi Display Module's internal state is done with a Structure that makes use of the Iteration Module (↑13.13, Iteration). This example illustrates how to plot a graph consisting of three segments on the Multi Display with the help of an Iteration Module. The corresponding Structure is shown in the figure below.

First let's begin by choosing the type of graphical object for plotting our lines. This is specified at the "Obj" input port of the Multi Display Module. For Obj = -3 the Multi Display Module draws a line from the coordinates "X1" and "Y1" of an object to the coordinates "X1" and "Y1" of the next object of the same type. Let's use that, since this way you can draw three line segments by specifying the "X1" and "Y1" coordinates of four objects. Connect a Constant with the value "-3" to the "Obj" input port and make sure that the Ignore Object checkbox in the Multi Display Module's Function page has been engaged. This way all graphical objects have the same object type of "-3".

Now let's turn to making use of the Iteration Module. For each object you need it to trigger the sending of the object's "Idx", "X1", and "Y1" values to the corresponding input ports. Note that the "Idx" value has to arrive first. All in all, since we decided to work with four objects (to draw three segments), we need the Iteration Module to send four Events. To make things easier, let's agree that the "X1" coordinates of the objects are the following, in increasing order of "Idx" value: "0", "1", "2", and "3". These values come in handy, since they can directly be generated by the Iteration Module. Connect a button (labeled "Iter-

ate") in to the "In" input port of the Iteration Module. It should be in Trigger Mode and have its "Max" value in the Function page set to "0". Pressing the button on the Instrument Panel sends an Event with the value "0" to the Iteration Module, which then in turn should output Events carrying the values "0", "0 + 1", "0 + 2", and "0 + 3". You see already that the increment value, as specified at the "Inc" input port should be "1" and that "N" should be "3" (for three additional output Events). This is shown in the Structure below.

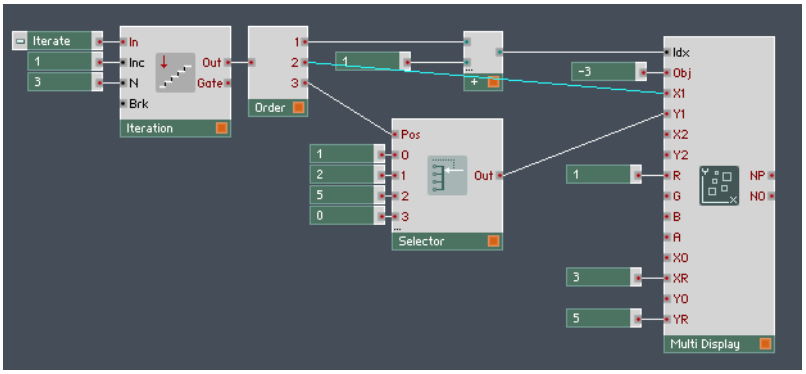


Fig. 1.58 The Structure for drawing three line segments using the Multi Display Module.

Next, use an Order Module (↑13.12, Order) at the Iteration Module's (↑13.13, Iteration) output to cause the "Idx" value to be sent first for each object and only then the "X1" and "Y1" values. Use the "1" output port of the Order Module for the "Idx" value. Since the "Idx" value starts with "1" for the first object and the values from your Iteration Module start with "0", you need to add "1" to the signal from the Order Module's "1" output port before routing it to the "Idx" input port of the Multi Display Module. Now you can connect the "2" output port of the Order Module directly to the "X1" input port of the Multi Display Module since that is how we chose the segment intervals for our plot to be.

Last, you need to specify the "Y1" value for each object. This is most easily done by using the "3" output port of the Order Module (↑13.12, Order) as the "Pos" value for a Selector Module (↑5.1, Selector) where each of the four "channel" input ports has the value corresponding to the "Y2" value of the corresponding graphical object. Such an implementation is shown in the figure above. In that example, the resulting plot would be segments between the points (0, 1), (1, 2), (2, 5), and (3, 0). Connect the output of the Selector Module to the "Y1" input port of the Multi Display Module.

To be able to see all of the plot, set the X Range parameter of the Multi Display Module at its "XR" input port to "3" and the Y Range parameter at the "YR" input port to "5" (since the greatest "Y1" value used here is "5" and the Y Origin parameter is "0"). To make the lines appear in red, connect the Constant (↑4.1, Constant) "1" to the "R" input port and make sure that the Ignore RGB checkbox in the Multi Display Module's Function page has been engaged. The resulting plot done by the Multi Display Module's Panel representation is shown in the figure below.

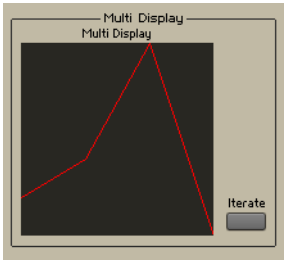


Fig. 1.59 A plot consisting of three line segments done by the Multi Display Module.

1.17 Poly Display

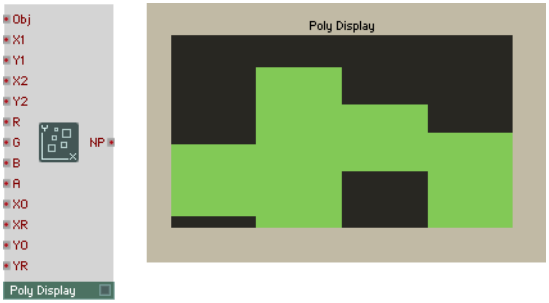


Fig. 1.60 A Poly Display Module

Since this goes for all of the "R", "G", and "B" input ports, all objects have the same color.

1.17.1 Overview

The Poly Display Module enables you to display and manipulate multiple graphical objects. Graphical objects can be chosen to be crosses, bars, rectangles, lines, and (animated) pictures. A range of parameters including an object's type, its coordinates, transparen-

cy, and color can be defined individually for each graphical object. The Properties of the Poly Display Module include a variety of options for customization of the display, including background color and picture. All Poly Display input ports accept Polyphonic Event signals, except for the "XO", "XR", "YO" and "YR" input ports, which accept Monophonic Event signals.
The major difference between the Multi Display and Poly Display Module is how the number of graphical objects is specified. For the Multi Display Module the number of objects is defined by the Number of Objects edit field in its Function page whereas for the Poly Display Module, that number is specified by the number Polyphonic Voices of the parent Instrument. The advantage of the Multi Display Module is that it can display any number of graphical objects, regardless of the number of Polyphonic Voices of the parent Instrument. Each object can then be addressed independently using the "Idx" (index) input port. In contrast, the Poly Display Module could be considered easier to program as it doesn't require any index-processing "code", and all objects can be addressed simultaneously by virtue of parallel Polyphonic Voice processing. However, the number of objects is limited to the number of Polyphonic Voices of the parent Instrument.

Application

When used in conjunction with the Mouse Area (↑1.18, Mouse Area) and Snap Value Array (↑14.27, Snap Value Array) Modules, the Poly Display Module allows sophisticated custom interface elements such as sequencers to be constructed. Use the To Voice (↑14.11, To Voice) and Voice Info (↑14.18, Voice Info) Modules, among others of the same kind, to address and extract information from individual Polyphonic Voices. To increase the number of graphical objects in your Poly Display, you need to increase the number of Polyphonic Voices of the parent Instrument. Please refer to subsection 9.3.2 in the Application Reference to learn how to change the number of Polyphonic Voices of an Instrument.

Please refer to subsection 9.3.4 in the Application Reference for more information on using Polyphonic Voices for parallel processing.

1.17.2 Ports

Input Ports

- **(Obj)** "Obj" (object) is the Polyphonic Event input port to specify the graphical appearance of the Poly Display Module's objects. The number of objects is equal to the number of Polyphonic Voices of the parent Instrument. Each Polyphonic Voice addresses one such object. Fractional "Obj" values are rounded to the nearest integer. Obj = -4 corresponds to a cross. The cross consists of a vertical and horizontal line. When the Center to X1, Y1 checkbox in the Function page is disengaged, the starting and end points for the horizontal line are given by "X1" and "X2", respectively and the starting and end points for the vertical line are given by the "Y1" and "Y2", respectively. Obj = -3 corresponds to a line from the coordinates "X1" and "Y1" to the coordinates "X1" and "Y1" of the object of the same type. Obj = -2 corresponds to a line drawn from the coordinates "X1" and "Y1" to the coordinates "X2" and "Y2" (when the Center to X1, Y1 checkbox in the Function page is disengaged). Obj = -1 corresponds to a bar whose opposite corners are given by the coordinates pairs (X1, Y1) and (X2, Y2) when the Center to X1, Y1 checkbox in the Function page is disengaged. Obj = 0 corresponds to a rectangle whose opposite corners are given by the coordinates pairs (X1, Y1) and (X2, Y2) when the Center to X1, Y1 checkbox in the Function page is disengaged. If you have loaded an animated picture with the Picture drop-down menu in the Function page you can send values in the range [1 … NP] to the "Obj" input port where "NP" is the number of animation frames in the picture. In this case the "Obj" value specifies which animation frame should be displayed at the location specified by the coordinate pair (X1, Y1) (when the Center to X1, Y1 checkbox in the Function page is disengaged). When the Center to X1, Y1 checkbox has been engaged in the Function page, the "X1" and "Y1" coordinates define the the center of the object.
- **(X1)** "X1" is the Event input port for specifying the X1 coordinate.
- **(Y1)** "Y1" is the Event input port for specifying the Y1 coordinate.
- **(X2)** "X2" is the Event input port for specifying the X2 coordinate.
- **(Y2)** "Y2" is the Event input port for specifying the Y2 coordinate.
- **(R)** "R" (red) is the Event input port for specifying the amount of red component in the RGB color of the object addressed by the Polyphonic Voice. The range of values at this input port is [0 … 1]. If this input port is disconnected, the default value, "1", is used for all objects.

- **(G)** "G" (green) is the Event input port for specifying the amount of green component in the RGB color of the object addressed by the Polyphonic Voice. The range of values at this input port is [0 … 1]. If this input port is disconnected, the default value, "1", is used for all objects.
- **(B)** "B" (blue) is the Event input port for specifying the amount of blue component in the RGB color of the object addressed by the Polyphonic Voice. The range of values at this input port is [0 … 1]. If this input port is disconnected, the default value, "1" is used for all objects.
- **(A)** "A" (alpha) is the Event input port for the object transparency. The range of values at this input port is [0 … 1]. A = 0 corresponds to a fully transparent object and A = 1 corresponds to a fully opaque object. If this input port is disconnected, the default value, "1" (fully opaque), is used for all objecs.
- **(XO)** "XO" (X origin) is the Event input port for the lowest visible "X" values. Connecting this input port to a control can be used for scrolling. Values incoming at this input port override the X Origin edit field in the Function page. When disconnected, the value from the X Origin edit field in the Function page is used.
- **(YO)** "YO" (Y origin) is the Event input port for the lowest visible "Y" values. Connecting this input port to a control can be used for scrolling. Values incoming at this input port override the Y Origin edit field in the Function page. When disconnected, the value from the Y Origin edit field in the Function page is used.
- **(XR)** "XR" (X range) is the Event input port for the horizontal viewing range (or zoom). Values incoming at this input port override the X Range edit field in the Function page. When disconnected, the value from the X Range edit field in the Function page is used.
- **(YR)** "YR" (Y range) is the Event input port for the vertical viewing range (or zoom). Values incoming at this input port override the Y Range edit field in the Function page. When disconnected, the value from the Y Range edit field in the Function page is used.

Output Ports

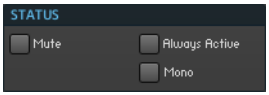
- **(NP)** "NP"(number of pictures) is the Event output port for reporting the number of animation frames in the loaded picture.

1.17.3 Properties: Function Page

Keeping the Poly Display Module Always Active

If the Poly Display Module is not connected to an active signal flow it will not display its internal state in the Panel representation. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Poly Display Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.



► To set the number of objects displayed by the Poly Display Module to "1", that is, to set the Module to Monophonic operation mode, go to its Function page and engage the Mono checkbox (shown in the figure above). In this case the Module will only accept Monophonic signals.

► To define the object location only by its center coordinates (for which the pair of values (X1, Y1) is used), go to the Module's Function page and engage the Center to X1, Y1 checkbox (shown in the screenshot below).

To load the (animated) picture to represent the object for "Obj" (object) values greater or equal to "1":

1. Go to the Module's Function page and open the Picture drop-down menu (shown in the figure above).
2. Now either choose the *-> Open from file...* menu entry to load a new picture file or choose a menu entry corresponding to a picture file that has already been loaded into the Ensemble by another Module.
3. This opens up the Picture Properties dialog window. This window shows the picture preview and lets you specify how REAKTOR should extract the animation frames from the loaded picture. Please refer to section 8.5 in the Application Reference for more information on working with the Picture Properties dialog window. Press the OK button when you are finished setting the picture properties.

Choosing the Origin and Range of Displayed Values

The origin and range of the X and Y coordinates in your Poly Display Module determine where (if a at all) your graphical objects, whose coordinates are set with the "X1", "Y1", "X2", and "Y2" input ports, will appear in the Poly Display. The displayed interval in the X dimension in the Poly Display Module's Panel representation is then [XO … XO + XR] where "XO" is set with the X Origin edit field and "XR" is set with the X Range edit field (shown in the screenshot below). Similarly, the displayed interval in the Y dimension in the Poly Display Module's Panel representation is then [YO … YO + YR] where "YO" is set with the Y Origin edit field and "YR" is set with the Y Range edit field (shown in the screenshot below).

You can also set these values directly from the corresponding input ports. In the case that these input ports are connected, the values in the corresponding edit fields are overridden.

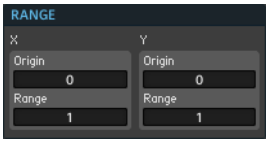


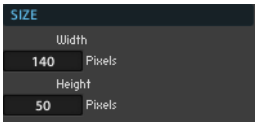
Fig. 1.61 Use the Range area of the Poly Display Module's Function page to set the range of coordinates that are displayed on the Module's Panel representation.

1.17.4 Properties: View Page

Setting the Poly Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for the graphical objects of your Poly Display Module.

► To set the width and height of the Poly Display Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

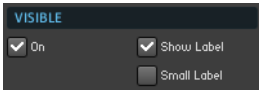


Choosing Label Size and Visibility Settings

► To reduce the size of your Multi Display Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

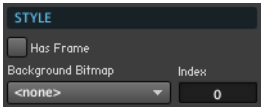
► To make the Multi Display Module's Panel representation disappear, disengage the On checkbox (shown in the figure below).



► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Poly Display Module's display disappear.

Turning on the Poly Display's Frame

► To turn on the frame for the Poly Display Module's Panel representation, engage the Has Frame checkbox in the View page (shown in the screenshot below).



► The Style area of the View page lets you toggle the visibility of the frame with the Has Frame checkbox and choose a background picture with the Background Bitmap drop-down menu and the Index edit field.

Loading a Background Picture for the Multi Display

To load a background picture for the Poly Display Module:

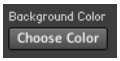
1. Go to the Module's View page and open the Background Bitmap drop-down menu (shown in the figure above).
2. Now either choose the *-> Open from file...* menu entry to load a new picture file or choose a menu entry corresponding to a picture file that has already been loaded into the Ensemble by another Module.
3. This opens up the Picture Properties dialog window. This window shows the picture preview and lets. If you have loaded a picture consisting of animation frames, the Picture Properties dialog window also lets you specify how REAKTOR should extract the

animation frames from the loaded picture. Please refer to section 8.5 in the Application Reference for more information on working with the Picture Properties dialog window. Press the OK button when you are finished setting the picture properties.

4. If you have loaded and configured an animated picture, you can choose the picture index of the animation frame to be used as a background picture with the Index edit field (shown in the screenshot above).

Choosing a Background Color

► To choose a background color for the Poly Display Module's Panel representation using the color picker dialog window of your OS, go to the Poly Display Module's View page. Then press the Choose Color button and use the OS-specific dialog window to choose your color of choice.



Setting Borders for the Poly Display's Panel Representation

Often you have some border for your background picture where you don't necessarily want the graphical objects to appear. With the Picture Borders edit fields (see screenshot below) you can specify the border area where the graphical objects should not appear. By increasing the border either on the left or right, or top or bottom, you increase the effective area of the Poly Display Module's Panel representation such that the area on which the graphical objects are shown corresponds to the dimensions specified with the Width and Height edit fields, as shown above.

► To specify the background picture's top, bottom, left, and right border (in pixels), use the corresponding edit fields in the View page, shown in the screenshot below.

1.17.5 Example: Displaying Pitch Values

Since the Poly Display Module gives you direct access to the values carried by Polyphonic Voices, it is ideal for tasks like displaying the pitch values carried by the Voices, for example. This example shows how to make a simple display that shows the pitch values carried by four Voices.

First, let's decide that we want to have a bar graph of the voice values, as can be seen in the last figure at the end of this example. For this, connect the constant value "-1" to the Poly Display Module's "Obj" input port. We want each bar to have unit width, start at Y1 = 0 and reach up to the value of the corresponding MIDI Note. Since Y1 = 0 for all Voices, just connect the Constant (↑4.1, Constant) "0" to the "Y1" input port. Also, the Note Pitch Module forwards the right MIDI Note values already carried by each Voice, so directly connect the Note Pitch Module (↑2.1, Note Pitch) to the "Y2" input port.
Next, let's send the "X1" and "X2" values. For the four Voices in the Instrument below, the pairs (X1, X2) should be the following (remember we wanted a bar width of "1"): (0, 0 + 1), (1, 1 + 1), (2, 2 + 1), and (3, 3 + 1). In essence, we just need one signal which has the values {0, 1, 2, 3} for its four Voices. This would be the signal for the "X1" input port. Then you just add the Constant "1" and forward the result to the "X2" input port. The Structure that achieves this is shown in the figure below.

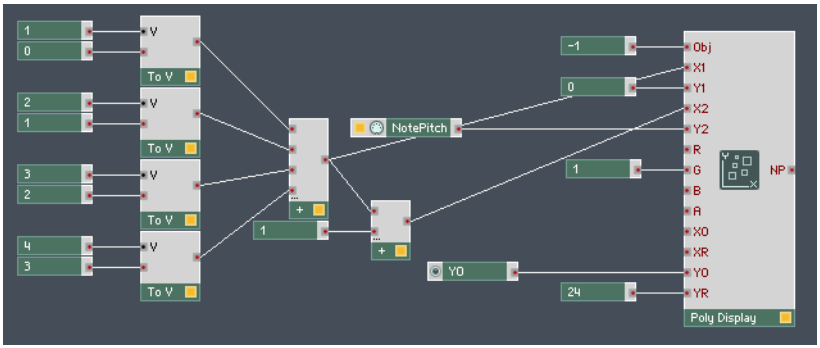


Fig. 1.62 The Structure for a simple Poly Display Structure that displays played MIDI Notes.

To set the values for each Voice separately, use To Voice Modules (↑14.11, To Voice) (although in this case you could achieve the same "X1" values using the Voice Info Module, see also ↑14.18, Voice Info). Place four To Voice Modules into the Structure, one for each Voice. Feed the values "1" to "4" to the "V" (Voice) input ports to designate the Voice which the To Voice Module addresses. The lower input port of each To Voice Module is meant for the value to which the Voice is set. Set the values as shown in the Structure above for each Voice: "0", "1", "2", and "3". Each To Voice Module outputs a Polyphonic signal where each Voice is zero except the Voice which it addresses. Use the Add Module (↑4.2, Add,

+) to sum up the Polyphonic signals from the To Voice Modules, Voice by Voice. This yields the desired signal for the "X1" input port. To get the signal for the "X2" input port, just add "1" and forward that to said port.
Next, to set all bars to have bright green coloring, forward the constant "1" to the "G" input port and make sure that the Ignore RGB checkbox in the Poly Display Module's Function page is engaged. If the Modules in the Structure have not yet been activated, then also engage the Always Active checkbox in the Function page.
One more thing: you need to make sure that the range of displayed values matches the incoming values. The X Origin and X Range values (set in the Poly Display Module's Function page) are "0" and "4", respectively (since we are using four Voices). For the Y Origin and Y Range values, it depends on which keys you are pressing and wish to be displayed. Let's choose the range of displayed MIDI Note values to be two octaves. This corresponds to a MIDI Note interval of "24", so connect a Constant (↑4.1, Constant) carrying this value to the "YR" input port. You might want some flexibility for the starting point of the displayed range of notes, so just create a Knob Module with the range [0 ... 127] and Stepsize "1" at the "YO" input port. The resulting Instrument panel of this example is depicted in the figure below.

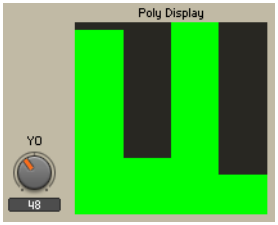


Fig. 1.63 The Poly Display displays played MIDI Notes starting with MIDI Note "48" and ending with MIDI Note "72."

1.18 Mouse Area

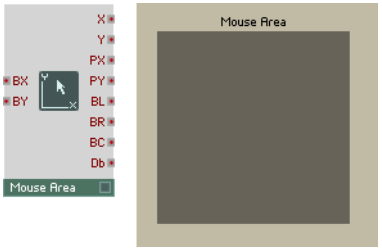


Fig. 1.64 A Mouse Area Module

1.18.1 Overview

The Mouse Area Module detects and outputs mouse actions that take place on its Panel representation. These mouse actions include button clicks, mouse drags, and changes in position. The Mouse Area "X" and "Y" output ports can either operate in Absolute or Incremental Mode (as selected in the Function page):

- If the Incremental Mouse Mode is turned off (Absolute Mode), the absolute mouse position on the Module's Panel representation is translated to values between the "Min" and "Max" values set in the Function page. The values are output to the "X" and "Y" output ports for the horizontal and vertical position, respectively. The display width and height define the range and resolution for mouse movements and the Mouse Reso edit field in the Function page has no effect.
- With the Incremental Mouse Mode turned on, the X and Y positions are also translated to values between the "Min" and "Max" values set in the Function page. However, these values are adjusted incrementally by multiple drags actions, in the same way in which the built-in REAKTOR knobs and faders work. Specifically, when the user begins a new drag, the "X" and "Y" output ports transmit values relative to where the previous drag ended, as opposed to the absolute position of the mouse cursor. The "BX" and "BY" input ports are only effective in Incremental Mode, and are used to set the incremental "base value" for subsequent drags (overriding the last mouse drag). The X and Y mouse resolution is given by the Mouse Reso edit field in the Function page.

In the View page you can change the appearance of the Mouse Area Module's Panel representation. Also, you can control which action causes the Panel representation to change from the "inactive" state to the "active" state.

Application

The Mouse Area Module can have its own outline and fill color, and can therefore be used as a panel interface element by itself. In this case you would use the Incremental Mouse Mode and connect the "BY" (and "BY") input ports to Snap Value Modules (↑14.26, Snap Value) to ensure that the incremental "base value" is set according to the last mouse movement saved in a Snapshot.
Most typically however, the Mouse Area Module used as an invisible (transparent) overlay on top of other Modules. Placing the Mouse Area Module over the Multi Display and Poly Display Modules allows for highly customized interface elements. Learning to creatively use the Mouse Area Module is highly recommended if you want to become good at programming your own REAKTOR GUIs!

1.18.2 Ports

Input Ports

- **(BX)** "BX" (base X) is the Event input port to set a new base value from which the incremental increase in the "X" value is calculated. Values at this input port should be within the range as defined by the edit fields in the Range area of the Function page. The "BX" (base X) value is only used if Incremental Mode has been activated.
- **(BY)** "BY" (base Y) is the Event input port to set a new base value from which the incremental increase in the "Y" value is calculated. Values at this input port should be within the range as defined by the edit fields in the Range area of the Function page. The "BY" (base Y) value is only used if Incremental Mode has been activated.

Output Ports

- **(X)** "X" (mouse X) is the Event output port for the horizontal (X) mouse position when the mouse is within the Mouse Area Module's display area in the Instrument Panel. Events with the current mouse position are only sent when the mouse button is depressed.

- **(Y)** "Y" (mouse Y) is the Event output port for the vertical (Y) mouse position when the mouse is within the Mouse Area Module's display area in the Instrument Panel. Events with the current mouse position are only sent when the mouse button is depressed.
- **(PX)** "PX" (position X) is the Event output port for the horizontal mouse position in pixels relative to the X Origin line which is at the left edge of the Mouse Area box. Moving the mouse to the left of the X Origin line outputs negative pixel values, moving it to the right outputs positive values. In contrast to the "X" output port, which is limited to movements within the Mouse Area box, the "PX" (position X) output port reports mouse positions all the way to the left and right edges of the screen. Note however, that the click-and-drag action must be started within the bounds of the Mouse Area Module's Panel representation.
- **(PY)** "PY" (position Y) is the Event output port for the vertical mouse position in pixels relative to the Y Origin line which is at the bottom edge of the Mouse Area box. Moving the mouse below of the Y Origin line outputs negative pixel values, moving it above that line outputs positive values. In contrast to the "Y" output port, which is limited to movements within the Mouse Area box, the "PX" (position X) output port reports mouse positions all the way to the bottom and top edges of the screen. Note however, that the click-and-drag action must be started within the bounds of the Mouse Area Module's Panel representation.
- **(BL)** "BL" (left button) is the Event output port for the left mouse button status. It is "1" when the left mouse button is depressed; otherwise it is "0". Events are sent from this output port only when the mouse is within the Mouse Area Module's display area in the Instrument Panel.
- **(BR)** "BR" (right button) is the Event output port for the right mouse button status. It is "1" when the right mouse button is depressed; otherwise it is "0". Events are sent from this output port only when the mouse is within the Mouse Area Module's display area in the Instrument Panel.
- **(BC)** "BC" (center button) is the Event output port for the center mouse button status. It is "1" when the center mouse button is depressed; otherwise it is "0". Events are sent from this output port only when the mouse is within the Mouse Area Module's display area in the Instrument Panel.
- **(Db)** "Db" (double click) is the Event output port for the double-click mouse action. An Event with the value "1" is sent when the Mouse Area Module's Panel representation area is double-clicked. After the double-click, no zero-valued Event is sent. Therefore you must probe this output for Events and not static values.

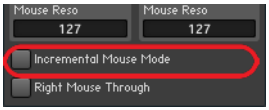
The right mouse button in Windows corresponds to a Ctrl+Click in Mac OS X. The center mouse button is generally not available in Mac OS X.

1.18.3 Properties: Function Page

Activating Incremental Mouse Mode

The output values for the Mouse Area Module are calculated in two different ways, depending on the operation mode. These two modes are discussed in the Module's Overview, above.

► To toggle between the two Mouse Area Module operation modes, engage or disengage the Incremental Mouse Mode checkbox accordingly. This checkbox is shown in the screenshot of the Function page below.



Setting the Range and Resolution for the Output Values

The range and resolution of the Mouse Area Module's values at the "X" and "Y" output ports are set in the Range area of the Module's Function page, shown in the screenshot below. The edit fields contained in this area function analogously to the ones contained in a conventional Knob or Fader Module. To learn how these edit fields affect the output values, please refer to subsection 8.2.1 in the Application Reference.

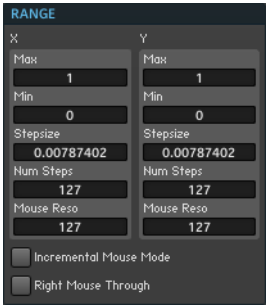
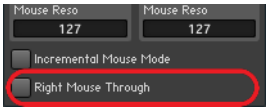


Fig. 1.65 The range and resolution of the Mouse Area Module's output values are set in the Range area of its Function page.

Engaging the Right Mouse Through Mode

► To keep the actions of the right mouse button from triggering Events at the Mouse Area Module's output and instead bring up the context menu, go to the Function page and engage the Right Mouse Through checkbox, as shown in the figure below.



The right mouse button in Windows corresponds to a Ctrl+Click in Mac OS X.

1.18.4 Properties: View Page

Setting the Mouse Area Size

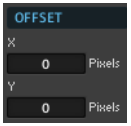
Depending on your Instrument Panel layout, you might want a larger or smaller Panel representation for your Mouse Area Module.

► To set the width and height of the Mouse Area Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

Fine Tuning the Mouse Area Placement in the Panel

The position of the Mouse Area on the Instrument Panel can be offset from the REAKTOR 4 x 4 grid.

► To shift the Mouse Area Module's Panel representation to the right or upwards by 1 to 3 pixels, enter the desired value into the X Offset or Y Offset edit field, respectively.

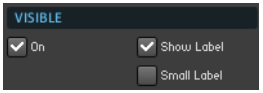


Choosing Label Size and Visibility Settings

► To reduce the size of your Mouse Area Module's Label, engage the Small Label checkbox (shown in the figure below).

► To make the Label disappear, disengage the Show Label checkbox (shown in the figure below).

► To make the Mouse Area Module's Panel representation disappear, disengage the On checkbox (shown in the figure below).

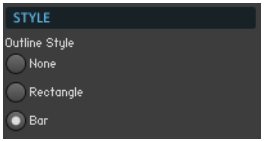


► Engage the Show Label checkbox to show the Label. Engage the Small Label checkbox to reduce the size of the Label of the Module's Panel representation. Disengage the On checkbox to make the Mouse Area Module's display disappear.

Changing the Outline Style

You can choose the Mouse Area's Panel representation to have three styles: None (invisible), Rectangle (1 pixel border surrounding the Mouse Area), or Bar (the Mouse Area is filled with the outline color).

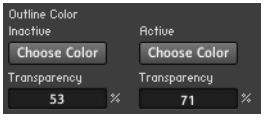
► To choose the style for the Mouse Area's Panel representation, go to the Module's View page and click on the corresponding radio button from the Outline Style Selector (shown in the screenshot below).



Choosing the Outline Color

The Mouse Area Module's Panel representation has two states: Active and Inactive. For each state you can define a color and the amount of transparency.

► To choose a color for the Inactive or Active state of the Mouse Area with the color picker of your OS, press the corresponding Choose Color button, as shown in the screenshot below.



► To set the amount of transparency for the Active and Inactive states, enter the desired value (in percent) into the Active Transparency and Inactive Transparency edit fields, respectively. These edit fields are shown in the screenshot above.

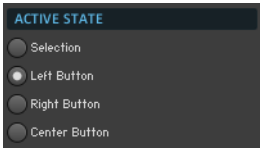
Defining the Active State

The Mouse Area Module lets you choose which mouse action turns on the Active state and how this state behaves. You have four options:

- **Selection:** Clicking on the Mouse Area with the left mouse button turns on the Active state. This state is kept (even if the mouse button is released) until something else on the Instrument Panel is selected.
- **Left Button:** Clicking on the Mouse Area with the left mouse button turns on the Active state as long as you keep the mouse button depressed.
- **Right Button:** Clicking on the Mouse Area with the right mouse button turns on the Active state as long as you keep the mouse button depressed.
- **Center Button:** Clicking on the Mouse Area with the center mouse button turns on the Active state as long as you keep the mouse button depressed.

The right mouse button in Windows corresponds to a Ctrl+Click in Mac OS X. The center mouse button is generally not available in Mac OS X.

► To define the behavior of your Mouse Area Module's Active state, go to its View page and click on the corresponding radio button from the Active State selector (shown in the figure below).



1.18.5 Example: Value Edit Field

This example shows how to build a custom value edit field that displays numbers with one decimal point precision. The edit field also enables you to change the values by clicking and dragging the mouse over the edit field and to reset the value to "0" by double clicking on the field. Start by creating a Macro with an output port labeled "Out". Make sure the Macro is set to Mononophonic Mode before you start building this example Structure. First we will start with building the value display. We want to display decimal numbers from -9.9 to 9.9, in 0.1 step increments. For this purpose you will need three Multi Picture Modules (↑1.11, Multi Picture): one for the sign (+ and -), one for the integers (0 to 9), and one for the first decimal place after zero (.0 to .9).

After inserting the three Multi Picture Modules into the Structure, designate one for the sign and go to its Function page. There, in the Select Picture menu you will find the *-> Open from file...* menu entry, click on it. This opens the Load Image File dialog box. Go to the *Tutorial Ensembles* folder and from there enter the folder labeled *Module Reference* where you will find the file "plus_minus.tga". Select the file and click the Open button. In the following Picture Properties dialog window, make sure that the Alpha Channel checkbox is activated. Next, make sure that the Animation Width and Animation Height edit fields both carry the value "32". You should now see a "+" and a "-" with transparent backgrounds next to each other in the Picture Preview display. If so, click the OK button to continue.

Fig. 1.66 The Picture Properties dialog window.

Now follow this same procedure for loading the digit animations to the other the Multi Picture Modules. Designate one of the Modules for integer digits and load the "numbers.tga" file from the same location as "plus_minus.tga" into the Multi Picture Module whilst making sure that the same settings for the Alpha Channel checkbox and Animation Height and Animation Width edit fields apply. For the third Multi Picture Module, repeat the aforementioned procedure, but this time loading the file "decimals.tga" into the Module. This last Module will be dedicated to display the first position after the decimal point. Go to the Instrument Panel and arrange the signs and digits in a way that makes sense when reading from left to right: first the sign, then the integer, and then the decimal fraction.

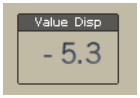


Fig. 1.67 The display part of the value edit field.

Now we need a way to split up the information that the value display should display into three parts: the sign (if the value is positive or negative), the integer part, and the decimal fraction. The first split will happen between the sign and the numbers. For that, insert a Rectify/Sign Module into the structure. The incoming value to the value display goes into the "In" input port of the Rectify/Sign Module. The "Sign" output port of that Module outputs a "1" if the incoming value is positive and "-1" if the incoming value is negative. We need to convert these values to specify the picture index of the animation loaded into the Multi Picuture Module. For that purpose, add an Add Module into the structure and use it to add "1" to the signal from the "Sign" output port of the Rectify/Sign Modue. The output of the Add Module will then be "0" if the value at the input of the Rectify/Sign Module is negative and "2" if the value is positive. After connecting the output port of the Add Module to the "Sel" input port of the Multi Picture Module, the "0" will address the first picture in the animation, the minus sign, and the value "2" will address the last picture of the animation, the plus sign. To test out the functionality of the value display, create a Knob Module at the "In" input of the Rectify/Sign Module by right-clicking on the "In" input port and clicking on the *Create Control* menu item. Set the range of the knob to [-9.9 ... 9.9] by going to the Module's Function page and setting the Max Value edit field to "9.9" and the Min Value edit field to "-9.9".

On to the numbers: insert two Modulo Modules (↑4.9, Modulo) into the structure. Connect the "|x|" output port of the Rectify/Sign Module the "A" input port of one Modulo Module. Since we want to split the number into integers and fractions, it is only natural to create a Constant with the value "1" at the "B" input port of the first Modulo Module. Now the "Div" output port of the Modulo Module carries the value that is the largest multiple of "1" and is at the same time smaller than the value received from the "|x|" output port of the Rectify/Sign Module. For example, if |x| = 5.2, then Div = 5 or if |x| = 3.8, then Div = 3. Connect the "Div" output port to the "Sel" input port of the Multi Picture Module that is meant to display the integer part of the incoming signal value. Since we want to display signals within the range -9.9 to 9.9, then the "Div" value has a range [0 ... 9] which exactly matches the indices of the pictures of the animation file "numbers.tga" loaded into that Multi Picture Module. Go to the Instrument Panel and move the knob that you previously

created and watch the sign and the integer numbers change in the Panel display of the Multi Picture Modules. Next, hook up the decimal fraction part of the incoming signal, namely the value sent to the "Mod" output of the Modulo Module, to the "A" input port of the second Modulo Module. Next, create a Constant with the value "0.1" at the "B" input port of that Modulo Module. Wire the "Div" output port that same Modulo Module to the "Sel" input of the last remaining Multi Picture Module. With the "B" input port of the first Modulo Module receiving a constant value of "0.1", the "Mod" output port of the Modulo Module outputs the remainder of the incoming value, divided by "0.1". To demonstrate this with the numbers used in the previous example, if |x| = 5.2, then for the first Modulo Module, Mod = 0.2 and sending that value to the "A" input of the second Modulo Module, we get 0.2 / 0.1 = 2 at the "Div" output of the second Modulo Module. In the same way, if |x| = 3.8, then for the first Modulo Module, Mod = 0.8 and when that value is sent to the "A" input port of the second Modulo Module, the "Div" output of that Module is 0.8 / 0.1 = 8. Note that since the range of the "Mod" output of the first Modulo Module is [0 … 0.9] and in the second Modulo Module we divide by "0.1", the range of the "Div" output of the second Modulo Module is [0 … 9]. In this case "Mod" has a range [0 … 9] which exactly matches the indices of the pictures of the animation file "decimals.tga" loaded into that Multi Picture Module. Go to the Instrument Panel and move the previously created knob to test out the functionality of the value display.

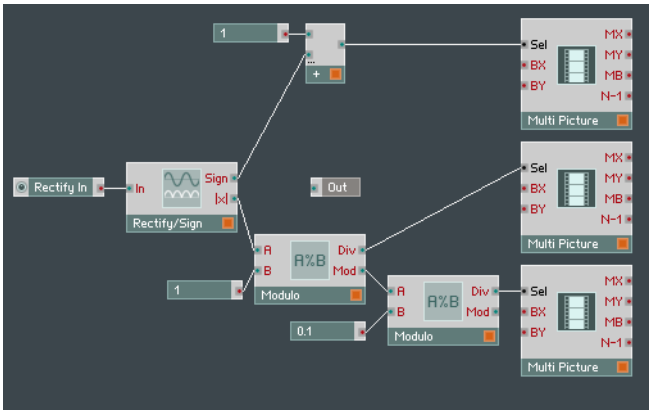


Fig. 1.68 The Structure for sorting out the sign and decimal parts of the value.

Now that you have a functioning value display, you probably want a way to manipulate the values directly through the value display itself, not with an extra knob cluttering your Panel layout. This can conveniently be achieved with the Mouse Area Module. Place it into the structure in front of the Rectify/Sign Module, leaving a bit of space for a Merge Module that you will insert later. By default, the Mouse Area Module is transparent in the Instrument Panel. To make testing the functionality easier you should make the mouse area visible. First, the Mouse Area Module needs to have a graphic element in the Instrument Panel. In the Appearance page of the Module, click on the Bar Outline Style selector to create a filled rectangle as the graphical element of the mouse area in the Instrument Panel. Now you have a 100 % transparent rectangle marking the mouse area in the Instrument Panel; for ease of manipulation you should to make the appearance more opaque. To do this, enter the value "80" into the Inactive Transparency edit field and the value "70" into the Active Transparency edit field of the Module's Appearance page. To add one more touch, click on the Left Button selector to make the active state of the mouse area appear during a left-click on the mouse area in the Instrument Panel. Now you should be able to see the mouse area in the Instrument Panel and see it become darker when you click on it. Unlock the Panel and move the mouse area over your value display. Set the dimensions of the mouse area to cover all digits of the value display using the Size X and Size Y edit fields in the Appearance page of the Mouse Area Module. After finding good settings for the dimensions, ("30" for Size X and "60" for Size Y should be good) lock the Panel again.

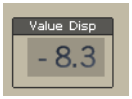


Fig. 1.69 The value edit field with the mouse area.

The Mouse Area Module can supply us with the values that we want to display with the structure that you built above. When you click and drag the mouse over the mouse area in the Instrument Panel, the "X" and "Y" output ports output events carrying the "x" and "y" coordinates, respectively, as specified in the Function page of the Module. For our purpose, we are only interested in clicking and dragging the mouse vertically to change the values in the value display since that seems the most intuitive setup for most Instrument layouts. For this reason we will leave the "X" output (and hence the settings in the Range X area of the Function page) alone and focus on the "y" coordinate. Connect the "Y" output port of the Mouse Area Module to the "In" input port of the Rectify/Sign Module. In order to be able to control the output values of the "X" and "Y" output ports of the Mouse Area

Module over the mouse area in the Instrument Panel much like a fader, that is, that clicking and dragging changes the output value, engage the Incremental Mouse checkbox in the Function page of the Mouse Area Module.
The range of values that we want to get out of the Mouse Area Module is [-9.9 ... 9.9]. To specify these parameters, go to the Function page of the Mouse Area Module and enter "-9.9" into the Min Y Value edit field and "9.9" into the Max Y Value edit field in the Range Y area. Next, we want to determine the precision with which one can manipulate the output values. The relevant parameters to this are edited with the Num Steps X and Mouse Reso X edit fields. If Num Steps X and Mouse Reso X are too large, you would have to drag the mouse all the way to the top of the screen to see any change in the output value. On the other hand, if these parameters are too small, dragging the mouse over the mouse area just a small distance already effects a great change in the output value. In order to have the right value increments at the output, the Num Steps X value should be an integer multiple of the Mouse Reso X value. Try playing around with the values and seeing the effect it has on the value display, but in the end settling with a value of "1000" in the Num Steps X edit field and "500" in the Mouse Reso X edit field should be fine.
You now have a structure that displays values and lets you directly manipulate the values by clicking and dragging the mouse over the display. We would like to add one additional feature to this setup, namely, the possibility to double-click the mouse area to reset the value to a certain number. The Mouse Area Module has an output port labeled "Db" that sends out an event with the value "1" every time the mouse area on the Panel is double-clicked. You now have two sources of events to determine the number that your value edit field is supposed to display. To combine them, place a Merge Module into the structure between the Mouse Area Module and the Rectify/Sign Module. Replace the wire to the "In" input port of the Rectify/Sign Module with a wire from the "Out" output port of the Merge Module. Next, connect the "Y" output port of the Mouse Area Module to the input port of the Merge Module and finally, while holding Ctrl, drag a wire from the "Db" output of the Mouse Area Module to the three dots on the left side of the Merge Module. The latter action creates a second port for the Merge Module. When you click and drag the mouse over the mouse area in the Instrument Panel, a value is sent to the Merge Module from the "Y" output port of the Mouse Area Module. This value is then forwarded in the form of an event (carrying the same value as was received from the "Y" output port) to the value display part of your structure which starts with the Rectify/Sign Module. If you now double click the mouse area in the Instrument Panel, the value "1" is sent to the Merge Module.

Now the output port of the Merge Module sends an event with the value "1". This should be reflected in the Instrument Panel where the value display should display "+1.0". Each new event at any of the input ports of the Merge Module triggers an event at the output port with the same value as the last incoming event. During initialization, the value from the lowest input port of the Merge Module is sent last to its output port so in the current structure, if you press the Run/Stop Audio button twice (turning the audio off and then on again) to reinitialize the structure, your value display will show the initialization value coming from the "Db" output port of the Mouse Area Module, namely "1". Depending on the circumstances, this might not be what you want, so changing the port order of the wires connected to the Merge Module can be a way to get the desired reinitiation behavior. In this case switching the port orders around will yield the initialization value "0" from the "Y" port of the Mouse Area Module to be sent to the Merge Module's output last.
Regardless of the port order of the wires at the Merge Module's input, you might not want the value edit field to be reset to "+1.0" when you double click the display. Having a reset-behavior of "0" might be much more natural. This can be changed very easily with the help of only one additional Module. Insert a Value Module into the structure and connect the "Db" output port of the Mouse Area Module to the "Trig" input port. Any event at the "Trig" input port will cause an event with the value lying at the "In" input port of the Value Module to be sent from its output port. In this case we leave the "In" input port disconnected which means that an event from the "Db" output port of the Mouse Area Module causes an event with the value "0" to be sent from the output of the Value Module. This is exactly what you want: you replaced an event carrying the value "1" with an even carrying the value "0"! Now replace the wire at the input port of the Merge Module that is connected to the "Db" output port of the Mouse Area Module with a wire to the output port of the Value Module. Now go to the Instrument Panel, make sure the value display is nonzero and double click it. Voila! The value display should now reset to the value "0".
Your value edit field is almost complete. You are probably asking, “Why almost and not fully complete??” The reason lies yet again in the initialization behavior. If you reinitialize the structure or load the Instrument anew, regardless of the value that the display was set to, it will always show "0". If you want to use this structure in an Instrument, it is desirable to have the Instrument's state upon saving fully recalled. Knob, fader, and value display settings count among the things you usually do not want to change when you reload an Instrument. This is where the Snap Value Module comes into play. Insert one into the structure and connect its input port to the output port of the Merge Module. The output port of the Snap Value Module should now be connected to the input port of the Rectify/

Sign Module and the "Out" Out Port that you added to your Macro's structure in the very beginning. The Snap Value Module takes the last value that was received at its input port and stores it in memory. When the structure is reinitialized or the Instrument is reloaded, this value will be sent to the output port of the Snap Value Module (and on to the structure downstream) instead of the initialization values from the Mouse Area Module. To ensure that the Snap Value Module properly deals with the values that it is supposed to store, go to its Function page and set the Max Value edit field to "9.9" and the Min Value edit field to "-9.9". This range is the one we have been using in this example. Also, to retain the functionality that we have achieved until now, that is, that the values from the Mouse Area Module are sent to the rest of the structure (and now, to the Out Port), make sure that the Event Thru checkbox in the Function page of the Snap Value Module is engaged. If the Event Thru checkbox would be disengaged, the values arriving at the Snap Value Module's input port would not be forwarded to the output port except during initialization. Your complete structure should now look like the following picture.

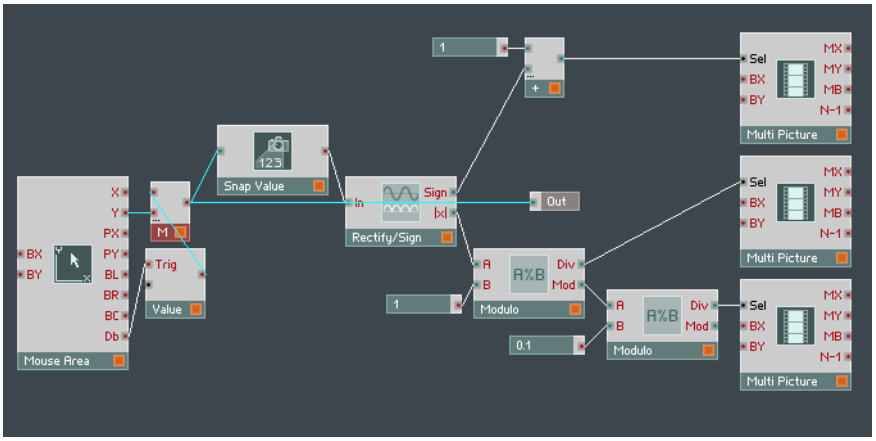


Fig. 1.70 The final Structure of the value edit field.

1.19 Stacked Macro

Fig. 1.71 A Stacked Macro

1.19.1 Overview

The Stacked Macro enables multiple graphical objects to share the same area of the Instrument Panel. Which objects are displayed within the area at any one time can then be controlled using the Panel Index Module (↑1.20, Panel Index Module).

Application

After inserting a Stacked Macro into your Structure, place it in the correct position on the panel, and set the desired size in the properties. Then insert 2 or more Macros ("normal" Macros, not additional Stacked Macros) inside the Stacked Macro, along with a Panel Index Module (↑1.20, Panel Index Module). Only one of the "normal" Macros (along with any panel elements inside it) will be visible at any one time. Which Macro is visible depends on the input value at the Panel Index Module's input port. Please refer to the Panel Index Module for an example on how to use the Stacked Macro feature.

To discover the index number of a Macro, first unlock the Panel by clicking on the Panel Lock button. Then right-click on the Stacked Macro frame on the Instrument panel - the list of Macros and their corresponding indices will be displayed in the context menu.

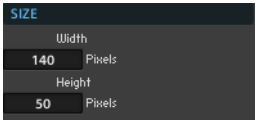
1.19.2 Properties: View Page

Setting the Stacked Macro Display Size

Depending on your Instrument Panel layout, you might want a larger or smaller display area for your Stacked Macro.

Note that Macros inside the Stacked Macro which have a larger display area than that set in the Stacked Macro's View page will appear "clipped" on the Instrument Panel.

► To set the width and height of the Stacked Macro, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



Other than the feature of switching between displayed Macros on a fixed area, the Stacked Macro functions analogously to the "normal" Macro. Please refer to section 8.4 in the Application Reference for more information on the View page of the Stacked Macro.

1.20 Panel Index Module

Fig. 1.72A Panel Index Module

1.20.1 Overview

The Panel Index Module is a key component in activating the functionality of Stacked Macros (↑1.19, Stacked Macro). Into each Stacked Macro exactly one Panel Index Module needs to be placed. Additionally, this Panel Index Module needs to be active. The Panel Index Module receives at its input port values in the range [0 ... N - 1] (where "N" denotes the number of Macros inside the Stacked Macro) which determine which Macro should appear within the Stacked Macro's area in the Instrument Panel.

Application

The Stacked Macro feature (which includes the Panel Index Module) enables multiple graphical objects to share the same area of the Instrument Panel.

1.20.2 Ports

- **(In)** "In" is the Event input port which receives values that determine which Macro should appear within the Stacked Macro's area in the Instrument Panel. The range of these values should be [0 ... N - 1] (where "N" denotes the number of Macros inside the Stacked Macro, see also ↑1.19, Stacked Macro).

1.20.3 Properties: Function Page

Keeping the Panel Index Module Always Active

If the Panel Index Module is not connected to an active signal flow, then the Stacked Macro Functionality is disabled. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Panel Index Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

1.20.4 Example: 2 Stacked Macros

This example shows how to place 2 Macros in the Structure that one the Instrument Panel appear in the same area. Which Macro is displayed, can be chosen using the List Module. First, however, place a Stacked Macro into the Structure, as shown in the figure below, and then navigate into the Macro by double-clicking it.

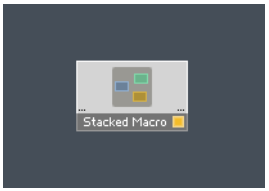


Fig. 1.73 Macros placed inside the Stacked Macro can be displayed one at a time using the Panel Index Module.

Next, place two Macros into the Structure, naming them "Macro 1" and "Macro 2", corresponding to the order in which they were placed inside the Structure. Now place a Panel Index Module into the Structure. Engage the Always Active checkbox in its Function page and connect it to a List Module (↑1.3, List), for example, to be able to address from the Instrument Panel, the contents of which Macro is to be displayed in the area of the Stacked Macro Module's (↑1.19, Stacked Macro) Panel representation. This is shown in the Structure below. Sending the value "1" from the List Module to the Panel Index Module causes the first Macro to be displayed, and sending the value "2" causes the second Macro to be displayed.

Fig. 1.74 The Macros receive their "Panel Index" according to the order in which they were placed inside the Stacked Macro.

Now place some contents with Panel representations inside the two Macros so that you can test out the Stacked Macro feature from the Instrument Panel. The resulting Instrument Panel is shown in the figure below, where a Lamp Module (↑1.5, Lamp) was placed inside the first Macro.

To increase the area of the Stacked Macro (↑1.19, Stacked Macro), go to its View page and increase the values in the Width and Height edit fields.

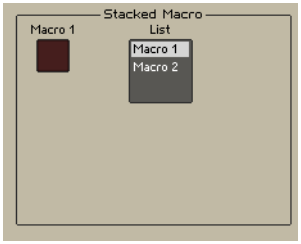


Fig. 1.75 Choosing the "Macro 1" entry from the List Module causes the value "1" to be sent to the Panel Index Module. This displays the contents of the first Macro.

2 MIDI In

MIDI In Modules route MIDI messages from external MIDI devices (and other Instruments) REAKTOR from external MIDI devices. There are separate Modules for various MIDI data types including MIDI notes, note velocity, Pitchbend, Mono and Poly Aftertouch, controllers, program changes, and MIDI clock. REAKTOR also generates separate gate messages for MIDI note events and there are Modules for that. Some MIDI In Modules can also be used in an IC or OSC connection.

2.1 Note Pitch

Fig. 2.1 A Note Pitch Module

2.1.1 Overview

The Note Pitch Module is a Polyphonic Event source for the pitch value of MIDI Note On events. Each Polyphonic Voice that has received a Note On Event from the parent Instrument's MIDI In channel carries the pitch value of that note in the MIDI pitch scale. The range of the output values of this Module is set Function page. The resolution of the output values is such that 128 steps fit in this interval. A MIDI Note Off event has no effect on the output of the Note Pitch Module.

Application

Use the Note Pitch Module to inform your Oscillator, Sampler, and other Modules which keys have been pressed on your MIDI or computer keyboard. Usually you connect the output of the Note Pitch Module to the "P" (pitch) input ports of the aforementioned Modules. When using the default range of Min = 0 to Max = 127 and controlling an oscillator's "P" (pitch) input port you will play in the common equal tempered tuning. One unit corresponds to one semitone and middle C is at 60. By setting "Min" and "Max" in the Function page to different values, pitch can be skewed or scaled, e.g. for playing in quarter tones.

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

2.1.2 Ports

- **(Out)** "Out" is the Polyphonic Event output port for the MIDI pitch values of the incoming MIDI notes.

2.1.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is scaled to the range [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.2 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Note Pitch Module's Function page.

Setting the Range of Input Values

The range of incoming MIDI notes over which the output range [Min ... Max] is scaled, can also be limited with the Lower Limit and Upper Limit edit fields. These edit fields are shown above in the screenshot of the Function page. The output values of the Note Pitch Module are limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

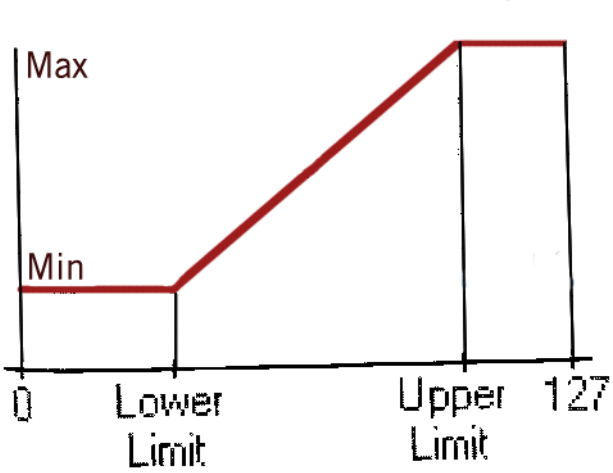


Fig. 2.3 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing values from the Note Pitch Module. The values are interpolated linearly between “Min” and “Max.”

The values for the Lower Limit and Upper Limit edit fields lie in the range [0 ... 127]. Also, the value for Upper Limit edit field must be greater than that set in the Lower Limit edit field. However, the value in the Max edit field can be smaller than the value in the Min edit field to achieve inverted operation. If opposite characteristics are set for two sources, a crossfade effect can be programmed (see below).

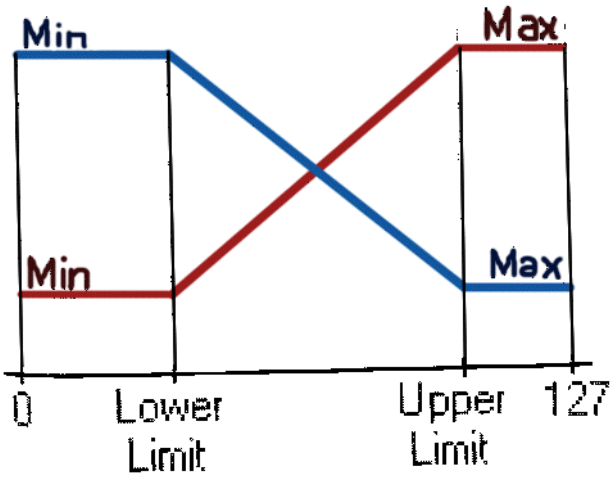


Fig. 2.4 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing values from the Note Pitch Module. Setting opposite characteristics for two sources creates a crossfade effect between them, going from low to high MIDI note values.

A switch with an adjustable threshold level can be emulated by setting the Lower Limit and Upper Limit edit fields to neighboring MIDI values, for example, “63” and “64”. When the input value to such a source is “64”, the value in the Min edit field will be output. Otherwise, the output value is the value in the Max edit field.

2.1.4 Example: Sampler with Trigger Button

The simplest sampler Instrument can be built using the Sampler Module (↑7.1, Sampler)which receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its trigger signal from a Button Module (↑1.2, Button). The pitch values of incoming MIID Notes are forwarded from the Note Pitch Module's output to the Sampler Module's "P" (pitch) input port. The Sampler Module's amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Ampl", as shown in the Structure below. Note that you can create a Knob Module (or sometimes a Button or Fader Module) for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry. Make sure that the Button at the "Trig" (trigger) input port has been set to Trigger Mode in its Function page.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into the Sampler Module's Sample Map.

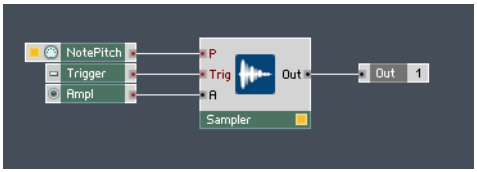


Fig. 2.5 The Structure for a simple sampler Instrument.

2.2 Pitchbend

Fig. 2.6 A Pitchbend Module

2.2.1 Overview

The Pitchbend Module is a Monophonic Event source for MIDI Pitchbend (pitchwheel change) events. The range of the output values of this Module is set Function page. The resolution of the output values is such that 16384 steps fit in this interval. When the pitchbend controller is in its neutral position, the output value is always "0".

Application

The Pitchbend Module lets you program your Instrument in such a way that it may recognize incoming MIDI messages that stem from your MIDI controller's pitch wheel. Add the output of this Module to the pitch value received from a Note Pitch Module and feed it, for example, to an oscillator's "P" (pitch) input port. The range of the output values for upward or downward pitchbend can be set independently in the Function page. This means that with a range of Min = -1 to Max = 1 you can then change the pitch with the pitchbend controller up or down one semitone. On the other hand, a range defined by Min = -12 and Max = 12 means you can pitchbend ±12 semitones which corresponds to ±1 octave.

2.2.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for the pitchbend values that are in the range [Min … Max], set in the Module's Function page.

2.2.3 Properties: Function Page

The output values of the Pitchbend Module are scaled to the range [Min … Max].

► To set the "Min" and "Max" values, to the Module's Function page and enter the desired values into the Min and Max edit fields, respectively.

2.2.4 Example: Note Pitch with Pitchbend

For a simple implementation of the Pitchbend parameter, refer to the Structure shown in the figure below. The "Min" and "Max" values for the Pitchbend Module have been set to "-2" and "2", respectively. As a result, the outgoing pitch values can be "bent" up to 2 semitones above or below the incoming pitch from the Note Pitch Module (↑2.1, Note Pitch).

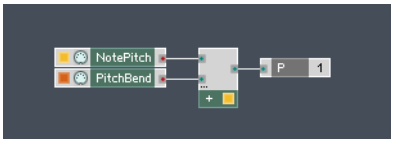


Fig. 2.7 A simple implementation of MIDI Pitchbend.

2.3 Gate

Fig. 2.8 A Gate Module

2.3.1 Overview

The Gate Module is a Polyphonic Event source for MIDI Note On and Note Off events. A Note On event sets the output to a value determined by the key velocity. The range of the output values to which the incoming range of MIDI velocities (from no velocity (Note Off) to maximum velocity) is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min …. Max]. When a Note Off event arrives at the parent Instrument's MIDI In, the Gate Mod-

ule sends an Event with the value "0" (a Gate Off Event), independent of the "Min" and "Max" settings in the Function page. If velocity sensitivity of your Instrument is to be disabled, the "Min" and "Max" values should be set to the same value.

Application

The Gate Module is a key Module in every synthesizer that reacts to incoming MIDI keyboard signals. In the simplest implementation, the Gate Module is connected to the "G" (gate) input port of an Envelope Module. The output of the Envelope Module is then multiplied with the oscillator source signal which is then sent to the output, perhaps after further signal processing with filters and effects. In such a case pressing a key on your MIDI or Computer keyboard triggers the envelope, "letting through" the oscillator signal to the next part of the Structure. Releasing the key releases the envelope, "closing" the path of the oscillator signal to the next part of the Structure. The Gate Module is also used in sampler instruments to trigger Sampler Modules.

2.3.2 Ports

- **(Out)** "Out" is the Polyphonic Event output port for the Gate On and Gate Off Events. The Gate On Events carry the value of the velocity with which the key was pressed. These values are scaled to fit the range [Min … Max] set in the Function page. Gate Off Events carry the value "0."

2.3.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min … Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.9 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Gate Module's Function page.

Setting the Range of Input Values

The range of incoming velocities to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The velocity output value of the Gate Module is limited to "Min" for incoming MIDI values below "Lower Limit" and to "Max" for incoming MIDI values above "Upper Limit". The range between the two limits is interpolated linearly between "Min" and "Max" (see below).

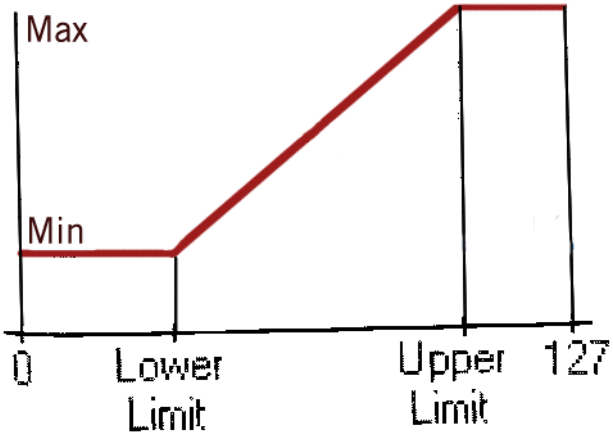


Fig. 2.10 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing velocity values from the Gate Module. The values are interpolated linearly between "Min" and "Max".

The values for the Lower Limit and Upper Limit edit fields lie in the range [0 ... 127]. Also, the value for Upper Limit edit field must be greater than that set in the Lower Limit edit field. However, the value in the Max edit field can be smaller than the value in the Min edit field to achieve inverted operation. If opposite characteristics are set for two sources, a crossfade effect can be programmed (see below).

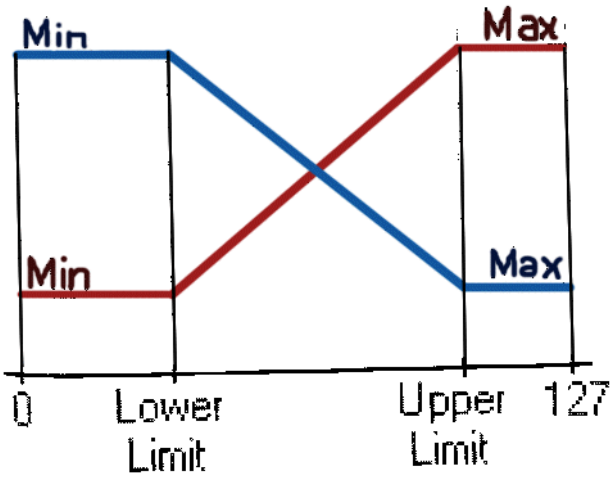


Fig. 2.11 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing velocity values from the Gate Module. Setting opposite characteristics for two sources creates a crossfade effect between them, going from low to high MIDI velocity values.

A switch with an adjustable threshold level can be emulated by setting the Lower Limit and Upper Limit edit fields to neighboring MIDI velocity values, for example, “63” and “64”. When the input value to such a source is “64”, the value in the Min edit field will be output. Otherwise, the output value is the value in the Max edit field.

2.3.4 Example: Loop Sampler

This example illustrates a sample Instrument with frequency modulation and variable (loop) starting position and loop length. As can be seen in the figure below, the Sampler Loop Module (↑7.3, Sampler Loop) has been used. It receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its trigger signal from the Gate Module. When a key on your computer or MIDI keyboard is pressed, a Gate On signal is sent from the Gate Module, carried by the Polyphonic Voice assigned to the key. As a result sample playback is started for this Voice until the key is released. When the key is released, a Gate Off signal is sent from the Gate Module and stops the sample playback (at the "G" (gate) input port. Its amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Ampl", as shown in the Structure below. Note that you can

create a Knob Module (or sometimes a Button or Fader Module) for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry.
A sine oscillator has been used as the frequency modulation signal. The pitch and amplitude of this signal can be controlled at the corresponding input ports of its source, the Sine Module (↑6.14, Sine Oscillator). The starting position of sample playback, starting position of the loop and loop length can also be controlled using Knob Modules (↑1.1, Fader/Knob) labeled "St", "LS", and "LL", respectively. All three knobs have the range [0 … 1]. This range is multiplied by the length of the currently loaded sample, retrieved from the "Len" (length) output port, and is then forwarded to the corresponding input port. This way the three parameters at the input ports are always in the range of the currently loaded sample.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into a Sampler Module's Sample Map.

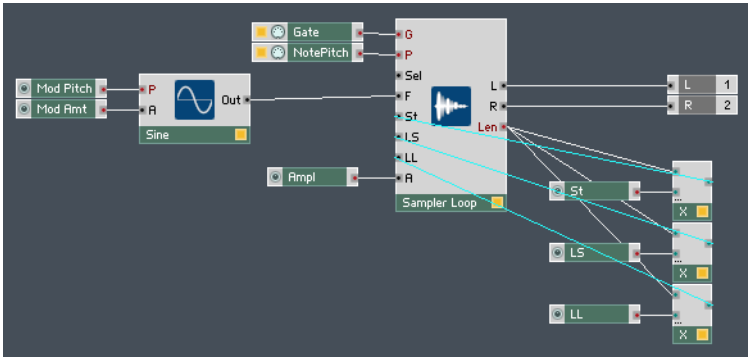


Fig. 2.12 The Structure for a frequency modulated loop sampler Instrument.

2.4 Single Trig Gate

Fig. 2.13 Single Trig Gate Module

2.4.1 Overview

The Single Trig Gate Module is a Monophonic Event source for MIDI Note On and Note Off events. A Note On event sets the output to a value determined by the key velocity. The range of the output values to which the incoming range of MIDI velocities (from no velocity (Note Off) to maximum velocity) is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min .... Max]. When all keys on the MIDI or computer keyboard are released, the Note Off event from the last key release causes the Single Trig Gate Module to send an Event with the value "0" (a Gate Off Event) from its output port, independent of the "Min" and "Max" settings in the Function page. If velocity sensitivity of your Instrument is to be disabled, the "Min" and "Max" values should be set to the same value.
The difference between this Module and the Gate Module (↑2.3, Gate) is that for the Single Trig Module only the first note produces an Event and thus triggers (or retriggers) a connected envelope, sample, or similar. A new note that is started while the previous one is still held (legato play) does not generate an Event at this Module's output port and therefore does not retrigger any envelopes, samplers, or similar.

Application

The Single Trig Gate Module should be applied in the same way as the Gate Module (↑2.3, Gate). However, use the Single Trig Gate Module for your Gate signals if you want to use the legato style of playing. Most synthesizers offer different modes of playing within their networks. This means that when you choose the "legato play" option from the Instrument Panel, most likely the internal Structure of the Instrument is rerouted such that Gate signals are received from the Single Trig Gate Module, instead of the Gate Module (↑2.3, Gate).

2.4.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for the Gate On and Gate Off Events. These Events are only sent when no other key is pressed. The Gate On events carry the value of the velocity with which the new key was pressed. The values are scaled to fit the range [Min ... Max] set in the Function page. The Gate Off Events carry the value "0".

2.4.3 Properties: Function Page

Setting the Range of Output Values

The range of the velocity output signal is scaled to the interval [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.14 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Single Trig Gate Module's Function page.

Setting the Range of Input Values

The range of incoming velocities to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The output velocity value of the Single Trig Gate Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

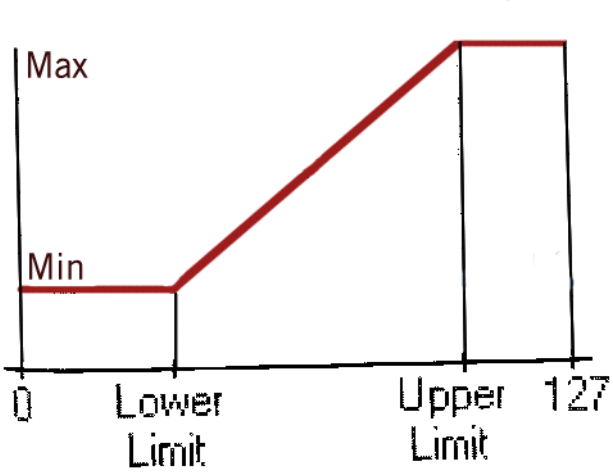


Fig. 2.15 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing velocity values from the Single Trig Gate Module. The values are interpolated linearly between “Min” and “Max.”

The velocity values for the Lower Limit and Upper Limit edit fields lie in the range [0 ... 127]. Also, the value for Upper Limit edit field must be greater than that set in the Lower Limit edit field. However, the value in the Max edit field can be smaller than the value in the Min edit field to achieve inverted operation. If opposite characteristics are set for two sources, a crossfade effect can be programmed (see below).

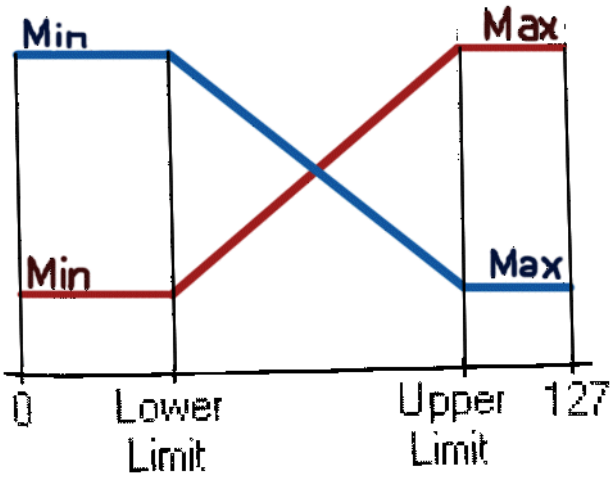


Fig. 2.16 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing velocity values from the Single Trig Gate Module. Setting opposite characteristics for two sources creates a crossfade effect between them, going from low to high MIDI velocity values.

A switch with an adjustable threshold level can be emulated by setting the Lower Limit and Upper Limit edit fields to neighboring MIDI velocity values, for example, “63” and “64”. When the input value to such a source is “64”, the value in the Min edit field will be output. Otherwise, the output value is the value in the Max edit field.

2.4.4 Example: Note to Chord

The To Voice Modules (↑14.11, To Voice) give you direct access to the values carried by each Voice. A handy application is to generate chords from incoming Monophonic MIDI Notes. The Structure that implements this feature is shown in the figure below for an Instrument that has its number of Polyphonic Voices set to "3". The chord that is played when a key is pressed is the minor chord: the incoming MIDI Note is assigned to the first Voice, the minor third to the second Voice, and the perfect fifth to the third Voice. As can be seen in the Structure below, Add Modules (↑4.2, Add, +) and the consequent To Voice Modules are used to assign each Voice the corresponding transposed MIDI Note. The output of each To Voice Module carries a Polyphonic signal where each Voice, except the one

specified by the "V" input port, is zero. The outputs of the To Voice Modules (↑14.11, To Voice) are added together to result in a Polyphonic signal where each Voice has a defined value.
To get the base note of the chord, the "Nr" output port of the Channel Message Module (↑2.17, Channel Message) is used. Although this output can send values other than MIDI Notes, triggering the value to be used only when a Gate On Event from the Single Trig Gate Module arrives, ensures that only MIDI Note values are used. The Separator Module (↑13.14, Separator), on the other hand, ensures that only Gate On Events trigger the note number to be forwarded to the chord generation algorithm.

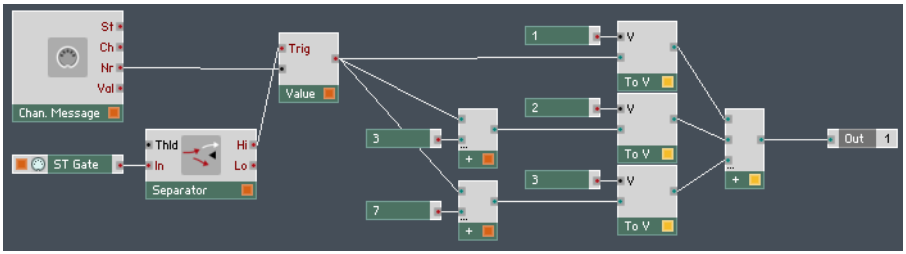


Fig. 2.17 The Structure to create a chord from an incoming Monophonic MIDI Note.

2.5 Sel Note Gate

Fig. 2.18 A Sel Note Gate Module

2.5.1 Overview

The Sel Note Gate Module is a Monophonic event source for selected Note On and Note Off events. A Note On event which has the MIDI note number selected in the Function page sets the output to a value determined by the key velocity. The range of the output values to which the incoming range of MIDI velocities (from no velocity (Note Off) to maximum velocity) is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min .... Max]. When a Note Off event arrives at the parent Instrument's MIDI In, the Sel Note Gate Module sends an Event with the value "0" (a Gate Off Event), independent of the "Min" and "Max" settings in the Function page. If velocity sensitivity of your Instrument is to be disabled, the "Min" and "Max" values should be set to the same value.

Application

Whereas the Gate and Single Trig Gate Modules (↑2.4, Single Trig Gate) respond to all MIDI note values, the Sel Gate Module only responds to one MIDI note value. You could use a number of Sel Gate Module's to build a highly customized live Instrument where each key triggers a different sampler or envelope.

2.5.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for the Gate On Events. These Events are only sent when the MIDI note value of the incoming MIDI note is equal to the one set in the Module's Function page. The velocity values of such a MIDI note are scaled to fit the range [Min … Max] set in the Function page.

2.5.3 Properties: Function Page

Setting the Incoming MIDI Note Value

The Sel Note Gate Module only outputs Gate On Events of MIDI notes that are the same pitch as defined by the Note edit field in the Module's Function page.

► To change the MIDI note to which the Sel Note Gate Module should react, go to the Module's Function page and enter the corresponding note value in the MIDI scale into the Note edit field (shown in the screenshot below).

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min … Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

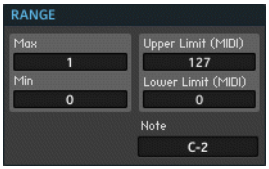


Fig. 2.19 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Sel Note Gate Module's Function page.

Setting the Range of Input Values

The range of incoming velocities to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The output value of Sel Note Gate Module is limited to "Min" for incoming MIDI values below "Lower Limit" and to "Max" for incoming MIDI values above "Upper Limit". The range between the two limits is interpolated linearly between "Min" and "Max" (see below).

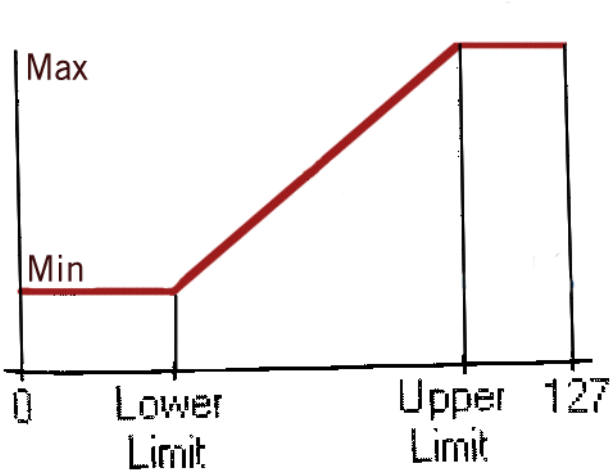


Fig. 2.20 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing values from the Sel Note Gate Module. The values are interpolated linearly between "Min" and "Max."

The values for the Lower Limit and Upper Limit edit fields lie in the range [0 ... 127]. Also, the value for Upper Limit edit field must be greater than that set in the Lower Limit edit field. However, the value in the Max edit field can be smaller than the value in the Min edit field to achieve inverted operation. If opposite characteristics are set for two sources, a crossfade effect can be programmed (see below).

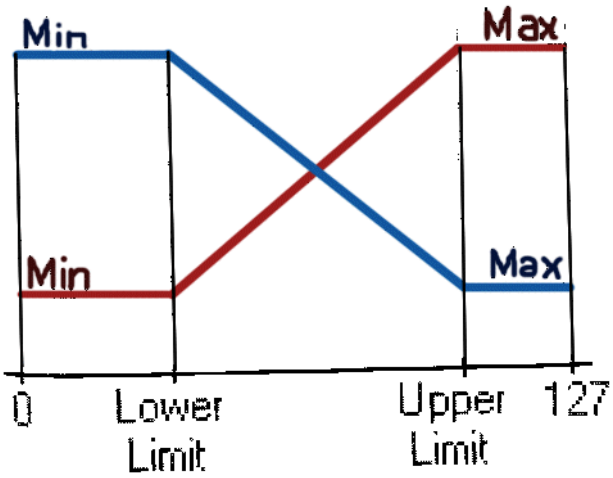


Fig. 2.21 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing values from the Sel Note Gate Module. Setting opposite characteristics for two sources creates a crossfade effect between them, going from low to high MIDI velocity values.

A switch with an adjustable threshold level can be emulated by setting the Lower Limit and Upper Limit edit fields to neighboring MIDI velocity values, for example, “63” and “64”. When the input value to such a source is “64”, the value in the Min edit field will be output. Otherwise, the output value is the value in the Max edit field.

2.5.4 Example: Tapedeck Playback

This example shows how you could use a single key on your computer or MIDI keyboard to trigger an action inside the Structure. In this case, the gate signal from the Sel Note Gate Module has been used to serve as the Gate signal for tapedeck playback at the Tapedeck 1 Channel Module (↑14.1, Tapedeck 1 Channel). Note you choose in the Sel Note Gate Module's Function page the MIDI Note for which a Gate signal is forwarded.

For a Sampler Module playback of different samples according to the MIDI Note that is played is achieved using the Sample Map.

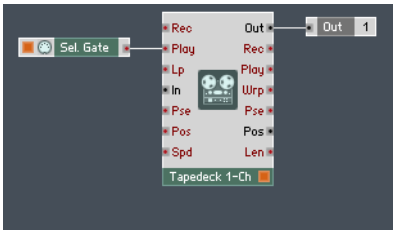


Fig. 2.22 Using the Sel Note Gate Module to have exactly one MIDI Note send the Gate signal for tapedeck playback.

2.6 On Velocity

Fig. 2.23 On Velocity Module

2.6.1 Overview

The On Velocity Module is a Polyphonic Event source for the velocity of MIDI Note On events. A Note On event sets the output to a value determined by the key velocity. The range of the output values to which the incoming range of MIDI velocities (from no velocity (Note Off) to maximum velocity) is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min .... Max]. If velocity sensitivity of your Instrument is to be disabled, the "Min" and "Max" values should be set to the same value.

Application

The On Velocity Module is useful when you want to quasi-statically scale certain signals according to the velocity. The difference between the On Velocity Module and the Gate Module is that upon receiving a Note Off Event, the On Velocity Module does *not* send a zero-valued Event from its output port. That's why you would need the On Velocity Module when you want your scaling characteristic to extend beyond the Note Off event (and not be abruptly cut off by a zero valued Gate Off Event).

2.6.2 Ports

- **(Out)** "Out" is the Polyphonic Event output port for the Gate On Events. These Events carry the value of the velocity with which the key was pressed. The values are scaled to fit the range [Min ... Max] set in the Function page.

2.6.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.24 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the On Velocity Module's Function page.

Setting the Range of Input Values

The range of incoming velocities to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The velocity output value of the On Velocity Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

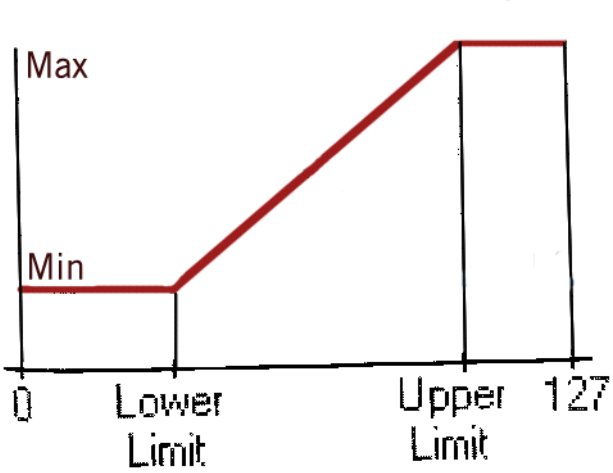


Fig. 2.25 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing velocity values from the On Velocity Module. The values are interpolated linearly between “Min” and “Max.

The values for the Lower Limit and Upper Limit edit fields lie in the range [0 ... 127]. Also, the value for Upper Limit edit field must be greater than that set in the Lower Limit edit field. However, the value in the Max edit field can be smaller than the value in the Min edit field to achieve inverted operation. If opposite characteristics are set for two sources, a crossfade effect can be programmed (see below).

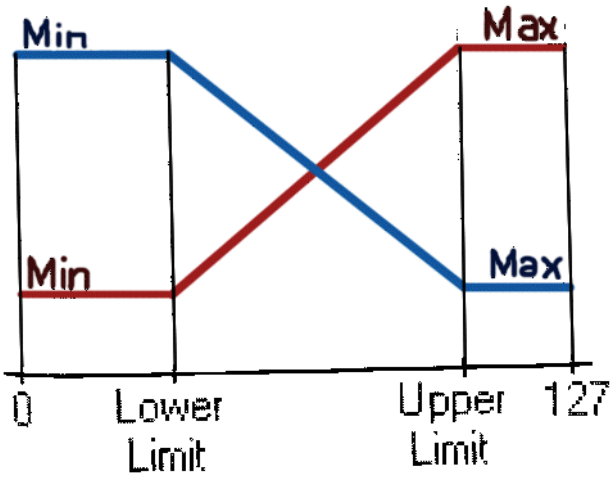


Fig. 2.26 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing velocity values from the On Velocity Module. Setting opposite characteristics for two sources creates a crossfade effect between them, going from low to high MIDI velocity values.

A switch with an adjustable threshold level can be emulated by setting the Lower Limit and Upper Limit edit fields to neighboring MIDI velocity values, for example, “63” and “64”. When the input value to such a source is “64”, the value in the Min edit field will be output. Otherwise, the output value is the value in the Max edit field.

2.6.4 Example: 6-Step Pitch Sequencer

This example shows how to use a 6-Step Sequencer Module (↑8.1, 6-Step Sequencer) to create a sequencer of pitch values along with a Gate signal of uniform velocity. To start, insert a 6-Step Sequencer Module and a Sync Pulse Module (↑2.15, Sync Pulse) into the Structure, as shown in the figure below. For the Sequencer Module's input ports from "S1" to "S6", create controls simply by right-clicking the input ports and choosing the *Create Control* menu entry. Knobs with the standard MIDI range and resolution are created, perfectly suitable for sequencing pitch values to an oscillator, for example. Next, use the *Create Control* menu entry to create a Button Module (↑1.2, Button) at the "Rst" (reset) input port. This enables you to reset the sequencer position. Insert an On Velocity Module (↑2.6,

On Velocity), set it to Monophonic Mode, and connect it to the "GA" (gate amplitude) input port. This lets you set the velocity of Gate On signals (outgoing at the "G" output port) to the velocity with which the last MIDI or computer keyboard key was pressed.

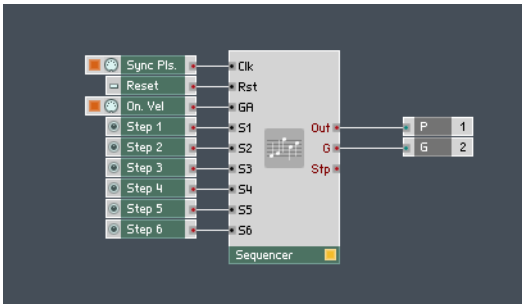


Fig. 2.27 The Structure for a 6-step pitch value sequencer.

Next, connect the Sync Pulse Module (↑2.15, Sync Pulse) to the "Clk" (clock) input port of the Sequencer Module (↑8.1, 6-Step Sequencer). Go to the Sync Pulse Module's Function page and choose the rate at which the sequencer runs from the Rate drop-down menu, shown in the figure below. The duration in this case does not make a difference. The rate will be synchronized to the MIDI Clock and clock Events will only be output to the Sequencer Module if the MIDI Clock is running. The setup shown in this example has the following functionality. When the MIDI Clock is running, the Sequencer Module will go through each step at the rate determined by the Sync Pulse Module. At each step, the "Out" output port will send an Event carrying the value as determined by the knob connected to the input port of the corresponding step, as well as a Gate On signal that carries the velocity value set by the velocity of the last pressed MIDI key.

Note that you could also use the Clock Module (↑2.14, Clock) and to get the desired rate using the Event Frequency Divider Module (↑13.4, Event Frequency Divider). The advantage of that method is that you can choose the frequency from the Instrument Panel.

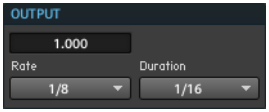


Fig. 2.28 Use the Rate drop-down menu in the Sync Pulse Module's Function page to choose the rate of clock Events sent to the Sequencer Module.

2.7 Off Velocity

Fig. 2.29 Off Velocity Module

2.7.1 Overview

The Off Velocity Module is a Polyphonic event source for the velocity of MIDI Note Off events. A Note Off event sets the output to a value determined by the key release velocity. The range of output values to which the incoming range of MIDI release velocities (from no velocity to maximum velocity) is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min ... Max].

Application

The Off Velocity Module is necessary if you with to scale (probably release relevant) parameters in your Structure to the release velocity of your MIDI keyboard. Note however, that only very few keyboards actually send MIDI key release velocity messages so in most cases this Module will not receive the necessary information from your MIDI controller.

2.7.2 Ports

- **(Out)** "Out" is the Polyphonic Event output port for the key release velocity Events. These Events are sent when the key is released and carry the value of the release velocity with which the key was released. The values are scaled to fit the range [Min ... Max] set in the Function page.

2.7.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.30 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Off Velocity Module's Function page.

Setting the Range of Input Values

The range of incoming release velocities to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The velocity output value of the Off Velocity Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

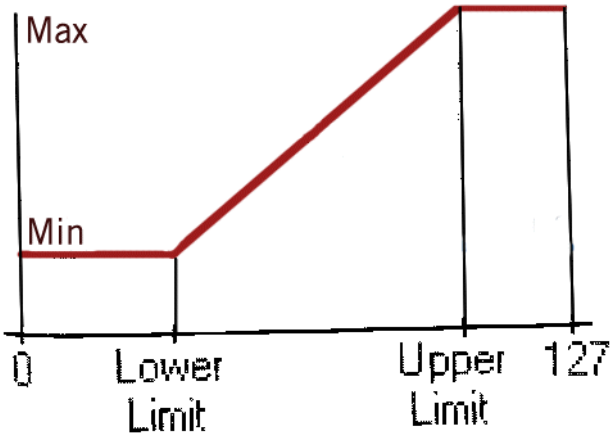


Fig. 2.31 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing release velocity values from the Off Velocity Module. The values are interpolated linearly between “Min” and “Max.

The values for the Lower Limit and Upper Limit edit fields lie in the range [0 ... 127]. Also, the value for Upper Limit edit field must be greater than that set in the Lower Limit edit field. However, the value in the Max edit field can be smaller than the value in the Min edit field to achieve inverted operation. If opposite characteristics are set for two sources, a crossfade effect can be programmed (see below).

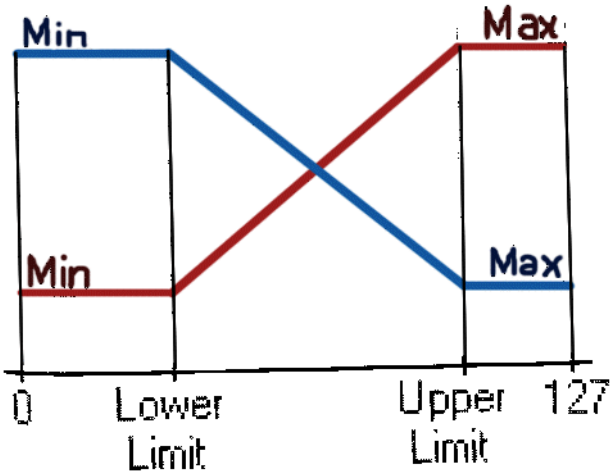


Fig. 2.32 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing release velocity values from the Off Velocity Module. Setting opposite characteristics for two sources creates a crossfade effect between them, going from low to high MIDI velocity values.

A switch with an adjustable threshold level can be emulated by setting the Lower Limit and Upper Limit edit fields to neighboring MIDI velocity values, for example, “63” and “64”. When the input value to such a source is “64”, the value in the Min edit field will be output. Otherwise, the output value is the value in the Max edit field.

2.7.4 Example: Gate Off with Off Velocity

This example shows a Structure with which you can send the Off Velocity value when a Gate Off Event is received. Use the Separator Module (↑13.14, Separator) to sort out the Gate Off Event (and block the Gate On Event) by connecting the "Lo" output port with Thld = 0. Let this signal be used as the trigger signal for the Value Module (↑13.15, Value) which then sends the Off Velocity, retrieved from the Off Velocity Module.

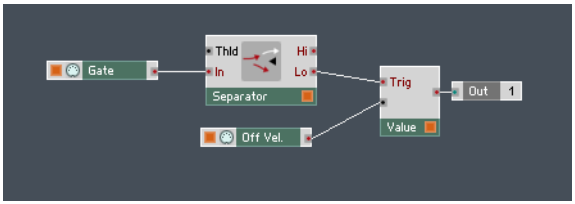


Fig. 2.33 The Structure for sending a Gate Off Event with Velocity Off value.

2.8 Controller

Fig. 2.34 Controller Module

2.8.1 Overview

The Controller Module is a Monophonic Event source for MIDI Controller events. This Module "listens" only to MIDI signals with a certain CC value, set in the Function page. A MIDI event with the set CC value, incoming at the parent Instruments MIDI In, sets the Controller Module's output to a value determined by the position of the controller (e.g. modulation wheel). The range of the output values to which the incoming range of MIDI controller values (in the range [0 ... 127]) is scaled, is set with the Min and Max edit fields in the Function page. The resolution of the output signal is determined by the Step Size edit field in the Function page.
The current position of all MIDI controllers (and faders) of an Instrument can be stored in a Snapshot from where it can be recalled at a later time. If the Snap Isolate checkbox is engaged, the Controller Module's output value will not respond to Snapshot recall.

Application

Use the Controller Module to tie certain MIDI messages from certain controllers such as faders or knobs to specific processes in your Instrument's Structure. For example, the MIDI modulation wheel has the CC value "1". You could then set the Controller/Note edit field to "1" and use the Controller Module's output to control the vibrato amount or some other modulation parameter.

2.8.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for the values of the selected MIDI controller messages. The values are scaled to fit the range [Min ... Max] set in the Function page.

2.8.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

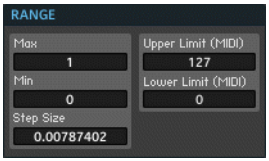


Fig. 2.35 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Controller Module's Function page.

Setting the Range of Input Values

The range of incoming release velocities to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The output value of the Controller Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

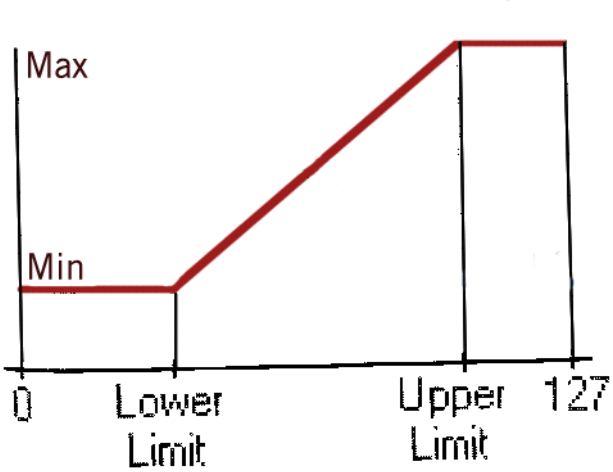


Fig. 2.36 The x-axis of the graph shows the incoming MIDI values whereas the y-axis shows the outgoing release velocity values from the Controller Module. The values are interpolated linearly between “Min” and “Max.

Setting the Step Size of the Output Values

► To set the resolution of the output values in the range [Min … Max], enter the corresponding increment (step size) value into the Step Size edit field, shown in the screenshot below.

Setting the CC Value of the Forwarded MIDI Message

The Controller Module "listens" to MIDI messages with a set CC value.

► To set the CC value of the MIDI messages that reach the Controller Module, enter the desired value into the Controller/Note edit field shown in the screenshot below.

The CC values of some standard MIDI controllers are:

- **1** Modulation Wheel

- **2** Breath Controller
- **7** Volume
- **10** Panpot
- **64** Sustain Switch
- **65** Portamento Switch
- **66** Hold Switch (Sostenuto)

See your MIDI keyboard's MIDI implementation chart to find out which controllers it can transmit.

Activating Soft Takeover

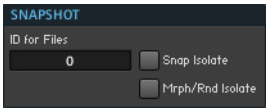
When the Controller Module's output value is operated by an external MIDI controller, it normally jumps straight to the received controller value. Such a jump can be quite noticeable in the sound, depending on the controlled parameter (for example, amplifier level). Thus, such jumps can be undesirable. You can avoid such behavior with the Soft Takeover feature. It causes the control to only move when the value received via MIDI reaches or goes past the current position.

► To activate Soft Takeover, engage the Soft Takeover checkbox, shown in the figure above.

Keeping the Controller Value Isolated from the Snapshot

The current position of all MIDI controllers (and faders) of an Instrument can be stored in a Snapshot from where it can be recalled at a later time.

► To keep the controller value from being saved with Snapshots, go to the Function page and engage the Snap Isolate checkbox. This way the Controller Module's output value will not respond to Snapshot recall.



Please refer to section 4.1in the Application Reference for more information on Snapshot recall.

2.8.4 Example: Master Volume

This example shows how to control the Ensemble's master volume level using the MIDI Controller that is usually assigned for such a task, MIDI CC 7. The Structure for this functionality is shown in the figure below. The Controller Module's output signal is per default in the range [0 … 1]. This range needs to be converted to the logarithmic scale for the Master Tune/Level Module's (↑14.16, Master Tune/Level) "Lvl" input port. Use the Log (A-to-Lvl) Module (↑4.17, Log (A-to-Lvl)) for this task (shown in the Structure below).

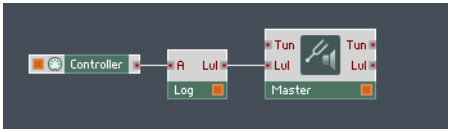


Fig. 2.37 The Controller Module can be used to send specific MIDI Controller messages to parts of your Structure, such as the "Lvl" input port of the Master Tune/Level Module.

Last, you need to set the CC value from which the Controller Module receives its messages. This is done with the Controller/Note edit field in the Controller Module's Function page, shown in the figure below. The standard MIDI Controller number for the volume is "7", as reflected by the edit field shown below.

Fig. 2.38 Use the Controller/Note edit field to set the CC number of the MIDI Controller from which the Controller Module is to receive its messages.

2.9 Channel Aftertouch

Fig. 2.39 A Channel Aftertouch Module

2.9.1 Overview

The Channel Aftertouch Module is a Monophonic Event source for MIDI Channel Aftertouch events. An incoming MIDI Channel Aftertouch event sets the output to a value determined by the pressure on all keys. The range of output values to which the incoming

range of Aftertouch values [0 … 127] is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min … Max].

Application

If you wish your Instrument to be able to receive Channel Aftertouch messages from your MIDI keyboard, placing this Module into the Instrument's Structure is necessary. You could use Channel Aftertouch messages to modulate the cutoff frequency or resonance of a filter, or even the modulation index of an FM synth. Note that not all MIDI keyboards send Channel Aftertouch messages.

2.9.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for Channel Aftertouch values. These Events are sent after keys are pressed. The values are scaled to fit the range [Min … Max] set in the Function page.

2.9.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min … Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.40 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Channel Aftertouch Module's Function page.

Setting the Range of Input Values

The range of incoming Aftertouch values to which the output values [Min … Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The output value of the Channel Aftertouch Module is

limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

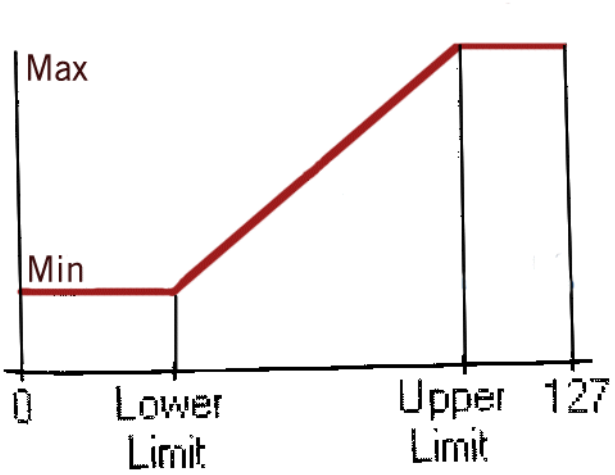


Fig. 2.41 The x-axis of the graph shows the incoming MIDI Channel Aftertouch values whereas the y-axis shows the outgoing values from the Channel Aftertouch Module. The values are interpolated linearly between “Min” and “Max."

2.9.4 Example: Channel Aftertouch for Filter Cutoff

The example Structure below shows how you could use the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) to modulate the filter cutoff pitch with the signal from a Channel Aftertouch Module. Just multiply the output of the Channel Aftertouch Module with the value which determines the amount of Aftertouch modulation for the cutoff signal and add it to the regular cutoff pitch value, received from the "P Cutoff" Knob Module (↑1.1, Fader/Knob). Feed the result to the Filter Module's "P" (pitch) input port. Note that the final modulation amount also depends on the "Min" and "Max" settings in the Channel Aftertouch Module's Function page. For the Channel Aftertouch Module, the modulation is applied to all Voices.

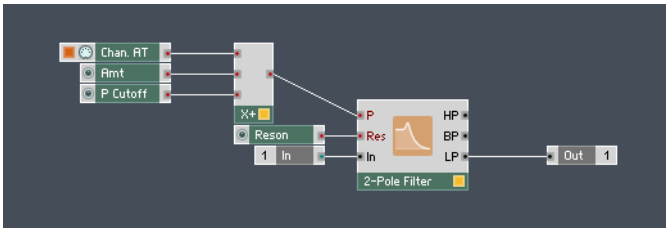


Fig. 2.42 The Structure for modulating a filter cutoff pitch with a Monophonic MIDI Aftertouch signal.

2.10 Poly Aftertouch

Fig. 2.43 A Poly Aftertouch Module

2.10.1 Overview

The Poly Aftertouch Module is a Polyphonic Event source for MIDI Channel Aftertouch events. An incoming MIDI Channel Aftertouch event sets the output value of a Polyphonic Voice to a value determined by the pressure on the corresponding key. The range of output values to which the incoming range of Aftertouch values [0 … 127] is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min … Max].

Application

If you wish your Instrument to be able to receive Polyphonic Channel Aftertouch messages from your MIDI keyboard, placing this Module into the Instrument's Structure is necessary. You could use Polyphonic Channel Aftertouch messages to modulate the cutoff frequency or resonance of a filter, or even the modulation index of an FM synth for each Polyphonic Voice separately. Note that only very few MIDI keyboards send Polyphonic Channel Aftertouch messages.

2.10.2 Ports

- **(Out)** "Out" is the Polyphonic Event output port for Channel Aftertouch values. These Events are sent after each key is pressed. The values are scaled to fit the range [Min … Max] set in the Function page.

2.10.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.44 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Poly Aftertouch Module's Function page.

Setting the Range of Input Values

The range of incoming Aftertouch values to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The output value of the Poly Aftertouch Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

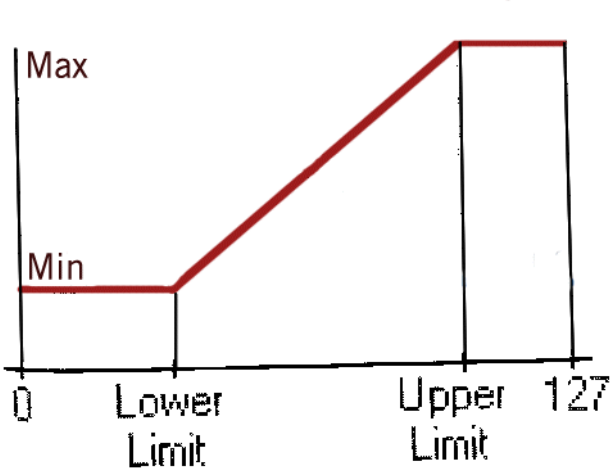


Fig. 2.45 The x-axis of the graph shows the incoming MIDI Channel Aftertouch values whereas the y-axis shows the outgoing values from the Poly Aftertouch Module. The values are interpolated linearly between “Min” and “Max.

2.10.4 Example: Poly Aftertouch for Filter Cutoff

The example Structure below shows how you could use the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) to modulate the filter cutoff pitch with the signal from a Poly Aftertouch Module. Just multiply the output of the Poly Aftertouch Module with the value which determines the amount of Aftertouch modulation for the cutoff signal and add it to the regular cutoff pitch value, received from the "P Cutoff" Knob Module (↑1.1, Fader/Knob). Feed the result to the Filter Module's "P" (pitch) input port. Note that the final modulation amount also depends on the "Min" and "Max" settings in the Poly Aftertouch Module's Function page. For the Poly Aftertouch Module, the modulation is applied only to the Voice whose corresponding key receives the Aftertouch messages.

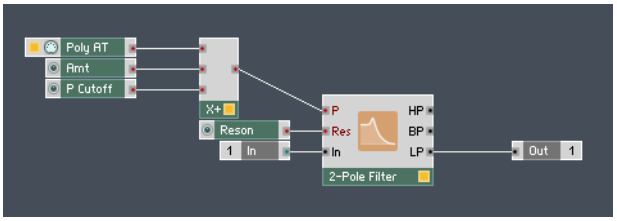


Fig. 2.46 The Structure for modulating a filter cutoff pitch with a Polyphonic MIDI Aftertouch signal.

2.11 Sel. Poly Aftertouch

Fig. 2.47 Sel Poly Aftertouch Module

2.11.1 Overview

The Sel Poly Aftertouch Module is a Monophonic Event source for selected MIDI Poly Aftertouch events. An Aftertouch message from a pressed key with the MIDI note number selected in the Function page sets the output to a value determined by the pressure on the key. The range of output values to which the incoming range of Aftertouch values [0 … 127] is scaled, is set with the Min and Max edit fields in the Function page. The resolution of the output signal is determined by the Step Size edit field in the Function page.
The current position of the Aftertouch value of the Instrument can be stored in a Snapshot from where it can be recalled at a later time. If the Snap Isolate checkbox is engaged, the Controller Module's (↑2.8, Controller) output value will not respond to Snapshot recall.

There are very few keyboards that generate Poly Aftertouch events.

Application

Whereas the Poly Aftertouch Module responds to Aftertouch messages from all MIDI notes, the Sel Poly Aftertouch Module only responds to Aftertouch messages from one MIDI note. You could use a number of Sel Poly Aftertouch Module's to build a highly customized live Instrument where the Aftertouch signals of different keys have a different modulation effect in your Instrument's Structure.

2.11.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for the values of the selected MIDI note's Aftertouch messages. The values are scaled to fit the range [Min ... Max] set in the Function page.

2.11.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

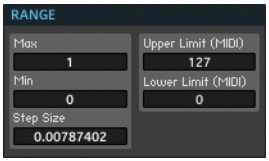


Fig. 2.48 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Sel Poly Aftertouch Module's Function page.

Setting the Range of Input Values

The range of incoming Aftertouch values to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The output value of the Sel Poly Aftertouch Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

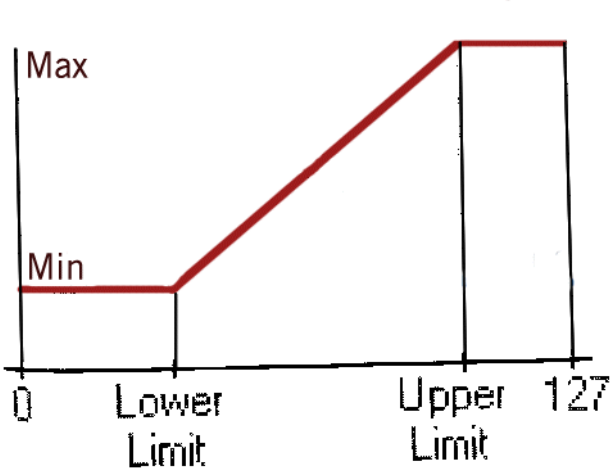


Fig. 2.49 The x-axis of the graph shows the incoming MIDI Channel Aftertouch values whereas the y-axis shows the outgoing values from the Sel Poly Aftertouch Module. The values are interpolated linearly between “Min” and “Max."

Setting the Step Size of the Output Values

► To set the resolution of the output values in the range [Min … Max], enter the corresponding increment (step size) value into the Step Size edit field, shown in the screenshot below.

Setting the MIDI Note Value of the Forwarded Aftertouch Message

The Sel Poly Aftertouch Module "listens" to MIDI messages from a key with a specific MIDI Note value.

► To set the MIDI Note value of the MIDI messages that reach the Sel Poly Aftertouch Module, enter the desired value into the Controller/Note edit field shown in the screenshot below.

Activating Soft Takeover

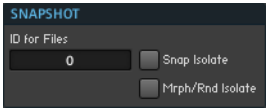
When the Sel Poly Aftertouch Module's output value is operated by an external MIDI controller, it normally jumps straight to the received controller value. Such a jump can be quite noticeable in the sound, depending on the controlled parameter (for example, amplifier level). Thus, such jumps can be undesirable. You can avoid such behavior with the Soft Takeover feature. It causes the control to only move when the value received via MIDI reaches or goes past the current position.

► To activate Soft Takeover, engage the Soft Takeover checkbox, shown in the figure above.

Keeping the Aftertouch Value Isolated from the Snapshot

The current value of the Aftertouch message can be stored in a Snapshot from where it can be recalled at a later time.

► To keep the Aftertouch value from being saved with Snapshots, go to the Function page and engage the Snap Isolate checkbox. This way the Sel Poly Aftertouch Module's output value will not respond to Snapshot recall.



Please refer to section 4.1in the Application Reference for more information on Snapshot recall.

2.11.4 Example: Sel Poly Aftertouch for Filter Cutoff

The example Structure below shows how you could use the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) to modulate the filter cutoff pitch with the signal from a Sel Poly Aftertouch Module. Just multiply the output of the Sel Poly Aftertouch Module with the value which determines the amount of Aftertouch modulation for the cutoff signal and add it to the regular cutoff pitch value, received from the "P Cutoff" Knob Module (↑1.1, Fader/Knob). Feed the result to the Filter Module's "P" (pitch) input port. Note that the final modulation amount also depends on the "Min" and "Max" settings in the Sel Poly Aftertouch Module's Function page. For the Sel Poly Aftertouch Module, the modulation is applied only to the Voice that carries the MIDI Note specified by the Controller/Note edit field in the Sel Poly Aftertouch Module's Function page.

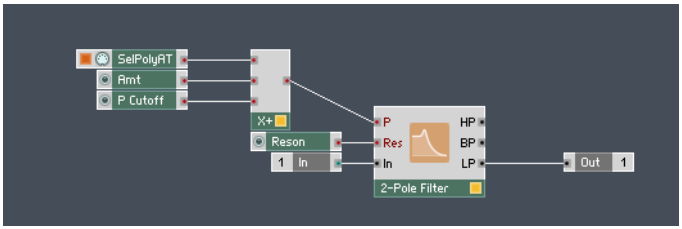


Fig. 2.50 The Structure for modulating a filter cutoff pitch with a Polyphonic MIDI Aftertouch signal received from only one specific MIDI key.

2.12 Program Change

Fig. 2.51 A Program Change Module

2.12.1 Overview

The Program Change Module is a Monophonic Event source for MIDI Program Change messages. A MIDI Program Change event sets the Module's output to the value determined by the transmitted program number. The range of output values to which the incoming range of Aftertouch values [0 ... 127] is scaled, is set with the Min and Max edit fields in the Function page. The resolution is such that 128 steps fit into the interval set in the Function page, [Min ... Max].

Application

The Program Change Module is very useful when you want to use MIDI Program Change messages to change the settings of specific parts of the Structure instead of recalling a whole Instrument (or Ensemble) Snapshot. When using this Module, you probably want to disengage the Recall by MIDI checkbox in the parent Instrument's Function page so that the MIDI Program Change events do not also recall the parent Instrument's Snapshots.

2.12.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for Program Change messages. The values are scaled to fit the range [Min ... Max] set in the Function page.

2.12.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min ... Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties.

Fig. 2.52 The “Min”, “Max”, “Lower Limit”, and “Upper Limit” values can be set in the corresponding edit fields of the Program Change Module's Function page.

Setting the Range of Input Values

The range of incoming MIDI Program Change values to which the output values [Min ... Max] are scaled, can also be limited with the Lower Limit and Upper Limit edit fields (shown in the Function page screenshot above). The output value of the Program Change Module is limited to “Min” for incoming MIDI values below “Lower Limit” and to “Max” for incoming MIDI values above “Upper Limit”. The range between the two limits is interpolated linearly between “Min” and “Max” (see below).

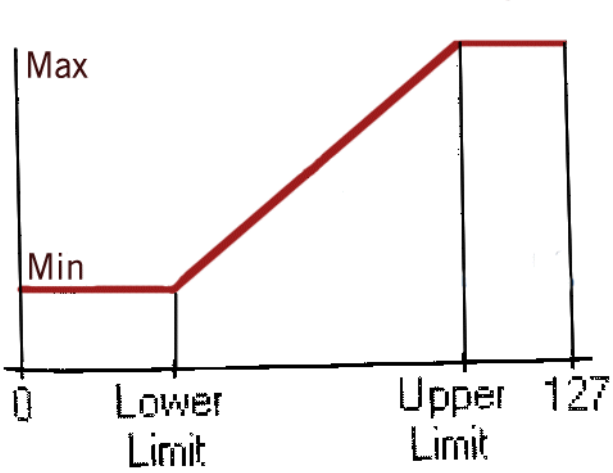


Fig. 2.53 The x-axis of the graph shows the incoming MIDI Program Change values whereas the y-axis shows the outgoing values from the Program Change Module. The values are interpolated linearly between “Min” and “Max."

2.12.4 Example: Snapshot Recall and Program Change

This example illustrates how to forward MIDI Program Change messages to the Snapshot Module (↑14.23, Snapshot) to recall REAKTOR Snapshots and conversely, how to have REAKTOR Snapshot recall operations trigger MIDI Program Change messages to be sent from REAKTOR. The Structure that achieves this is shown in the Structure below.

The Program Change Module should send a value in the range [1 … 128] to address one of the 128 Snapshots in the active Bank to be recalled. For that, go to the Program Change Module's Function page and set the "Min" and "Max" values to "1" and "128", respectively. Use the Order Module (↑13.12, Order) to first specify at the "Snp" input port the Snapshot number to be recalled (upon receiving a Program Change Event) and only then trigger the Snapshot recall operation at the "Recl" input port. Similarly, Use the Value Module (↑13.15, Value) to send the Event carrying the recalled Snapshot number to the Program Change Out Module (↑3.7, Program Change Out) when a Snapshot recall operation takes place. The "Min" and "Max" values in the Program Change Out Module's (↑3.7, Program Change Out) Function page should also be set to "1" and "128", respectively.

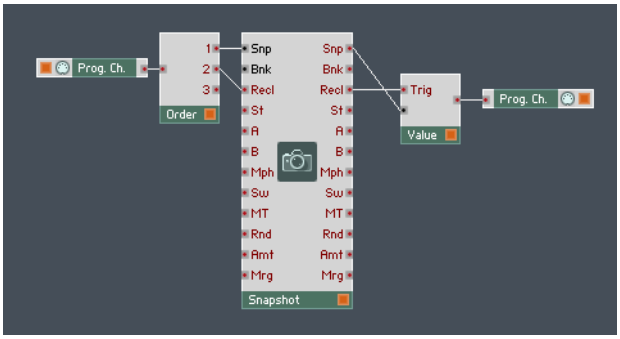


Fig. 2.54 You can use the Program Change (Out) and Snapshot Modules to link Snapshot operations to external MIDI devices.

2.13 Start/Stop

Fig. 2.55 Start/Stop Module

2.13.1 Overview

The Start/Stop Module is a Monophonic Event source for MIDI Clock Start and Stop events. The output Events serve to synchronize processes in your Structure (like LFOs, Sequencers, etc.) with external MIDI devices or the internal Master Clock. The Start/Stop Module's "G" (gate) output port sends a Monophonic Gate signal. Gate On corresponds to a running clock and Gate Off corresponds to a stopped clock. The Gate On value can be set with the Output Value edit field in the Function page. The output signal jumps to the Output value when the Run button in the Main Toolbar is pressed or when a MIDI Start event is received at the parent Instruments MIDI In. The signal jumps back to zero when the Stop button is pressed or when a MIDI Stop event is received.

Please refer to subsection 3.3.3 in the Application Reference for more information on operating the Run and Stop buttons.

Application

The Start/Stop Module is typically connected to the reset input of sequencers and Event dividers which are synchronized with the MIDI Clock to force a synchronized start. In more rare cases you might want to synchronize an LFO or an oscillator to MIDI Clock Start events. Use a Separator Module (↑13.14, Separator) to separate Clock start Events from Clock Stop Events.

2.13.2 Ports

- **(G)** "G" (gate) is the Monophonic Event output port for a MIDI Clock Start and MIDI Clock Stop messages. The output is set to the value specified in the Function page when the MIDI Clock is running and to zero when the Clock is stopped.
- **(Rst)** "Rst" (reset) is the Monophonic Event output port for MIDI Clock Reset messages. An Event with the value "1" is sent when the Clock starts from the beginning of the song (i.e. not when the Clock starts running again after being paused). Use this output port to reset sequencers, for example.

2.13.3 Properties: Function Page

Setting the Gate On Output Value

► To set the value of the Event at the "G" (gate) output port when the Clock is running, go the Function page and enter the desired value into the Output Value edit field, as shown in the figure below.

2.13.4 Example: Multiplex Sequencer

This example shows how to build a sequencer with variable length, but with a maximum of 16 steps. The first part of the Structure is shown in the figure below. There you see that the core of the sequencer is a Multiplex 16 Module and the values of the steps are received from Knob Modules (↑1.1, Fader/Knob) at the input ports "0" to "15". The sequence length of the step at which the sequencer "folds back" to the first step, is set with the knob labeled "Length" at the "Len" input port. The actual position of the sequencer is

received at the "Pos" (position) input port in the form of a regular Events with incremental values. The value of the Event determines the sequencer position where values greater than "Len" are "folded back" to the set sequencer range.

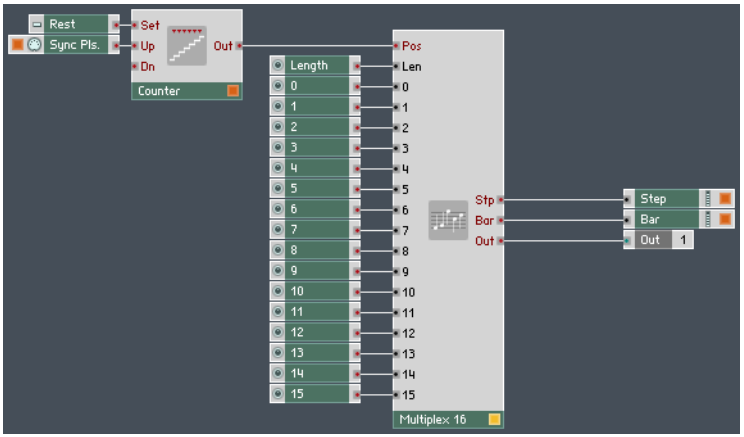


Fig. 2.56 16 step sequencer

In this example, the "Pos" values are ultimately received from the Sync Clock Module (↑2.15, Sync Pulse) which sends clock Events at regular time intervals, synchronized to the MIDI Clock. The rate of outgoing Events can be set at the Sync Clock Module's (↑2.15, Sync Pulse) Function page, with the Rate drop-down menu. In order to get regular Events carrying incremental values, the Counter Module (↑13.2, Counter) has been used. Each Event from the Sync Clock Module (↑2.15, Sync Pulse) increments the output of the Counter Module (↑13.2, Counter) by one, which then ends up being used as the "Pos" value for the Multiplex 16 Module. At the Counter Module's "Set" input port you should have a Button Module (↑1.2, Button). In its Function page, set the Button Module to Trigger Mode and its "Max" value to zero. This enables you to reset the sequencer to its first position from the Instrument Panel.
As an additional feature, the XY Module (↑1.14, XY) has been used to graphically track the current sequencer position. The final Structure with this implemented is shown in the first figure at the end of the example and the final Panel representation of the sequencer is shown at the last figure at the end of the example. Please look at that figure to see where we are headed with the XY Module. The XY Module should be configured to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). This corresponds to going to its View

page and choosing the *Rectangle* menu entry from the Object Type drop-down menu (shown in the figure below). Also, you don't need a cursor and therefore choose the *None* menu entry from the Cursor Type drop-down menu (also shown in the figure below). While you are already in the XY Module's View page, disengage the Show Label and Show Value checkboxes and set the Height and Width edit fields so that the XY Module's Panel representation looks similar to what is depicted in the last figure of this example.

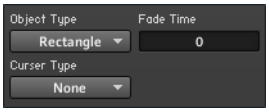


Fig. 2.57 To draw a rectangle from (X1, Y1) to (X2, Y2), choose the Rectangle menu entry from the Object Type drop-down menu.

The rectangle object type causes the XY Module's Panel representation to draw a rectangle from the coordinates (X1, Y1) to the coordinates (X2, Y2). We want to turn the XY Module (↑1.14, XY) into a row of 16 "imaginary" rectangles, each corresponding to one of the 16 steps in the sequencer. This is done the easiest when each box has the dimensions 1 by 1. For this reason, it is necessary to set the range of displayed values correctly in the XY Module's Function page. Set the "Min" and "Max" values for the X coordinate to "0" and "16", respectively. "Min" and "Max" for the Y coordinate should be "0" and "1", respectively. This is shown in the figure below.

Fig. 2.58 Use the Min and Max edit fields to set the range for the X and Y coordinates to [0 … 16] and [0 … 1], respectively.

Now that you have configured the XY Module (↑1.14, XY), you just need to feed the correct values from the Multiplex 16 Module to the input ports for the corresponding coordinates. Remember you wish to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). As shown in the Structure depicted below, connect the "Stp" (step) output port off the Multiplex 16 Module to the "X1" input port of the XY Module, and the value Step + 1 (using an Add Module, see also ↑4.2, Add, +) to the "X2" input port. Furthermore, connect the constant value "1" to the "Y2" input port. Your resulting Structure should look like the first figure below and its Instrument Panel counterpart should look similar to what is

shown in the second figure below. Numeric Display Modules (↑1.8, Meter) have been added at the "Stp" and "Bar" output ports to give you quantitative visual access to the step and bar positions of the sequencer.

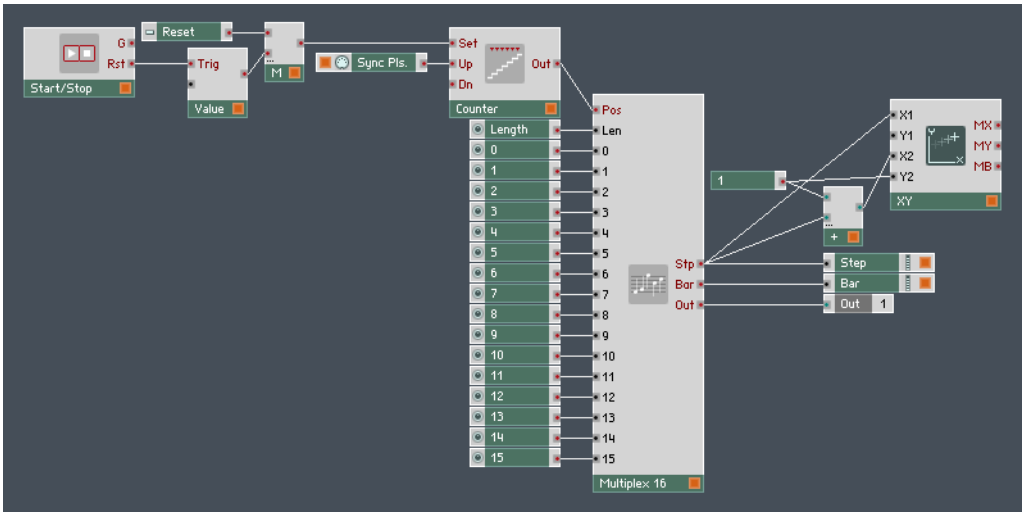


Fig. 2.59 The Structure for a simple sequencer with variable length and a simple step position display.

An additional feature to add would be that resetting the MIDI Clock also causes the sequencer position to be reset as well. Let's use the Start/Stop Module as that source of "reset" Events. For this, a reset operation should send the value "0" to the Counter Module's (↑13.2, Counter) "Set" input port. Since the "Rst" output port sends an Event with value "1", use a Value Module (↑13.15, Value) to change the Event's value to "0" and the a Merge Module to merge this reset signal with that received from the "Reset" button.

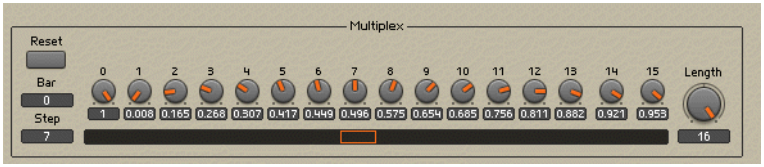


Fig. 2.60The Instrument Panel of the simple sequencer.

2.14 Clock

Fig. 2.61 Clock Module

2.14.1 Overview

The Clock Module is a Monophonic Event source for a clock signal which corresponds to the external MIDI Clock or the internal Master Clock. For each 96th note an Event is sent from the output port. The value of that Event can be set with the Output Value edit field in the Function page.

Application

Usually you would use the Clock Module to run sequencers (Sequencer Modules, ↑8, Sequencer, for example). In some cases your sequencers use 96th note increments. In most cases, however, you want to use sixteenth notes, eighth notes, quarter notes, half notes, and so on. To get Clock Events at these rates you need to feed the output of the Clock Module into the "Trig" (trigger) input port of an Event Frequency Divider Module (↑13.4, Event Frequency Divider). This is the difference between the Clock Module and the Sync Pulse Module: with the help of the Event Frequency Divider Module (↑13.4, Event Frequency Divider) you can extract Events that run at arbitrary rates from the Clock Module whereas the repetition rate of the Sync Pulse Module is set in its Function page.

2.14.2 Ports

- **(Out)** "Out" is the Monophonic output port for the Events which come at a rate of 96th notes synchronized to the MIDI Clock. The values of these Events are set in the Function page.

2.14.3 Properties: Function Page

Setting the Output Value

► To set the value of the 96th note Events at the output port, go the Function page and enter the desired value into the Output Value edit field, as shown in the figure below.

2.14.4 Example: 1/16th Notes Event Source

The Clock Module is a source of Events that run at a constant rate of 96th notes, synced to the MIDI Clock. To get arbitrary note divisions, combine the Clock Module with the Frequency Divider Module (↑13.4, Event Frequency Divider). As shown in the Structure below, connect the Clock Module to the Frequency Divider Module's "Trig" input port. To get a Gate On Events (followed by Gate Off Events) running at the rate of sixteenth notes, send the value "6" to the "N" input port. This is because 96 / 6 = 16. To allocate an equal time interval for the Gate On values as to the Gate Off values at the Frequency Divider Module's (↑13.4, Event Frequency Divider) output, send the value "0.5" to the "W" (width) input port.

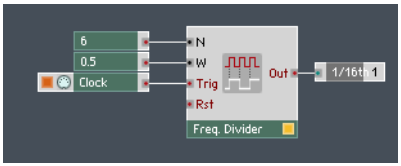


Fig. 2.62 Use the Frequency Divider Module to get arbitrary note divisions from the Clock Module.

2.15 Sync Pulse

Fig. 2.63 Sync Pulse Module

2.15.1 Overview

The Sync Pulse Module is a Monophonic Event source for a clock signal which is derived from an external MIDI Clock or the internal Master Clock. The output of the Sync Pulse Module is a Monophonic Gate signal. The Gate On value can be set with Output Value edit field in the Function page. The Gate Off value is zero. At each beat the output signal jumps to the Gate On value and back to zero after a certain time interval (duration). The rate of beats and the duration are set in the Function page. The Module is activated and deactivated by MIDI Start and MIDI Stop Events, respectively.

Application

Use the Sync Pulse Module to drive sequencers at a static rate. You can also connect the output port to the "G" (gate) input port of any Sampler or Envelope Module to trigger these Modules at a static rate. Of course, triggering happens in synch to the MIDI Clock and only when the MIDI Clock is running.

2.15.2 Ports

- **(Out)** "Out" is the Monophonic Event output port for the Gate On and Gate Off signals which are synchronized to the MIDI Clock and take place with a rate and duration set in the Function page.

2.15.3 Properties: Function Page

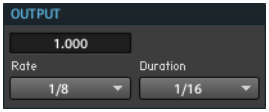
Setting the Value of the Gate On Events

► To set the value of the Gate On Events at the output port, go the Function page and enter the desired value into the Output Value edit field, as shown in the figure below.

Setting the Rate and Duration of the Gate On Events

► To set the rate of the clock signal (quarter, eighth, sixteenth notes, etc.), choose the desired rate from the Rate drop-down menu (shown in the figure below).

► To set the duration of the Gate signal (eighth, sixteenth note, etc.), choose the desired duration from the Duration drop-down menu (shown in the figure below).



2.15.4 Example: 6-Step Pitch Sequencer

This example shows how to use a 6-Step Sequencer Module (↑8.1, 6-Step Sequencer) to create a sequencer of pitch values along with a Gate signal of uniform velocity. To start, insert a 6-Step Sequencer Module and a Sync Pulse Module into the Structure, as shown in the figure below. For the Sequencer Module's input ports from "S1" to "S6", create controls simply by right-clicking the input ports and choosing the *Create Control* menu entry. Knobs with the standard MIDI range and resolution are created, perfectly suitable for sequencing pitch values to an oscillator, for example. Next, use the *Create Control* menu entry to create a Button Module (↑1.2, Button) at the "Rst" (reset) input port. This enables you to reset the sequencer position. Insert an On Velocity Module (↑2.6, On Velocity), set it to Monophonic Mode, and connect it to the "GA" (gate amplitude) input port. This lets you set the velocity of Gate On signals (outgoing at the "G" output port) to the velocity with which the last MIDI or computer keyboard key was pressed.

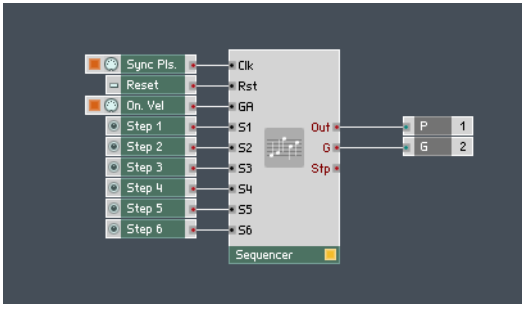


Fig. 2.64 The Structure for a 6-step pitch value sequencer.

Next, connect the Sync Pulse Module to the "Clk" (clock) input port of the Sequencer Module. Go to the Sync Pulse Module's Function page and choose the rate at which the sequencer runs from the Rate drop-down menu, shown in the figure below. The duration in this case does not make a difference. The rate will be synchronized to the MIDI Clock and clock Events will only be output to the Sequencer Module (↑8, Sequencer) if the MIDI

Clock is running. The setup shown in this example has the following functionality. When the MIDI Clock is running, the Sequencer Module (↑8, Sequencer) will go through each step at the rate determined by the Sync Pulse Module. At each step, the "Out" output port will send an Event carrying the value as determined by the knob connected to the input port of the corresponding step, as well as a Gate On signal that carries the velocity value set by the velocity of the last pressed MIDI key.

Note that you could also use the Clock Module (↑2.14, Clock) and to get the desired rate using the Event Frequency Divider Module (↑13.4, Event Frequency Divider). The advantage of that method is that you can choose the frequency from the Instrument Panel.

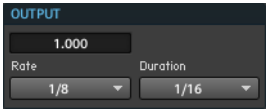


Fig. 2.65 Use the Rate drop-down menu in the Sync Pulse Module's Function page to choose the rate of clock Events sent to the Sequencer Module.

2.16 Song Position

Fig. 2.66 Song Position Module

2.16.1 Overview

The Song Position Module acts as a source for the song position value, counted in 96th notes from the song start. The song position is determined with the MIDI Clock which is either an incoming external MIDI Clock signal or the internal Master Clock. When you initially press the Run button in the Main Toolbar, the song position starts running from zero. Pressing the Run button again pauses the Clock and the song position stays put. Press the Run button a third time to continue running the song position from the current position. If instead you press the Stop button, you reset the song position to zero.

Application

Use the Song Position Module in a sequencer that plays back patterns that are dependent on the song position. This way you can arrange whole songs within REAKTOR and without the help of a host sequencer.
Typically you connect either output signal to the "A" input port of the Modulo Module (↑4.9, Modulo) and a Constant (↑4.1, Constant) of "6" to the "B input port, yielding a 16th note count at the "Div" (divide) output port.

2.16.2 Ports

Output Ports

- **(96)** "96" is the Event output port for the integer count of 96th notes (that's 24 Events per quarter note). Use the Modulo Module (↑4.9, Modulo) to get counts of other denominations.
- **(96a)** "96a" is the Audio output port for the sample-accurate fractional song position measured in 96th notes (24 per quarter note).

2.16.3 Example: Beat and Bar Counter

The example, shown in the Structure below, keeps track of the number of measures from the song start and the current count, in the 4/4 time signature. The "96" output port of the Song Position outputs Events with the song position value in 96th notes from the song start. To get the number of quarter notes from the song start, use the Modulo Module (↑4.9, Modulo) with the value "24" at its "B" input port (since 96 / 24 = 4). This outputs the number of quarter notes from the song start at the first Modulo Module's "Div" output port. Use another Modulo Module (↑4.9, Modulo), this time with "4" at the "B" input port, to calculate the number of measures (bars) from the song start. The value "4" at the "B" input port is because in our time signature we have four beats per bar. The number of bars from the song start is then output at the "Div" output port and the current beat in quarter notes ("0", "1", "2", or "3") is given at the "Mod" output port.

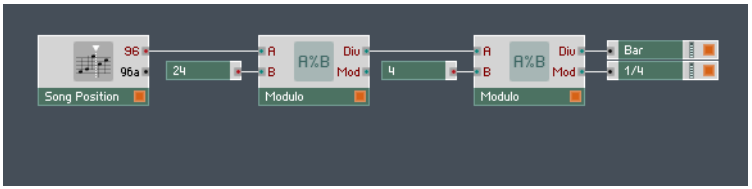


Fig. 2.67 Use the Song Position Module in conjunction with Modulo Modules to keep track of the beat and measure count.

2.17 Channel Message

Fig. 2.68 Channel Message Module

2.17.1 Overview

The Channel Message Module is a Monophonic Event source for incoming MIDI channel messages from an external MIDI device (keyboard or sequencer etc.), or internally from other Instruments within the Ensemble. The output order of the output ports (see below) is strictly defined, from top to bottom. This ensures that the message type (e.g. control change) and message source (e.g. controller 7 on channel 1) are always reported before the actual value.

Application

The Channel Message Module gives you an overview of what is happening at the parent Instrument's MIDI In port. You could, for example, use the MIDI Channel value at the "Ch" output port to route MIDI messages from different MIDI channels to different parts of the Structure, even different Instruments in an Ensemble. You can also use the Channel Message Module as a replacement for several Controller Modules (↑2.8, Controller). This way you only need one Module to monitor incoming MIDI controller messages.

2.17.2 Ports

Output Ports

- **(St)** "St" is the Event output port which reports the type of MIDI message received. 0 = Note Off, 1 = Note On, 2 = Poly Aftertouch, 3 = Control Change, 4 = Program Change, 5 = Channel Aftertouch, and 6 = Pitchbend.
- **(Ch)** "Ch" (MIDI channel number) is the Event output port that sends the MIDI channel number of the incoming messages. The channel values are in the range [1 … 16].
- **(Nr)** "Nr" (MIDI note number) is the Event output port that sends the MIDI note number, CC (Control Change) number, or Program Change value of the incoming MIDI message. The range of these value is [0 … 127].
- **(Val)** "Val" (value) is the Event output port that sends the velocity value of a note, the pressure value of an Aftertouch message, or the value of a CC (Control Change) or Pitchbend message. For external MIDI messages, the values are 7-bit quantized (14-bit for Pitchbend). For internal MIDI messages, the messages are sent as 32-bit floating-point values.

2.17.3 Properties: Function Page

Setting the Range of Output Values

The range of the output signal is mapped to the interval [Min … Max]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties. The default setting is Min = 0 and Max = 1. Another common setting is 0 to 127, which can aid interpretability of values in some situations (Program Change for example).

Fig. 2.69 Use the Min and Max edit fields in the Range area of the Function page to set the range of output values [Min … Max].

2.17.4 Example: Note to Chord

The To Voice Modules (↑14.11, To Voice) give you direct access to the values carried by each Voice. A handy application is to generate chords from incoming Monophonic MIDI Notes. The Structure that implements this feature is shown in the figure below for an Instrument that has its number of Polyphonic Voices set to "3". The chord that is played when a key is pressed is the minor chord: the incoming MIDI Note is assigned to the first Voice, the minor third to the second Voice, and the perfect fifth to the third Voice. As can be seen in the Structure below, Add Modules (↑4.2, Add, +) and the consequent To Voice Modules (↑14.11, To Voice) are used to assign each Voice the corresponding transposed MIDI Note. The output of each To Voice Module carries a Polyphonic signal where each Voice, except the one specified by the "V" input port, is zero. The outputs of the To Voice Modules are added together to result in a Polyphonic signal where each Voice has a defined value.

To get the base note of the chord, the "Nr" output port of the Channel Message Module is used. Although this output can send values other than MIDI Notes, triggering the value to be used only when a Gate On Event from the Single Trig Gate Module (↑2.4, Single Trig Gate) arrives, ensures that only MIDI Note values are used. The Separator Module (↑13.14, Separator), on the other hand, ensures that only Gate On Events trigger the note number to be forwarded to the chord generation algorithm.

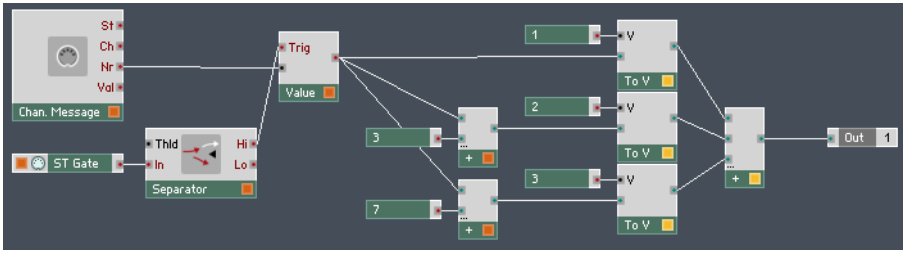


Fig. 2.70 The Structure to create a chord from an incoming Monophonic MIDI Note.

3 MIDI Out

REAKTOR can generate as well as process MIDI messages. MIDI Out Modules enable you to send MIDI messages to other REAKTOR Instruments as well as external MIDI devices. Each Module in this category corresponds to a MIDI In Module. Some MIDI Out Modules can also be used in an internal or OSC connection. Although the MIDI Out Modules are not so often used when building basic synthesizers, the realm of MIDI effects and algorithmic sequencers offers a wide array of musical possibilities, most of which can be implemented in REAKTOR.

3.1 Note Pitch/Gate

Fig. 3.1 A Note Pitch Module

3.1.1 Overview

The Note Pitch/Gate Module converts Monophonic or Polyphonic Events to MIDI Note events. Each Event at the "G" (gate) input port generates a MIDI messages at the parent Instrument's MIDI Out which REAKTOR forwards either internally to another Instrument or to an external MIDI device. The value of the Event at the "G" (gate) input port specifies the note velocity where "0" corresponds to a Note Off Event. The pitch of the outgoing MIDI Note is determined by the value at the "P" (pitch) input port. The range of the input values at the "P" (pitch) input port is set Function page.

Application

Use the Note Pitch/Gate Module to when you want to control external MIDI devices or other Instruments with a REAKTOR sequencer or some other type of creative MIDI effect. For example, you could build your own custom envelope follower type of effect that converts an incoming Audio signal into a MIDI signal that you can use to control a hardware synthesizer.

3.1.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Polyphonic Audio input port for the MIDI pitch values of the incoming MIDI notes.
- **(G)** "G" (gate) is the Polyphonic Event input port for Gate On and Gate Off Events. The values of the Events are interpreted as the note velocity where "0" corresponds to a MIDI Note Off event.

3.1.3 Properties: Function Page

Setting the Range of Input Values

The range of the values incoming at the "P" (pitch) input port [Min … Max] is mapped to the range of the outgoing MIDI signal [0 … 127]. An incoming Event with a value equal to "Min" sets the MIDI Note Pitch to "0", an Event with value equal to "Max" sets the MIDI Note Pitch to "127", as shown in the second figure below.

► To set the values for "Min" and "Max", enter the desired values into the corresponding edit fields in the Note Pitch/Gate Module's Function page, as shown in the screenshot below.

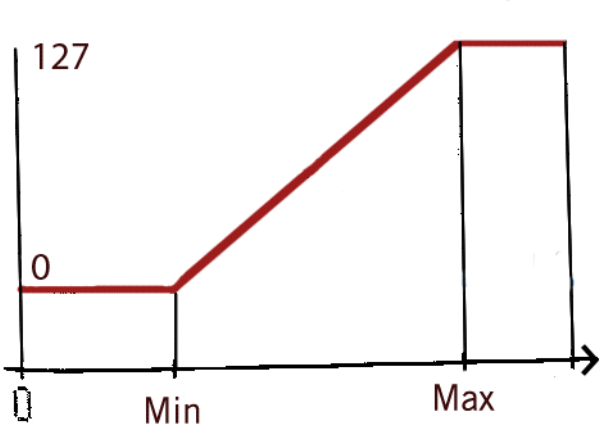


Fig. 3.2 The x-axis of the graph shows the values incoming at the "P" (pitch) input port whereas the y-axis shows the outgoing MIDI Note values. These values are interpolated linearly between over the interval [Min ... Max].

3.1.4 Example: Gate, Pitch, and Pitchbend Out

This example shows how to employ the Note Pitch/Gate Module and the Pitchbend Out Module (↑3.2, Pitchbend Out) to send this information to an external MIDI device from your Instrument's MIDI Out. The Structure for such an application is shown in the figure below, where a Macro has been employed to process MIDI signals (perhaps an arpeggiator of sorts). The Polyphonic pitch and Gate signals are forwarded to the "P" and "G" input ports of the Note Pitch/Gate Module. The range of the pitch signal has been set to [0 ... 127] which is also the default range for the "P" input port, set in the Note Pitch/Gate Module's Function page. The Gate signal at the "G" input port should be in the range [0 ... 1]. The MIDI Effect Macro also generates Pitchbend signals which are in the range [-2 ... 2]. This range of values, arriving at the Pitchbend Out Module (↑3.2, Pitchbend Out), should be the same as the range set in its Function page. Note that the Pitchbend signal should be Monophonic.

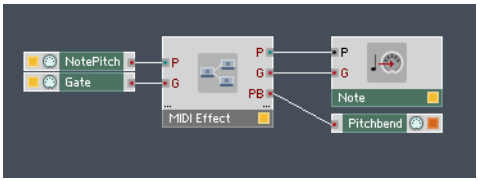


Fig. 3.3 This Structure shows how to send signals generated by a MIDI effect to the Instrument's MIDI Out.

3.2 Pitchbend Out

Fig. 3.4 A Pitchbend Module

3.2.1 Overview

The Pitchbend Out Module is a converts Monophonic or Polyphonic Events to MIDI Pitchbend (pitchwheel change) events. The range of input values [Min ... Max] is set in the Function page and is mapped to the output range [-8192 ... 8191]. Each Event at the input port generates a MIDI message at the MIDI Out of the parent Instrument. Depending on the Instrument's MIDI Out settings, this message is relayed to another REAKTOR Instrument or is forwarded to an external MIDI device.

Application

The Pitchbend Out Module lets you control another Instrument or a hardware synthesizer's pitch as if with a pitchwheel. Use this Module if you want to build a note sequencer that also sends MIDI Pitchbend values for each note.

3.2.2 Ports

- **(In)** "In" is the Polyphonic Event input port for Pitchbend values that are in the range [Min ... Max], as set in the Module's Function page.

3.2.3 Properties: Function Page

Setting the Range of Input Values

The range of input values [Min ... Max] is mapped to the range of the outgoing MIDI Pitchbend values [-8192 ... 8291]. An incoming Event with a value equal to "Min" sets the MIDI Pitchbend value to "-8192", an Event with value equal to "Max" sets the MIDI Pitchbend value to "8191", as shown in the second figure below.

► To set the values for "Min" and "Max", enter the desired values into the corresponding edit fields in the Pitchbend Out Module's Function page, as shown in the screenshot below.

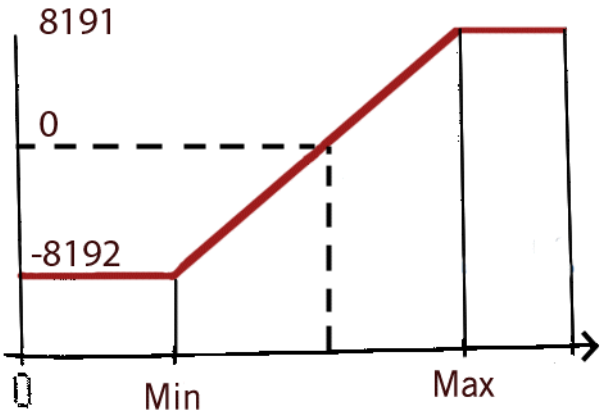


Fig. 3.5 The x-axis of the graph shows the values at the input port whereas the y-axis shows the outgoing MIDI Pitchbend values. These values are interpolated linearly between over the interval [Min ... Max]. An outgoing MIDI Pitchbend value of "0" corresponds to a neutral pitchwheel position.

3.2.4 Example: Gate, Pitch, and Pitchbend Out

This example shows how to employ the Note Pitch/Gate Module and the Pitchbend Out Module to send this information to an external MIDI device from your Instrument's MIDI Out. The Structure for such an application is shown in the figure below, where a Macro has been employed to process MIDI signals (perhaps an arpeggiator of sorts). The Polyphonic pitch and Gate signals are forwarded to the "P" and "G" input ports of the Note Pitch/Gate Module. The range of the pitch signal has been set to [0 … 127] which is also the default range for the "P" input port, set in the Note Pitch/Gate Module's Function page. The Gate signal at the "G" input port should be in the range [0 … 1]. The MIDI Effect Macro also generates Pitchbend signals which are in the range [-2 … 2]. This range of values, arriving at the Pitchbend Out Module, should be the same as the range set in its Function page. Note that the Pitchbend signal should be Monophonic.

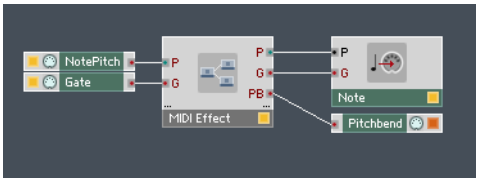


Fig. 3.6 This Structure shows how to send signals generated by a MIDI effect to the Instrument's MIDI Out.

3.3 Controller Out

Fig. 3.7 Controller Out Module

3.3.1 Overview

The Controller Out Module converts a Monophonic Event signal to MIDI CC (Control Change) messages. Each incoming Event generates a MIDI message at the MIDI Out of the parent Instrument. The range of input values [Min … Max] is set in the Function page and is mapped to the MIDI output range [0 … 127].

Depending on the Instrument's MIDI Out settings, this message is relayed to another REAKTOR Instrument or is forwarded to an external MIDI device. You can keep these messages from being relayed to the external MIDI device by engaging the Disable for External MIDI checkbox in the Function page. The MIDI CC value (number of the controller) is also set in the Function page.

Application

The Controller Out Module enables you to link certain parameters in your Instrument's Structure to the MIDI CC (Control Change) of other Instruments or external MIDI devices. This way you can build a custom sequencer in REAKTOR for your favorite hardware of software synthesizer. With the Controller Out Module, each step of the sequencer can then address the exactly the MIDI CC parameters that are relevant for your synthesizer.

3.3.2 Ports

- **(In)** "In" is the Monophonic Event input port for the MIDI CC (Controller Change) values that are in the range [Min … Max], as set in the Module's Function page.

3.3.3 Properties: Function Page

Keeping the Controller Out Module Always Active

The Controller Out Module only forwards values to the parent Instrument's MIDI Out if it is part of an active signal flow. Connecting it directly to a Knob Module, for example, does not suffice. For such a case the Always Active feature is useful.

► To keep the Controller Out Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Setting the Range of Input Values

The range of the values at the input port [Min … Max] is mapped to the range of the outgoing MIDI signal [0 … 127]. An incoming Event with a value equal to "Min" sets the MIDI message value to "0", an Event with value equal to "Max" sets the MIDI message value to "127", as shown in the second figure below.

► To set the values for "Min" and "Max", enter the desired values into the corresponding edit fields in the Controller Out Module's Function page, as shown in the screenshot below.

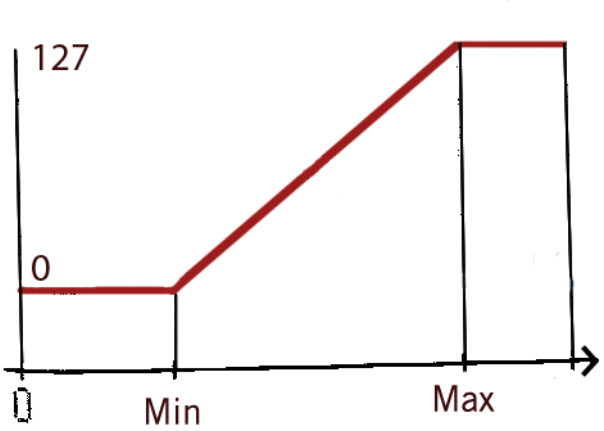
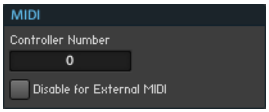


Fig. 3.8 The x-axis of the graph shows the values at the input port whereas the y-axis shows the outgoing MIDI CC message values. These values are interpolated linearly between over the interval [Min ... Max].

Setting the CC Value of the Forwarded MIDI Message

The Controller Out Module sends MIDI messages with a set CC (Controller Change) value.

► To set the CC value of the MIDI messages that sent from the parent Instrument's MIDI Out, enter the desired value into the Controller Number edit field shown in the screenshot below.



You can keep the MIDI CC messages from being sent to external MIDI devices by engaging the Disable for External MIDI checkbox.

The CC values of some standard MIDI controllers are:

- **1** Modulation Wheel
- **2** Breath Controller
- **7** Volume
- **10** Panpot
- **64** Sustain Switch
- **65** Portamento Switch
- **66** Hold Switch (Sostenuto)

See your MIDI keyboard's MIDI implementation chart to find out which controllers it can receive.

Disabling the MIDI CC Messages for External MIDI

Sometimes you want the MIDI CC messages from a certain Controller Out Module to reach only internal REAKTOR Instruments and not external MIDI devices.

- To keep the MIDI CC messages of a certain Controller Out Module from being sent to external MIDI devices, go to its Function page and engage the Disable for External MIDI checkbox (shown in the screenshot above).

3.3.4 Example: Modulation Wheel with Knob

This example shows how to send MIDI CC messages from your Instrument's MIDI Out. In particular, the CC value of "1" has been chosen, which by convention corresponds to Modulation Wheel messages. So although you can choose the CC value to be anything you like in the range [1 … 128], this example deals with Modulation Wheel messages. The Structure, however, would look the same for all cases. It is shown in the figure below. Just insert a Controller Out Module into the Structure and connect a Knob Module (↑1.1, Fader/Knob) to it (if that is what you want to use to control the MIDI CC messages). Next, go to the Controller Out Module's Function page and set the Controller Number edit field to "1". Also, engage the Always Active checkbox to activate the Module (if need be). Last, make sure that the "Min" and "Max" values for the Controller Out Module match the range of values set for the Knob Module, and you are set to send MIDI CC messages from your Structure and even Instrument Panel!

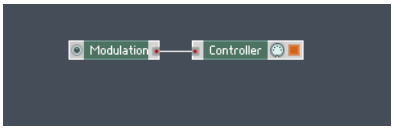


Fig. 3.9 Use the Controller Out Module to send MIDI CC messages.

3.4 Channel Aftertouch Out

Fig. 3.10 Channel Aftertouch Out Module

3.4.1 Overview

The Channel Aftertouch Out Module converts a Monophonic Event signal to MIDI Channel Aftertouch events. Each incoming Event generates a MIDI Channel Aftertouch message at the MIDI Out of the parent Instrument. The range of input values [Min ... Max] is set in the Function page and is mapped to the MIDI output range [0 ... 127]. Depending on the Instrument's MIDI Out settings, this message is relayed to another REAKTOR Instrument or is forwarded to an external MIDI device.

Application

If you wish REAKTOR to send Channel Aftertouch messages to another Instrument or an external MIDI device such as a hardware synthesizer, placing this Module into the Instrument's Structure is necessary.

3.4.2 Ports

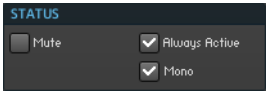
- **(In)** "In" is the Monophonic Event input port for the Aftertouch values that are in the range [Min ... Max], as set in the Module's Function page.

3.4.3 Properties: Function Page

Keeping the Channel Aftertouch Out Module Always Active

The Channel Aftertouch Out Module only forwards values to the parent Instrument's MIDI Out if it is part of an active signal flow. Connecting it directly to a Knob Module, for example, does not suffice. For such a case the Always Active feature is useful.

► To keep the Controller Out Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Setting the Range of Input Values

The range of the values at the input port [Min ... Max] is mapped to the range of the outgoing MIDI signal [0 ... 127]. An incoming Event with a value equal to "Min" sets the MIDI message value to "0", an Event with value equal to "Max" sets the MIDI message value to "127", as shown in the second figure below.

► To set the values for "Min" and "Max", enter the desired values into the corresponding edit fields in the Channel Aftertouch Out Module's Function page, as shown in the screenshot below.

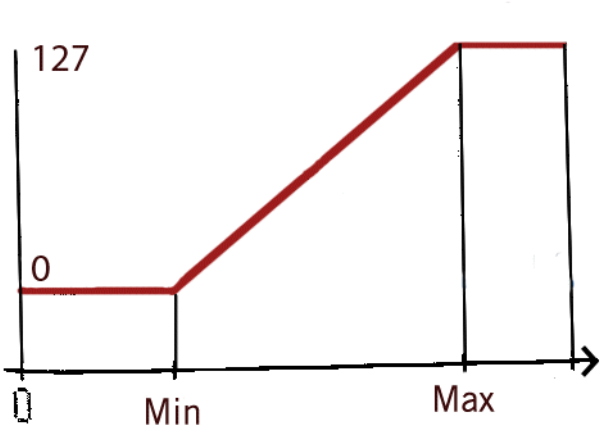


Fig. 3.11 The x-axis of the graph shows the values at the input port whereas the y-axis shows the outgoing MIDI Channel Aftertouch message values. These values are interpolated linearly between over the interval [Min ... Max].

3.4.4 Example: Envelope to Channel Aftertouch

This example shows how to turn an envelope signal into Channel Aftertouch MIDI messages which are send from the parent Instrument's MIDI Out. The Structure for this is shown in the figure below. First, insert an ADSR Module (↑9.13, ADSR - Env) with the desired curve parameters and Gate signal from the Gate Module (↑2.3, Gate). Note that since there usually are no Aftertouch messages after all keys have been released, the "R" parameter for the ADSR Module (↑9.13, ADSR - Env) has been left disconnected. Next, you need to turn the Polyphonic Audio output signal of the ADSR Module into a Monophonic Event signal. Do this with the help of an Event Smoother (↑14.15, Event Smoother) and Event Voice Combiner All Module (↑14.4, Event V.C. All). Route the result to the Channel Aftertouch Out Module.

Presumably, the range of the envelope signal is [0 … 1]. Make sure that the "Min" and "Max" values in the Channel Aftertouch Out Module's Function page also have been set to "0" and "1" so that the incoming values are correctly converted for the outgoing MIDI messages. Also, if your Module's are not active, engage the Always Active checkbox in the Channel Aftertouch Out Module's Function page.

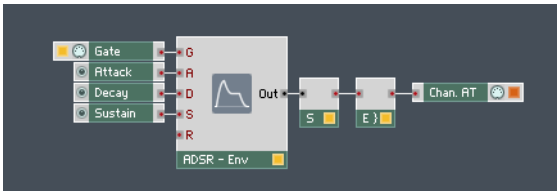


Fig. 3.12 Sending Channel Aftertouch messages in the shape of an envelope.

3.5 Poly Aftertouch Out

Fig. 3.13 Poly Afertouch Out Module

3.5.1 Overview

The Poly Aftertouch Out Module assigns each Polyphonic MIDI Note at the "P" (pitch) input port a MIDI Channel Aftertouch value according to the value of the corresponding Voice at the "AT" (aftertouch) input port. Each Event at the "AT" (aftertouch) input port generates a MIDI message at the parent Instrument's MIDI Out. The value of the Event at the "AT" (aftertouch) input port specifies the note's Aftertouch pressure and should be in the range [0 … 1]. The range of the input values at the "P" (pitch) input port is set Function page.

Application

The Poly Aftertouch Out Module enables you to send Polyphonic MIDI Channel Aftertouch messages to other REAKTOR Instruments or external MIDI devices. There are very few MIDI devices that can handle Poly Aftertouch messages.

3.5.2 Ports

Input Ports

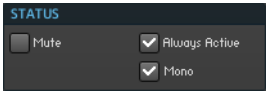
- **(P)** "P" (pitch) is the Polyphonic Audio input port for the MIDI pitch values of the keys that the outgoing Poly Channel Aftertouch messages belong.
- **(AT)** "AT" (aftertouch) is the Polyphonic Event input port for the Poly Channel Aftertouch values. The incoming values should be in the range [0 … 1] and are mapped to the outgoing MIDI range [0 … 127].

3.5.3 Properties: Function Page

Keeping the Poly Aftertouch Out Module Always Active

The Poly Aftertouch Out Module only forwards values to the parent Instrument's MIDI Out if it is part of an active signal flow. Connecting it directly to a Knob Module, for example, does not suffice. For such a case the Always Active feature is useful.

► To keep the Poly Aftertouch Out Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Setting the Range of Input Values

The range of the values incoming at the "P" (pitch) input port [Min ... Max] is mapped to the range of the outgoing MIDI signal [0 ... 127]. An incoming Event with a value equal to "Min" sets the MIDI Note Pitch to "0", an Event with value equal to "Max" sets the MIDI Note Pitch to "127", as shown in the second figure below.

► To set the values for "Min" and "Max", enter the desired values into the corresponding edit fields in the Poly Aftertouch Out Module's Function page, as shown in the screenshot below.

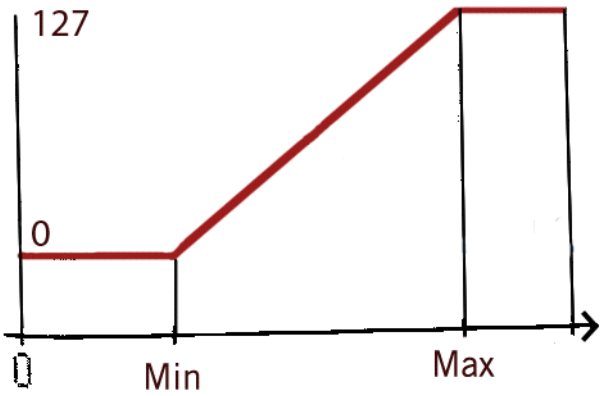


Fig. 3.14 The x-axis of the graph shows the values incoming at the "P" (pitch) input port whereas the y-axis shows the outgoing MIDI Note values. These values are interpolated linearly between over the interval [Min ... Max].

3.5.4 Example: Envelope to Poly Aftertouch

This example shows how to turn an envelope signal into Poly Aftertouch MIDI messages which are send from the parent Instrument's MIDI Out. The Structure for this is shown in the figure below. First, insert an ADSR Module (↑9.13, ADSR - Env) with the desired curve parameters and Gate signal from the Gate Module (↑2.3, Gate). Note that since there usu-

ally are no Aftertouch messages after a key has been released, the "R" parameter for the ADSR Module (↑9.13, ADSR - Env) has been left disconnected. Next, you need to turn the Polyphonic Audio output signal of the ADSR Module (↑9.13, ADSR - Env) into an Event signal. Do this with the help of an Event Smoother (↑14.15, Event Smoother) Module. Route the result to the "AT" input port of the Poly Aftertouch Out Module. The pitch to which each envelope signal belongs to can directly be received from the Note Pitch Module (↑2.1, Note Pitch). This is because the Gate signal for the envelope and the note pitch are tied to the same Voice. Just connect the output of the Note Pitch Module (↑2.1, Note Pitch) to the "P" input port of the Poly Aftertouch Out Module.
The pitch values coming from the Note Pitch Module (↑2.1, Note Pitch) are in the range [0 … 127]. Make sure that the (incoming) pitch range set in the Function page of the Poly Aftertouch Out Module is the same. Also, make sure that the range of the envelope signal that reaches the Poly Aftertouch Out Module's "AT" input port is [0 … 1] so that these values are converted correctly to the MIDI scale. If your Module's are not active, engage the Always Active checkbox in the Poly Aftertouch Out Module's Function page.

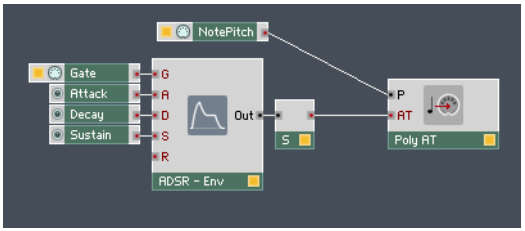


Fig. 3.15 Sending Poly Aftertouch messages in the shape of an envelope.

3.6 Sel Poly Aftertouch Out

Fig. 3.16 Sel Poly Aftertouch Out Module

3.6.1 Overview

The Sel Poly Aftertouch Out Module converts a Monophonic Event signal to MIDI Poly Aftertouch events. Each incoming Event generates a MIDI message at the MIDI Out of the parent Instrument. The MIDI Note number of the key to which the incoming Aftertouch value (key pressure) is sent, is set in the Function page. The range of input values [Min …

Max] is set in the Function page and is mapped to the MIDI output range [0 … 127]. An Event with value equal to "Min" sets the Aftertouch value to "0" and an Event with value equal to "Max" sets the Aftertouch value to "127". Depending on the Instrument's MIDI Out settings, this message is relayed to another REAKTOR Instrument or is forwarded to an external MIDI device.

Application

The Sel Poly Aftertouch Out Module enables you to send Polyphonic MIDI Channel Aftertouch messages to specific MIDI Notes of other REAKTOR Instruments or external MIDI devices. There are very few MIDI devices that can handle Poly Aftertouch messages.

3.6.2 Ports

(In) "In" is the Monophonic Event input port for the Aftertouch values that are in the range [Min … Max], as set in the Module's Function page.

3.6.3 Properties: Function Page

Keeping the Sel Poly Aftertouch Out Module Always Active

The Sel Poly Aftertouch Out Module only forwards values to the parent Instrument's MIDI Out if it is part of an active signal flow. Connecting it directly to a Knob Module, for example, does not suffice. For such a case the Always Active feature is useful.

► To keep the Sel Poly Aftertouch Out Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Setting the Range of Input Values

The range of the values at the input port [Min … Max] is mapped to the range of the outgoing MIDI signal [0 … 127]. An incoming Event with a value equal to "Min" sets the MIDI Aftertouch value "0", an Event with value equal to "Max" sets the MIDI Aftertouch value to "127", as shown in the second figure below.

► To set the values for "Min" and "Max", enter the desired values into the corresponding edit fields in the Sel Poly Aftertouch Out Module's Function page, as shown in the screenshot below.

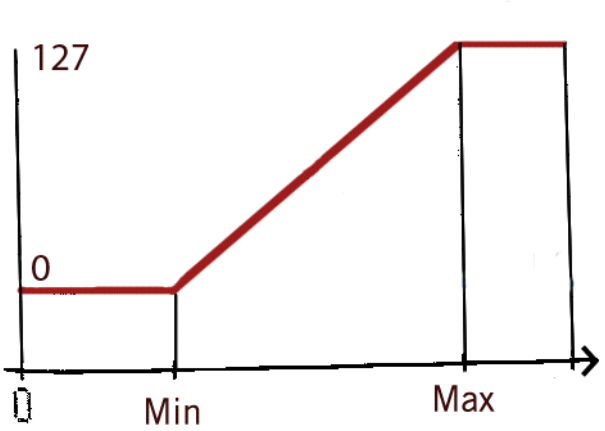


Fig. 3.17 The x-axis of the graph shows the values at the input port whereas the y-axis shows the outgoing MIDI Aftertouch message values. These values are interpolated linearly between over the interval [Min ... Max].

Setting the MIDI Note Value of the Aftertouch Message

The Sel Poly Aftertouch Out Module sends Aftertouch messages only to one key with a specific MIDI Note value.

► To set the MIDI Note value of the Aftertouch messages outgoing from the Sel Poly Aftertouch Out Module, enter the desired value into the Midi Note edit field shown in the screenshot below.

3.6.4 Example: Envelope to Selective Aftertouch

This example shows how to turn an envelope signal into Aftertouch MIDI messages which are sent from the parent Instrument's MIDI Out to one particular MIDI Note. The Structure for this is shown in the figure below. First, insert an ADSR Module (↑9.13, ADSR - Env) with the desired curve parameters and Gate signal from the Sel Gate Module (↑2.5, Sel Note Gate) (which probably is set to receive its Gate signals from the same MIDI Note). There usually are no Aftertouch messages after the key has been released, so the "R" parameter for the ADSR Module (↑9.13, ADSR - Env) has been left disconnected. Next, you need to turn the Audio output signal of the ADSR Module (↑9.13, ADSR - Env) into an Event signal. Do this with the help of an Event Smoother (↑14.15, Event Smoother) Module. Route the result to the Sel Poly Aftertouch Out Module, and make sure that all Modules are set to Monophonic Mode.
Presumably, the range of the envelope signal is [0 ... 1]. Make sure that the "Min" and "Max" values in the Sel Poly Aftertouch Out Module's Function page also have been set to "0" and "1" so that the incoming values are correctly converted for the outgoing MIDI messages. Now make sure that the MIDI Note edit field of the Sel Poly Aftertouch Module refers to the MIDI note to which you wish to send the Aftertouch messagse and that the Note edit field of the Sel Gate Module's (↑2.5, Sel Note Gate) Function page refers to the MIDI note from which you wish to receive the Gate message. Last, if your Module's are not active, engage the Always Active checkbox in the Sel Aftertouch Out Module's Function page.

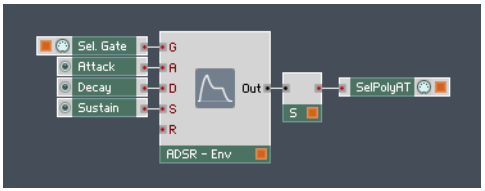


Fig. 3.18 Sending Aftertouch messages in the shape of an envelope to one articular MIDI Note.

3.7 Program Change Out

Fig. 3.19 Program Change Out Module

3.7.1 Overview

The Program Change Out Module converts a Monophonic Event signal to MIDI Program Change messages. Each incoming Event generates a MIDI message at the parent Instrument's MIDI Out. The range of input values [Min … Max] is set in the Function page and is mapped to the MIDI output range [0 … 127]. An Event with value equal to "Min" forwards a Program Change to "0" and an Event with value equal to "Max" forwards a Program Change to "127".
Depending on the Instrument's MIDI Out settings, these Program Change messages are relayed to another REAKTOR Instrument or are forwarded to an external MIDI device. Start/ Stop Out

Application

The Program Change Out Module is necessary if you want to control the Snapshots or patches of other Instruments or external MIDI devices, respectively. You might want to build a sequencer within REAKTOR that controls a hardware synthesizer and whose patterns are tied to specific patches in that synthesizer. The Program Change Out Module would be the tool to help you realize such a setup.

3.7.2 Ports

- **(In)** "In" is the Monophonic Event input port for the Program Change values that are in the range [Min … Max], as set in the Module's Function page.

3.7.3 Properties: Function Page

Setting the Range of Input Values

The range of the values at the input port [Min … Max] is mapped to the range of the outgoing MIDI Program Change messages [0 … 127]. An incoming Event with a value equal to "Min" sends a Program Change value of "0", an Event with value equal to "Max" sends a Program Change value of "127", as shown in the second figure below.

► To set the values for "Min" and "Max", enter the desired values into the corresponding edit fields in the Program Change Out Module's Function page, as shown in the screenshot below.

3.7.4 Example: Snapshot Recall and Program Change

This example illustrates how to forward MIDI Program Change messages to the Snapshot Module (↑14.23, Snapshot) to recall REAKTOR Snapshots and conversely, how to have REAKTOR Snapshot recall operations trigger MIDI Program Change messages to be sent from REAKTOR. The Structure that achieves this is shown in the Structure below.

The Program Change Module (↑2.12, Program Change) should send a value in the range [1 … 128] to address one of the 128 Snapshots in the active Bank to be recalled. For that, go to the Program Change Module's (↑2.12, Program Change) Function page and set the "Min" and "Max" values to "1" and "128", respectively. Use the Order Module (↑13.12, Order) to first specify at the "Snp" input port the Snapshot number to be recalled (upon receiving a Program Change Event) and only then trigger the Snapshot recall operation at the "Recl" input port. Similarly, Use the Value Module (↑13.15, Value) to send the Event carrying the recalled Snapshot number to the Program Change Out Module when a Snapshot recall operation takes place. The "Min" and "Max" values in the Program Change Out Module's Function page should also be set to "1" and "128", respectively.

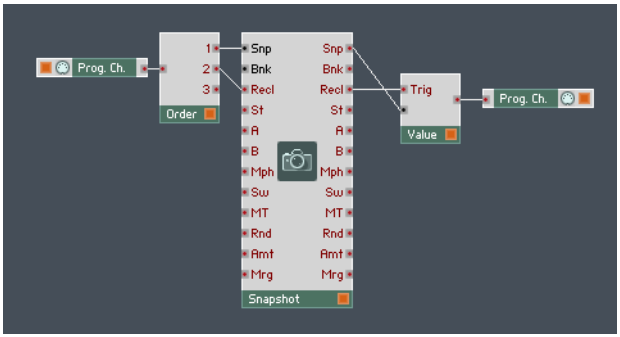


Fig. 3.20 You can use the Program Change (Out) and Snapshot Modules to link Snapshot operations to external MIDI devices.

3.8 Start/Stop Out

Fig. 3.21 Start/Stop Out Module

3.8.1 Overview

The Start/Stop Out Module converts Monophonic Events into MIDI Clock Start, Stop, and Reset messages. The output messages serve to synchronize external MIDI devices with processes in your Structure (like LFOs, Sequencers, etc.). The Start/Stop Out Module's "G" (gate) input port converts positive (Gate On) Events into MIDI Clock Start messages and zero valued or negative (Gate Off) Events into MIDI Clock Stop Messages at the parent Instrument's MIDI Out. A positive Event at the "Rst" (reset) input port sends a MIDI Clock Reset message from the parent Instrument's MIDI Out. Such a message causes the song position to be reset to zero.

Application

You might have a sequencer that is started, stopped, and reset with separate toggle buttons (in the Instrument Panel). Then, in addition to the timing relevant sequencer components, you would also connect the Start/Stop Out Module to these buttons. This way you can communicate to external MIDI devices whether your sequencer is running or not and whether the song position should be reset or not.

3.8.2 Ports

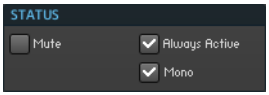
- **(G)** "G" (gate) is the Monophonic Event input port for Gate On and Gate Off Events which correspond to MIDI Clock Start and MIDI Clock Stop messages, respectively. Positive Events are counted as Gate On Events and negative or zero valued Events are counted as Gate Off Events.
- **(Rst)** "Rst" (reset) is the Monophonic Event input port for positive Events which trigger MIDI Clock Reset messages to be sent from the parent Instrument's MIDI Out. A MIDI Clock Reset message resets the song position.

3.8.3 Properties: Function Page

Keeping the Start/Stop Out Module Always Active

The Start/Stop Out Module only forwards values to the parent Instrument's MIDI Out if it is part of an active signal flow. Connecting it directly to a Knob Module, for example, does not suffice. For such a case the Always Active feature is useful.

► To keep the Start/Stop Out Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.



3.8.4 Example: MIDI Start/Stop Messages from Panel

The Structure shown below illustrates how you can easily send MIDI Clock Start and Stop messages from the Instrument Panel using the Start/Stop Out Module and two Button Modules (↑1.2, Button). Just right-click each input port and choose the *Create Control* menu entry. Rename each button as you wish and make sure that the "Reset" button is in Trigger Mode (set in its Function page). Also, to activate all Modules, engage the Always Active checkbox in the Start/Stop Out Module's Function page.

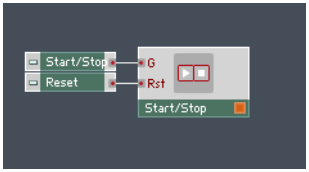


Fig. 3.22 With the Start/Stop Out Module you can directly send MIDI Clock Start, Stop, and Reset messages from the parent Instrument's MIDI Out.

3.9 Clock Out

Fig. 3.23 Clock Out Module

3.9.1 Overview

The Clock Out Module converts positive input Events to (1/96) MIDI Clock signals at the parent Instrument's MIDI Out.

Application

Use the Clock Out Module to synchronize the MIDI Clock of an external MIDI device to the internal Structure of an Instrument.

For an example regarding the Clock Out Module, please refer to the example in the next entry, the Song Position Out Module (↑3.10, Song Position Out).

3.9.2 Ports

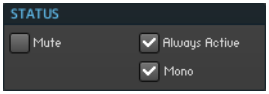
(In) "In" is the Monophonic Event input port for positive valued Events. Such an Event creates a MIDI Clock event at the parent Instrument's MIDI Out.

3.9.3 Properties: Function Page

Keeping the Clock Out Module Always Active

The Clock Out Module only forwards values to the parent Instrument's MIDI Out if it is part of an active signal flow. Connecting it directly to a Knob Module, for example, does not suffice. For such a case the Always Active feature is useful.

► To keep the Clock Out Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.



3.10 Song Position Out

3.10.1 Overview

The Song Position Out Module sends song position values to the parent Instrument's MIDI Out. The MIDI song position value is taken from the "Pos" (position) input port and is counted in units of 96th notes. The song position value is only sent to the MIDI Out when a positive valued Event arrives the "Trig" input port.

Application

Use the Song Position Out Module to control external sequencers which accept song position messages and for which you have arranged a sequence of patterns.

3.10.2 Ports

Input Ports

- **(Trig)** "Trig" (trigger) is the Event input port for trigger Events. Every Event with a positive value at this input port creates a MIDI song position event with the "Pos" value at the parent Instrument's MIDI Out.
- **(Pos)** "Pos" (song position) is the Audio input port for the song position value (as a multiple of 96th notes). This value is forwarded to the parent Instrument's MIDI Out when a trigger Event arrives at the "Trig" (trigger) input port.

3.10.3 Example: Song Position and Clock from Structure

This example shows how to build a source for clock signals running at the rate of 96th notes in respect to a BPM value that you choose from the Instrument Panel. These clock Events are output to the parent Instrument's MIDI Out, including the song position value calculated from these clock Events. The final Structure that achieves this functionality is shown in the figure below.

First, set up the source of your clock signals. This should be a Ramp Module (↑6.36, Ramp Oscillator) working in Hi-Res Mode (as chosen by the Mode drop-down menu in its Function page). The amplitude of the ramp is arbitrary, so just set it to "1" by connecting a Constant (↑4.1, Constant) to its "A" (amplitude) input port. Then you need to calculate the frequency of the ramp. The song position Events and clock events should run at the rate of 96th notes. Therefore, the frequency value (at the Ramp Module's (↑6.36, Ramp Oscillator) "F" input port) should be F = BPM * 96 / 60. This is because BPM / 60 gives you the

number of Events (or beats) per second (the units that the "F" input port uses) and since you want 96th notes (or 96 notes per beat or second), multiply the previous result by "96". This calculation is reflected by the Structure shown below.

Next, you need to generate a trigger signal from the Ramp Module's output. This is most easily done with the combination of the Differentiator Module (↑10.22, Differentiator) and the A to E Trig Module (↑14.8, A to E (Trig)). The Differentiator Module's (↑10.22, Differentiator) output signal exhibits a positive zero crossing whenever the incoming (ramp) signal goes from a falling slope to a rising slope. This happens at the transient part of the ramp and at the frequency specified at the Ramp Module's (↑6.36, Ramp Oscillator) "F" input port. The positive zero crossing at the Differentiator Module's (↑10.22, Differentiator) output is then used to trigger an Event at the A to E Trig Module's (↑14.8, A to E (Trig)) output port. The Event carries the value set at that Module's "A" input port. Since the Clock Out Module (↑3.9, Clock Out) only interprets positive valued Events as a clock signal, we can use a toggle Button Module (↑1.2, Button) at the "A" input port with "on" and "off" values "1" and "0" to turn the clock on and off. Again, this is shown in the Structure below where the button at the "A" input port is labeled "Run / Stop". So to forward a clock signal to the parent Instrument's MIDI Out, just connect the "E" output port to the Clock Out Module (↑3.9, Clock Out).

For the song position value we will use a Counter Module (↑13.2, Counter). The Song Position Out Module receives and sends the song position value in units of 96th notes. Since the rate of Events at the A to E Trig Module's (↑14.8, A to E (Trig)) "E" output port is already such, we can use these Events to increment the Counter Module's (↑13.2, Counter) output by "1" for each 96th note Event incoming at the "Up" input port. Again, the Counter Module's (↑13.2, Counter) internal state only is incremented for positive valued Events at the "Up" input port, so by setting the "A" input port of the A to E Trig Module (↑14.8, A to E (Trig)) to "0" with the "off" state of the "Run / Stop" button, we can pause the advance of the song position. As a result, the output of the Counter Module (↑13.2, Counter) is the song position counted in 96th notes, as is needed for the Song Position Out Module. Now you just need to use the Order Module (↑13.12, Order) to first send the song position value from its "1" output port to the "Pos" input port of the Song Position Out Module. Only then do you send the Event from the Order Module's (↑13.12, Order) "2" output port to the Song Position Out Module's "Trig" input port. To be able to reset the song position, connect a Button Module to the Counter Module's "Set" input port. In its Function page, the Button Module's (↑1.2, Button) "Max" and "Min" values should be set to "0" and it should be in Trigger Mode.

Last, make sure that all Modules are active. This can be ensured by engaging the Always Active checkbox in both the Clock Out Module's (↑3.9, Clock Out) and the Song Position Out Module's Function page.

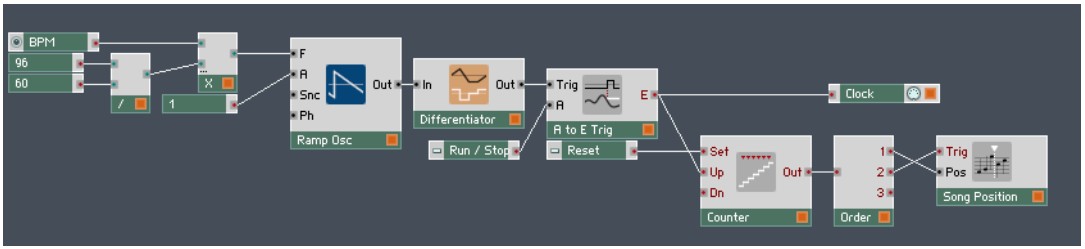


Fig. 3.24 Use the Clock Out and Song Position Out Modules to send MIDI Clock and MIDI Song Position messages from the parent Instrument's MIDI Out.

3.11 Channel Message Out

Fig. 3.25 Channel Message Out Module

3.11.1 Overview

The Channel Message Out Module converts Monophonic Events to MIDI channel messages which are sent to an external MIDI device (sequencer, synthesizer, etc.) or internally to other Instruments within the Ensemble. The MIDI Channel Messages are transmitted to the parent Instrument's MIDI Out every time an Event arrives at the "St" input port and according to what values lay at the other input ports. Thus, the desired message value and destination must be correctly defined with appropriate values at the other input ports before Events arrive at the "St" input port. All input ports accept Monophonic signals only.

Application

The Channel Message Module (↑2.17, Channel Message) gives you direct access to send all types of MIDI messages (except MIDI Clock messages). A Channel Message Out Module can replace any number of Controller Out Modules Controller Modules (↑2.8, Controller). This way you only need one Module to control outgoing MIDI controller messages.

3.11.2 Ports

Output Ports

- **(St)** "St" is the Event input port which determines the type of MIDI message sent and also triggers the message to be sent. 0 = Note Off, 1 = Note On, 2 = Poly Aftertouch, 3 = Control Change, 4 = Program Change, 5 = Channel Aftertouch, and 6 = Pitchbend.
- **(Ch)** "Ch" (MIDI channel number) is the Event input port that determines the MIDI channel number of the outgoing messages. The channel values are in the range [1 … 16].
- **(Nr)** "Nr" (MIDI note number) is the Event input port that sets the MIDI note number, CC (Control Change) number, or Program Change value of the outgoing MIDI message. The range of these value is [0 … 127].
- **(Val)** "Val" (value) is the Event input port that sets the velocity value of a note, the pressure value of an Aftertouch message, or the value of an outgoing CC (Control Change) or Pitchbend message.

3.11.3 Properties: Function Page

Keeping the Channel Message Out Module Always Active

The Channel Message Out Module only forwards values to the parent Instrument's MIDI Out if it is part of an active signal flow. Connecting it directly to a Knob Module, for example, does not suffice. For such a case the Always Active feature is useful.

► To keep the Channel Message Out Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Setting the Range of Input Values

The range of the input signal at the "Val" (value) input port [Min … Max] is mapped to the MIDI interval [0 … 127]. The values for "Min" and "Max" are set with the corresponding edit fields in the Function page of the Module's Properties. The default setting is Min = 0 and Max = 1. Another common setting is 0 to 127, which can aid in sending MIDI Program Change messages.

Fig. 3.26 Use the Min and Max edit fields in the Range area of the Function page to set the range of output values [Min … Max].

3.11.4 Example: Sending Multiple MIDI CC Messages

This example illustrates how to use the Channel Message Out Module to send MIDI CC messages to multiple MIDI Controllers. The Structure that fulfills this requirement is shown in the figure below. Let's say you have three Knob Modules (↑1.1, Fader/Knob) (labeled "CC1", "CC 12", and "CC 64"), the values of which you wish to forward to the corresponding MIDI Controller. Then the most convenient way to do this is by using the Channel Message Out Module.

First, for each change in a knob's value, you first need to send to the Channel Messages Out Module's "Nr" input port the CC number to which the Control Change message is to be sent. Then you need to send the new control value to the "Val" input port and only then the trigger signal to the "St" input port. Note that the trigger Event needs to carry the value "3" for MIDI CC messages. This way the Channel Message Out Module can correctly identify and process the values at its "Nr" and "Val" input ports.

The correct order of Events (described above) arriving at the Channel Message Out Module has been achieved with the use of Order (↑13.12, Order), Value (↑13.15, Value), and Merge Modules (↑13.16, Merge). Upon a change in a knob's value, the Order Module (↑13.12, Order) first triggers a Value Module (↑13.15, Value) from its "1" output port. This Value Module then sends an Event carrying the CC value to the Channel Message Out Module's "Nr" input port. Events from different sources are merged using the Merge Module (↑13.16, Merge). The next step is to merge the control values from the "2" output ports of the Order Modules (↑13.12, Order) using a Merge Module and then send the merged signal to the "Val" input port. Last, the "3" output ports of the Order Modules are merged and trigger the Event carrying the value "3" to be sent to the "St" input port.

Note that the ranges of the knobs have all been set to [0 … 1], in correspondence to the "Min" and "Max" values set in the Channel Message Out Module's Function page. Furthermore, if the Modules in the Structure are inactive, engage the Always Active checkbox in the Channel Message Out Module's Function page.

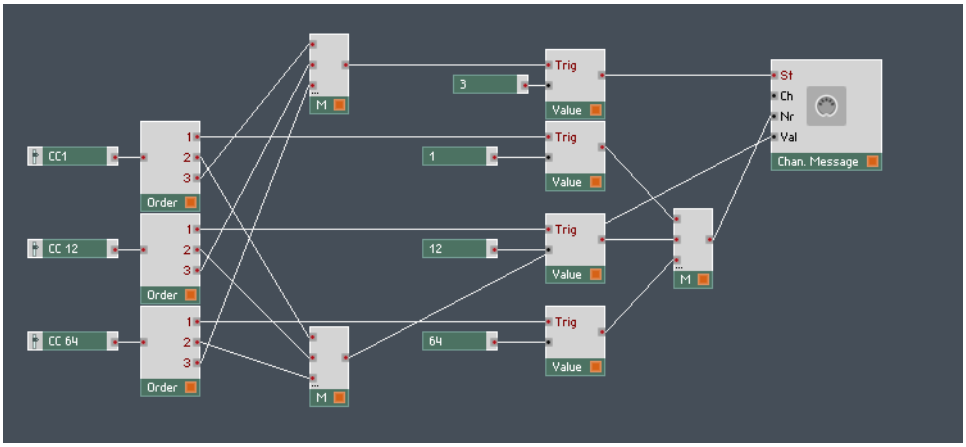


Fig. 3.27 Use the Channel Messages conveniently send multiple MIDI CC messages.

4 Math

You figure out the equation and REAKTOR does the math. There are Modules here for common operations such as addition, subtraction, multiplication, and division as well as for less-familiar mathematical functions such as absolute value, arc-tangent, and reciprocal square root. Some of the Math Modules are also used for exponential and logarithmic scale conversion to match REAKTOR's Module-input scaling. For example, some REAKTOR modules have linear frequency inputs (measured in Hertz) while others have exponential frequency inputs (measured in semitones and adjusted for MIDI note numbering). There are Modules here for converting between those two formats.
All Modules in this section are hybrid Modules, meaning they can be used as either Audio or Event Modules depending on their inputs. You can identify hybrid input and output ports by their green color. Many of the Modules (Add and Multiply, for example) also have dynamic inputs as indicated by three small dots at the bottom-left of the Module icon. When a wire is dragged to an empty part of the in-port region (the left edge of the Module icon) of a dynamic-input Module while holding down the Ctrl key in Windows (Cmd key in Mac OS X), a new input port is created automatically.

4.1 Constant

Fig. 4.1 Costant Module

4.1.1 Overview

The Constant Module is a monophonic source for a constant value. The value can be either positive or negative and is set with the Value edit field in the Function page of the Properties Tab. The Module only sends an Event once, that is for Initialization when it is activated. The Initialization Order among Constant Modules in the same Structure may depend on the Modules that they are connected to. Read the example for the Merge Module for details on this. You can create a Constant with the value "1" at any input port of a Module by right-clicking on that input port and clicking on the *Create Constant* menu entry. You can then change the value of the Constant in the Function page of that Constant's Properties.

Application

Typical usage cases for the Constant Module include:

- Setting parameters for other Modules, like the amplitude for an oscillator, to a constant value (see example).
- Modifying incoming signals by using the Constant Module in conjunction with other Math Modules.

For brevity, the Constant Module is referred to in context simply as Constant, as in "add a Constant with the value "1" to the Structure."

4.1.2 Ports

Output Ports:

- **(Out)** "Out" is the output port for the constant value signal.

4.1.3 Properties: Function Page

Changing the Constant Value

In order to change the value of the output, enter a new number into the Value edit field or the Label edit field.

4.1.4 Example: Constant Pitch and Amplitude

A common use for a Constant Module is the setting of parameters for other Modules to a constant value. In the example illustrated by the screenshot below two Constant Modules with values "69" and "1" are used to set the pitch and amplitude, respectively, of a Sine Module (↑6.14, Sine Oscillator) to constant values. Note that you can either create the Constants by right-clicking the Structure background and then clicking on the *Built-In Module > Math > Constant* menu entry or by right-clicking on an input port and then clicking on the *Create Constant* menu entry.

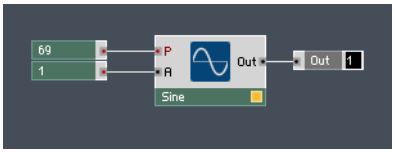


Fig. 4.2 Two Constant Modules with values "69" and "1" are used to set the pitch and amplitude of a Sine wave, respectively.

4.2 Add, +

Fig. 4.3 The Add Module

4.2.1 Overview

With the Add Module you can add two or more audio or Event signals. The output signal is the sum of the input signals: Out = In1 + In2 + In3 + ... The Add Module has dynamic input ports. This means that additional input ports can be created (if all existing input ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Add Module. The minimum number of input ports can be defined with the Min Port Groups edit field in the Function page. An Event at any input port triggers an Event at the output port.

Application

Typical usage cases for the Constant Module include:

- Performing arithmetic operations with signal values.
- Using the Add Module as a simple multi-channel mixer where the level of all channels is fixed at 0 dB (see example).

You also can use the Add Module to subtract values from a signal by setting the value to be subtracted as a negative Constant.

4.2.2 Ports

Input Ports:

- **(In1)** "In1" is the input port for the first summand of the output signal.
- **(In2)** "In2" is the input port for the second summand of the output signal.
- **(...)** "In3, In4, ..." are the dynamic inputs for additional summands for the output signal. When all the input ports area already connected and a wire is dragged to an empty part of the in-port region (the left edge of the Module icon) while holding down the Ctrl key in Windows (Cmd key in Mac OS X), a new input port is created automatically. The maximum number of input ports for the Add Module is 40.

Output Ports:

- **(Out)** "Out" is the output port for the sum of all the input signals.

4.2.3 Properties: Function Page

Changing the Minimum Number of Input Ports

To change the minimum number of input ports, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of input ports for the Add Module is 40.

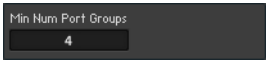


Fig. 4.4 Use the Min Num Ports Groups edit field to set the minimum number of input ports appearing on Add Module.

4.2.4 Example: Mixing Two Signals

You can mix two signals using the Add Module. Say you have two signals, In1 and In2, that you want to mix together. For that you need to place an Add Module into the Structure. Then connect the sources of the two signals (here we have two In Ports with the labels "In1" and "In2") to the two input ports of the Add Module. The output of the Add Module is a mix of the two signals In1 and In2 with the level of each fixed at 0 dB. You can now use the mix by connecting the output port of the Add Module to another Module. Here the signal was just routed to the "Out" Out Port, as seen in the picture.

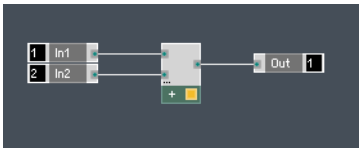


Fig. 4.5 Use the Add Module to mix two signals together.

4.3 Subtract

Fig. 4.6 Subtract Module

4.3.1 Overview

The Subtract Module subtracts one audio or Event signal from another. The output signal is the difference between the second input signal and the first input signal: Out = In1 - In2. An Event at either input port triggers an Event at the output port.

Application

Some usages of the Subtract Module include:

- Determining the difference between two values.
- Transposing MIDI notes (see example).

4.3.2 Ports

Input Ports

- **(In1)** "In1" is the input port for signal the signal from which the signal In2 is subtracted from.
- **(In2)** "In2" is the input port for the signal to be subtracted from signal In1.

Output Ports

- **(Out)** "Out" is the output port for the result of the subtraction: Out = In1 - In2.

4.3.3 Example: Transposing

The Subtract Module can be used to transpose notes, here in the logarithmic pitch scale. In the picture below you see a Structure where a Note Pitch Module has been used to detect the pitch that has been played with the keyboard. You see in the picture that the note with the value "56" in the logarithmic pitch scale has been. This corresponds to the note G3 in the MIDI scale. If we want to transpose this pitch down an octave, we need to subtract the value "12" from it, since an octave contains 12 semitones (see the explanation of the logarithmic pitch scale for more details on this). This is easily done with the Subtract Module by connecting the signal carrying the pitch to be transposed to the higher input port and the Constant with the value "12" to the lower input port. The output of the Subtract Module is in this case 56 - 12 = 44, the note G2 in the MIDI scale.

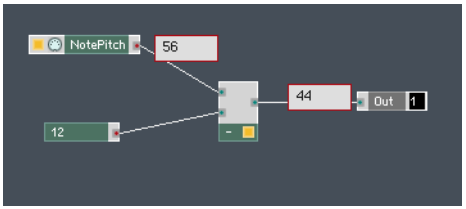


Fig. 4.7 In this example the Subtract Module is used to transpose the incoming pitch down 12 semitones.

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

4.4 Invert, -x

Fig. 4.8 Invert Module

4.4.1 Overview

The Invert Module multiplies an audio or an Event signal by the value "-1". The output signal is then the inverse of the input signal: Out = –In. You can also use the Multiply Module to invert a signal but using the Invert Module makes for a cleaner Structure. Simply inverting a sound has no audible effect, except when the inverted sound is combined in

some way with the uninverted sound. Inverting control signals generally has a very obvious effect, like for ramp signals, as shown in the example. An Event at the input port triggers an Event at the output port.

4.4.2 Ports

Input Ports

- **(In)** "In" is the input port for the signal to be inverted.

Output Ports

- **(Out)** "Out" is the output port for the inverted signal.

4.4.3 Example: Inverting an LFO

An example of inverting control signals is sending the output signal of an LFO Module through the Invert Module. The first figure below illustrates an instant where the value "0.995722" is sent from the "Sin" output port of the LFO Module. Connecting a wire from the "Sin" output port to the input port of the Invert Module yields the inverse value of at the output port of the Invert Module, "-0.995722". The second figure qualitatively shows the waveforms of the sine and inverted sine signals that arrive at the "Out" and "Inv" Out Ports, respectively.

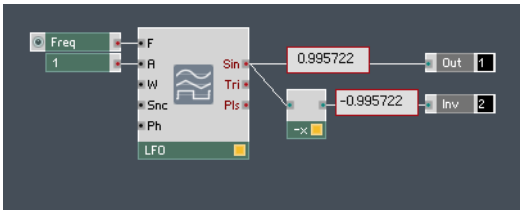


Fig. 4.9 The Invert Module changes the sign of the incoming signal.

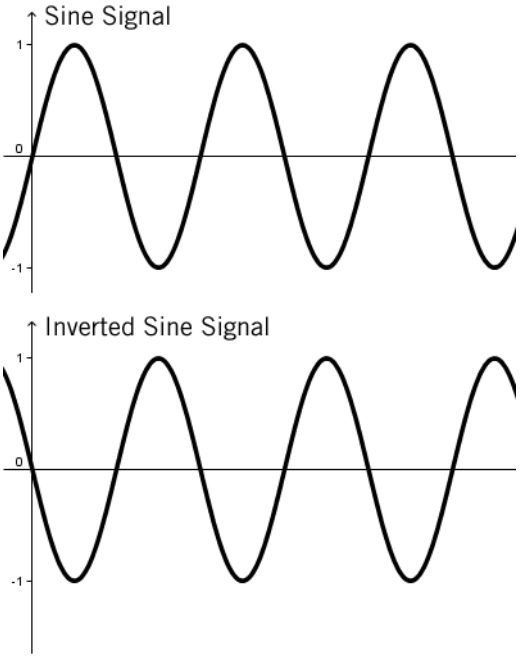


Fig. 4.10 The above plot shows a sine signal with its inverted counterpart plotted below.

4.5 Multiply, X

Fig. 4.11 Multiply Module

4.5.1 Overview

The Multiply Module multiplies two or more audio or Event signals. The output signal is the product of the input signals: Out = In1 * In2 * In3 * .... When a zero is connected to one of the inputs the output value is always zero. The Multiply Module has dynamic input ports. This means that additional input ports can be created if all existing input ports are already connected by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while

dragging a wire from an output port of another Module to the three dots on the Multiply Module. The minimum number of input ports can be defined with the Min Num Port Groups edit field in the Function page of the Properties Tab. An Event at any input port triggers an Event at the output port.

Application

Typical usage cases for the Multiply Module include:

- An amplifier controlled by a signal (corresponds to VCA in analog synthesizers): An audio signal is fed to one input port and an amplification factor to the other. This application is demonstrated in the example.
- Applications where you need to compute the square of a signal when the same signal is fed to two inputs: Out = In * In = In^2. This is the case for quadratic shaping curves where Out = 1 - (1 - In) ^{2}.
- Ring modulation: when two different sounds are connected to the inputs, the output is the ring modulation of the two sounds.

4.5.2 Ports

Input Ports:

- **(In1)** "In1" is the input port for the multiplicand for the output signal.
- **(In2)** "In2" is the input port for the second multiplicand for the output signal.
- **(...)** "In3, In4, ..." are the dynamic inputs for additional summands for the output signal. If all the input ports are already connected a wire is dragged to an empty part of the in-port region (the left edge of the Module icon) while holding down the Ctrl key in Windows (Cmd key in Mac OS X), a new input port is created automatically. The maximum number of input ports for the Multiply Module is 40.

Output Ports:

- **(Out)** "Out" is the output port for the product of the incoming signals.

4.5.3 Properties: Function Page

Changing the Minimum Number of Input Ports

To change the minimum number of input ports, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of input ports for the Multiply Module is 40.

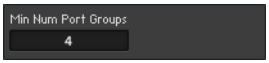


Fig. 4.12 Use the Min Num Ports Groups edit field to set the minimum number of input ports appearing on the Multiply Module.

4.5.4 Example: Simple Amplifier

The Multiply Module can be used as an amplifier controlled by a signal. In this case the amplifier is controlled by the output value of a Knob Module. The audio or Event signal from the In Port is fed to one input port of the Multiply Module and the amplification factor to the other.

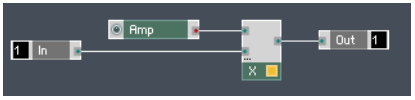


Fig. 4.13 A simple amplifier can be constructed by using the Multiply Module to multiply an incoming signal with a value received from a knob (here labeled "Amp").

4.6 Mult/Add (a * b + c), X+

Fig. 4.14 Mult/Add Module

4.6.1 Overview

The Mult/Add Module is a combined multiplier-adder. The output is the result of the operation (a * b) + c where "a" signifies the highest lying input port, "b" the middle input port, and "c" the lowest lying input port. An Event at any input port triggers an Event at the

output port. You can also use the Multiply Module and Add Module to perform the "multiply and add" operation but the Mult/Add Module is more convenient and makes for a cleaner Structure. An Event at any input port triggers an Event at the output port.

Application

A typical usage case for the Mult/Add Module is the crossfader Structure.

4.6.2 Ports

Input Ports

- **(a)** "a" is the highest input port is the input for the variable "a" for the output equation (a * b) + c.
- **(b)** "b" is the middle input port is the input for the variable "b" for the output equation (a * b) + c.
- **(c)** "c" is the middle lowest input port is the input for the variable "c" for the output equation (a * b) + c.

Output Ports

- **(Out)** "Out" is the output port for the signal (a * b) + c with the variables defined by the input values as listed above.

4.6.3 Example 1: LFO With Envelope

The Mult/Add Module is efficiently employed whenever a multiplication and addition are both needed in the corresponding form. One example is a control signal that consists of an LFO that has a gate triggered envelope and a constant value offset. First, the LFO signal, coming from the "Sin" output port of the LFO Module, is multiplied with the envelope signal from the "Out" output port of the AD - Env Module. This is done using the upper two input ports of the Mult/Add Module and causes the LFO"s amplitude to change according to the attack and decay settings of the AD - Env Module. If we want the control signal to start at the value "0.1" and also approach the value "0.1" (as opposed to "0") after the envelope decay time has been surpassed, then we can use the lowest input port of the Mult/Add Module and connect a Constant with the value "0.1" to it, as seen in the picture below.

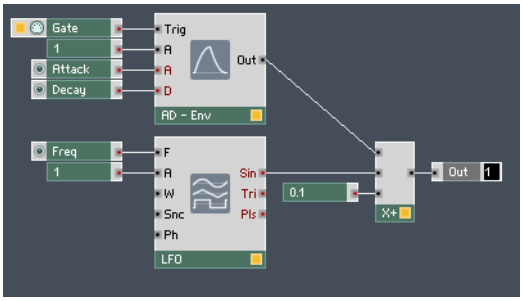


Fig. 4.15 You can use the Mult/Add Module on a signal to modulate it with an offset.

4.6.4 Example 2: Simple Crossfader

A frequent use case for the Mult/Add Module is a linear crossfader between two signals, In1 and In2. To put this functionality in a Macro, create a Macro with two input ports labeled "In1" and "In2" and an output port labeled "Out". The basic mathematical representation of this simple mixer is as follows:

Out = In2 × x + In1 × (1 - x)

where "x" represents a Knob Module with a range [0 … 1] and In1 and In2 the respective input signals. When x = 0 only the In1 signal is forwarded to the output and when x = 1 only the In2 signal is let through.

You see that the general form of the equation is a × b + c and hence you can save some CPU resources by using the Mult/Add Module. Insert the Mult/Add Module and a Knob Module into the Macro. Now connect the uppermost input port of the Mult/Add Module, "a", to the "In2" input port and the middle input port, "b", to a Knob Module. In the Knob Module's Function page, set the Min edit field to "0" and the Max edit field to "1". While you are at it, change label of the Module in the Label edit field to "Fade". Now add a Subtract Module and a Multiply Module to the structure. Connect the output of the Knob Module to the upper input of the Subtract Module. Create a Constant with the value "1" and connect its output to the lower input of the Subtract Module. You now have the (1 - x) at the output of the Subtract Module, where "x" is the value set with the knob labeled "Fader". Now use the Multiply Module to multiply the output of the Subtract Module with the signal from the "In2' input port. Finally, connect the output of the Multiply Module to the lowest input port of the Mult/Add Module and connect the output of this Module to the

Out Port. Your structure should now look similar to the following picture. You now have a simple linearly crossfading mixer in the form of a macro. You might want to use this when you wish to mix between two simple signals, like two Oscillators, for example.

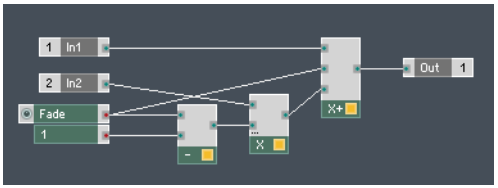


Fig. 4.16 The Structure for a simple crossfade effect.

For two arbitrarily complex audio signals or to pan between left and right stereo channels, a parabolic curve is used. The difference to the mixer with the linear curve is that when the "Fade" setting is x = 0.5, more than just 50 % of both signals is mixed together to keep the overall volume from being noticeably reduced in the case of two very different input signals. Try building this parabolic mixer using the logic you developed in the previous example to implement the following mathematical description of the parabolic mixer:
$Out = In2 \times (1 - (1 - x)^2) + In1 \times (1 - x^2)$

If you don't know yet how to calculate the square of a signal, x^2, then take a look at the overview of the Multiply Module.

4.7 Reciprocal, 1 / x

Fig. 4.17 Reciprocal Module

4.7.1 Overview

The output of the Reciprocal Module is one divided by the input. If the input signal is exactly zero the output is also set to zero. An Event at the input port triggers an Event at the output port.

Application

Processing sound signals does not normally yield anything useful. The Module is rather meant for processing control signals (which do not come near zero). Normalization, as shown in the example, is such a processing operation.

Be careful with small input values (near zero) because the output values become very large.

4.7.2 Ports

Input Ports

- **(In)** "In" is the input port for the signal by which "1" is to be divided.

Output Ports

- **(Out)** "Out" is the output port sending the value of "1" divided by the input signal value.

4.7.3 Example: Efficient Normalization

Let's say we have two signals, "a" and "b", that both have the range [0 … c] and we want them to have the range [0 … 1]. The process by which we achieve this is called normalization. Dividing, in this case by "c", is a common way to normalize control signals. The normalized signals "n1" and "n2" are then defined in respect to the signals "a" and "b" by the following two equations:

n1 = a / c

n2 = b / c

If several signals need to be normalized to the same value, that is, to be divided by the same value, using a Reciprocal Module in conjunction with Multiply Modules will use less CPU resources than explicitly dividing every signal by the same value. This is because multiplying requires less CPU resources than dividing. So to normalize the two signals, "a" and "b", in the most efficient way we need to build the Structure shown in the picture below.

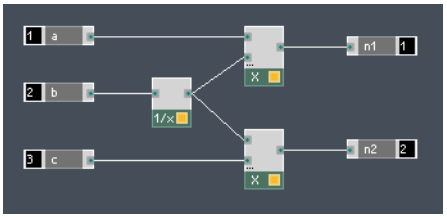


Fig. 4.18 One possible use for the Reciprocal Module is to normalize two signals with the same scaling factor.

4.8 Divide, /

Fig. 4.19 Divide Module

4.8.1 Overview

The Divide Module divides the value at the upper input port by the value at the lower input port. If the input signal is exactly zero the output is also set to zero. Be careful with small values (near zero) at the lower input port because the output values become very large. An Event at either input port triggers an Event at the output port.

Application

Dividing by sound signals does not normally yield anything useful. Whenever possible, use a Multiply Module instead of a Divide Module for audio signals because the CPU load will be significantly less (see the Reciprocal Module example for more information on this).

4.8.2 Ports

Input Ports

- **(In1)** "In1" is the input port for the numerator of the fraction evaluated by the Module.
- **(In2)** "In2" is the input port for the denominator of the fraction which is evaluated by the Module.

Output Ports

- **(Out)** "Out" is the output port for the numeric value of the fraction defined by Out = In1 / In2.

4.8.3 Example: Scaling

A control signal can be scaled using the Divide Module. In the picture below the signal is scaled to be 10 times smaller. The control signal is carried by the wire from the In Port to the upper input port of the Divide Module. A Constant with the value "10" is connected to the bottom input port of the Divide Module. This causes the signal sent to the Out Port to be 10 times smaller than the signal that is received from the In Port.

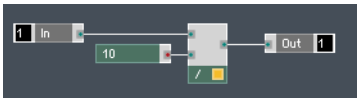


Fig. 4.20 In this example the Divide Module is employed to reduce the signal amplitude by a factor of ten.

4.9 Modulo

Fig. 4.21 Modulo Module

4.9.1 Overview

The Modulo Module computes the integer division of the two input values and also the remainder. An Event at either one of the input ports creates an Event at each output port.

The Event from the "Mod" output port is sent before the "Div" output port.

Application

This Module is commonly used in:

- Custom value displays
- Sequencers; to see a basic example of how to calculate the measure number and the count from the song position, go to the example.

4.9.2 Ports

Input Ports:

- **(A)** "A" is the hybrid input port for the signal A that is to be divided by B.
- **(B)** "B" is the hybrid input for the signal B used for dividing A. In most applications B is an integer.

Output Ports:

- **(Div)** "Div" (division) is the hybrid output port for the integer division of A and B. This is the largest integer smaller than or equal to A / B.
- **(Mod)** "Mod" (modulo) is the hybrid output port for the modulo of A and B. This is the remainder of the integer division of A and B. The range of this output port is [0 ... B) (excluding B).

4.9.3 Example: Measure and Count Calculator

Sometimes sequencers are controlled by the song position. The "Pos" Input Port of the sequencer receives the number of beats that the current song position is from the beginning of the song. If you are writing a waltz that has a 3/4 time signature and your sequencer has length "3", then you will need to divide the song position by "3" to determine how many measures have passed since the beginning of the song and which count you are on (1, 2, or 3). This can easily be done with the Modulo Module, as shown in the picture below. If the song position is "92", then the number of measures that have passed since the beginning is given by the integer division of 92 / 3. The result is the largest multiple of "3" that is less or equal to "92", divided by "3". The largest multiple of "3" that is less than equal to "92" is "90" and 90 / 3 = 30, the value at the "Div" output. The count is then given by 92 % 3 = 2 (also known as the remainder of the integer division), the value at the "Mod" output port.

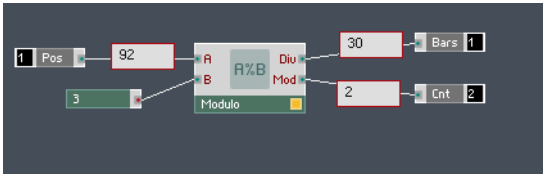


Fig. 4.22 Use the Modulo Module to determine from the song position ("Pos") value how many measures have passed since the beginning of the song and which count you are on.

4.10 Rectifier, |x|

Fig. 4.23 Rectifier Module

4.10.1 Overview

The Rectifier Module takes the absolute value of the input signal, that is, negative values are inverted and become positive and positive values are left unchanged. An Event at the input port triggers an Event at the output port.

Application

An application of the Rectifier Module is to combine it with a low pass filter to create a signal level detector.

4.10.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the signal to be rectified. Negative values become positive with equal magnitude.

Output Ports

- **(Out)** "Out" is the hybrid output port for the rectified signal.

4.10.3 Example: Changing Polarity

The Rectifier Module can be used to change the polarity of a control signal. Say you have a bipolar knob that has a range [-1 … 1] to control one parameter and you want that same knob to control another parameter that has range [0 … 1]. You can transform the output signal from the Knob Module using the Rectifier Module as shown in the picture below When you turn the knob from "0" to "1", both the signal at the "+/-" and "+" Out Port will go from "0" to "1". On the other hand if you turn the knob from "0" to "-1", the signal at the "+/-" Out Port will go from "0" to "-1" and the signal at the "+" Out Port will go from "0" to "1".

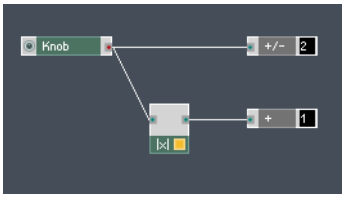


Fig. 4.24 A possible use for the Rectifier Module is to change the polarity of a control signal from bipolar to unipolar.

4.11 Rectify/Sign

Fig. 4.25 Rectify/Sign Module

4.11.1 Overview

The input signal is rectified, that is, negative values are inverted and become positive and positive values are left unchanged. Information about the sign of the incoming signal value is sent from the "Sign" output port. An Event at the input port triggers Events at both output ports.

The Event from the "|x|" output port is sent before the "Sign" output port.

Application

A common application for the Rectify/Sign Module is to separate the sign and the absolute value of value for a custom value display.

4.11.2 Ports

Input Ports:

- **(In)** "In" is the hybrid input port for the audio or Event signal to be rectified and analyzed for the sign.

Output Ports:

- **(Sign)** "Sign" is the hybrid output port for the sign of the input signal. It sends the value "1" for positive input signal values and the value "-1" for negative input signal values.
- **(I x I)** "I x I" (absolute value) is the hybrid output port for the rectified signal. Negative input values become positive with equal magnitude. Positive input values are left unchanged.

4.11.3 Example: Sign and Magnitude Decomposition

The Rectify/Sign Module in essence breaks down an incoming value into two parts: its magnitude and its sign. By multiplying the values at the output ports together, we can again reconstruct the value that was initially sent into the Rectify/Sign Module. The picture below illustrates this property as a Structure.

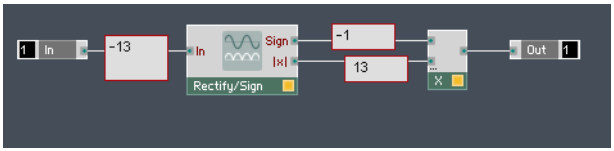


Fig. 4.26 The Rectify/Sign Module retrieves two pieces of information from the incoming value: its sign and its absolute value.

4.12 Compare

Fig. 4.27 Compare Module

4.12.1 Overview

The Compare Module represents a comparing logic function. It compares the magnitudes of the values at the two input ports and sets the values at the two output ports according to the result of the comparison. For the Compare Module, the two comparison operators are "greater than" and "less than or equal to". An Event at either input port triggers an Event at both output ports.

The Event from the ">" output port is sent before the "<=" output port.

Application

The Compare Module can be used in conjunction with the Signal Path Modules like the Selector Module to control the routing of signals. An implementation of this technique is presented in the example.

4.12.2 Ports

Input Ports

- **(A)** "A" is the hybrid input port for the first of two values to be compared.
- **(B)** "B" is the hybrid input port for the second of two values to be compared.

Output Ports

- **(>)** ">" (greater) is the hybrid output port for the result of the comparison A > B ("A" greater than "B"). If A > B is true, the value at the output port is set to "1". Otherwise (if A > B is false) the output port sends the value "0".
- **(<=)** "<=" (less or equal) is the hybrid output port for the result of the comparison A <= B ("A" less than or equal to "B"). If A <= B is true, the value at the output port is set to "1". Otherwise (if A <= B is false) the output port sends the value "0".

4.12.3 Example: Routing

The Compare Module can be used in conjunction with the Selector Module to control the routing of signals. In this example a control signal is compared to a constant value "2" and depending on the result of the comparison, different values are sent to the Out Port. The comparison is done with the Compare Module. In the figure below you see a Structure of this signal flow where the value sent from the In Port is "4". Since "4" is sent to the "A" input port and "2" to the "B" input port of the Compare Module, the comparison A > B or 4 > 2 is a logically true statement. Hence the value sent from the ">" output port is "1". Since "Pos" is "1", the value from the "1" input port of the Selector Module, "7", is forwarded to its output port. If, for example, the value at the In Port would be "1", then an analogous discussion to the previous one would let you deduce that the value "3" is sent to the Out Port.

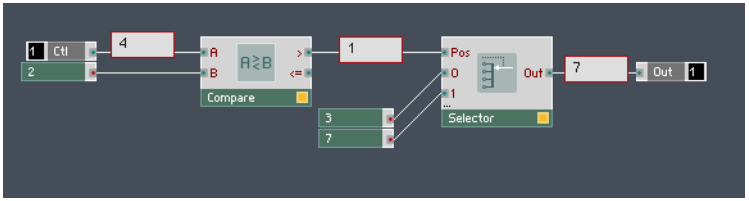


Figure 236 The Compare Module can be used in conjunction with the Selector Module to variably route a signal depending on the value of the incoming control signal (at the "Ctl" input port).

4.13 Compare/Equal

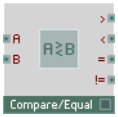


Fig. 4.28 Compare/Equal Module

4.13.1 Overview

The Compare/Equal Module represents a comparing logic function. It compares the magnitudes of the values at the two input ports and sets the values at the four output ports according to the result of the comparison. For the Compare/Equal Module, the four comparison operators are "greater than", "less than", "equal to", and "not equal to". An Event at either input port triggers an Event at all output ports.

The Events are sent from the output ports in the following order: first ">", then "<", then "=", and finally "!=".

Application

The Compare Module can be used in conjunction with the Signal Path Modules like the Relay Module to control the routing of signals. An implementation of this technique is presented in the example.

4.13.2 Ports

Input Ports:

- **(A)** "A" is the hybrid input port for the first of two values to be compared.
- **(B)** "B" is the hybrid input port for the second of two values to be compared.

Output Ports:

- **(>)** ">" (greater) is the hybrid output port for the result of the comparison A > B ("A" greater than "B"). If A > B is true, the value at the output port is set to "1". Otherwise (if A > B is false) the output port sends the value "0".
- **(<)** "<" (less) is the hybrid output port for the result of the comparison A < B ("A" less than "B"). If A < B is true, the value at the output port is set to "1". Otherwise (if A < B is false) the output port sends the value "0".
- **(=)** "=" (equal) is the hybrid output port for the result of the comparison A = B ("A" equal to "B"). If A = B is true, the value at the output port is set to "1". Otherwise (if A = B is false) the output port sends the value "0".
- **(!=)** "!=" (not equal) is the hybrid output port for the result of the comparison A != B ("A" not equal to "B"). If A != B is true, the value at the output port is set to "1". Otherwise (if A != B is false) the output port sends the value "0".

Note that the values at the (=) and (!=) output ports are always opposite: if (=) is "1" then (!=) is "0" and vice versa.

4.13.3 Example: Routing

The Compare/Equal Module can be used in conjunction with the Relay Module to control the routing of signals. In this example a control signal is compared to a constant value "2" and depending on the result of the comparison, different values are sent to the Out Port. The comparison is done with the Compare/Equal Module. In the figure below you see a Structure of this signal flow where the value sent from the In Port is "2". Since "2" is sent to the "A" input port and "2" to the "B" input port of the Compare/Equal Module, the comparison A = B or 2 = 2 is a logically true statement. Hence the value sent from the "=" output port is "1". Since "Ctl" is "1", the value from the "1" input port of the Relay Module, "6", is forwarded to its output port. If, for example, the value at the In Port would be "1", then an analogous discussion to the previous one would let you deduce that the value "0"

is sent to the Out Port. This is because in the case that there is only two input ports for the Relay Module, a value of "0" at the "Ctl" input port causes the value "0" to be sent from the output port of the Relay Module.

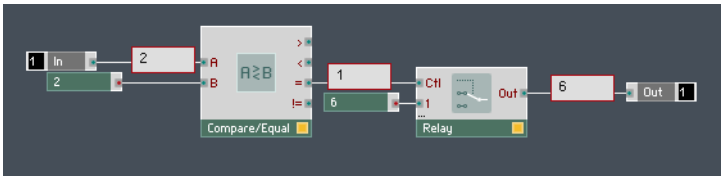


Fig. 4.29 This Structure shows an example of conditional signal processing utilizing the Compare/Equal and Relay Modules.

4.14 Quantize

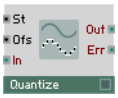


Fig. 4.30 Quantize

4.14.1 Overview

The Quantize Module quantizes audio and Event signals to a quantization grid with adjustable step size and offset. The input signal is rounded to the nearest quantization step before being forwarded to the "Out" output port. The difference between this output signal value and the input signal value is forwarded to the "Err" output port.

The Event from the "Out" output port is sent before the Event from the "Err" output port.

Application

Useful applications for the Quantize Module include:

- Quantization of an LFO Rate to sync with a clock signal..
- Quantization of the frequency of an oscillator to be a multiple of the frequency of another oscillator (see example).

4.14.2 Ports

Input Ports:

- **(St)** "St" (step size) is the hybrid input port for controlling the size of the quantization step. When set to zero, the signal is not quantized.
- **(Ofs)** "Ofs" (offset) is the hybrid input port for the offset value of the quantization grid.
- **(In)** "In" is the hybrid input port for the signal to be quantized.

Output Ports:

- **(Out)** "Out" is the hybrid output port for the quantized signal.
- **(Err)** "Err" (error) is the hybrid output port for the quantization error that is generated by the rounding: Err = Out - In.

4.14.3 Example: Frequency Quantization

The Quantize Module can be used to quantize the frequency of one "slave" oscillator to the frequency of another "master" oscillator. The Structure in the picture below elucidates how this can be done in a context where two sinus waves (from the "master" and "slave" oscillators) are sent through a ring modulator to create a harmonic spectrum with a fundamental tone and one overtone.

To quantize the "slave" oscillator frequency, the output port of the Knob Module that sets the "slave" oscillator frequency is connected to the "In" input port of the Quantize Module. For the output signal of the Quantize Module to be a multiple of the "master" oscillator frequency set by the Knob Module labeled "Master", the quantization grid of the Quantize Module needs to comprise multiples of the "master" oscillator frequency. In other words, the step size of the quantization grid needs to be specified by the output port of the Knob Module that sets the "master" oscillator frequency. For this purpose the output port of the Knob Module labeled "Master" is connected to the "St" input port of the Quantize Module. The output port of the Quantize Module now sends a "slave" oscillator frequency value that has been quantized to be a multiple of the "master" oscillator frequency.

For example, if the "Master" knob sends the value "250" to the "St" input port of the Quantize Module, the resulting quantization grid is -250, 0, 250, 500, 750, 1000, ... (note that negative values are allowed). In this case, if the "Slave" knob sends the value "600" to the "In" input port of the Quantize Module, the closest quantization grid value is "500" and so the "Out" output port of the Quantize Module sends the value "500".

To detune the "slave" oscillator frequency from being a perfectly tuned harmonic of the "master" oscillator frequency, you can send a small value to the "Ofs" input port of the Quantize Module. This is not recommended for normal musical purposes, since the actual detuning of the two frequencies in semitones depends on the frequencies.

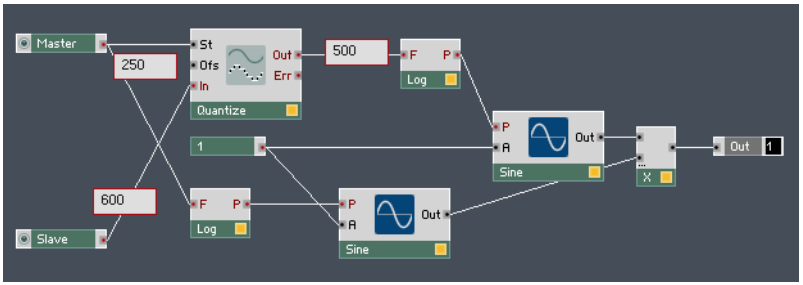


Fig. 4.31 You can use the Quantize Module to quantize the frequencies of a "slave" oscillator to a grid determined by the frequency of a "master" oscillator.

4.15 Exp (Lvl-to-A) Module

Fig. 4.32 Exp (Lvl-to-A) Module

4.15.1 Overview

The Exp (Lvl-to-A) Module is an exponential function for converting logarithmic signal level values in dB to linear amplitude values. An Event at the input port triggers an Event at the output port.

Application

Use this Module to change level values given in decibel (for example in a mixer) to linear values that, for example, can be connected to the "A" input port of envelopes, samplers, and oscillators.

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

4.15.2 Ports

Input Ports

- **(Lvl)** "Lvl" (level) is the hybrid input port for logarithmic signal level values in dB (decibel) to be converted to linear amplitude values. The typical range for the input values is [-50 ... 10].

Output Ports

- **(A)** "A" (amplitude) is the hybrid output port for a linear amplitude control signal..

4.15.3 Example: Oscillator Amplitude in dB

If you are used to specifying signal levels in dB you might want to control your oscillator levels on the panel with knobs that use the decibel scale. The Oscillator Modules in REAKTOR have an "A" input port for the amplitude which works in linear scale, so you have to convert your values specified in the logarithmic scale with the "Level" Knob Module to linear amplitude values. For this, you use the Exp (Lvl-to-A) Module as shown in the picture below.

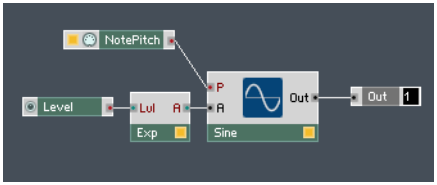


Fig. 4.33 Use the Exp (Lvl-to-A) Module to convert a logarithmic amplitude level to a linear amplitude value.

4.16 Exp (P-to-F)

Fig. 4.34 Exp (P-to-F) Module

4.16.1 Overview

The Exp (P-to-F) Module is an exponential function for converting logarithmic pitch values in semitones to linear frequency values in Hz. An Event at the input port triggers an Event at the output port.

Application

Use this Module to change pitch values given in semitones (for example MIDI notes from a Note Pitch Module) to linear frequency values that, for example, can be used for frequency shifting (see example) or are relevant for FM Synhesis.

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

4.16.2 Ports

Input Ports

- **(P)** "P" (pitch) is the hybrid input port for logarithmic pitch values in semitones to be converted to linear amplitude values. The typical range for the input values is [0 … 127].

Output Ports

- **(F)** "F" (frequency) is the hybrid output port for a frequency control signal in Hz (Hertz).

4.16.3 Example: Frequency Shifting

For some purposes, like the synthesis of drum sounds, inharmonic spectra are necessary. This can be achieved with frequency shifting because it allows spectra with arbitrary frequencies (within the system's range, of course) to be created. Ring modulation, in contrast, creates spectra that have frequencies that are multiples of the components of the modulator frequency, offset by the carrier frequency. What is relevant for us in this context is to know how to convert notes from the Note Pitch Module that come in the logarithmic pitch scale to the linear frequency scale so that we can add or subtract any number of Hertz from the original pitch. This is done with the Exp (P-to-F) Module by connecting the output port of the Note Pitch Module to the "P" input port of the Exp (P-to-F) Module and connecting the "F" output port of the Exp (P-to-F) Module to a Math Module to shift the

frequency. Since the oscillators receive the pitch information from the "P" input port and in the logarithmic pitch scale, you need to convert the frequencies back to the logarithmic pitch scale. This is done with the Log (F-to-P) Module, as shown in the figure below. Connect the wires carrying the new frequencies to the "F" input port of Log (F-to-P)Modules and the "P" output ports of the Log (F-to-P)Modules to the "P" input ports of your oscillators.

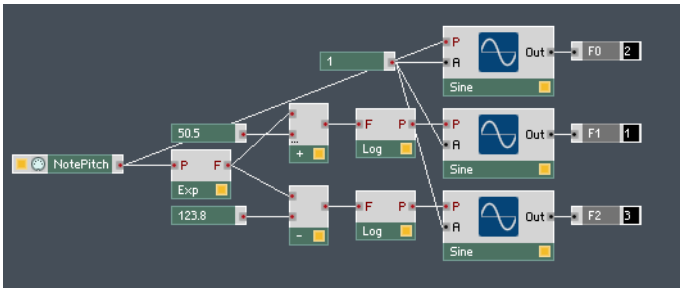


Fig. 4.35 Use Exp (P-to-F) Module to convert a pitch to a frequency. After shifting the frequencies use the Log (F-to-P) Module to convert the new frequencies back to pitch values.

4.17 Log (A-to-Lvl)

Fig. 4.36 Log (A-to-Lvl) Module

4.17.1 Overview

The Log (A-to-Lvl) Module is a logarithm function for converting linear amplitude values to logarithmic signal level values in dB. An Event at the input port triggers an Event at the output port.

Application

You can use this Module to:

- Change linear amplitude values to level values given in decibel that, for example, can be displayed with a level meter.
- Convert linear time values into logarithmic time values for the "A", "D", "S", "R", and other inputs ports for the decay times of envelopes.

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

4.17.2 Ports

Input Ports:

- **(A)** "A" (amplitude) is the hybrid input port for the linear amplitude value to be converted to the logarithmic level value in dB. The typical range for the input values is [0 … 1000].

Output Ports:

- **(Lvl)** "Lvl" (level) is the hybrid output port for the level in dB. The typical range for the output values is [-60 … 0].

4.17.3 Example: Level Meter

Although the amplitudes for Oscillator Modules in REAKTOR are specified at the Module's "A" input port, most musicians are used to monitoring the signal levels in the decibel scale. One possibility to monitor the amplitude (which has a range [0 … 1]) in dB is to connect the "Ampl" Knob Module to the input port of a Log (A-to-Lvl) Module and the output port of the Log (A-to-Lvl) Module to the input port of a Meter Module, as shown in the first figure below. Since the typical range of the "Lvl" output port's values is [-60 … 0], go to the Function page of the Meter Module and set the Max dB Value edit field to "0" and the Min dB Value edit field to "-60". Also, engage the Always Active checkbox to make the Meter Module active independent of the fact if the signal path is ultimately connected to the Audio Out Module or not. The second figure below shows how the meter looks in the Panel View. Note that the current example is only to demonstrate the functionality of the Log (A-to-Lvl) Module, since by replacing the Meter Module with the Level Meter Module you eliminate the need for the Log (A-to-Lvl) Module. The Level Meter Module intrinsically converts the linear amplitude into dB.

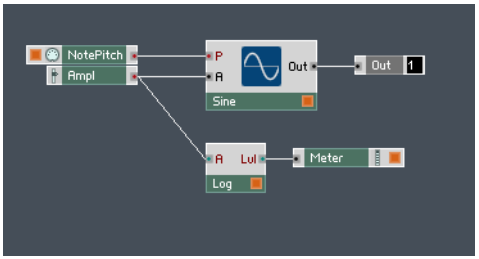


Fig. 4.37 The Log (A-to-Lvl) Module is used to convert linear amplitude values to logarithmic level values as displayed by the Meter Module.

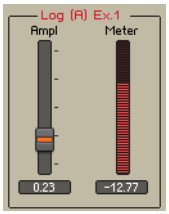


Fig. 4.38 The Meter Module displays the incoming amplitude level values in the logarithmic scale (in dB).

4.18 Log (F-to-P)

Fig. 4.39 Log (F-to-P) Module

4.18.1 Overview

The Log (F-to-P) Module is a logarithm function for converting linear frequency values in Hz to logarithmic pitch values in semitones. An Event at the input port triggers an Event at the output port.

Application

You can use this Module is you want to specify the frequency of an oscillator exactly in Hz.

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

4.18.2 Ports

Input Ports

- **(F)** "F" (frequency) is the hybrid input port for the linear frequency value to be converted to the logarithmic pitch value in semitones. The typical range for the input values is [0 … 5000].

Output Ports

- **(P)** "P" (pitch) is the hybrid output port for the logarithmic pitch in semitones. The typical range for the output values is [0 … 100].

4.18.3 Example: Cutoff Frequency

For some purposes, like setting the cutoff frequency of a filter to a certain frequency in Hz, it is necessary to convert the frequency to the logarithmic pitch scale so that it can be routed to the "P" input port of a filter. This is done with the Log (F-to-P) Module by connecting the wire carrying the frequency value to the "F" input port of the and connecting the "P" output port to the "P" input port of the Filter Module as shown in the figure below. Here the 2-Pole Notch Module filters out frequencies around 325.7 Hz from the incoming signal.

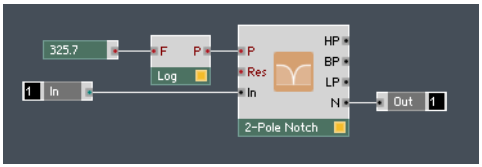


Fig. 4.40 Use the Log (F-to-P) Module to convert a precise cutoff frequency value to a logarithmic pitch value which is then fed to the "P" input port of a Filter Module.

4.19 Power

Fig. 4.41 Power Module

4.19.1 Overview

The Power Module represents a power function. The output port delivers the result of the first input port "X" to the power of the second input port "Y" (usually denoted as X^Y). An Event at either input port triggers an Event at the output port.

Application

The Power Module is mostly applied for mathematical operations having to do with scaling and compensation, such as when stacking Voices for Unison playing.

4.19.2 Ports

Input Ports:

- **(X)** "X" is the hybrid input port for the basis of the power.
- **(Y)** "Y" is the hybrid input port for the exponent of the power.

Output Ports:

- **(X^Y)** "X^Y" is the hybrid output port for the result of the calculation X^Y.

4.19.3 Example: Power of 2

This example shows you how to calculate a power expression. Here, the value "2" was used for the base and the value "8" was used for the power. The picture below shows you how to calculate the resulting power expression, 2^8 using the Power Module. The result is 256, as you can verify with your calculator.

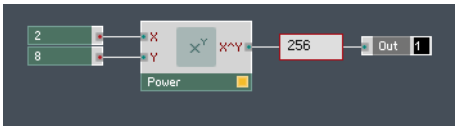


Fig. 4.42 The Power Module lets you calculate the results of power expressions such as 2^8.

4.20 Square Root

Fig. 4.43 Suqare Root Module

4.20.1 Overview

The Square Root Module represents the square root function. The output port delivers the result of the square root of the value at the input port. For negative input values, the output is zero. An Event at either input port triggers an Event at the output port.

Application

The Square Root Module is mostly applied for mathematical operations having to do with scaling of control and modulation signals, for example.

The square root can also be calculated using the Power Module. Just set the value at the "Y" input port to "0.5".

4.20.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value of which the square root is to be taken.

Output Ports

- **(Out)** "Out" is the hybrid output port for the square root of the input value.

4.20.3 Example: Square Root of 81

This example shows you how to calculate the square root of a value. Here, the input value "81" was used. The figure below shows you how to calculate the resulting square root of 81 using the Square Root Module. The result is 9, as you can easily verify.

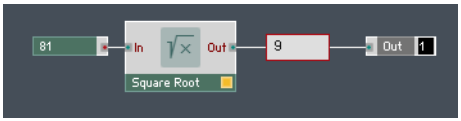


Fig. 4.44 The Square Root Module lets you calculate expressions involving the square root function.

4.21 1 / Square Root

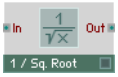


Fig. 4.45 1/Square Root Module

4.21.1 Overview

The 1 / Square Root Module calculates the reciprocal of the square root of the value at the input port. For negative input values the output is zero. An Event at either input port triggers an Event at the output port.

Application

The 1 / Square Root Module is mostly used to calculate mathematical equations in REAKTOR.

The reciprocal square root can also be calculated using the Power Module. Just set the value at the "Y" input port to "-0.5".

4.21.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value of which the reciprocal square root is to be taken.

Output Ports

- **(Out)** "Out" is the hybrid output port for the reciprocal square root of the input value.

4.21.3 Example: Reciprocal Square Root of 81

This example shows you how to calculate the reciprocal of the square root of a value. Here, the input value "81" was used. The figure below shows you how to calculate the resulting reciprocal of the square root of 81 using the 1 / Square Root Module. The result is 1 / 9 or 0.11111, as you can verify with your calculator.

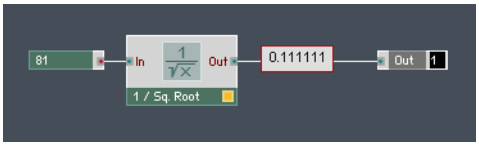


Fig. 4.46 The 1 / Square Root Module enables you to calculate the equivalent of taking the inverse of a value and then taking its square root.

4.22 Sine

Fig. 4.47 Sine Module

4.22.1 Overview

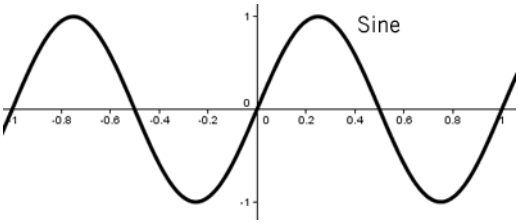
The Sine Module calculates the trigonometric sine function. Both the input and output are scaled to a range [-1 … 1]. To calculate an input based on degrees, divide the input value by 360 before sending it to the input port. To calculate an input based on radians, divide the input value by 2*Pi (approximately 6.283) before sending it to the input port. The figure below shows a graph of the output values of the sine wave for an input range [-1 … 1]. An Event at the input port triggers an Event at the output port.

Application

Common uses of the Sine Module are:

- Sine waveshaping of audio signals (see the example for details on this)

- Spreading voices out in the stereo panorama, as described in the tutorial in subsection 9.3.4 of the Application Reference.



4.22.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value for the scaled argument of the sine function.

Output Ports

- **(Out)** "Out" is the hybrid output port for the scaled sine function of the input value. The output ranges from 1 (for an input value of ".25") to -1 (for an input value of ".75"). This input range comprises half a wave cycle.

4.22.3 Example: Waveshaping

A common use of the Sine Module is the sine waveshaper. Waveshaping is used to apply distortion effects for audio signals. A Structure that implements a sine waveshaper would look like the one shown in the picture below. The audio signal from the In Port is multiplied by the factor "Drive" and then fed to the sine waveshaper. The distorted signal is then routed to the Out Port. For very low input values the sine shaper has an almost linear response (leaving the signal unaffected) but for higher values the distortion is clearly audible. Set the range of the Knob Module labeled "Drive" to [0 ... 20] by going to its Function page and setting the Max Value edit field to "20" and the Min Value edit field to "0". You can now increase or decrease the strength of this distortion effect by increasing or decreasing the "Drive" parameter, respectively.

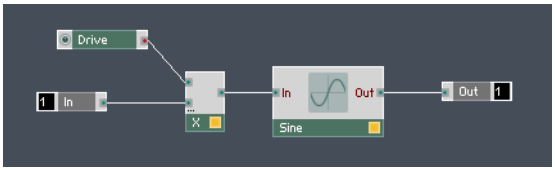


Fig. 4.48 A common application for the trigonometric sine function is found in waveshaping.

4.23 Sine/Cosine

Fig. 4.49 Sine/Cosine Modules

4.23.1 Overview

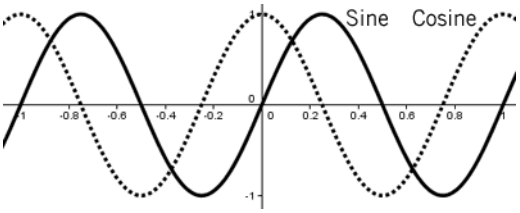
The Sine/Cosine Module calculates the trigonometric sine and cosine functions. Both the input and output are scaled to a range [-1 … 1]. To calculate an input based on degrees, divide the input value by 360 before sending it to the input port. To calculate an input based on radians, divide the input value by 2*Pi (approximately 6.283) before sending it to the input port. The figure below shows a graph of the output values of the sine wave (solid line) and cosine wave (dotted line) for an input range [-1 … 1]. An Event at the input port triggers an Event at both output ports.

The Event from the "Sin" output port is sent before the Event from the "Cos" output port.

Application

Common uses for the Sine/Cosine Module include:

- Plotting of a circle, as shown in the example.
- Creating a scope with a circular trace. This is a specific example of when you might want to plot a circle.



4.23.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value for the scaled argument of the sine function and cosine functions.

Output Ports

- **(Sin)** "Sin" (sine) is the hybrid output port for the scaled sine function of the input value. The output ranges from 1 (for an input value of ".25") to -1 (for an input value of ".75"). This input range comprises half a wave cycle.
- **(Cos)** "Cos" (cosine) is the hybrid output port for the scaled cosine function of the input value. The output ranges from 1 (for an input value of "0") to -1 (for an input value of ".50"). This input range comprises half a wave cycle.

4.23.3 Example: Circle

This example shows how the Sine/Cosine Module is used to plot a circle with the help of Ramp Osc Module and an XY Module. The actual plotting is done with the XY Module and as the name suggests, you need to define the x- and y-coordinate of each point that you want to plot. The simplest way to define the x- and y-coordinates of all points on the unit circle is with the following two equations:

- x = sin(f)
- y = cos(f)

where "f" stands for a parameter that has range [-1 … 1]. You see that the circle is defined by the cosine and sine functions, both of which use the same parameter as the argument. No doubt, the Sine/Cosine Module is the most efficient way to calculate the x- and y-coordinates from these equations.

Note that the range [-1 ... 1] for the parameter "f" is only valid within REAKTOR since the Modules use scaled versions of the sine and cosine functions.

The Sine/Cosine Module needs something to specify the parameter "f" at its input port before it can calculate anything. A Ramp Osc Module is perfect for that because we can have a parameter "f" that periodically scans through the range [-1 ... 1] at a frequency of our choice. As seen in the picture below, Ramp Osc Module was chosen to output a ramp with an amplitude of 1 (range [-1 ... 1]) and a frequency of 100 Hz. The Constants at the "A" and "F" inputs, respectively, serve the purpose of specifying these values. The values at the "Sin" and "Cos" output ports of the Sine/Cosine Module now correspond to the values of the x- and y- coordinates, respectively, of points on the unit circle and are routed to the "X1" and "Y1" input ports of the XY Module, respectively. In order for the XY Module to properly plot the circle, go the XY Module's View page and choose the *Scope* menu item from the Object Type menu The second picture below shows the Panel View of the XY Module plotting a unit circle.

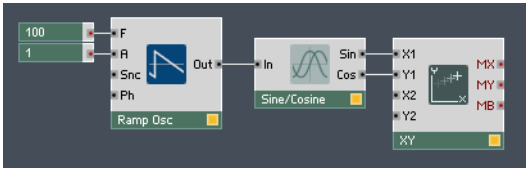


Fig. 4.50 The "Sin" and "Cos" outputs of the Sine/Cosine Module can conveniently be used to plot a circle.

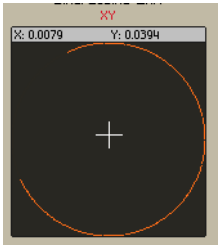


Fig. 4.51 The XY Module's Panel representation shows that a circle is plotted when the Sine/Cosine and Ramp Osc Modules are used as discussed above.

4.24 ArcSin Module

Fig. 4.52 Arc Sin Module

4.24.1 Overview

The ArcSin Module calculates the inverse sine function. The arc sine of "x" is a number between "-0.25" and "0.25", the sine of which is "x". Since the sine function"s output range is [-1 … 1], the arc sine function is only valid for inputs within that range. For input values < -1 the ArcSin returns -0.25 and for arguments >1, it returns 0.25. An Event at the input port triggers an Event at the output port.

Application

The ArcSin Module is mostly used to calculate mathematical equations in REAKTOR.

4.24.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value for the scaled argument of the arc sine function.

Output Ports

- **(Out)** "Out" is the hybrid output port for the scaled arc sine function of the input value. The output ranges from -0.25 (for an input value of "-1") to 0.25 (for an input value of "1").

4.24.3 Example: Arc Sine As Inverse of Sine

This example demonstrates how the ArcSin Module acts as the inverse of the sine function. In the picture below you see a Knob Module with range [-1 … 1] connected to the input port of a ArcSin Module and the output port of the ArcSin Module connected to the input port of a Sine Module. Since the arc sine function is the inverse for the sine function, any value in the range [-1 … 1] that is fed to the ArcSin Module will again be reproduced (up to a certain error stemming from the floating point format) at the output port of

the serially connected Sine Module. In the figure below the input value is "0.14" and the output value at the Sine Module is "0.139992", the same value with an error of "0.000008".

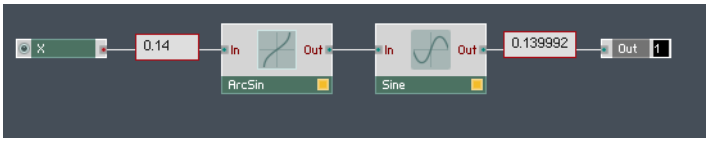


Fig. 4.53 ArcSin represents the inverse trigonometric function of the sine function. When the two corresponding Modules are connected in series, they approximately cancel each other out.

4.25 ArcCos Module

Fig. 4.54 ArcCos Module

4.25.1 Overview

The ArcCos Module calculates the inverse cosine function. The arc cosine of "x" is a number between "0" and "0.50", the cosine of which is "x". Since the cosine function's output range is [-1 … 1], the arc sine function is only valid for inputs within that range. For input values < -1 the ArcCos Module returns 0.50 and for arguments >1, it returns 0.

Application

The ArcCos Module is mostly used to calculate mathematical equations in REAKTOR.

4.25.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value for the scaled argument of the arc cosine function.

Output Ports

- **(Out)** "Out" is the hybrid output port for the scaled arc cosine function of the input value. The output ranges from 0.5 (for an input value of "-1") to 0 (for an input value of "1").

4.25.3 Example: Arc Cosine As Inverse of Cosine

This example demonstrates how the ArcCos Module acts as the inverse of the sine function. In the figure below you see a Knob Module with range [-1 … 1] connected to the input port of a ArcCos Module and the output port of the ArcCos Module connected to the input port of a Sine/Cosine Module. Since the arc cosine function is the inverse for the cosine function, any value in the range [-1 … 1] that is fed to the ArcCos Module will again be reproduced (up to a certain error stemming from the floating point format) at the "Cos" output port of the serially connected Sine/Cosine Module. In the picture below the input value is "0.8" and the output value at the "Cos" output port of the Sine/Cosine Module is "0.799959", the same value with an error of "0.000041".

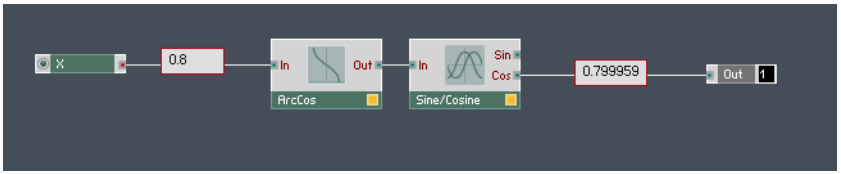


Fig. 4.55 ArcCos represents the inverse trigonometric function of the cosine function. When the two corresponding Modules are connected in series, they approximately cancel each other out.

4.26 ArcTan Module

Fig. 4.56 ArcTan Module

4.26.1 Overview

The ArcTan Module calculates the inverse tangent function. The arc tangent of "x" is a number between "-0.25" and "0.25", the tangent of which is "x". Since the tangent function's output range comprises all floating point numbers in REAKTOR, the arc tangent function is valid for all input values. An Event at the input port triggers an Event at the output port.

Application

The ArcTan Module can be used as an alternative to the Saturator Module (↑12.1, Saturator).

4.26.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value for the scaled argument of the arc tangent function.

Output Ports

- **(Out)** "Out" is the hybrid output port for the scaled arc tangent function of the input value. The output ranges from -0.25 to 0.25.

4.26.3 Example: Arc Tangent as Inverse of Tangent

This example demonstrates how the ArcTan Module acts as the inverse of the tangent function. In the picture below you see a Knob Module with range [-1000 ... 1000] connected to the input port of a ArcTan Module and the output port of the ArcTan Module connected to the input port of a Sine/Cosine Module. Since the arc tangent function is the inverse for the tangent function, any value in the range [-1000 ... 1000] that is fed to the ArcTan Module will again be reproduced (up to a certain error stemming from the floating point format) at the output port of the Divide Module serially connected Sine/Cosine Module. The Divide Module serves to use the equation sin(x) / cos (x) = tan(x) to derive the tangent function from the sine and cosine functions of the Sine/Cosine Module. In the figure below the input value is "-320" and the output value at the output port of the Divide Module is "-320.012", the same value with an error of "0.012".

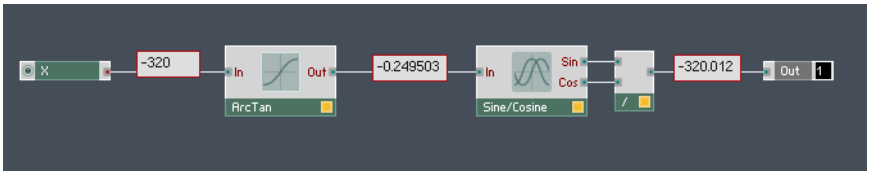


Fig. 4.57 ArcTan represents the inverse trigonometric function of the tangent function. When the two corresponding functions are applied in series to an incoming signal, they approximately cancel each other out.

5 Signal Path

Signal Path Modules allow both control and audio data to be flexibly routed in REAKTOR Structures. These include Mixer Modules, input and output Selector Modules, remote-control switches (known as Relay Modules), Crossfader Modules, and Distributor Modules.

5.1 Selector

Fig. 5.1 Selector module

5.1.1 Overview

The Selector Module forwards one input signal or a mix of two input signals to its output port depending on the value at the "Pos" input port. The input signals are connected to the input ports below the "Pos" input port. These dynamic input ports are called "channel" input ports. Additional "channel" input ports can be created (if all existing "channel" input ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Selector Module. The minimum number of "channel" input ports can be defined with the Min Num Port Groups edit field in the Function page of the Properties Tab. An Event at selected input ports triggers an Event at the output port.

Controlling the Routing

One can distinguish between three cases for the functionality of the Module depending on the value at the "Pos" input port:

- When the value at the "Pos" input port is an integer "n", the signal at the "n"-th "channel" input port, counting from the top, is forwarded to the output port.
- When the "Pos" value lies between two integers, the Module's behavior depends on the Interpolation setting in the Function page of the Properties Tab. There are three possibilities: linear and sine interpolation (this means that a mix of the two input signals whose "channel" index is closest to the "Pos" value is sent to the output port) and no interpo-

lation (only the input signal with the „channel" index closest to the "Pos" value is sent to the output port). The difference between linear and sine interpolation lies in the way that the values between the two "channels" are calculated.

- If the value at the "Pos" input port is greater than the number of "channel" input ports, the Module's behavior depends on the setting of the Wrap checkbox in the Function page of the Properties Tab. If the Wrap checkbox is disengaged, the signal at the lowest lying "channel" input port is forwarded to the output port. Otherwise, the "Pos" value wraps around the number of "channel" input ports, "Max", so that Max + 1 becomes "0", Max + 2 becomes "1", and so on.

Application

Applications of the Selector Module include:

- Using the Selector Module in conjunction with a Compare Module or a Compare/Equal Module to selectively route signals. See the example for the Compare Module for details on this.
- Feeding the "channel" input ports of the Selector Module with the signal from the output ports of the Multi-Tap Delay Module and the "Pos" input port with the output signal of a Ramp Osc Module. This creates a nice delay effect.

5.1.2 Ports

Input Ports:

- **(Pos)** "Pos" (position) is the hybrid input port for selecting the "channel" input port(s) which are forwarded to the output port. Pos = 0 selects In0 (the first "channel"), Pos = 1 selects In1 (the second "channel"), and so on. Pos = 0.5 gives a mix of In0 and In1. The exact crossfading curve between the two "channels" depends on the Interpolation setting in the Function page of the Properties Tab. The typical range for the "Pos" value is [0 ... Max] where "Max" is the number of "channel" input ports.
- **(In0)** "In0" is the first hybrid "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Pos" input port.
- **(In1)** "In1" is the second hybrid "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Pos" input port.

- **(...)** "In2, In3, ..." are the dynamic "channel" input ports. Additional "channel" input ports can be created (if all existing "channel" input ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Selector Module. The maximum number of "channel" input ports for the Selector Module is 16.

Output Ports:

- **(Out)** "Out" is the output port for one input signal or a mix of two input signals depending on the value at the "Pos" input port.

5.1.3 Properties: Function Page

Changing the Minimum Number of Input Ports

To change the minimum number of "channel" input ports, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of "channel" input ports for the Selector Module is 16.

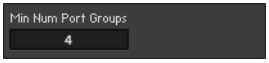


Fig. 5.2 Use the Min Num Ports Groups edit field to set the minimum number of "channel" input ports appearing on the Selector Module.

Changing the Interpolation Setting

If the "Pos" value is between two integers the signal at the output port depends on the Interpolation setting (shown in the second picture below):

► To have no interpolation, that is, for the input signal with the „channel" index closest to the "Pos" value to be sent to the output port, click on the No Interpolation radio button.

► To have linear interpolation, that is, for the "Pos" value to linearly crossfade between the two input signals whose "channel" index is closest to the "Pos" value, click on the Linear Interpolation radio button. The linear interpolation curve is represented by the thin lines in the picture below.

► To have sine interpolation, that is, for the "Pos" value to linearly crossfade between the two input signals whose "channel" index is closest to the "Pos" value, click on the Sine Interpolation radio button. The sine interpolation curve is shown as the thick lines in the picture below. As seen on the graph, sine interpolation curve delivers a higher level for the when the crossfading parameter is in the center position.

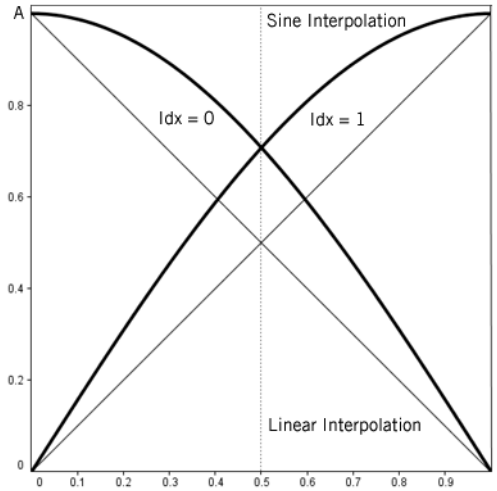


Fig. 5.3 Crossfading curves for linear and sine interpolation.

In the picture above, the vertical axis represents the amplitude factor by which the corresponding signal is multiplied and the horizontal axis represents the value of "Pos", with a range from the lower "channel" index (here, "0") up to the higher "channel" index (here, "1"). In the graph you see that at Pos = 0 the signal from "channel" "0" (falling curve) is multiplied by "1" and the signal from "channel" "1" (rising curve) is multiplied by "0", that is, only the signal from "channel" "0" is passed to the output. Furthermore, at Pos = 1 the signal from "channel" "0" is multiplied by "0" and the signal from "channel" "1" is multiplied by "1", that is, only the signal from "channel" "1" is passed to the output. The graph shows how the signal amplitudes behave between these "Pos" values. The thin lines represent a linear crossfading curve and the thick lines represent a sine crossfading curve.

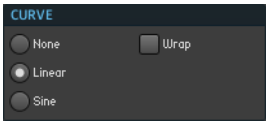


Fig. 5.4 Use the Curve radio button selector to choose the Module's Interpolation setting. Engage the Wrap checkbox to activate the Wrapping feature.

Wrapping the "Pos" Value

If you want the value at the "Pos" input port to "wrap" around the number of "channel" input ports, engage the Wrap checkbox, shown in the figure above. This means that for "Pos" greater than the number of "channel" input ports, "Max", the new "Pos" value is equal to the old value minus a multiple of "Max" so that the new "Pos" value is in the range [0 ... Max]. For example, Max + 1, becomes 0, Max + 2 becomes 2, and so on. For "Pos" less than zero the new "Pos" value is equal to the negative value plus a multiple of "Max". If the Wrap checkbox is disengaged and Pos > Max, the signal at the "channel" input port with the greatest index is forwarded to the output port and if Pos < 0, the signal at the smallest index is forwarded to the output port.

5.1.4 Example: Linear Interpolation Curve

In this example, the Selector Module has Constants connected to the "channel" input ports and a "Pos" value of "2.25". The Interpolation setting is set to "linear" with the Linear Interpolation selector in the Function page. Since 2 < Pos < 3, the output signal will be a mix between the signals from the "2" and "3" channel input ports. In other words, the mix is a linearly crossfaded signal between the constants "6" and "7" with the crossfading parameter "0.25", yielding the output value "6.25".

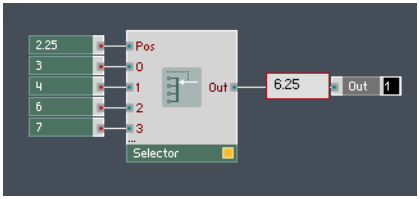


Fig. 5.5 Here the output value has been linearly interpolated between "6" and "7".

5.2 Relay

Fig. 5.6 Relay Module

5.2.1 Overview

The Relay Module acts as a router that selects between two signals at the input ports below the "Ctl" input port, called "channel" input ports. For a value "Ctl" > 0, the upper "channel" input signal is sent to the output. Otherwise the lower "channel" input signal is sent to the output, unless the lower "channel" input port is not present. In that case a zero signal is produced at the output port. The "channel" input ports support dynamic input port management: if the existing "channel" input port is already connected, an additional "channel" input port can be created by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Relay Module. The minimum number of "channel" input ports can be defined with the Min Num Port Groups edit field in the Function page of the Properties Tab. An Event at any input port triggers an Event at the output port.

Application

This Module can be substituted by a Selector Module with one or two "channel" input ports. The Relay Module is only available for backward compatibility because the "Ctl" signal is interpreted differently.

5.2.2 Ports

Input Ports

- **(Ctl)** "Ctl" (control) is the hybrid input port for selecting the "channel" input port which is forwarded to the output port. For a value "Ctl" > 0, the upper "channel" input signal is sent to the output. Otherwise the lower "channel" input signal is sent to the output, unless the lower "channel" input port is not present. In that case a zero signal is produced at the output port.

- **(In0)** "In0" is the upper hybrid "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Ctl" input port.
- **(In1)** "In1" is the lower hybrid "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Ctl" input port.

Output Ports

- **(Out)** "Out" is the output port for an input signal depending on the value at the "Ctl" input port.

5.2.3 Properties: Function Page

Changing the Minimum Number of Input Ports

To change the minimum number of "channel" input ports, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of "channel" input ports for the Selector Module is 2.

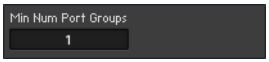


Fig. 5.7 Use the Min Num Ports Groups edit field to set the minimum number of "channel" input ports appearing on the Relay Module.

5.2.4 Example: Choose One Signal

In this example a Relay Module is employed to route one of two Constants to the output. The control signal telling the Relay Module which Constant to forward to the output comes from the In Port labeled "Ctl". Since the "Ctl" signal carries the value "1" (> 0), the signal from the upper "channel" input port, the constant "8", is forwarded to the output.

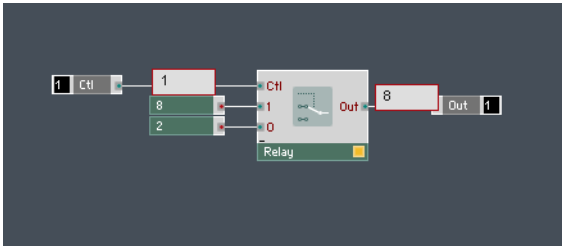


Fig. 5.8 A "Ctl" value of "1" at the corresponding Relay input port causes the value at the upper "channel" input port to be forwarded to the Relay Module's output.

5.3 Crossfade

Fig. 5.9 Crossfade Module

5.3.1 Overview

The Crossfade Module crossfades between two input signals and forwards the resulting mix to the corresponding output port. The two input signals are taken from a pair of input ports below the "X" input port, called "channel" input ports. Since the Crossfade Module supports dynamic port management, additional pairs of "channel" input ports can be created if all existing "channel" input port pairs already have at least one connection. Additional output ports can be created if all existing ports already have a connection. This is done by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an input or output port of another Module to the three dots on the left edge of the Crossfade Module for input ports, and on the right edge of the Crossfade Module for output ports. The minimum number of "channel" pairs can be defined with the Min Num Port Groups edit field in the Function page of the Properties Tab. An Event at any input port triggers an Event at the output port.

Controlling the Routing

The value at the "X" input port determines the mix of the two "channel" input signals at the corresponding output port:

- When the value at the "X" input port is "0", only the signal at the higher lying port of a pair of "channel" input ports is forwarded to the output port.
- When the value at the "X" input port is "1", only the signal at the lower lying port of a pair of "channel" input ports is forwarded to the output port.
- For values of "X" between "0" and "1", the mix depends on the Interpolation setting in the Function page of the Properties Tab.

Application

The Crossfade Module is best applied in cases where you need to crossfade between two types of signals with only one parameter. The two signals would then correspond to the dry and wet signals in an effect, for example.

5.3.2 Ports

Input Ports

- **(X)** "X" (crossfade) is the hybrid input port for the mixing ratio which is determined by values between "0" (Out = In0) and "1" (Out = In1). Values beyond this range are clipped.
- **(In0 0)** "In0 0" is a hybrid input port for the first of two "channel" input signals to be mixed for the corresponding output port.
- **(In0 1)** "In1 1" is a hybrid input port for the second of two "channel" input signals that are mixed for the corresponding output port.
- **(...)** "In1 0, In1 1, ..." are the dynamic "channel" input ports. Additional pairs of "channel" input ports can be created (if all existing "channel" input ports have at least one connected port) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the left edge of the Crossfade Module. The maximum number of "channel" input port pairs for the Crossfade Module is 8.

Output Ports

- **(O)** "O" is the hybrid output for the mix of the "channel" input signals at the input ports labeled with "0".
- **(...)** "1, 2, ..." are the dynamic output ports for the mix of the "channel" input signals at the input ports labeled with "1", "2", and so on, respectively. Since the Crossfade Module supports dynamic port management, additional pairs of output ports can be created (if all existing output ports already have a connection) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an input port of another Module to the three dots on the right edge of the Crossfade Module.

5.3.3 Properties: Function Page

Changing the Minimum Number of Port Groups

To change the minimum number of "channel" input ports, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of "channel" input port pairs for the Crossfade Module is 8.

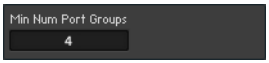


Fig. 5.10 Use the Min Num Ports Groups edit field to set the minimum number of "channel" input port appearing on the Crossfade Module.

Changing the Interpolation Setting

The "X" input port accepts values in the range [0 … 1]. Values outside this range are clipped. The crossfading curve between the two "channel" input signals depends on the Interpolation setting, set with the Curve radio button selector shown in the figure below:

► To have linear interpolation, that is, for the "X" value to linearly crossfade between a pair of two input signals, click on the Linear Interpolation radio button. The linear interpolation curve is represented by the thin lines in the picture below.

► To have sine interpolation, that is, for the "X" value to crossfade between the pair of two input signals according to a sine function, click on the Sine Interpolation radio button. The sine interpolation curve is shown as the thick lines in the picture below. As seen on the graph, sine interpolation curve delivers a higher level when the crossfading parameter is in the center position.

Fig. 5.11 Use the Curve radio button selector to choose the Module's Interpolation setting.

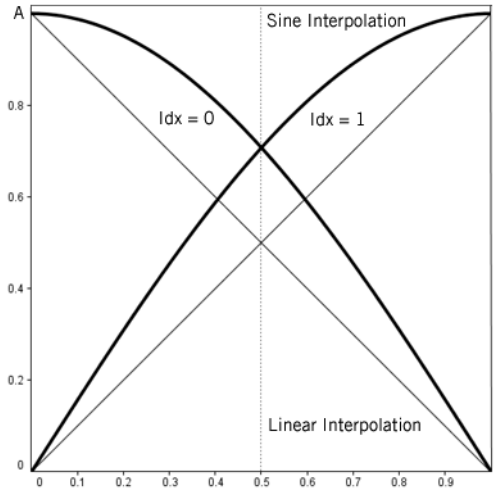


Fig. 5.12 Crossfading curves for linear and sine interpolation.

The vertical axis represents the amplitude factor by which the corresponding signal is multiplied and the horizontal axis represents the value of "X", from the lower "channel" index, "0", to the higher "channel" index, "1". In the graph you see that at X = 0 the signal from "channel" "0" (falling curve) is multiplied by "1" and the signal from "channel" "1" (rising curve) is multiplied by "0", that is, only the signal from "channel" "0" is passed to the output. Furthermore, at X = 1 the signal from "channel" "0" is multiplied by "0" and the signal from "channel" "1" is multiplied by "1", that is, only the signal from "channel" "1" is passed to the output. The graph shows how the signal amplitudes behave between these "X" values. The thin lines represent a linear crossfading curve and the thick lines represent a sine crossfading curve.

5.3.4 Example: Parallel Crossfading Between Two Signals

In this example you have two pairs of dry and wet signals that you want to mix using the same parameter. The first pair is labeled "dry0" and "wet0", and the signals carry the values "2" and "4", respectively. The second pair is labeled "dry1" and "wet1", and the signals carry the values "5" and "6", respectively. The crossfading parameter "X" is equal to "0.49" and the Linear Interpolation radio button has been selected in the Function page of the Crossfade Module. Now you have a case where the output port "0" is a linearly crossfaded mix between the signals "dry0" and "wet0" and the output port "1" carries a linearly crossfaded

mix between the signals "dry1" and "wet1". Since the parameter "X" is set to "0.49", the signal at the "0" output port is a linearly crossfaded signal between the constants "2" and "4" with the crossfading parameter "0.49", yielding the output value "2.98". The same way, , the signal at the "1" output port is a linearly crossfaded signal between the constants "5" and "6" with the crossfading parameter "0.49", yielding the output value "5.49". The picture below shows how the Structure to this example would look like in REAKTOR.

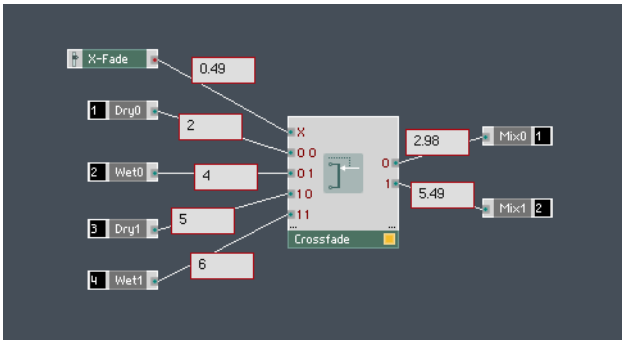


Fig. 5.13 The Crossfade Module in this example performs a linear crossfade between Dry0 and Wet0, and Dry1 and Wet1, respectively.

5.4 Distributor

Fig. 5.14 Distributor Module

5.4.1 Overview

The Distributor Module distributes the signal at the "In" input port between one or two output ports, depending on the value at the "Pos" input port. Since the Distributor Module has dynamic output ports, additional output ports can be created (if all existing output ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an input port of another Module to the three dots on the right edge of the Distributor Module. The minimum number of input ports can be defined with the Min Num Port Groups edit field in the Function page of the Properties Tab. An Event at any input port triggers Events only at the output ports that are sending nonzero signals.

The output ports send Events in the order of their index, from smallest to greatest. For example, an Event at an input port causes the "0" output port first to send the Event, followed by the "1" output port (granted that the "Pos" value is in the range between of the said port indices).

Controlling the Routing

One can distinguish between three cases for the functionality of the Module depending on the value at the "Pos" input port:

- When the value at the "Pos" input port is an integer "n", the input signal is forwarded to the "n"-th output port.
- When the "Pos" value lies between two integers, the Module"s behavior depends on the Interpolation setting in the Function page of the Properties Tab. There are three possibilities: linear and sine interpolation (this means that the input signal is distributed between the two output ports whose index is closest to the "Pos" value) and no interpolation (the input signal is only sent to the output port with its index closest to the "Pos" value. The difference between linear and sine interpolation lies in the way that the signal is distributed between the two output ports.
- If the value at the "Pos" input port is greater than the number of output ports, the Module's behavior depends on the setting of the Wrap checkbox in the Function page of the Properties Tab. If the Wrap checkbox is disengaged, the input signal is forwarded to the lowest lying output port. Otherwise, the "Pos" value wraps around the number of output ports, "Max", so that Max + 1 becomes "0", Max + 2 becomes "1", and so on.

Application

The Distributor Module is used to dynamically route signals, such as choosing a "send" effect from a number of effects, connected to the output ports, for an audio signal.

5.4.2 Ports

Input Ports

- **(Pos)** "Pos" (position) is the hybrid input port for selecting the output port(s) between which the signal is distributed. Pos = 0 selects "0" (the first output port), Pos = 1 selects "1" (the second output port), and so on. Pos = 0.5 evenly distributes the input signal between the output ports "0" and "1". In this case, the exact levels of the signals

at these output ports depend on the Interpolation setting in the Function page of the Properties Tab. The typical range for the "Pos" value is [0 ... Max] where "Max" is the number of output ports.

- **(In)** "In" is the hybrid input port for the input signal to be distributed between one or two output ports depending on the value at the "Pos" input port.

Output Ports

- **(0)** "0" is the first hybrid output port to which the input signal may be sent depending on the value at the "Pos" input port.
- **(1)** "1" is the second hybrid output port to which the input signal may be sent depending on the value at the "Pos" input port.
- **(...)** "2, 3, ..." are the dynamic output ports. Additional output ports can be created (if all existing output ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the right edge of the Distributor Module. The maximum number of output ports for the Distributor Module is 16.

5.4.3 Properties: Function Page

Changing the Minimum Number of Output Ports

To change the minimum number of output ports, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of output ports for the Distributor Module is 16.

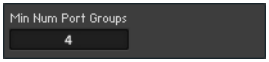


Fig. 5.15 Use the Min Num Ports Groups edit field to set the minimum number of output ports appearing on the Distributor Module.

Changing the Interpolation Setting

If the "Pos" value is between two integers the signal is distributed between two output ports according to the Interpolation setting, set with the Curve radio button selector, shown in the second figure below:

- To have no interpolation, that is, for the input signal only to be sent to the output port with an index closest to the "Pos" value, click on the No Interpolation radio button.

- To have linear interpolation, that is, for the input signal to be distributed with a linear crossfading curve between two output ports, click on the Linear Interpolation radio button. The input signal is distributed between the two output ports whose index is closest to the "Pos" value. The linear interpolation curve is represented by the thin lines in the picture below.
- To have sine interpolation, that is, for the input signal to be distributed with a sine crossfading curve between two output ports, click on the Sine Interpolation radio button. The input signal is distributed between the two output ports whose index is closest to the "Pos" value. The sine interpolation curve is represented by the thick lines in the picture below. As seen on the graph, sine interpolation curve delivers a higher level for the when the crossfading parameter is in the center position.

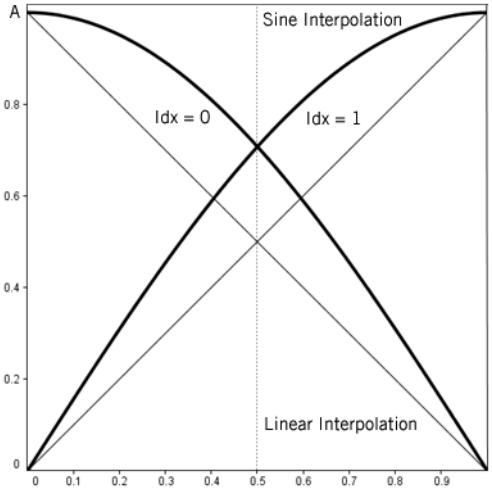


Fig. 5.16 Crossfading curves for linear and sine interpolation.

In the picture above, the vertical axis represents the amplitude factor by which the corresponding signal is multiplied and the horizontal axis represents the value of "Pos", with a range from the lower port index (here, "0") up to the higher port index (here, "1"). In the graph you see that at Pos = 0 the signal at output port "0" (falling curve) is the input signal multiplied by "1" and the signal at output port "1" (rising curve) is the input signal multiplied by "0". In other words, that is, the input signal is only passed to the "1" output port. Furthermore, at Pos = 1 the signal at output port "0" (falling curve) is the input signal multiplied by "0" and the signal at output port "1" (rising curve) is the input signal

multiplied by "1". The graph shows how the signal amplitudes behave between these "Pos" values. The thin lines represent a linear crossfading curve and the thick lines represent a sine crossfading curve.

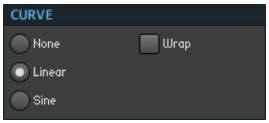


Fig. 5.17 Use the Curve radio button selector to choose the Module's Interpolation setting. Engage the Wrap checkbox to activate the Wrapping feature.

Wrapping the "Pos" Value

If you want the value at the "Pos" input port to "wrap" around the number of "channel" output ports, engage the Wrap checkbox shown in the figure above. This means that for "Pos" greater than the number of output ports, "Max", the new "Pos" value is equal to the old value minus a multiple of "Max" so that the new "Pos" value is in the range [0 ...Max]. For example, Max + 1, becomes 0, Max + 2 becomes 2, and so on. For "Pos" less than zero, the new "Pos" value is equal to the negative value plus a multiple of "Max". If the Wrap checkbox is disengaged and Pos > Max, the signal is sent to the output port with the greatest index and if Pos < 0, the signal is sent to the output port with the smallest index.

5.4.4 Example: Distribution Between 4 Ports

This example portrays a case where a Distributor Module distributes an incoming constant signal at the "In" input port, carrying the value "3", to four output ports. Since the value at the "Pos" input port is "2.5", the input signal is distributed between the "2" and "3" output ports. The Linear Interpolation selector has been selected in the Function page of the Crossfade Module and since the crossfading parameter is "0.5", the "2" and "3" output ports carry the same value, "1.5". The "0" and "1" output ports are sending a zero signal.

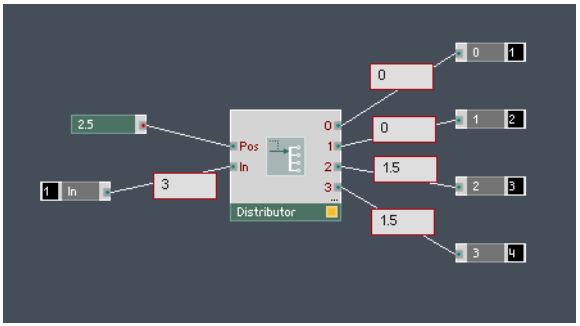


Fig. 5.18 The Distributor Module in this example distributes the input signal (the value "3") between four output ports. The "Pos" value causes the value to be linearly interpolated between the "2" and "3" output ports.

5.5 Stereo Pan

Fig. 5.19 Stereo Pan Module

5.5.1 Overview

The Stereo Pan Module acts as stereo panner for an adjustable number of „channel" input signals which are connected to the "In" input ports. The "channel" input signals are panned between the left and right stereo channels according to the value at the "Pan" input port. The sum of the "L" and "R" output port values is always exactly twice the value at the corresponding "channel" input port. The "channel" input signal is split across the two corresponding output ports at a variable ratio. The Stereo Pan Module has dynamic port management. Additional "channel" input ports can be created if all existing input ports have a connection. Additional output ports can be created if all existing output port pairs already have at least one connection. The ports are created by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port or input port of another Module to the three dots on the left or right edge of the Stereo Pan Module for "channel" input ports or output ports, respectively. The minimum number of "channel" input ports can be defined with the Min Num Port Groups edit field in the Function page of the

Properties Tab. An Event at the "Pan" input port triggers Events at all output ports. An Event at a "channel" input port triggers Events only at the "L" and "R" output ports associated with that particular "channel".

The order of the Events at the output ports goes from top to bottom.

Application

Applications of the Stereo Pan Module include:

- Placing individual voices of a Polyphonic Instrument in the stereo field, as shown in the tutorial in subsection 9.3.4 of the Application Reference.
- Panning Event related signal processing such as LFOs or Envelopes in the stereo field.

5.5.2 Ports

Input Ports

- **(Pan)** "Pan" is the input port for the control signal that specifies the position of the "channel" input signals in the stereo panorama. When Pan = -1, the signals are completely panned to the left stereo channel, when Pan = 0, the signals are in the center of the stereo field, and when Pan = 1, the signals are completely panned to the right stereo channel.
- **(In0)** "In0" is the "channel" input port for the first audio signal to be amplified and to be positioned in the stereo panorama.
- **(...)** "In1, In2, ..." are the dynamic input ports. Additional input ports be created (if all existing input ports already have a connection) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the left side of the Stereo Pan Module. The maximum number of input ports for the Stereo Pan Module is 8.

Output Ports

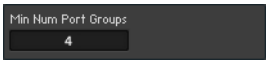
- **(L0)** "L0" (left) is the output port for the left stereo channel of the mixed signal.
- **(R0)** "R0" (right) is the output port for the right stereo channel of the mixed signal. **(...)** "L1, R1, L2, R2, ..." are the dynamic output ports. Additional output ports be created (if all existing output port groups already have at least one connection) by holding

down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an input port of another Module to the three dots on the right side of the Stereo Pan Module. The maximum number of output port groups for the Stereo Pan Module is 8.

5.5.3 Properties: Function Page

Changing the Minimum Number of Port Groups

► To change the minimum number of input port groups, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of port groups for the Stereo Pan Module is 8



Changing the Interpolation Setting

The "Pan" input port accepts values in the range [-1 … 1]. Values outside this range are clipped. The crossfading curve between the left and right "Pan" settings depends on the Interpolation setting, set with the Curve radio button selector shown in the figure below:

► To have linear interpolation, that is, for the "Pan" value to linearly crossfade from left panning to right panning, click on the Linear Interpolation radio button. The linear interpolation curve is qualitatively represented by the thin lines in the picture below.

► To have sine interpolation, that is, for the "Pan" value to crossfade from left panning to right panning according to a sine function, click on the Sine Interpolation radio button. The sine interpolation curve is qualitatively shown as the thick lines in the picture below. As seen on the graph, sine interpolation curve delivers a higher overall output level when the crossfading parameter is in the center position.

► Use the Curve radio button selector to choose the Module's Interpolation setting.

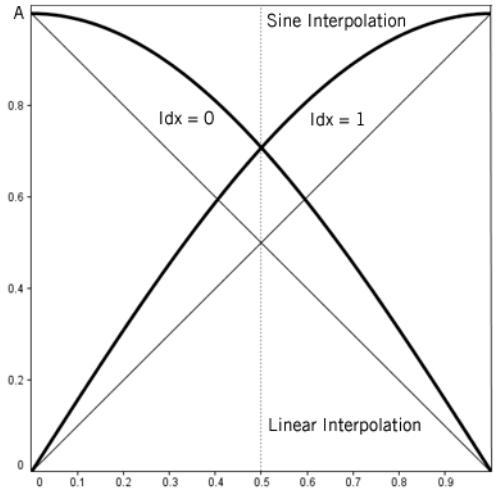


Fig. 5.20 Qualitative Crossfading curves for linear and sine interpolation. From left to right, the graph goes from left panning to center panning to right panning.

5.5.4 Example: Panning Constant Signals

This example demonstrates that the sum of the "L" and "R" output port values of the Stereo Pan Module is always exactly twice the value at the corresponding "channel" input port. As shown in the picture below, we have two "channel" input ports labeled "In1" and "In2", and a "Pan" parameter of "-0.25". Since the value carried by "channel" input signal "In1" is "3", the range for the values at the output ports "L1" and "R1" is [0 ... 6]. The maximum value is twice the value at the corresponding "channel" input port. For a Pan = -1 the "channel" input signal with double amplitude would only be sent to the left stereo channel and a zero signal would be sent to the right stereo channel, that is, "L1" = 6 and "R1" = 0. For a Pan = 1 the "channel" input signal with double amplitude would only be sent to the right stereo channel and a zero signal would be sent to the left stereo channel, that is, "L2" = 0 and "R1" = 6. All other values for "Pan" results in output signal levels that are linearly interpolated between these two extreme cases. Hence, a "Pan" value of "-0.25" yields a signal carrying the value "3.75" for the "L1" output port and a signal carrying the value "2.25" for the "R1" output port

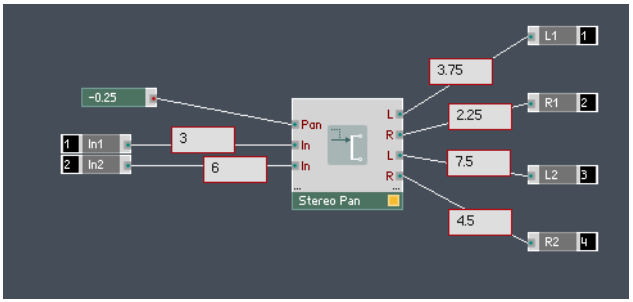


Fig. 5.21 The Stereo Pan Module distributes the In1 signal between the first pair of "L" and "R" output ports and the In2 signal between the second pair of "L" and "R" output ports.

5.6 Mixer

Fig. 5.22 Mixer Module

5.6.1 Overview

The Mixer Module acts as an amplifier or mixer for an adjustable number of „channel" input signals. The "channel" input signals are connected to the "In" input ports and are amplified (or attenuated) by the values (in dB) at the respective "Lvl" input ports and then summed to form the output signal at the output port. The Mixer Module has dynamic input port management. Additional input port pairs ("Lvl" and "In") can be created (if all existing input port pairs already have at least one connection) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the left edge of the Mixer Module. The minimum number of input port pairs can be defined with the Min Num Port Groups edit field in the Function page of the Properties Tab.

Application

The Mixer Module can be used, for example, to mixdown the oscillator section of a synthesizer (with each oscillator having its own signal level) before sending the mix to the filters. A qualitatively similar implementation of this is shown in the example.

5.6.2 Ports

Input Ports

- **(LvlO)** "LvlO" (level) is the logarithmic control input port for controlling the amplifier gain factor for the first "channel" input port, given in dB. When the value is negative, the output signal level is smaller than the input signal level.
- **(InO)** "InO" is the "channel" input port for the audio signal to be amplified.
- **(...)** "In1, Lvl1, In2, Lvl2, ..." are the dynamic input ports. Additional input port pairs can be created (if all existing input port pairs already have at least one connection) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the left side of the Mixer Module. The maximum number of input port pairs for the Mixer Module is 20.

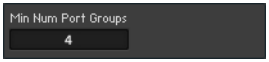
Output Ports

- **(Out)** "Out" is the output port for the mixed signal: Out = In0 · 10 ^ (LvlO / 20) + In1 · 10 ^ (Lvl1 / 20) + ...

5.6.3 Properties: Function Page

Changing the Minimum Number of Input Port Pairs

► To change the minimum number of output ports, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of input port pairs for the Mixer Module is 20.



5.6.4 Example: Mixing Two Oscillators

The Mixer Module can be used to mixdown the oscillator section of a synthesizer (with each oscillator having its own signal level) before sending the mix to the filters. In this example the oscillator section consists of a Sine Module and a Sawtooth Module, both sending an oscillator waveform with unit amplitude and the same pitch, determined by the Note Pitch Module. The output signals of the two oscillators are connected to the "In" ports of the Mixer Module. The signal level for the sine wave and the sawtooth wave is specified by the "Sine Level" and "Saw Level" Knob Modules, respectively. The signal at

the output port of the Mixer Module is then a mix of both sine and oscillator waves with each having a signal level that you can specify with their corresponding Knob Modules. Such a Structure is shown in the picture below.

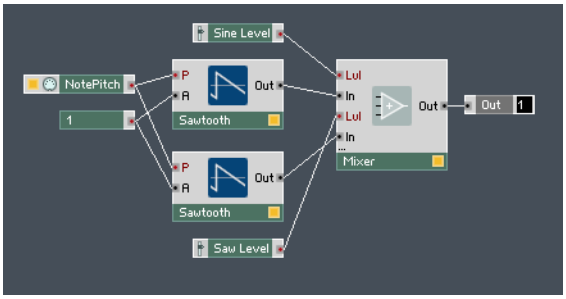


Fig. 5.23 For example, you can use the Mixer Module to mix two oscillator signals.

5.7 Stereo Mixer

Fig. 5.24 Stereo Mixer Module

5.7.1 Overview

The Stereo Mixer Module acts as an amplifier or mixer with stereo output for an adjustable number of „channel" input signals which are connected to the "In" input ports. The "channel" input signals are amplified (or attenuated) by the values (in dB) at the respective "Lvl" input ports and panned between the left and right stereo channels according to the value at the respective "Pan" input port. Then the left and right stereo channels are each summed up separately and sent to the "L" and "R" output port, respectively. The Stereo Mixer Module has dynamic input port management. Additional input port groups ("Lvl", "Pan", and "In") can be created (if all existing input port groups already have at least connection) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the left edge of the Stereo Mixer Module. The minimum number of input port groups can be defined with the Min Num Port Groups edit field in the Function page of the Properties Tab.

Application

The Stereo Mixer Module can be used, for example, to mixdown the drums of the sampler section of a drum instrument. Each drum sample can have its own signal level and location in the stereo panorama. The output signal is then a stereo mix of the drums that can be sent to the effects for further treatment. A qualitatively similar implementation of this is shown in the example.

5.7.2 Ports

Input Ports:

- **(LvlO)** "LvlO" (level) is the logarithmic control input port for controlling the amplifier gain factor of the first "channel" input port, given in dB. When the value is negative, the output signal level is smaller than the input signal level.
- **(PanO)** "PanO" is the input port for the control signal that specifies the position of the corresponding "channel" input signal in the stereo panorama. When Pan = -1, the signal is completely panned to the left stereo channel, when Pan = 0, the signal is in the center of the stereo field, and when Pan = 1, the signal is completely panned to the right stereo channel.
- **(InO)** "InO" is the "channel" input port for the audio signal to be amplified and to be positioned in the stereo panorama.
- **(...)** "In1, Lvl1, Pan1, In2, Lvl2, Pan2, ..." are the dynamic input ports. Additional input port groups can be created (if all existing input port groups already have at least one connection) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the left side of the Stereo Mixer Module. The maximum number of input port groups for the Stereo Mixer Module is 13.

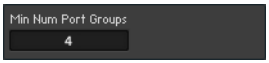
Output Ports:

- **(L)** "L" (left) is the output port for the left stereo channel of the mixed signal.
- **(R)** "R" (right) is the output port for the right stereo channel of the mixed signal.

5.7.3 Properties: Function Page

Changing the Minimum Number of Input Port Groups

► To change the minimum number of input port groups, type the desired number into the Min Num Port Groups edit field (shown below). The maximum number of input port groups for the Stereo Mixer Module is 13.



5.7.4 Example: Stereo Drums

The Stereo Mixer Module can be used to mixdown the drums of the sampler section of a drum instrument. Each drum sample can have its own signal level and location in the stereo panorama. The output signal is then a stereo mix of the drums that can be sent to the effects for further treatment. In this example, two Sampler Modules with the labels "Kick" and "Snare" were employed to provide the monophonic signal of a kick drum and snare drum, respectively, for the Stereo Mixer Module. The pitch and triggering signals are received externally, perhaps from a sequencer, through the "P" and "Tr" In Ports. Since we are manually specifying the signal level and panorama position of both drums with the Stereo Mixer, both "A" input ports of the Sampler Modules receive a Constant with the value "1". You can now use the Knob Modules, as seen in the picture below, to specify both the signal level and the positioning of the signal in the stereo field for both drums separately, before they are mixed down to the two stereo channels, "L" and "R".

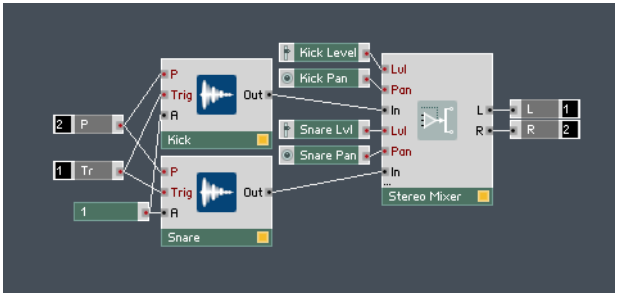


Fig. 5.25 Use the Stereo Mixer Module to mix stereo signals such as output signals of stereo Sampler Modules.

6 Oscillators

An oscillator is a signal generating process that results in a periodic audio signal. Oscillators are usually used as audio signal sources. The only other alternatives for generating audio signals in REAKTOR are playing samples using Samplers or using an external audio signal. The oscillator section includes the usual, single-cycle waveform generators (sine, pulse, sawtooth, and so on) as well as impulse, step, noise, and table-driven generators for generating audio signals. There are also a couple of Modules, like the Clock Oscillator Module and Ramp Module, which are used for purposes other than audio signal generation.
All of REAKTOR's oscillators can run at any frequency, from 0 Hz (standstill) through the entire audio range right up to the limit set by the sample rate. Since very low frequencies are allowed, these oscillators are also suited for use as LFOs, although the audio rate processing requires more CPU resources. Furthermore, oscillators with waveforms that, have strong transients (like the Sawtooth Module) have anti-aliasing in REAKTOR and are therefore not well suited for use as LFOs. When you do decide to use an oscillator as an LFO to modulate another Module"s Event input port (for example "P" as opposed to "F"), you need to insert an A to E (perm) Module to convert the audio signal to an Event signal.

6.1 Sawtooth Oscillator

Fig. 6.1 Sawtooth Module

6.1.1 Overview

The Sawtooth Module is an oscillator with a sawtooth waveform, as shown on the Structure icon of the Module. It features logarithmic pitch control at the "P" input port and linear amplitude modulation at the "A" input port.

Application

The sawtooth oscillator delivers a signal rich in harmonics which is a good starting point for subtractive synthesis.

6.1.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the sawtooth waveform.

6.1.3 Example: Rehearsing the Oscillator

This example shows how to get a sawtooth waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Sawtooth Module is sent to the "P" input port from the output port of the Note Pitch Module.

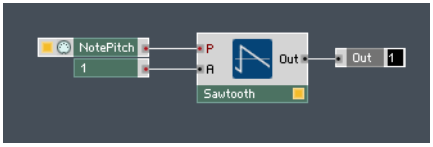


Fig. 6.2 A simple implementation of the Sawtooth Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

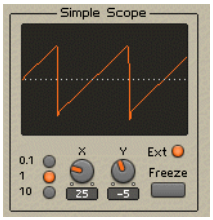


Fig. 6.3 The sawtooth waveform.

6.2 Saw FM Oscillator

Fig. 6.4 Saw FM Module

6.2.1 Overview

The Saw FM Module is an oscillator with a sawtooth waveform, as shown on the Structure icon of the Module. It features logarithmic pitch control at the "P" input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis.

Application

The sawtooth oscillator delivers a signal rich in harmonics which is a good starting point for subtractive synthesis. Since the Saw FM Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.

6.2.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 ... 127] where "69" corresponds to "Concert A" or 440 Hz.

- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated sawtooth waveform.

6.2.3 Example: Rehearsing the Oscillator

This example shows how to get a sawtooth waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by a sine wave. The pitch information for the Saw FM Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

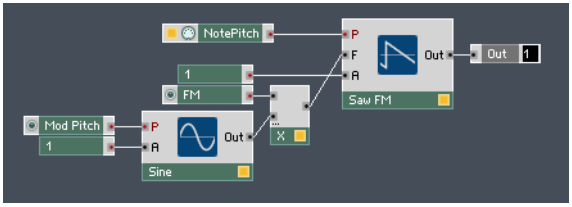


Fig. 6.5 A simple implementation of the Saw FM Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

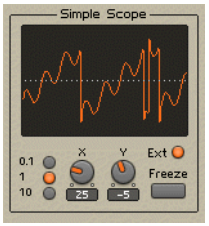


Fig. 6.6 An example of a frequency modulated sawtooth waveform.

6.3 Saw Sync Oscillator

Fig. 6.7 Saw Sync Module

6.3.1 Overview

The Saw Sync Module is an oscillator with a sawtooth waveform, as shown on the Structure icon of the Module. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. In addition, this Module supports phase synchronization (also known as hard sync): a positive zero crossing of the signal at the "Snc" input port causes the phase of the sawtooth oscillator to be reset to the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive. The signal at the "Snc" input port is usually another oscillator. When the synchronizing oscillator has the same frequency as the synchronized oscillator, there is no audible effect. The frequency of the sawtooth sync oscillator can be linearly modulated by an audio signal at the "F" input port, making FM synthesis possible.

Application

The sawtooth oscillator delivers a signal rich in harmonics which is a good starting point for subtractive synthesis. Also, the oscillator can be hard synced to another oscillator to create the classic oscillator sync sound. Since the Saw Sync Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.

6.3.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the phase synchronization.
- **(Snc)** "Snc" (sync) is the Audio input port for controlling synchronization of the waveform. A positive crossing of the signal at this input port restarts the oscillator phase from the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive.
- **(Ph)** "Ph" (phase) is the input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. The range for the "Ph" input port is [-1 … 1] where "-1" corresponds to a phase of -180 degrees and "1" corresponds to a phase 180 degrees.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated and/or hard synchronized sawtooth waveform.

6.3.3 Example: Rehearsing the Oscillator

This example shows how to get a sawtooth waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation and hard synchronization by two separate sine waves. The pitch information for the Saw Sync Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module. The Saw Sync Module is hard synchronized to a second sine wave, the

pitch of which is determined by the "Snc Pitch" Knob Module. The phase to which the sawtooth wave is reset upon a positive zero-crossing at the "Snc" input port is determined by the "SncPhase" Knob Module at the "Ph" input port.

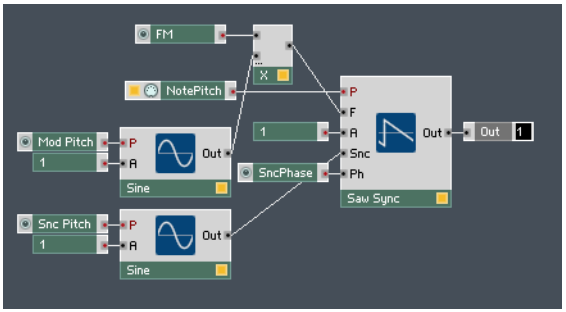


Fig. 6.8 A simple implementation of the Saw Sync Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.9 An example of a hard synchronized sawtooth waveform.

6.4 Saw Pulse Oscillator

Fig. 6.10 Saw Pulse Module

6.4.1 Overview

The Saw Pulse Module is an oscillator with a variable waveform that morphs between a sawtooth and an impulse train. You can morph between the sawtooth and impulse waveforms by changing the slope of the ramp with a signal at the "Slp" input port. The Saw Pulse Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

The Saw Pulse Module delivers a signal with variable richness in harmonics which is a good starting point for subtractive synthesis. Morphing the waveform from a sawtooth to an impulse train decreases the amount of lower harmonics and increases the amount of higher harmonics in the signal spectrum.

6.4.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the "Slp" value.
- **(Slp)** "Slp" (slope) is the Audio input port for controlling the slope of the waveform. A value of "0" produces a normal sawtooth waveform, a larger value shortens the ramp to an impulse. The range for the "Slp" input port is [0 … 20] where "0" corresponds to the sawtooth waveform and "20" corresponds to the impulse train.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the sawtooth/pulse waveform.

6.4.3 Example: Rehearsing the Oscillator

This example shows how to get a sawtooth/pulse waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Saw Pulse Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

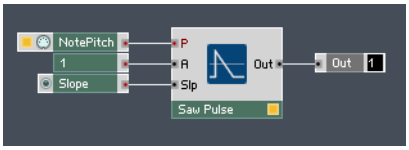


Fig. 6.11 A simple implementation of the Saw Pulse Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

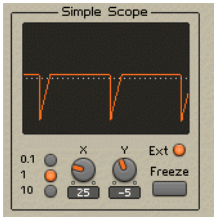


Fig. 6.12 A possible saw pulse Waveform.

6.5 Bi-Saw Oscillator

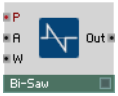


Fig. 6.13 Bi-Saw Module

6.5.1 Overview

The Bi-Saw Module is an oscillator with a bipolar sawtooth waveform featuring pulse width modulation at the "W" input port. The positive zero-crossing of the waveform is locked to zero-phase, so the pulse width modulation results in a modulation of the slope of the ramps. A positive zero crossing of a signal happens when signal values go from negative to positive. The Bi-Saw Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

The Bi-Saw Module delivers a signal with variable richness in harmonics which is a good starting point for subtractive synthesis. Morphing the waveform from a "W" value of "0" to a a "W" value of "1" decreases the amount of lower harmonics and increases the amount of higher harmonics in the signal.

6.5.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(W)** "W" (width) is the Audio input port for controlling the width of the zero signal between individual sawtooth ramps (also known as pulse width modulation or PWM). The positive zero-crossing of the waveform is locked to zero-phase, so the pulse width modulation results in a modulation of the slope of the ramps. A positive zero crossing of a signal happens when signal values go from negative to positive. The range of values for

"W" is about [0 … 6] where the upper bound can be stretched a bit. For W = 0 you get a normal sawtooth waveform and larger values for "W" result in steeper ramps between and a longer zero signal between these ramps.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with a bipolar sawtooth waveform.

6.5.3 Example: Rehearsing the Oscillator

This example shows how to create a bipolar sawtooth waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Bi-Saw Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width value is determined with a Knob Module labeled "P-Width" at the "W" input port.

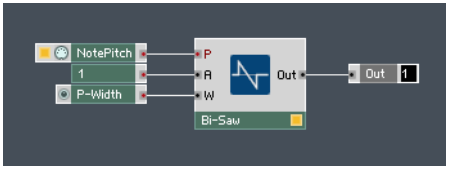


Fig. 6.14 A simple implementation of the Bi-Saw Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

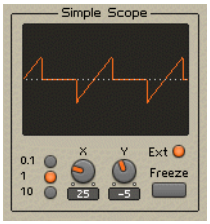


Fig. 6.15 A possible bi-saw waveform.

6.6 Triangle Oscillator

Fig. 6.16 Triangle Module

6.6.1 Overview

The Triangle Module is an oscillator with a symmetric triangle waveform, as shown on the Structure icon of the Module. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

The triangle oscillator delivers a signal that has only odd harmonics present, with amplitudes that decrease as the square of the harmonic number. This yields a mellow timbre, somewhere between a sine and pulse wave. An upwardly transposed triangle wave can be mixed to a sum of sine waves, for example, to add harmonic content. It can also be useful as a modulating signal in frequency modulation or ring moduation where the harmonic density should not be too great, as might be the case with the sawtooth waveform.

6.6.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 ... 127] where "69" corresponds to "Concert A" or 440 Hz.

- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the triangle waveform.

6.6.3 Example: Rehearsing the Oscillator

This example shows how to get a triangle waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Triangle Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

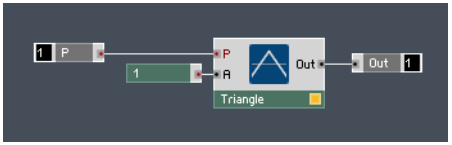


Fig. 6.17 A simple implementation of the Triangle Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

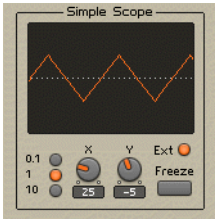


Fig. 6.18 The triangle waveform.

6.7 Tri FM Oscillator

Fig. 6.19 Triangle FM Module

6.7.1 Overview

The Triangle FM Module is an oscillator with a symmetric triangle waveform, as shown on the Structure icon of the Module. It features logarithmic pitch control at the "P" (pitch) input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis.

Application

The triangle oscillator delivers a signal that has only odd harmonics present, with amplitudes that decrease as the square of the harmonic number. This yields a mellow timbre, somewhere between a sine and pulse wave. An upwardly transposed triangle wave can be mixed to a sum of sine waves, for example, to add harmonic content. It can also be useful as a modulating signal in frequency modulation or ring moduation where the harmonic density should not be too great, as might be the case with the sawtooth waveform. Since the Triangle FM Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.

6.7.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.

- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated triangle waveform.

6.7.3 Example: Rehearsing the Oscillator

This example shows how to get a triangle waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by a sine wave. The pitch information for the Triangle FM Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

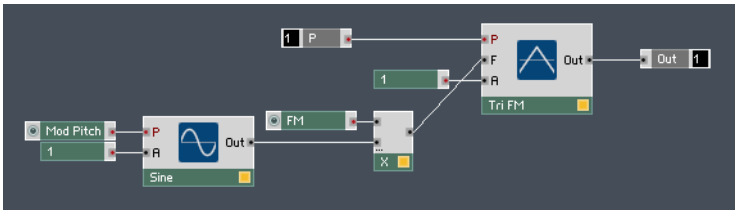


Fig. 6.20 A simple implementation of the Tri FM Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.21 An example of a frequency modulated triangle waveform.

6.8 Tri Sync Oscillator

Fig. 6.22 Tri Sync Module

6.8.1 Overview

The Tri Sync Module is an oscillator with a triangle waveform, as shown on the Structure icon of the Module. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. In addition, this Module supports phase synchronization (also known as hard sync): a positive zero crossing of the signal at the "Snc" input port causes the phase of the triangle oscillator to be reset to the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive. The signal at the "Snc" input port is usually another oscillator. When the synchronizing oscillator has the same frequency as the synchronized oscillator, there is no audible effect. The frequency of the triangle sync oscillator can be linearly modulated by an audio signal at the "F" input port, making FM synthesis possible.

Application

The triangle oscillator delivers a signal that has only odd harmonics present, with amplitudes that decrease as the square of the harmonic number. This yields a mellow timbre, somewhere between a sine and pulse wave. An upwardly transposed triangle wave can be mixed to a sum of sine waves, for example, to add harmonic content. It can also be useful as a modulating signal in frequency modulation or ring moduation where the harmonic

density should not be too great, as might be the case with the sawtooth waveform. Since the Triangle Sync Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis. Also, the oscillator can be hard synced to another oscillator to create the classic oscillator sync sound.

6.8.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the phase synchronization..
- **(Snc)** "Snc" (sync) is the Audio input port for controlling synchronization of the waveform. A positive crossing of the signal at this input port restarts the oscillator phase from the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive.
- **(Ph)** "Ph" (phase) is the input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. The range for the "Ph" input port is [-1 … 1] where "-1" corresponds to a phase of -180 degrees and "1" corresponds to a phase 180 degrees.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated and/or hard synchronized triangle waveform.

6.8.3 Example: Rehearsing the Oscillator

This example shows how to get a triangle waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation and hard synchronization by two separate sine waves. The pitch information for the Tri Sync Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of

the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module. The Tri Sync Module is hard synchronized to a second sine wave, the pitch of which is determined by the "Snc Pitch" Knob Module. The phase to which the triangle wave is reset upon a positive zero-crossing at the "Snc" input port is determined by the "SncPhase" Knob Module at the "Ph" input port.

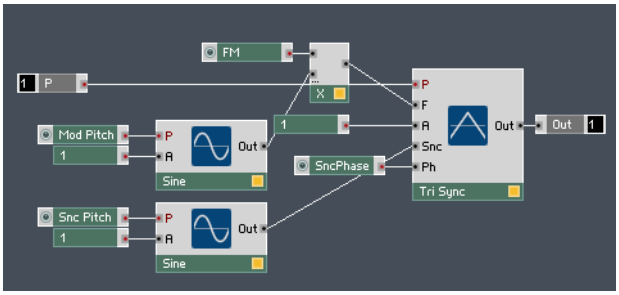


Fig. 6.23 A simple implementation of the Tri Sync Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.24 An example of a hard synchronized triangle waveform.

6.9 Tri/Par Symm Oscillator

Fig. 6.25 Tri/Par Symm Module

6.9.1 Overview

The Tri/Par Symm Module is an oscillator that generates two basic waveforms: parable and triangle. You can change the symmetry of the waveforms from completely symmetric (less harmonics) to asymmetric (more harmonics) by controlling the value at the "W" input port. The Tri/Par Symm Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

The Tri/Par Symm Module delivers signals with variable richness in harmonics. These output signals are good as modulating signals in frequency modulation or ring modulation. They can also be a starting point for subtractive synthesis.

6.9.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(W)** "W" (width) is the Event input port for controlling the symmetry of the waveform. A value of "-1" produces a falling-ramp sawtooth with a linear ramp at the "Tri" output port and a parabolic ramp at the "Par" output port. "0" produces a triangle waveform at the "Tri" output port and a parabolic waveform at the "Par" output port. "+1" produces a rising-ramp sawtooth with a linear ramp at the "Tri" output port and a parabolic ramp at the "Par output port.

Output Ports

- **(Tri)** "Tri" (triangle) is the output port for the audio signal with the variable triangle waveform.
- **(Par)** "Par" (parable) is the output port for the audio signal with the variable parable waveform.

6.9.3 Example: Rehearsing the Oscillator

This example shows how to create triangle and parabolic waveforms with variable symmetry, constant unit amplitude, and a pitch set with the MIDI controller or the keyboard. The pitch information for the Tri/Par Symm Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The symmetry value is determined with a Knob Module labeled "Symm" at the "W" input port.

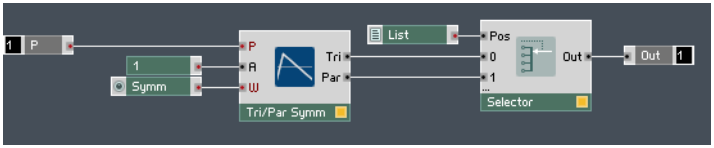


Fig. 6.26 A simple implementation of the Tri/Par Symm Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

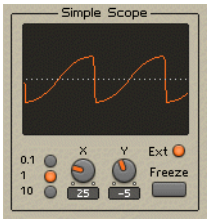


Fig. 6.27 A possible waveform of the Tri/Par Symm Module with the "Par" setting.

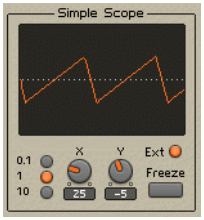


Fig. 6.28 A possible waveform of the Tri/Par Symm Module with the "Tri" setting.

6.10 Parabol Oscillator

Fig. 6.29 Parabol Module

6.10.1 Overview

The Parabol Module is an oscillator with a parabolic waveform, as shown on the Structure icon of the Module. The waveform consists of two parabolas that meet at zero level. The Parabol Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

The parabolic waveform sounds like a sine wave with some added odd numbered overtones at very low level. In many cases the parabolic waveform can be used as replacement for a sine generator with less computational load.

6.10.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" is the output port for audio signal with the parabolic waveform.

6.10.3 Example: Rehearsing the Oscillator

This example shows how to get a parabolic waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Parabol Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

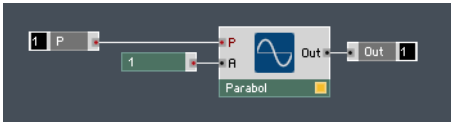


Fig. 6.30 A simple implementation of the Parabol Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

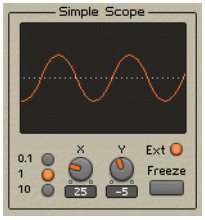


Fig. 6.31 The waveform of the parabol oscillator.

6.11 Par FM Oscillator

Fig. 6.32 Par FM Module

6.11.1 Overview

The Par FM Module is an oscillator with a parabolic waveform, as shown on the Structure icon of the Module. The waveform consists of two parabolas that meet at zero level. It features logarithmic pitch control at the "P" (pitch) input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis.

Application

The parabolic waveform sounds like a sine wave with some added odd numbered overtones at very low level. In many cases the parabolic waveform can be used as replacement for a sine generator with less computational load. Since the Par FM Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.

6.11.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated parabolic waveform.

6.11.3 Example: Rehearsing the Oscillator

This example shows how to get a parabolic waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by a sine wave. The pitch information for the Par FM Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

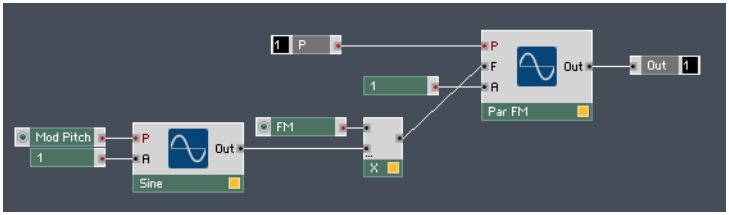


Fig. 6.33 A simple implementation of the Par FM Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

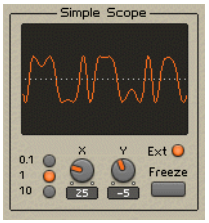


Fig. 6.34 An example of a frequency modulated parabol oscillator.

6.12 Par Sync Oscillator

Fig. 6.35 Par Sync Module

6.12.1 Overview

The Par Sync Module is an oscillator with a parabolic waveform, as shown on the Structure icon of the Module. The waveform consists of two parabolas that meet at zero level. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. In addition, this Module supports phase synchronization (also known as hard sync): a positive zero crossing of the signal at the "Snc" input port causes the phase of the parabolic oscillator to be reset to the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive. The signal at the "Snc" input port is usually another oscillator. When the synchronizing oscillator has the same frequency as the synchronized oscillator, there is no audible effect. The frequency of the parabolic sync oscillator can be linearly modulated by an audio signal at the "F" input port, making FM synthesis possible.

Application

The parabolic waveform sounds like a sine wave with some added odd numbered overtones at very low level. In many cases the parabolic waveform can be used as replacement for a sine generator with less computational load. Since the Par Sync Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis. Also, the oscillator can be hard synced to another oscillator to create the classic oscillator sync sound.

6.12.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 ... 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A ... A], depending on the phase synchronization..
- **(Snc)** "Snc" (sync) is the Audio input port for controlling synchronization of the waveform. A positive crossing of the signal at this input port restarts the oscillator phase from the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive.
- **(Ph)** "Ph" (phase) is the input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. The range for the "Ph" input port is [-1 ... 1] where "-1" corresponds to a phase of -180 degrees and "1" corresponds to a phase 180 degrees.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated and/or hard synchronized triangle waveform.

6.12.3 Example: Rehearsing the Oscillator

This example shows how to get a parabolic waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation and hard synchronization by two separate sine waves. The pitch information for the Par Sync Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module. The Par Sync Module is hard synchronized to a second sine wave, the pitch of which is determined

by the "Snc Pitch" Knob Module. The phase to which the parabolic wave is reset upon a positive zero-crossing at the "Snc" input port is determined by the "SncPhase" Knob Module at the "Ph" input port.

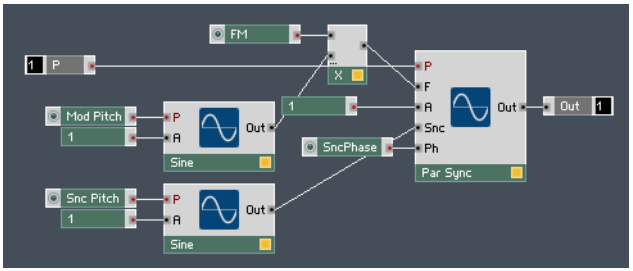


Fig. 6.36 A simple implementation of the Par Sync Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.37 An example of the waveform of a hard synchronized parabol oscillator.

6.13 Par PWM Oscillator

Fig. 6.38 Par PWM Module

6.13.1 Overview

This variable waveform is also particularly effective when used as an LFO. The Par PWM Module is an oscillator with a variable parabolic waveform. The parabolic waveform consists of two parabolas that meet at zero level. You can change the ratio between the length of the upper parabolic signal and the lower parabolic signal with an Event signal to the "W" input port. The Par PWM Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

The Par PWM Module delivers a signal with variable richness in harmonics which is a good starting point for subtractive synthesis. Morphing the waveform from a "W" value of "-1" to a value of "0" decreases the amount of harmonics in the signal; morphing from a "W" value of "0" to "+1" increases the amount of harmonics in the signal.

6.13.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the pulse width setting.
- **(W)** "W" (width) is the Event input port for controlling the waveform of the parabolic oscillator. At W = 0 the parabolic waveform consists of two parabolas that meet at zero level. By changing "W" within the range [-1 … 1] you can change the ratio between the

length of the upper parabolic signal and the lower parabolic signal. For W = -1 you get a waveform solely consisting of downward parabolas and for W = +1 you get a waveform solely consisting of upward parabolas.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with a variable parabolic waveform.

6.13.3 Example: Rehearsing the Oscillator

This example shows how to create a variable parabolic waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Par PWM Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width value is determined with a Knob Module labeled "P-Width" at the "W" input port.

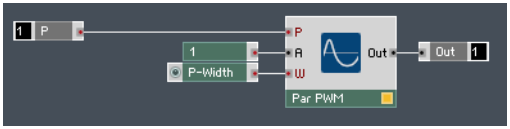


Fig. 6.39 A simple implementation of the Par PWM Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

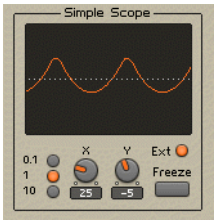


Fig. 6.40 A typical waveform of a Par PWM Module.

6.14 Sine Oscillator

Fig. 6.41 Sine Module

6.14.1 Overview

The Sine Module is an oscillator with a sinus waveform, as shown on the Structure icon of the Module. A sine signal has one harmonic at the frequency of the oscillator. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

Typical applications for the Sine Module include:

- Having the Sine Module as a modulating oscillator in FM synthesis or ring modulation
- Using the Sine Module as a sub-oscillator to introduce a deep fundamental frequency to a sound coming from the oscillators section in a synthesizer.

If the sine tone does not need to be completely pure, the parabolic oscillator makes a good replacement with less computational load.

6.14.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" Audio signal output for the sine waveform.

6.14.3 Example: Rehearsing the Oscillator

This example shows how to get a sine waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Sine Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

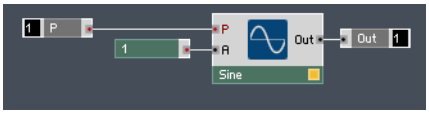


Fig. 6.42 A simple implementation of the Sine Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

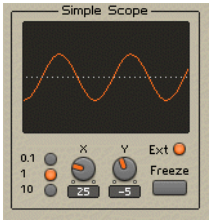


Fig. 6.43 The sine waveform.

6.15 Sine FM Oscillator

Fig. 6.44 Sine FM Module

6.15.1 Overview

The Sine FM Module is an oscillator with a sinus waveform, as shown on the Structure icon of the Module. A sine signal has one harmonic at the frequency of the oscillator. It features logarithmic pitch control at the "P" (pitch) input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis.

Application

Typical applications for the Sine FM module include:

- Using it as a modulating oscillator in FM synthesis or ring modulation.
- Using it as a sub-oscillator to introduce a deep fundamental frequency to a sound coming from the oscillators section in a synthesizer.
- Since the Sine FM Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.

If the sine tone does not need to be completely pure, the parabolic oscillator makes a good replacement with less computational load.

6.15.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated sine waveform.

6.15.3 Example: Rehearsing the Oscillator

This example shows how to get a sine waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by another sine wave. The pitch information for the Sine FM Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

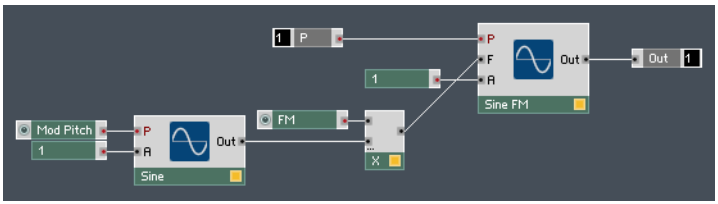


Fig. 6.45 A simple implementation of the Sine FM Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.46 An example of a frequency modulated sine waveform.

6.16 Sine Sync Oscillator

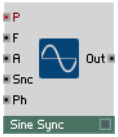


Fig. 6.47 Sine Sync Module

6.16.1 Overview

The Sine Sync Module is an oscillator with a sinus waveform, as shown on the Structure icon of the Module. A sine signal has one harmonic at the frequency of the oscillator. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. In addition, this Module supports phase synchronization (also known as hard sync): a positive zero crossing of the signal at the "Snc" input port causes the phase of the sine oscillator to be reset to the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive. The signal at the "Snc" input port is usually another oscillator. When the synchronizing oscillator has the same frequency as the synchronized oscillator, there is no audible effect. The frequency of the sine sync oscillator can be linearly modulated by an audio signal at the "F" input port, making FM synthesis possible.

Application

Typical uses for the Sine Sync Module include:

- The sine oscillator delivers a signal with a single harmonic at the frequency of the oscillator. It can be used as a modulating oscillator in FM synthesis or ring modulation or as a sub-oscillator to introduce a deep fundamental frequency to a sound coming from the oscillators section in a synthesizer.
- Since the Sine Sync Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.
- The oscillator can be hard synced to another oscillator to create the classic oscillator sync sound.

If the sine tone does not need to be completely pure, the parabolic oscillator makes a good replacement with less computational load.

6.16.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the phase synchronization..
- **(Snc)** "Snc" (sync) is the Audio input port for controlling synchronization of the waveform. A positive crossing of the signal at this input port restarts the oscillator phase from the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive.
- **(Ph)** "Ph" (phase) is the input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. The range for the "Ph" input port is [-1 … 1] where "-1" corresponds to a phase of -180 degrees and "1" corresponds to a phase 180 degrees.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated and/or hard synchronized sine waveform.

6.16.3 Example: Rehearsing the Oscillator

This example shows how to get a sine waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation and hard synchronization by two separate sine waves. The pitch information for the Sine Sync Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module. The Sine Sync Module is hard synchronized to a second sine wave, the

pitch of which is determined by the "Snc Pitch" Knob Module. The phase to which the triangle wave is reset upon a positive zero-crossing at the "Snc" input port is determined by the "SncPhase" Knob Module at the "Ph" input port.

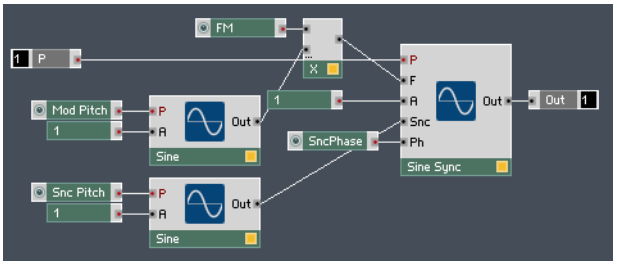


Fig. 6.48 A simple implementation of the Sine Sync Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

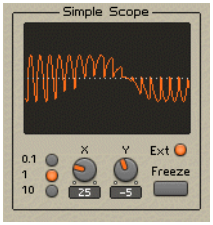


Fig. 6.49 An example of a hard synchronized sine waveform.

6.17 Sine 4x Oscillator

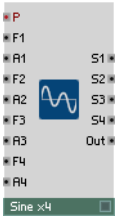


Fig. 6.50 Sine 4x Module

6.17.1 Overview

The Sine 4x Module comprises four sine oscillators with sine waveforms. For each of these oscillators the amplitude and relative frequency to the fundamental pitch can be specified at the dedicated input ports. The fundamental pitch is controlled with a signal at the "P" (pitch) input port and the pitches for the four oscillators are controlled at the "F1", "F2", "F3", and "F4" input ports. All pitch values are in the logarithmic pitch scale. The amplitudes for the oscillators are controlled in the linear scale at the "A1", "A2", "A3", and "A4" input ports. You can either use the sum of the sinus signals at the "Out" output port or tap each individual sine signal at the "S1", "S2", "S3", and "S4" output ports.

Application

Each sine oscillator delivers a signal with a single harmonic at the frequency of that oscillator. This way you can use a signal with 4 harmonics that you specify at the input ports or 4 individual harmonics. These can be used as modulating signals in FM synthesis or ring modulation. The Multi-Sine Module can be substituted by the Additive Bank Module. The Multi-Sine Module is only available for backward compatibility.

6.17.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 ... 127] where "69" corresponds to "Concert A" or 440 Hz.

- **(F1)** "F1" (frequency 1) is the Audio input port for controlling the frequency of the first oscillator. The value at the input port is given as a harmonic number relative to the fundamental pitch at the "P" (pitch) input port. For example, a value of "1" results in a frequency for the first oscillator that is the frequency of the fundamental. The typical range for this input port is [0 … 20].
- **(A1)** "A1" (amplitude 1) is the Audio input port for controlling the amplitude of the first oscillator. In other words, the signal at the "S1" output port has the range [-A1 … A1]. Since this is an Audio input port, it can be used for ring modulation.
- **(F2)** "F2" (frequency 2) is the Audio input port for controlling the frequency of the second oscillator. The value at the input port is given as a harmonic number relative to the fundamental pitch at the "P" (pitch) input port. For example, a value of "3" results in a frequency for the second oscillator that is triple the frequency of the fundamental. The typical range for this input port is [0 … 20].
- **(A2)** "A2" (amplitude 2) is the Audio input port for controlling the amplitude of the second oscillator. In other words, the signal at the "S2" output port has the range [-A2 … A2]. Since this is an Audio input port, it can be used for ring modulation.
- **(F3)** "F3" (frequency 3) is the Audio input port for controlling the frequency of the third oscillator. The value at the input port is given as a harmonic number relative to the fundamental pitch at the "P" (pitch) input port. For example, a value of "1" results in a frequency for the third oscillator that is the frequency of the fundamental. The typical range for this input port is [0 … 20].
- **(A3)** "A3" (amplitude 3) is the Audio input port for controlling the amplitude of the third oscillator. In other words, the signal at the "S3" output port has the range [-A3 … A3]. Since this is an Audio input port, it can be used for ring modulation.
- **(F4)** "F4" (frequency 4) is the Audio input port for controlling the frequency of the fourth oscillator. The value at the input port is given as a harmonic number relative to the fundamental pitch at the "P" (pitch) input port. For example, a value of "2" results in a frequency for the fourth oscillator that is double the frequency of the fundamental pitch. The typical range for this input port is [0 … 20].
- **(A4)** "A4" (amplitude 4) is the Audio input port for controlling the amplitude of the fourth oscillator. In other words, the signal at the "S4" output port has the range [-A4 … A4]. Since this is an Audio input port, it can be used for ring modulation.

Output Ports

- **(S1)** "S1" (sine 1) is the output port for the first sine oscillator.

- **(S2)** "S2" (sine 2) is the output port for the second sine oscillator.
- **(S3)** "S3" (sine 3) is the output port for the third sine oscillator.
- **(S4)** "S4" (sine 4) is the output port for the fourth sine oscillator.
- **(Out)** "Out" is the output port for the sum of the four sine oscillators.

6.17.3 Example: Rehearsing the Oscillator

This example shows how to get five signals with four of them carrying a sine wave and the last one carrying the sum of the previous four signals. The amplitudes and frequencies of the sine waves can be controlled using Knob Modules and the fundamental pitch is set with the MIDI controller or the keyboard. The pitch information for the Multi-Sine Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. All of the signals from the output ports of the Multi-Sine Module are sent to a Selector Module where the signal to be forwarded to the Out Port is chosen on the instrument panel with a List Module, the output of which is connected to the "Pos" input port of the Selector Module.

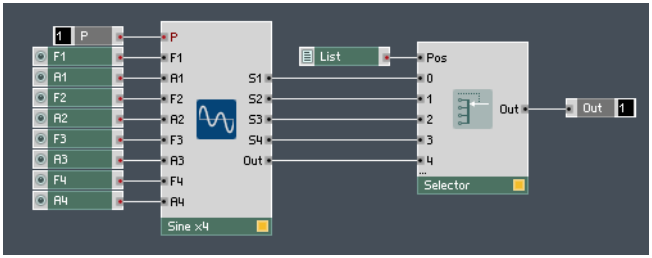


Fig. 6.51 A simple implementation of the Sine 4x Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.52 An example of a waveform composed of 4 additive sine waves.

6.18 Sine Bank

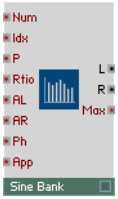


Fig. 6.53 Sine Bank Module

6.18.1 Overview

The Sine Bank Module is a bank of parallel sine oscillators, called partials. This Module enables you to perform additive synthesis with REAKTOR. For each partial, its frequency ratio to the fundamental, its amplitude, and its phase can be specified at the “Rtio”, “AL” and “AR”, and “Ph” input ports, respectively. Setting these parameters for each partial is done serially and with the use of the “Idx” input port to specify to which partial the incoming parameters should be sent. The fundamental pitch of the Sine Bank is specified at the “P” input; it corresponds to the frequency in respect to which the frequencies of all the partials are calculated. Also, the number of partials that the Sine Bank Module generates has to be specified, either in the Function page of the Module's Properties or at the “Num” input port. After sending all the parameters to the desired partials, an “apply” Event has to be sent to the “App” input port to effectively apply these changes to the partials. The resulting stereo Audio signal is sent to the “L” and “R” output ports for the left and right stereo channels, respectively. Also, in the Function page of the Module's Properties, one can specify the smoothing time and smoothing quality. The smoothing quality setting has an effect on the Module's CPU usage.

The default parameters for the partials are such that the “L” output sends a pulse wave with the pitch of the fundamental (set at the “P” input port) and the “R” output sends a saw wave with the pitch one octave above the fundamental. By adding both signals together you get a saw wave at the pitch of the fundamental.

Application

The Sine Bank Module is meant for additive synthesis. In theory, it is possible to synthesize anything using additive synthesis. In practice, this is often not possible because of the sheer amount of data that would have to be sent to the partials of the Sine Bank Module. Nevertheless, with the right parameters, the synthesis of a very broad range of sounds becomes available. Often a large number of partials are used; the maximum number of partials is 10'000. Usually much less partials than this maximum number are used, mainly because of CPU limitations. In a case where there are a lot of partials it is most effective to make use of the Iterator Module (found in the *Built-In Module > Event Processing* submenu) to supply all partials with the necessary parameters between the processing of subsequent audio samples.

Please refer to section 12.3 in the Application Reference for a short tutorial on creating a simple modal synthesizer.

6.18.2 Ports

Input Ports

- **(Num)** “Num” (number of partials) is the Event input port for setting the number of active partials in the Sine Bank Module. This value should be a positive integer in the range [0 … Max] where “Max” corresponds to the number set in the Max. Number edit field in the Function page of the Module's Properties. If this input port is left disconnected, the Sine Bank Module automatically sets the number of partials to the value to “Max”. A higher number of active partials requires more CPU resources. Often the CPU load is increased for multiples of 4 partials.
- **(Idx)** “Idx” (index) is the Event input port for specifying the partial number to which the subsequent parameters arriving at the “Rtio”, “AL”, “AR”, and “Ph” input ports should be routed. This value should be a positive integer in the range [1 … Num] where “Num” denotes the number of active partials (see “Num” input port).

- **(P)** “P” (pitch) is the Event input port for controlling the fundamental pitch of the Sine Bank Module. The frequency corresponding to the fundamental pitch is the frequency in respect to which the frequencies of all the partials are calculated. The values at this input port are set in the logarithmic pitch scale and are usually in the range [0 … 127].
- **(Rtio)** “Rtio” (frequency ratio) is the Event input port for setting the ratio of the partial's frequency to the fundamental frequency determined at the “P” input port. If the “Rtio” value of a partial is “1”, it means that the ratio between the partial's frequency and the fundamental frequency is “1”, that is, the partial has the fundamental frequency. If the “Rtio” value of a partial is “2”, then the partial has double the frequency of the fundamental. In general, the frequency of the partial (“f”) can be calculated from the fundamental frequency (“F”) and the frequency ratio (“Rtio”) with the following formula: f = F * Rtio. As can be seen from the formula, negative “Rtio” values cause negative frequencies for the partials. Negative frequencies cause the phase of the oscillator to run backwards.
- **(AL)** “AL” (amplitude, left channel) is the Event input port for specifying the partial amplitude in the left stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “L” output port. The typical range is [0 … 1], but negative values are also allowed; they cause the oscillator waveform of the corresponding partial to be inverted.
- **(AR)** “AR” (amplitude, right channel) is the Event input port for specifying the partial amplitude in the right stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “R” output port. The typical range is [0 … 1], but negative values are also allowed; they cause the oscillator waveform of the corresponding partial to be inverted.
- **(Ph)** “Ph” (phase) is the Event input port for specifying the phase (position in the waveform) to which the oscillator is reset when a positive “apply” Event arrives at the “App” input port. The range for the “Ph” input port is [-1 … 1] where “-1” corresponds to a phase of -180 degrees and “1” corresponds to a phase 180 degrees.
- **(App)** “App” (apply) is the Event input port for applying the latest parameter values that have arrived at the various input ports to the actual oscillators inside the Sine Bank Module. The parameter values sent to the Sine Bank Module are stored in a temporary buffer and only applied to the partials once an Event arrives at the “App” input

port. If the value of the Event is greater than zero, the apply operation also applies the latest values that have been sent to the “Ph” input port and forces the smoothers to jump to the values of the corresponding partial amplitude parameters. Thus, a positive valued Event at the “App” input port works like a “reset”. If the “apply” Event is less or equal to zero, all new parameters except the phase are applied.

Output Ports

- **(L)** “L” (left stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AL” parameter.
- **(R)** “R” (right stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AR” parameter.
- **(Max)** “Max” (maximum number of partials) is the Event output port for the value set in the Max. Number edit field of the Module's Function page.

6.18.3 Properties: Function Page

Setting the Maximum Number of Partials

The maximum number of partials that the Sine Bank Module generates has to be specified in the Max. Number edit field in the Module's Function page, shown in the figure below. The actual number of active partials is specified at the “Num” input port. “Num” values larger than the value set in the Max. Number edit field are clipped. If the “Num” input port is left disconnected, the Sine Bank Module automatically sets the number of partials to the value specified by the Max. Number edit field. The default value for this edit field is “128”. Also, the value in the Max. Number edit field is sent to the “Max” output port. A higher number of active partials requires more CPU resources.

Due to internal optimization, for many computers the CPU load is increased when the number of partials reaches a multiple of 4 partials.

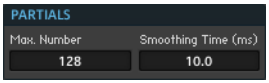


Fig. 6.54 The maximum number of partials and the smoothing time are set in the Partials area of the Sine Bank Module's Function page.

Setting the Smoothing Time

The smoothing time defines the time-interval over which the partial amplitudes are faded between different values. This parameter can be set in the Smoothing Time edit field. The units for this edit field are milliseconds and the default value is 10 ms.

Setting the Smoothing Quality

You can specify the quality of the smoothing curves with the Smoothing Quality drop-down menu (see screenshot below). Higher smoothing quality causes the Module to respond better to fast modulations of the partial amplitudes. However, increasing the smoothing quality also increases the CPU load.

Fig. 6.55 The Smoothing Quality drop-down menu lets you increase the quality of the Module's sound in response to fast modulations of the partial amplitudes.

6.19 Pulse Oscillator

Fig. 6.56 Pulse Module.

6.19.1 Overview

The Pulse Module is an oscillator with a pulse waveform with variable pulse width. It features logarithmic pitch control at the "P" (pitch) input port, linear amplitude modulation at the "A" input port, and pulse width modulation (PWM) at the "W" input port. With a width value of "0", the pulse oscillator delivers a signal with a broad harmonic spectrum, consisting only of odd harmonics. This gives it a sound that has a "hollow" timbre, characteristic of wind instruments. Changing the width value increases the amount of even harmonics in the signal.

Application

Applications for the Pulse Module include:

- You can connect an audio signal such as the output of another oscillator to the "W" input of the Pulse Module. This creates the classic PWM-sound.

- The output signal can be used as a modulating oscillator in FM synthesis or ring modulation.
- The pulse wave is also a good starting point for subtractive synthesis because of its broad spectrum.

A nonzero value at the "W" input port results in an output signal with a maximum amplitude greater than the value at the "A" input port, so make sure to choose your amplitude value accordingly or to downscale the output signal before routing it to the Audio Out Module.

6.19.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the pulse width setting.
- **(W)** "W" (width) is the Audio input port for controlling the pulse width (PWM) of the waveform. The range of the values at this input port is [-0.98 … 0.98]. The ratio between the low and high states of the pulse waveform in relation to the value at the "W" input port is given as Low : High = (1 + W) / (1 – W). So, a "W" value of "0" yields a symmetric pulse waveform (a Low : High ratio of 50 : 50) at the output port. A "W" value of "-0.33" yields a Low : High ratio of 33 : 66, "-0.5" yields a Low : High ratio of 25 : 75, and "-0.98" already yields a waveform that can be considered an impulse train. Changing the "W" value from "0" to another value effectively introduces even harmonics into the output signal.

Output Ports

- **(Out)** "Out" is the output port for the audio signal carrying the pulse waveform.

6.19.3 Example: Rehearsing the Oscillator

This example shows how to get a pulse waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information to the Pulse Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width is determined by a value sent to the "W" input port of the Pulse Module from the Knob Module labeled "P-Width".

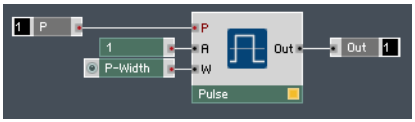


Fig. 6.57 A simple implementation of the Pulse Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

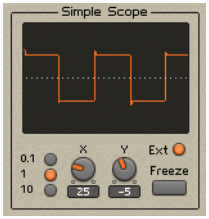


Fig. 6.58 A pulse waveform.

6.20 Pulse FM Oscillator

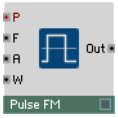


Fig. 6.59 Pulse FM Module

6.20.1 Overview

The Pulse FM Module is an oscillator with a pulse waveform with variable pulse width. It features logarithmic pitch control at the "P" (pitch) input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis. The Pulse FM Module also supports pulse width modulation (PWM) at the "W" input port. With a width value of "0", the pure (non frequency modulated) pulse oscillator delivers a signal with a broad harmonic spectrum, consisting only of odd harmonics. This gives it a sound that has a "hollow" timbre, characteristic of wind instruments. Changing the width value increases the amount of even harmonics in the signal.

Application

Applications for the Pulse FM Module include:

- You can connect an audio signal such as the output of another oscillator to the "W" input of the Pulse FM Module. This creates the classic PWM-sound.
- The output signal can be used as a modulating oscillator in FM synthesis or ring modulation.
- The pulse wave is also a good starting point for subtractive synthesis because of its broad spectrum.
- Since there is an Audio input port for linear frequency control, the Pulse FM Module can be used as a carrier oscillator in FM synthesis.

A nonzero value at the "W" input port results in an output signal with a maximum amplitude greater than the value at the "A" input port, so make sure to choose your amplitude value accordingly or to downscale the output signal before routing it to the Audio Out Module.

6.20.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the pulse width setting.
- **(W)** "W" (width) is the Audio input port for controlling the pulse width (PWM) of the waveform. The range of the values at this input port is [-0.98 … 0.98]. The ratio between the low and high states of the pulse waveform in relation to the value at the "W" input port is given as Low : High = (1 + W) / (1 – W). So, a "W" value of "0" yields a symmetric pulse waveform (a Low : High ratio of 50 : 50) at the output port. A "W" value of "-0.33" yields a Low : High ratio of 33 : 66, "-0.5" yields a Low : High ratio of 25 : 75, and "-0.98" already yields a waveform that can be considered an impulse train. Changing the "W" value from "0" to another value effectively introduces even harmonics into the pure (non frequency modulated) pulse wave.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated pulse waveform.

6.20.3 Example: Rehearsing the Oscillator

This example shows how to get a pulse waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by another sine wave. The pitch information for the Pulse FM Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width is determined by a value sent to the "W" input port of the Pulse FM Module from the Knob Module labeled "P-Width". The

pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

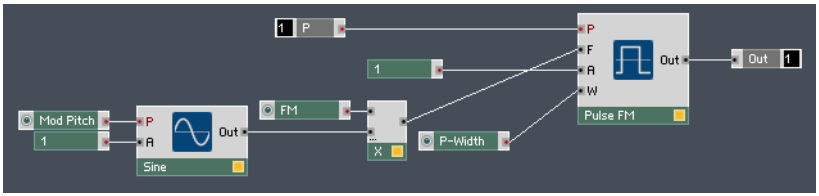


Fig. 6.60 A simple implementation of the Pulse FM Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

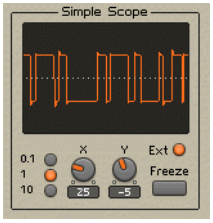


Fig. 6.61 An example of a frequency modulated pulse waveform.

6.21 Pulse Sync Oscillator

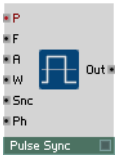


Fig. 6.62 Pulse Sync Module

6.21.1 Overview

The Pulse Sync Module is an oscillator with a pulse waveform with variable pulse width. It features logarithmic pitch control at the "P" (pitch) input port, linear amplitude modulation at the "A" input port, and pulse width modulation (PWM) at the "W" input port. With a width value of "0", the pure (non frequency modulated) pulse oscillator delivers a signal with a broad harmonic spectrum, consisting only of odd harmonics. This gives it a sound that has "hollow" timbre, characteristic of wind instruments. Changing the width value increases the amount of even harmonics in the signal. In addition, this Module supports phase synchronization (also known as hard sync): a positive zero crossing of the signal at the "Snc" input port causes the phase of the pulse sync oscillator to be reset to the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive. The signal at the "Snc" input port is usually another oscillator. When the synchronizing oscillator has the same frequency as the synchronized oscillator, there is no audible effect. The frequency of the pulse sync oscillator can be linearly modulated by an audio signal at the "F" input port, making FM synthesis possible.

Application

Applications for the Pulse Module include:

- You can connect an audio signal such as the output of another oscillator to the "W" input of the Pulse Sync Module. This creates the classic PWM-sound.
- The output signal can be used as a modulating oscillator in FM synthesis or ring modulation.
- The pulse wave is also a good starting point for subtractive synthesis because of its broad spectrum.
- Since there is an Audio input port for linear frequency control, the Pulse Sync Module can be used as a carrier oscillator in FM synthesis.
- The pulse oscillator can be hard synced to another oscillator to create the classic oscillator sync sound.

A nonzero value at the "W" input port results in an output signal with a maximum amplitude greater than the value at the "A" input port, so make sure to choose your amplitude value accordingly or to downscale the output signal before routing it to the Audio Out Module.

6.21.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has a range within [-A … A], depending on the pulse width and phase synchronization settings.
- **(W)** "W" (width) is the Audio input port for controlling the pulse width (PWM) of the waveform. The range of the values at this input port is [-0.98 … 0.98]. The ratio between the low and high states of the pulse waveform in relation to the value at the "W" input port is given as Low : High = (1 + W) / (1 – W). So, a "W" value of "0" yields a symmetric pulse waveform (a Low : High ratio of 50 : 50) at the output port. A "W" value of "-0.33" yields a Low : High ratio of 33 : 66, "-0.5" yields a Low : High ratio of 25 : 75, and "-0.98" already yields a waveform that can be considered an impulse train. Changing the "W" value from "0" to another value effectively introduces even harmonics into the pure (non frequency modulated or hard synced) pulse wave.
- **(Snc)** "Snc" (sync) is the Audio input port for controlling the synchronization of the waveform. A positive crossing of the signal at this input port restarts the oscillator phase from the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive.
- **(Ph)** "Ph" (phase) is the input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. The range for the "Ph" input port is [-1 … 1] where "-1" corresponds to a phase of -180 degrees and "1" corresponds to a phase 180 degrees.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated and/or hard synchronized pulse waveform.

6.21.3 Example: Rehearsing the Oscillator

This example shows how to get a pulse waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation and hard synchronization by two separate sine waves. The pitch information for the Pulse Sync Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width is determined by a value sent to the "W" input port of the Pulse Module from the Knob Module labeled "P-Width". The pitch of the frequency modulator (a Sine Module) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module. The Pulse Sync Module is hard synchronized to a second sine wave, the pitch of which is determined by the "Snc Pitch" Knob Module. The phase to which the triangle wave is reset upon a positive zero-crossing at the "Snc" input port is determined by the "SncPhase" Knob Module at the "Ph" input port.

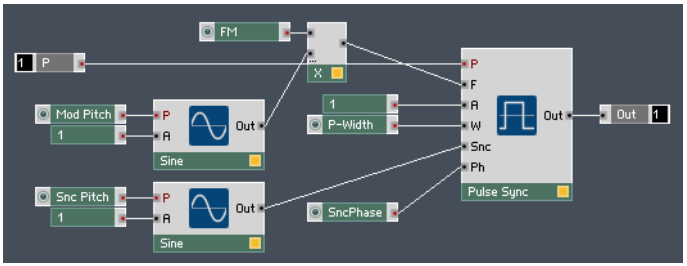


Fig. 6.63 A simple implementation of the Pulse Sync Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

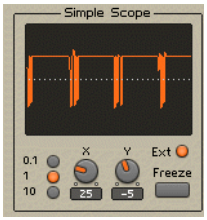


Fig. 6.64 An example of a hard synchronized pulse waveform.

6.22 Pulse 1-Ramp Oscillator

Fig. 6.65 Pulse 1-Ramp Module

6.22.1 Overview

The Pulse 1-Ramp Module is an oscillator with a variable trapezoidal waveform. Among other waveforms it is possible to a sawtooth and a pulse wave. You can morph between the different waveforms by changing the slope of the ramp and the pulse width with a signal at the "Slp" and "W" input ports, respectively. The falling edge of the pulse waveform is always vertical. The Pulse 1-Ramp Module features logarithmic pitch control at the "P" (pitch) input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis.

Application

The Pulse 1-Ramp Module delivers a signal with variable harmonic distribution, which is a good starting point for subtractive synthesis. Morphing the waveform from a sawtooth to pulse waveform eliminates even harmonics in the signal spectrum, giving the sound a hollow timbre characteristic to wind instruments.

6.22.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module is within the range [-A … A], depending on the slope and pulse width settings.
- **(W)** "W" (width) is the Audio input port for controlling the pulse width (PWM) of the waveform. The range of the values at this input port is [-0.98 … 0.98]. The ratio between the low and high states of the waveform in relation to the value at the "W" input port is given as Low : High = (1 + W) / (1 – W). So, a "W" value of "0" yields a Low : High ratio of 50 : 50 at the output port. A "W" value of "-0.33" yields a Low : High ratio of 33 : 66, "-0.5" yields a Low : High ratio of 25 : 75, and "-0.98" already yields a waveform that can be considered an impulse train. The final waveform strongly depends on the slope setting. If, for example, the slope is not steep enough, then the signal will not reach the maximum amplitude during the "high" state and the waveform will not be symmetrical in respect to the zero level.
- **(Slp)** "Slp" (slope) is the Audio input port for controlling the slope of the rising edge of the waveform. When the value sent to the "Slp" input is "0" (or when the input is not connected) the waveform does not rise and the output is always zero, that is, there is no audible signal at the output port. A "Slp" value of "1" (with W = 0) yields a sawtooth signal, a "Slp" value of "20" yields a waveform very close to a pulse wave. The typical range for the values at this input port is [1 … 20].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated sawtooth/pulse waveform.

6.22.3 Example: Rehearsing the Oscillator

This example shows how to get a trapezoidal waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by another sine wave. The pitch information for the Pulse 1-Ramp Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width and slope are determined by values sent to the "W" and "Slp" input ports of the Pulse 1-Ramp Module from Knob Modules labeled "P-Width" and "Slope", respectively. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

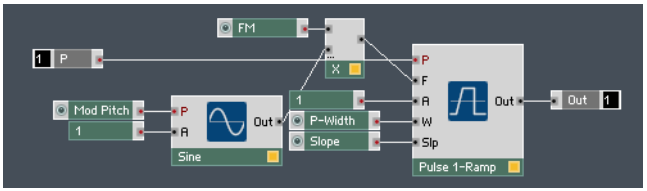


Fig. 6.66 A simple implementation of the Pulse 1-Ramp Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

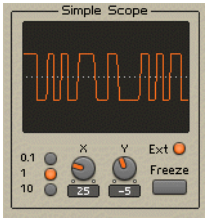


Fig. 6.67 An example of the output waveform of a frequency modulated Pulse 1-Ramp Module.

6.23 Pulse 2-Ramp Oscillator

Fig. 6.68 Pulse 2-Ramp Module

6.23.1 Overview

The Pulse 2-Ramp Module is an oscillator with a variable trapezoidal waveform. Among other waveforms, it is possible to create sawtooth, triangle, and pulse waves. You can morph between the different waveforms by changing the slope of the rising edge of the pulse, falling edge of the pulse, and the pulse width with a signal at the "Up", "Dn", and "W" input ports, respectively. The Pulse 2-Ramp Module features logarithmic pitch control at the "P" (pitch) input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis.

Application

The Pulse 2-Ramp Module delivers a signal with variable harmonic distribution, which is a good starting point for subtractive synthesis. Morphing the waveform from a sawtooth to a triangle or pulse waveform eliminates even harmonics in the signal spectrum, giving the sound a hollow timbre characteristic to wind.

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.

- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module is within the range [-A ... A], depending on the slope and pulse width settings.
- **(W)** "W" (width) is the Audio input port for controlling the pulse width (PWM) of the waveform. The range of the values at this input port is [-0.98 ... 0.98]. The ratio between the low and high states of the waveform in relation to the value at the "W" input port is given as Low : High = (1 + W) / (1 – W). So, a "W" value of "0" yields a Low : High ratio of 50 : 50 at the output port. A "W" value of "-0.33" yields a Low : High ratio of 33 : 66, "-0.5" yields a Low : High ratio of 25 : 75, and "-0.98" already yields a waveform that can be considered an impulse train. The final waveform strongly depends on the slope setting. If, for example, the slope is not steep enough, then the signal will not reach the maximum amplitude during the "high" state and the waveform will not be symmetrical in respect to the zero level.
- **(Up)** "Up" (upward slope) is the Audio input port for controlling the slope of the rising edge of the waveform. When the value sent to the "Slp" input is "0" (or when the input is not connected) the waveform does not rise and the output is always at the minimum value, that is, there is no audible signal at the output port. An "Up" value of "1" (with W = 0 and Dn = 20) yields a sawtooth signal, and an "Up" value of "20" would yield a waveform very close to a pulse wave. The typical range for the values at this input port is [1 ... 20].
- **(Dn)** "Dn" (downward slope) is the Audio input port for controlling the slope of the falling edge of the waveform. When the value sent to the "Dn" input is "0" (or when the input is not connected) the waveform does not fall and the output is always at the maximum value, that is, there is no audible signal at the output port. A "Slp" value of "1" (with W = 0 and Up = 20) yields a sawtooth signal, and a "Dn" value of "20" would yield a waveform very close to a pulse wave. The typical range for the values at this input port is [1 ... 20].

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated sawtooth/pulse waveform.

6.23.2 Example: Rehearsing the Oscillator

This example shows how to get a trapezoidal waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by another sine wave. The pitch information for the Pulse 2-Ramp Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width and slope are determined by values sent to the "W", "Up", and "Dn" input ports of the Pulse 2-Ramp Module from Knob Modules labeled "P-Width", "Up", and "Dn", respectively. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

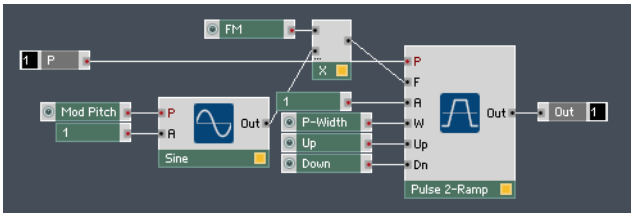


Fig. 6.69 A simple implementation of the Pulse 2-Ramp Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

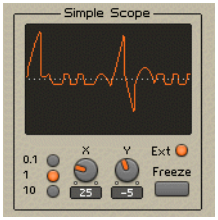


Fig. 6.70 An example of the output waveform of a frequency modulated Pulse 2-Ramp Module.

6.24 Bi-Pulse Oscillator

Fig. 6.71 Bi-Pulse Module

6.24.1 Overview

The Bi-Pulse Module is an oscillator with a bipolar pulse waveform with variable pulse width. It features logarithmic pitch control at the "P" (pitch) input port, linear amplitude modulation at the "A" input port, and pulse width modulation (PWM) at the "W" input port. With a width value of "0", the bipolar pulse oscillator delivers a signal with a broad harmonic spectrum, consisting only of odd harmonics. This gives it a sound that has a "hollow" timbre, characteristic of wind instruments. Changing the width value increases the amount of even harmonics in the signal.

Application

Applications of the Bi-Pulse Module include:

- You can connect an audio signal such as the output of another oscillator to the "W" input of the Bi-Pulse Module. This creates the classic PWM-sound.
- The output signal can be used as a modulating oscillator in FM synthesis or ring modulation.
- The bipolar pulse wave is also a good starting point for subtractive synthesis because of its broad spectrum.

An amplitude value of "1" results in an output signal with a maximum value around "8", so make sure to choose your amplitude value carefully before routing the signal to the Audio Out Module.

6.24.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(W)** "W" (width) is the Audio input port for controlling the pulse width (PWM) of the waveform. The range of the values at this input port is [-0.98 … 0.98]. The ratio between the low and high states of the pulse waveform in relation to the value at the "W" input port is given as Low : High = (1 + W) / (1 – W). So, a "W" value of "0" yields a symmetric pulse waveform (a Low : High ratio of 50 : 50) at the output port. A "W" value of "-0.33" yields a Low : High ratio of 33 : 66, "-0.5" yields a Low : High ratio of 25 : 75, and "-0.98" already yields a waveform that can be considered an impulse train. Changing the "W" value from "0" to another value effectively introduces even harmonics into the output signal.

Output Ports

- **(Out)** "Out" is the output port for the audio signal carrying the bipolar pulse waveform.

6.24.3 Example: Rehearsing the Oscillator

This example shows how to get a pulse waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information to the Bi-Pulse Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pulse width is determined by a value sent to the "W" input port of the Bi-Pulse Module from the Knob Module labeled "P-Width".

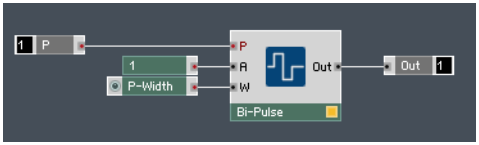


Fig. 6.72 A simple implementation of the Bi-Pulse Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

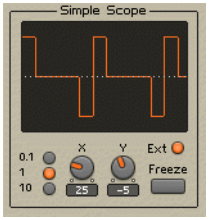


Fig. 6.73 A possible bi-pulse waveform.

6.25 Impulse Oscillator

Fig. 6.74 Impulse Module

6.25.1 Overview

The Impulse Module is an oscillator with an impulse train signal. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port.

Application

The impulse oscillator delivers a signal with a spectrum that has all harmonics with approximately equal amplitudes present. Because of its broad spectrum, the Impulse Module can be a starting point for subtractive synthesis.

An amplitude value of "1" causes an output signal with a maximum value around "8", so make sure to choose your amplitude value carefully before routing the signal to the Audio Out Module.

6.25.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 ... 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the amplitude of the impulse.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the impulse waveform.

6.25.3 Example: Rehearsing the Oscillator

This example shows how to get an impulse waveform with constant unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information for the Impulse Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

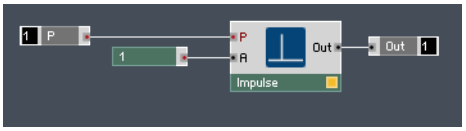


Fig. 6.75 A simple implementation of the Impulse Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemblej. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.76 The impulse waveform.

6.26 Impulse FM Oscillator

Fig. 6.77 Impulse FM Module

6.26.1 Overview

The Impulse FM Module is an oscillator with an impulse train signal. It features logarithmic pitch control at the "P" (pitch) input port, linear frequency modulation at the "F" input port, and linear amplitude modulation at the "A" input port. The "F" input port accepts audio signals and is meant for FM synthesis. It delivers a signal with a spectrum that has all harmonics with approximately equal amplitudes present.

Application

Typical applications for the Impulse FM Module include:

- Using the output signal as a starting point for subtractive synthesis.
- Modulating another oscillator in FM synthesis or ring modulation.
- Since the Impulse FM Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.

An amplitude value of "1" causes an output signal with a maximum value around "8", so make sure to choose your amplitude value carefully before routing the signal to the Audio Out Module.

6.26.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated impulse waveform.

6.26.3 Example: Rehearsing the Oscillator

This example shows how to get an impulse waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation by another sine wave. The pitch information for the Impulse FM Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module.

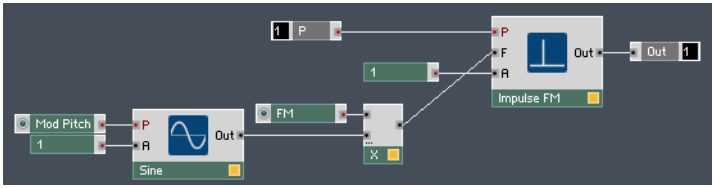


Fig. 6.78 A simple implementation of the Impulse FM Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope.

Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list

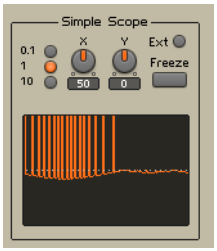


Fig. 6.79 An example of a frequency modulated impulse waveform.

6.27 Impulse Sync Oscillator

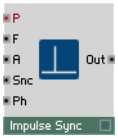


Fig. 6.80 Imulse Sync Module

6.27.1 Overview

The Impulse Sync Module is an oscillator with an impulse train signal. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. In addition, this Module supports phase synchronization (also known as hard sync): a positive zero crossing of the signal at the "Snc" input port causes the phase of the impulse sync oscillator to be reset to the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive. The signal at the "Snc" input port is usually another oscillator. When the synchronizing oscillator has the same frequency as the synchronized oscillator, there is no audible effect. The frequency of the impulse sync oscillator can be linearly modulated by an audio signal at the "F" input port, making FM synthesis possible. The pure (non-modulated) impulse wave has a spectrum that has all harmonics with approximately equal amplitudes present.

Application

Typical applications for the Impulse Sync Module include:

- Using the output signal as a starting point for subtractive synthesis.
- Modulating another oscillator in FM synthesis or ring modulation.
- The oscillator can be hard synced to another oscillator to create the classic oscillator sync sound.
- Since the Impulse FM Module has an Audio input port for linear frequency control, it can be used as a carrier oscillator in FM synthesis.

An amplitude value of "1" causes an output signal with a maximum value around "8", so make sure to choose your amplitude value carefully before routing the signal to the Audio Out Module.

6.27.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(F)** "F" (frequency) is the Audio input port for linear frequency modulation in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (pitch) input port. The sum is then the final oscillator frequency.
- **(A)** "A" (amplitude) is the Audio input port for controlling the amplitude of the impulse.
- **(Snc)** "Snc" (sync) is the Audio input port for controlling synchronization of the waveform. A positive crossing of the signal at this input port restarts the oscillator phase from the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive.
- **(Ph)** "Ph" (phase) is the input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. The range for the "Ph" input port is [-1 … 1] where "-1" corresponds to a phase of -180 degrees and "1" corresponds to a phase 180 degrees.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated and/or hard synchronized impulse waveform.

6.27.3 Example: Rehearsing the Oscillator

This example shows how to get an impulse waveform with constant unit amplitude, a pitch set with the MIDI controller or the keyboard, and frequency modulation and hard synchronization by two separate sine waves. The pitch information for the Impulse Sync Module is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. The pitch of the frequency modulator (a Sine Module, see ↑6.14, Sine Oscillator) is determined by the "Mod Pitch" Knob Module and the frequency modulation amount is determined by the "FM" Knob Module. The Impulse Sync Module is hard synchronized to a second sine wave, the pitch of which is determined by the "Snc Pitch" Knob Module. The phase to which the triangle wave is reset upon a positive zero-crossing at the "Snc" input port is determined by the "SncPhase" Knob Module at the "Ph" input port.

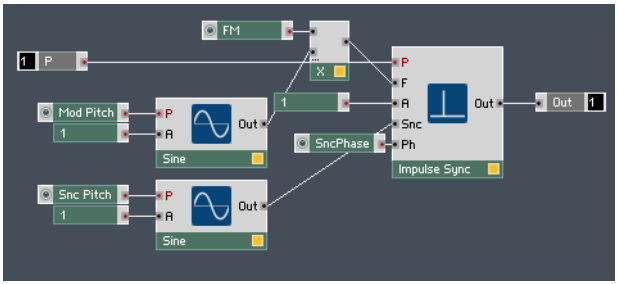


Fig. 6.81 A simple implementation of the Impulse Sync Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

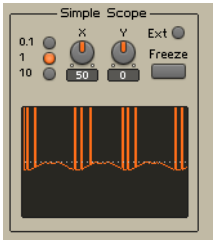


Fig. 6.82 An example of a hard synchronized impulse waveform.

6.28 4-Step Oscillator

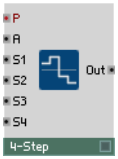


Fig. 6.83 4-Step Module

6.28.1 Overview

The 4-Step Module is an oscillator with a waveform consisting of four consequent steps. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all four steps can be set independent of the others.

Application

The 4-Step Module can be used to create any variety of stepped waveforms including the pulse and bi-pulse waveforms. By changing the strength of the transients between the pulses you can vary the presence of high harmonic components in the oscillator's output signal.

6.28.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(S1)** "S1" (step 1) is the Audio input for controlling the level of the first step.
- **(S2)** "S1" (step 2) is the Audio input for controlling the level of the second step.
- **(S3)** "S3" (step 3) is the Audio input for controlling the level of the third step.
- **(S4)** "S4" (step 4) is the Audio input for controlling the level of the fourth step.

Output Ports

- **(Out)** "Out" is the Audio signal output for the step waveform.

6.28.3 Example: Rehearsing the Oscillator

This example shows how to get a stepped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

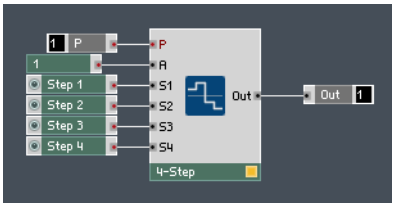


Fig. 6.84 A simple implementation of the 4-Step Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope.

Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

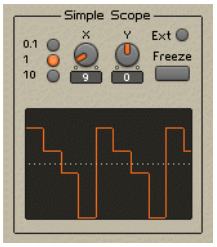


Fig. 6.85 A typical 4-step waveform.

6.29 5-Step Oscillator

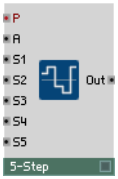


Fig. 6.86 5-Step Module

6.29.1 Overview

The 5-Step Module is an oscillator with a waveform consisting of five consequent steps. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all five steps can be set independent of the others.

Application

The 5-Step Module can be used to create any variety of stepped waveforms including the pulse and bi-pulse waveforms. By changing the strength of the transients between the pulses you can vary the presence of high harmonic components in the oscillator's output signal.

6.29.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(S1)** "S1" (step 1) is the Audio input for controlling the level of the first step.
- **(S2)** "S1" (step 2) is the Audio input for controlling the level of the second step.
- **(S3)** "S3" (step 3) is the Audio input for controlling the level of the third step.
- **(S4)** "S4" (step 4) is the Audio input for controlling the level of the fourth step.
- **(S5)** "S5" (step 5) is the Audio input for controlling the level of the fifth step.

Output Ports

- **(Out)** "Out" is the Audio signal output for the step waveform.

6.29.3 Example: Rehearsing the Oscillator

This example shows how to get a stepped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

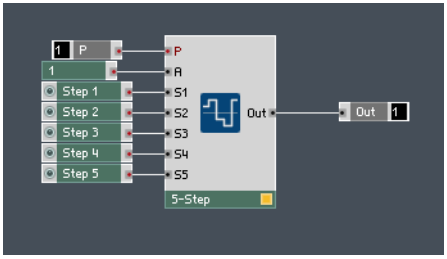


Fig. 6.87 A simple implementation of the 5-Step Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while fol-

lowing the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

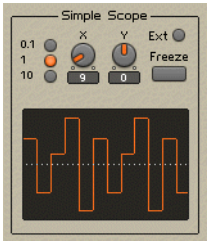


Fig. 6.88 A typical 5-step waveform.

6.30 6-Step Oscillator

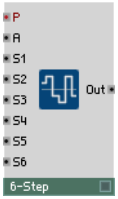


Fig. 6.89 6-Step Module

6.30.1 Overview

The 6-Step Module is an oscillator with a waveform consisting of six consequent steps. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all six steps can be set independent of the others.

Application

The 6-Step Module can be used to create any variety of stepped waveforms including the pulse and bi-pulse waveforms. By changing the strength of the transients between the pulses you can vary the presence of high harmonic components in the oscillator's output signal.

6.30.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(S1)** "S1" (step 1) is the Audio input for controlling the level of the first step.
- **(S2)** "S1" (step 2) is the Audio input for controlling the level of the second step.
- **(S3)** "S3" (step 3) is the Audio input for controlling the level of the third step.
- **(S4)** "S4" (step 4) is the Audio input for controlling the level of the fourth step.
- **(S5)** "S5" (step 5) is the Audio input for controlling the level of the fifth step.
- **(S6)** "S6" (step 6) is the Audio input for controlling the level of the sixth step.

Output Ports

- **(Out)** "Out" is the Audio signal output for the step waveform.

6.30.3 Example: Rehearsing the Oscillator

This example shows how to get a stepped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

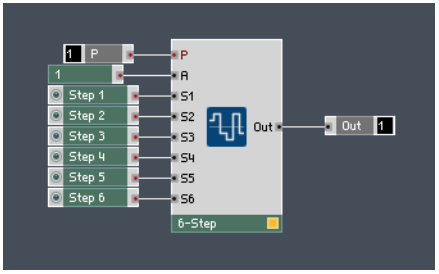


Fig. 6.90 A simple implementation of the 6-Step Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

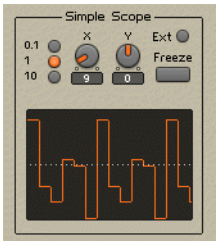


Fig. 6.91 A typical 6-step waveform.

6.31 8-Step Oscillator

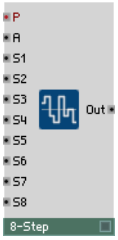


Fig. 6.92 8-Step Module

6.31.1 Overview

The 8-Step Module is an oscillator with a waveform consisting of eight consequent steps. It features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all eight steps can be set independent of the others.

Application

The 8-Step Module can be used to create any variety of stepped waveforms including the pulse and bi-pulse waveforms. By changing the strength of the transients between the pulses you can vary the presence of high harmonic components in the oscillator's output signal.

6.31.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(S1)** "S1" (step 1) is the Audio input for controlling the level of the first step.
- **(S2)** "S1" (step 2) is the Audio input for controlling the level of the second step.
- **(S3)** "S3" (step 3) is the Audio input for controlling the level of the third step.
- **(S4)** "S4" (step 4) is the Audio input for controlling the level of the fourth step.
- **(S5)** "S5" (step 5) is the Audio input for controlling the level of the fifth step.
- **(S6)** "S6" (step 6) is the Audio input for controlling the level of the sixth step.
- **(S7)** "S7" (step 7) is the Audio input for controlling the level of the seventh step.
- **(S8)** "S8" (step 8) is the Audio input for controlling the level of the eighth step.

Output Ports

- **(Out)** "Out" is the Audio signal output for the step waveform.

6.31.3 Example: Rehearsing the Oscillator

This example shows how to get a stepped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

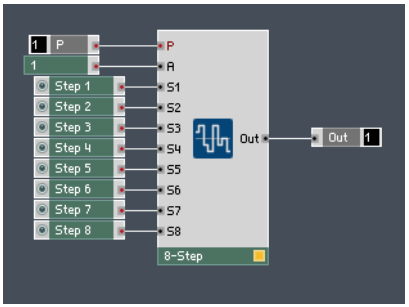


Fig. 6.93 A simple implementation of the 8-Step Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

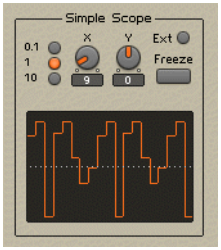


Fig. 6.94 A typical 8-step waveform.

6.32 4-Ramp Oscillator

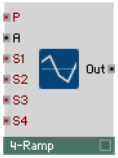


Fig. 6.95 4-Ramp Module

6.32.1 Overview

The 4-Ramp Module is an oscillator with a waveform consisting of four breakpoints which are connected by ramps. The level of the start and end of the ramps can be varied by adjusting the level of these breakpoints. The Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all four breakpoints can be set independent of the others.

Application

The 4-Ramp Module can be used to create any variety of ramped waveforms including the triangle waveform. The spectrum of this oscillator type is less bright than the one produced by the stepped oscillators. By changing the steepness of the ramps you can vary the presence of high harmonic components in the oscillator's output signal.

6.32.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 ... 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A ... A].
- **(S1)** "S1" (step 1) is the Event input for controlling the level of the first breakpoint.
- **(S2)** "S2" (step 2) is the Event input for controlling the level of the second breakpoint.
- **(S3)** "S3" (step 3) is the Event input for controlling the level of the third breakpoint.
- **(S4)** "S4" (step 4) is the Event input for controlling the level of the fourth breakpoint.

Output Ports

- **(Out)** "Out" is the Audio signal output for the ramp waveform.

6.32.3 Example: Rehearsing the Oscillator

This example shows how to get a ramped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

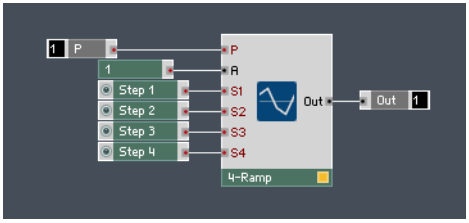


Fig. 6.96 A simple implementation of the 4-Ramp Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

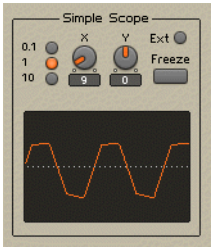


Fig. 6.97 A typical 4-ramp waveform.

6.33 5-Ramp Oscillator

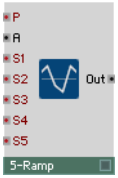


Fig. 6.98 5-Ramp Module

6.33.1 Overview

The 5-Ramp Module is an oscillator with a waveform consisting of five breakpoints which are connected by ramps. The level of the start and end of the ramps can be varied by adjusting the level of these breakpoints. The Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all five breakpoints can be set independent of the others.

Application

The 5-Ramp Module can be used to create any variety of ramped waveforms including the triangle waveform. The spectrum of this oscillator type is less bright than the one produced by the stepped oscillators. By changing the steepness of the ramps you can vary the presence of high harmonic components in the oscillator's output signal.

6.33.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 ... 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A ... A].
- **(S1)** "S1" (step 1) is the Event input for controlling the level of the first breakpoint.
- **(S2)** "S2" (step 2) is the Event input for controlling the level of the second breakpoint.
- **(S3)** "S3" (step 3) is the Event input for controlling the level of the third breakpoint.
- **(S4)** "S4" (step 4) is the Event input for controlling the level of the fourth breakpoint.

- **(S5)** "S5" (step 5) is the Event input for controlling the level of the fifth breakpoint.

Output Ports

- **(Out)** "Out" is the Audio signal output for the ramp waveform.

6.33.3 Example: Rehearsing the Oscillator

This example shows how to get a ramped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

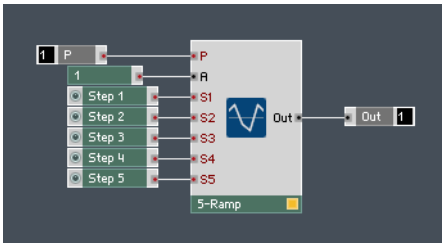


Fig. 6.99 A simple implementation of the 5-Ramp Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

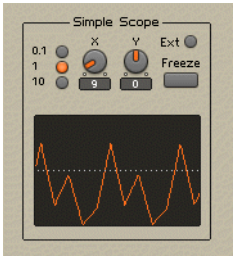


Fig. 6.100 A typical 5-ramp waveform.

6.34 6-Ramp Oscillator

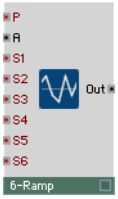


Fig. 6.101 6-Ramp Module

6.34.1 Overview

The 6-Ramp Module is an oscillator with a waveform consisting of six breakpoints which are connected by ramps. The level of the start and end of the ramps can be varied by adjusting the level of these breakpoints. The Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all six breakpoints can be set independent of the others.

Application

The 6-Ramp Module can be used to create any variety of ramped waveforms including the triangle waveform. The spectrum of this oscillator type is less bright than the one produced by the stepped oscillators. By changing the steepness of the ramps you can vary the presence of high harmonic components in the oscillator's output signal.

6.34.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(S1)** "S1" (step 1) is the Event input for controlling the level of the first breakpoint.
- **(S2)** "S2" (step 2) is the Event input for controlling the level of the second breakpoint.
- **(S3)** "S3" (step 3) is the Event input for controlling the level of the third breakpoint.

- **(S4)** "S4" (step 4) is the Event input for controlling the level of the fourth breakpoint.
- **(S5)** "S5" (step 5) is the Event input for controlling the level of the fifth breakpoint.
- **(S6)** "S6" (step 6) is the Event input for controlling the level of the sixth breakpoint.

Output Ports

- **(Out)** "Out" is the Audio signal output for the ramp waveform.

6.34.3 Example: Rehearsing the Oscillator

This example shows how to get a ramped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

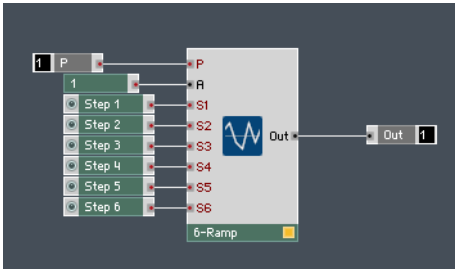


Fig. 6.102 A simple implementation of the 6-Ramp Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

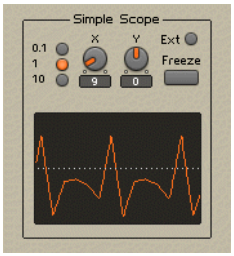


Fig. 6.103 A typical 6-ramp waveform.

6.35 8-Ramp Oscillator

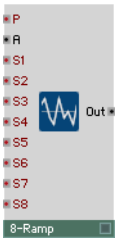


Fig. 6.104 8-Ramp Module

6.35.1 Overview

The 8-Ramp Module is an oscillator with a waveform consisting of eight breakpoints which are connected by ramps. The level of the start and end of the ramps can be varied by adjusting the level of these breakpoints. The Module features logarithmic pitch control at the "P" (pitch) input port and linear amplitude modulation at the "A" input port. The amplitudes of all eight breakpoints can be set independent of the others.

Application

The 8-Ramp Module can be used to create any variety of ramped waveforms including the triangle waveform. The spectrum of this oscillator type is less bright than the one produced by the stepped oscillators. By changing the steepness of the ramps you can vary the presence of high harmonic components in the oscillator's output signal.

6.35.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the pitch (oscillator frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module has the range [-A … A].
- **(S1)** "S1" (step 1) is the Event input for controlling the level of the first breakpoint.
- **(S2)** "S2" (step 2) is the Event input for controlling the level of the second breakpoint.
- **(S3)** "S3" (step 3) is the Event input for controlling the level of the third breakpoint.
- **(S4)** "S4" (step 4) is the Event input for controlling the level of the fourth breakpoint.
- **(S5)** "S5" (step 5) is the Event input for controlling the level of the fifth breakpoint.
- **(S6)** "S6" (step 6) is the Event input for controlling the level of the sixth breakpoint.
- **(S7)** "S7" (step 7) is the Event input for controlling the level of the seventh breakpoint.
- **(S8)** "S8" (step 8) is the Event input for controlling the eigth breakpoint.

Output Ports

- **(Out)** "Out" is the Audio signal output for the ramp waveform.

6.35.3 Example: Rehearsing the Oscillator

This example shows how to get a ramped waveform with a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

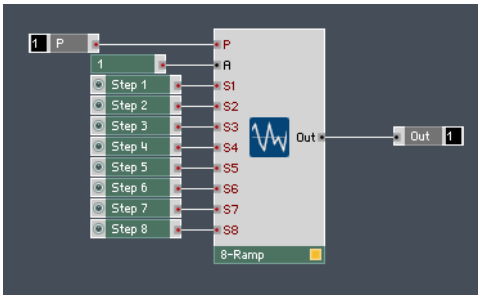


Fig. 6.105 A simple implementation of the 8-Ramp Module.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

Fig. 6.106 A typical 8-ramp waveform.

6.36 Ramp Oscillator

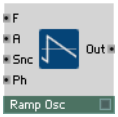


Fig. 6.107 Ramp Osc Module.

6.36.1 Overview

The Ramp Osc Module is an oscillator with a ramp waveform. It features linear frequency control at the "F" input port and linear amplitude control at the "A" input port. The ramp waveform is a linear ramp from "0" to "A" which then instantly wraps around back to "0". In the Function page you can choose from three available modes of operation, depending on the precision required for the ramp. The Ramp Osc Module also features phase synchronization (hard synch) at the "Snc" input port. A positive zero crossing of the signal at the "Snc" input port causes the phase of the ramp oscillator to be reset to the phase specified at the "Ph" input port.

Application

The Ramp Osc Module is typically used as a control signal, for example as a read or write pointer for an Audio Table Module that functions as a waveform oscillator. Such an example is shown in chapter 11 of the Application Reference. You can also use the ramp as a playback pointer for a sampler.

6.36.2 Properties: Function Page

Setting the Oscillator Mode

Depending on how the Ramp Osc Module is implemented in a Structure, different oscillator modes are required. To choose the oscillator mode, use the Mode drop-down menu, as shown in the picture below.

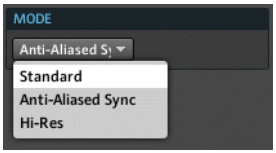


Fig. 6.108 The three operation modes of the Ramp Osc Module can be selected using the Mode drop-down menu in its Function page.

There are three available operation modes:

- **Standard:** There is no anti-aliasing, enabling you to use the ramp as a read or write pointer. However, in this mode for long samples there will be tempo/pitch inaccuracies due to floating point accumulation errors.
- **Anti-Aliased Sync:** This mode causes the output signal to be anti-aliased which makes the ramp oscillator usable as a source for an audible signal.
- **Hi-Res:** If the mode is set to Hi-Res te internal resolution is set to 48 bits and this enables ramp oscillator to be used as a phase source for long samples.

6.36.3 Ports

Input Ports

- **(F)** "F" (frequency) is the Audio input port to set the oscillator frequency in Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the ramp amplitude. In other words, the output signal of the Module is a ramp from "0" to "A"], depending on the phase synchronization.

- **(Snc)** "Snc" (sync) is the Audio input port for controlling synchronization of the waveform. A positive crossing of the signal at this input port restarts the oscillator phase from the phase specified at the "Ph" input port. A positive zero crossing of a signal happens when signal values go from negative to positive.
- **(Ph)** "Ph" (phase) is the input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. The range for the "Ph" input port is [-1 ... 1] where "-1" corresponds to a phase of -180 degrees and "1" corresponds to a phase 180 degrees.

Output Ports

- **(Out)** "Out" is the output port for the audio signal with the possibly frequency modulated and/or hard synchronized ramp waveform.

6.36.4 Example

For an example of the implementation of the Ramp Osc Module, please refer to chapter 11 of the Application Reference.

6.37 Clock Oscillator

Fig. 6.109 Clock Osc Module

6.37.1 Overview

The Clock Osc represents a free running monophonic clock source. By free running it is meant that this clock source is inherently independent from REAKTOR's MIDI Clock. The internal oscillator produces a regular stream of "On" and "Off" signal values in the form of Events. The value of the "On" Events is specified at the "A" input port. With the "Snc" input port you can reset (synchronize) the oscillator and with the "W" input port you can control the duration of the "On" value.

Application

A possible application of the Clock Osc is as a clock source for a sequencer. Another possibility is to create a timer, as shown in the example below. Note that with the appropriate Structure you can easily synchronize the Clock Osc Module to the MIDI Clock.

6.37.2 Ports

Input Ports

- **(F)** "F" (frequency) is the Event input port to set the frequency of the "On" Events in Hz. If you think in BPM and would like to convert a BPM value to its corresponding frequency in Hertz, use the following formula: Events-per-beat / 60 BPM. So, for 16th note Events at 4 beats to the bar (that is four 16ths per beat), you get F = 4 / 60 BPM = 0.0667 BPM.
- **(W)** "W" (width) is the Audio input port for the duration of the "On" Event. The range of the values at this input port is [-0.98 ... 0.98]. The ratio between the "On" and "Off" states of the output signal in relation to the value at the "W" input port is given as On : Off = (1 + W) / (1 – W). So, a "W" value of "0" yields a symmetric pulse waveform (a On : Off ratio of 50 : 50) at the output port. A "W" value of "-0.33" yields an On : Off ratio of 33 : 66, "-0.5" yields an On : Off ratio of 25 : 75, and "-0.98" already yields a waveform that can be considered an impulse train.
- **(Snc)** "Snc" (sync) is the Event input port for controlling synchronization of the clock. A positive Event at this input port resets the oscillator.
- **(A)** "A" (amplitude) is the Audio input port for controlling the value of the "On" Event. In other words, the output signal of the Module is series of values being either "0" or "A".
- **(Snc)** "Snc" Event input for synchronization of the waveform. A positive Event synchronizes the clock source.

Output Ports

- **(Out)** "Out" is the Event output for the clock signal, alternating between "On" value and zero.

6.37.3 Example: Simple Timer

The idea to use the Clock Osc Module to build a timer is not very farfetched. The Structure in the first picture below shows an implementation of a very simple timer. The Structure consists of three parts: the Clock Osc Module as the source of clock Events, the Counter Module as a counting mechanism, and the Numeric Module as a means of displaying the result.

In a nutshell, the Clock Osc Module sends Events with the value A = 1 at a frequency determined by the "Freq" knob to the "Up" input port of the Counter Module. The counter Module counts the number of positive Events at its "Up" input port (the zero-valued Events are ignored) and forwards the total number of Events to its output port. The Numeric Module is used to display that number on the Panel.

Additionally, a button labeled "Snc" has been created to reset the timer. In the resetting mechanism, first the clock needs to be reset by sending a positive Event to the "Snc' input port and second an Event with the value "0" needs to be sent to the "Set" input port of the Counter. The latter will cause the Counter state to be set to zero. The Order Module has been used to cause the Clock Osc Module to be reset before the Counter Module (please refer to subsection 9.2 in the Application Reference for more information on the order of Events). The Value Module is used to use the Event from the "Snc" button which carries the value "1" to trigger an Event with the value "0" to be sent to the "Set" input port.

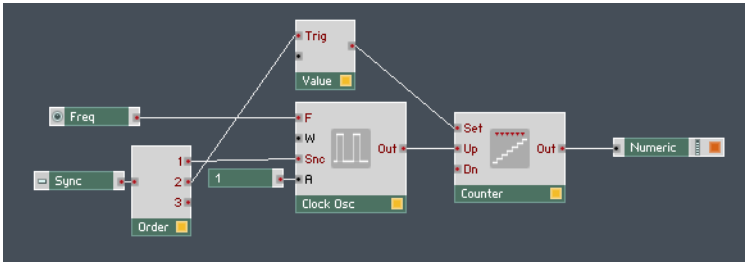


Fig. 6.110 This Structure is an implementation of a simple timer.

Fig. 6.111 The Panel representation of the timer.

6.38 Noise Oscillator

Fig. 6.112 Noise Module

6.38.1 Overview

The Noise Module is a white noise generator. It produces a random signal containing equal amounts of all possible frequency components. The signal consists of only two values, "A/2" and "–A/2" which appear in a random sequence.

Application

Since white noise contains equal amounts of all possible frequency components, it is an ideal starting point for subtractive synthesis. You can use the Noise Module to also produce pink noise and brown noise (also known as brownian noise). The amplitude reduction for pink noise is 3 dB per octave whereas the same parameter for brown noise is 6 dB per octave. You can connect the output of the Noise Module to filters to produce these types of noise. It should be noted, that pink noise is the type of noise perceived as most pleasing to the human ear and not coincidentally it is a type of statistical behavior that pops up in all kinds natural phenomena.

6.38.2 Ports

Input Ports

- **(A)** "A" (amplitude) is the Audio input for controlling the amplitude of the white noise signal. The output signal consists of two values "A/2" and "-A/2".

Output Ports

- **(Out)** "Out" is the output port for the Audio signal with a noise spectrum.

6.38.3 Example: Rehearsing the Oscillator

This example shows how to get a white noise signal with constant amplitude.

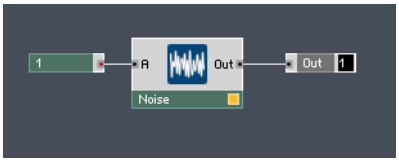


Fig. 6.113 Noise with an amplitude of "0.5"

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list

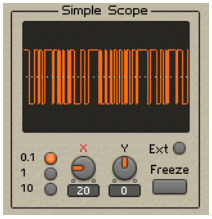


Fig. 6.114 The waveform corresponding to white noise.

6.39 Random Oscillator

Fig. 6.115 Random Module

6.39.1 Overview

The Random Module is a pseudorandom value generator. The output signal is a stepped waveform where the level of each step is pseudorandom in the amplitude interval. Each "level" has an equal probability of appearing. We say pseudorandom because the algorithm for generating the number is well-defined and therefore will always behave in a predictable

manner. However, if you don't know that the sequence of numbers is generated by an algorithm, they appear to be perfectly random. The seed for creating different pseudorandom sequences is set using the Set Random Module (↑14.24, Set Random). The Random Module works like a noise generator with uniform distribution followed by a sample & hold circuit clocked at a regular frequency.

Application

The Random Module can be used to generate a "lo-fi", broken circuit audio signal. You can also add it in small doses to your oscillators to add more grit to them if you are into that type of sound. Since the audible signal generated by the Random Module has strong high harmonic components it is also a good starting point for subtractive synthesis.

To add a random value to control signals such as envelopes, LFOs, or sequencer Event timing and velocity values, it is more efficient to use the Slow Random Module in the LFO, Envelope section!

6.39.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the step rate (sample and hold frequency) in the logarithmic pitch scale. The pitch value is given in semitones in the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the oscillator amplitude. In other words, the output signal of the Module is a pseudorandom value in the range [-A … A].

Output Ports

- **(Out)** "Out" Audio signal output for the pseudorandom step waveform.

6.39.3 Example: Rehearsing the Oscillator

This example shows how to get a pseudorandom waveform within the bounds of unit amplitude and a pitch set with the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module.

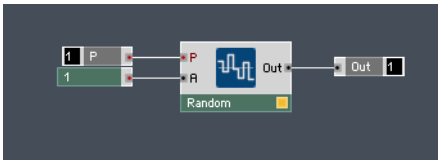


Fig. 6.116 A Structure for a pseudorandom signal with unit amplitude.

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

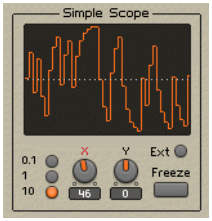


Fig. 6.117 The Random Module is used to generate a pseudorandom signal.

6.40 Geiger Oscillator

Fig. 6.118 Geiger Module

6.40.1 Overview

The Geiger Module generates an impulse train at random time intervals, much like a Geiger Counter radiation particle detector. The average rate of impulses can be controlled at the "P" (pitch) input port. The value at the "Rnd" input port controls the randomness of the

Event timing. The seed for creating different pseudorandom sequences for the Geiger Module is set using the Set Random Module (↑14.24, Set Random). This Module features both an Audio and an Event output port.

Application

The Geiger Module can be used in two distinct ways, depending on which output port you use. With the "Out" Event output port you can randomly trigger processes such as envelopes or sequencer patterns. Connect the "Out" Event output port to a "G" (gate) input port of an Envelope Module to have the impulses trigger an Envelope. At low pitch values you can use the signal from the "Clk" Audio output port for sound effects mimicking the well-known Geiger Counter, a popular motive in film productions. At higher pitches the sound resembles a zipper noise. A Geiger impulse train too has strong high harmonic components. This means that you can use the signal from the "Clk" output port as a starting point for subtractive synthesis.

6.40.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the average rate or density of the randomly occurring impulses in the logarithmic pitch scale. The typical range for the incoming pitch values at this input port is [-50 ... 50].
- **(Rnd)** "Rnd" (random) is the Event input port for controlling the randomness of the Event distribution. "0" corresponds to a completely random distribution, "1" corresponds to a completely regular impulse train. The typical range for incoming values at this input port is [0 ... 1].

Output Ports

- **(Clk)** "Clk" is the Audio output port for the Geiger impulse train.
- **(Out)** "Out" is the output port for the (randomly timed) Events.

6.40.3 Example: Rehearsing the Oscillator

This example shows how to get a Geiger impulse train with a pitch set by the MIDI controller or the keyboard. The pitch information is sent to the "P" (pitch) input port from the output port of the Note Pitch Module. Use a knob connected to the "Rnd" input port to control the randomness of the Geiger impulse train.

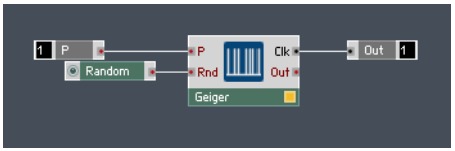


Fig. 6.119 Geiger impulse train

The best way to get acquainted with the oscillators in REAKTOR is to hear them in action while tweaking the different parameters. You can hear the sound of this oscillator from the "Clk" output port and compare it to the other oscillators in the Oscillator Rehearsal Ensemble. The Ensemble consists of an easy to use interface to select and tweak your oscillator of choice while following the change in its sound with your ears and the change in waveform on the scope. Since each oscillator can have its own set of controls besides pitch and amplitude, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an oscillator has been selected from the "Oscillators" list.

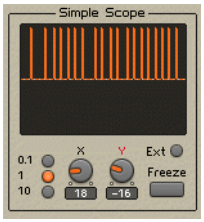


Fig. 6.120 The geiger waveform

7 Samplers

If the source of your Audio signal in REAKTOR is not an oscillator or an external Audio input signal, then it must be a Sampler Module. REAKTOR's Sampler Modules include basic sample players as well as sophisticated sample processors for granular synthesis, pitch and time shifting, and beat slicing. There's even one for picking out individual samples by number. For an overview of the different kinds of Sampler Modules and their Function and Appearance pages, please refer to chapter 10 in the Application Reference.

7.1 Sampler

Fig. 7.1 Sampler Module

7.1.1 Overview

The Sampler Module is a "conventional" sampler, as described in subsection 10.2.1 in the Application Reference. The playback speed of the sample is coupled to the playback pitch. If you send a higher pitch value to the "P" (pitch) input port of the Module and trigger a playback, then the playback speed is increased. The spectral components of the sample are transposed uniformly, that is, all components are transposed by the same amount. This means that by transposing a sample too much, the perceived timbre of the sample is strongly altered. For example, an adult human voice that is transposed upwards an octave will sound like a chipmunk. Sample management is carried out in the Sample Map Editor.

Please refer to chapter 6 in the Application Reference for more information on the Sample Map Editor.

Application

The Sampler Module is used for real-time polyphonic and transposed playback of samples in Sample Maps. You can build samplers that are triggered for a "one-shot" playback with the trigger signal coming from a button on the Instrument Panel or as an Event from a sequencer. You can also use edit the sample playback settings in the Sample Map Editor

and have the sample playback as a loop. This is useful when you wish to playback beat loops or long atmospheric samplers. Another application is to use the Sampler Module as an oscillator where the oscillator waveform is given by the loaded sample. For this you must have the Oscillator Mode checkbox in the Module's Function page engaged. The playback rate of the sample will then be adjusted to produce the correct pitch.

7.1.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the playback rate (pitch) at which the sample is played back and for selecting a sample from the loaded Sample Map. The "P" (pitch) value is given in the logarithmic pitch scale, in semitones belonging to the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz. If the pitch value is equal to the Root Key of the current sample, the sample will be played back at its original pitch.
- **(Trig)** "Trig" (trigger) is the Event input port for triggering a sample playback from the beginning of the sample. A positive Event at this input is considered a trigger.
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude and the selection of a sample from the Sample Map by velocity mapping. Values in the range [0 … 1] are mapped to the velocity range [0 … 127] in the Sample Map.. Since this is an Audio input port, it can be used for ring modulation.

Output Ports

- **(Out)** "Out" is the Audio output port delivering the sample player's output signal.

7.1.3 Example: Sampler with Trigger Button

The simplest sampler Instrument can be built using the Sampler Module (↑7.1, Sampler) which receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its trigger signal from a Button Module (↑1.2, Button). Its amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Ampl", as shown in the Structure below. Note that you can create a Knob Module (or sometimes a Button or Fader Module) for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry. Make sure that the Button at the "Trig" (trigger) input port has been set to Trigger Mode in its Function page.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into the Sampler Module's Sample Map.

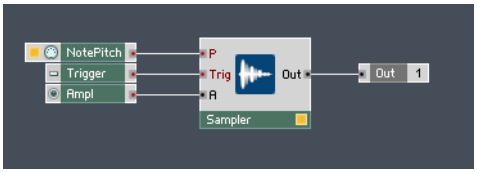


Fig. 7.2 The Structure for a simple sampler Instrument.

7.2 Sampler FM

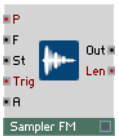


Fig. 7.3 Sampler FM Module

7.2.1 Overview

The Sampler FM Module is a "conventional" sampler, as described in subsection 10.2.1 in the Application Reference. The playback speed of the sample is coupled to the playback pitch. If you send a higher pitch value to the "P" (pitch) input port of the Module and trigger a playback, then the playback speed is increased. The spectral components of the sample are transposed uniformly, that is, all components are transposed by the same amount. This means that by transposing a sample too much, the perceived timbre of the sample is strongly altered. For example, an adult human voice that is transposed upwards an octave will sound like a chipmunk. Sample management is carried out in the Sample Map Editor. The "F" input port allows you to add frequency modulation to the sample playback. Additionally, use the "St" input port to control the starting point of the sample playback.

Please refer to chapter 6 in the Application Reference for more information on the Sample Map Editor.

Application

The Sampler FM Module is used for real-time polyphonic and transposed playback of samples in Sample Maps. You can build samplers that are triggered for a "one-shot" playback with the trigger signal coming from a button on the Instrument Panel or as an Event from a sequencer. You can also use edit the sample playback settings in the Sample Map Editor and have the sample playback as a loop. This is useful when you wish to playback beat loops or long atmospheric samplers. Another application is to use the Sampler FM Module as an oscillator where the oscillator waveform is given by the loaded sample. For this you must have the Oscillator Mode checkbox in the Module's Function page engaged. The playback rate of the sample will then be adjusted to produce the correct pitch. In addition to the usual sound you get from sample playback at different speeds, you get the added benefit of being able to produce an FM'ish sound by using the Sampler FM Module instead of simply using the Sampler Module.

7.2.2 Ports

Input Ports

- **(P)** "P" (pitch) is the Event input port for controlling the playback rate (pitch) at which the sample is played back and for selecting a sample from the loaded Sample Map. The "P" (pitch) value is given in the logarithmic pitch scale, in semitones belonging to the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz. If the pitch value is equal to the Root Key of the current sample, the sample will be played back at its original pitch.
- **(F)** "F" (frequency) is the linear Audio input port for modulating the sample playback rate in the linear frequency scale. When this input port is connected to another oscillator, the frequency modulation effect is achieved — the same as for the oscillator modules in REAKTOR. If a large negative value is present, the sample is played backwards.
- **(St)** "St" (start position) is the Audio input port for controlling the starting point of the sample playback. The position is set in milliseconds from the start of the sample. The length of the loaded sample in milliseconds is delivered by the "Len" output port. This means that although the samples in your Sample Map might have different lengths, you can use the value from the "Len" output port and multiply it by a parameter in the range [0 … 1]. The result will yield you values in the full range of the current sample, in milliseconds, which you then feed to the "St" input port. If this input port is left unconnected, then the default value "0" is used.

- **(Trig)** "Trig" (trigger) is the Event input port for triggering a sample playback from the beginning of the sample. A positive Event at this input is considered a trigger.
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude and the selection of a sample from the Sample Map by velocity mapping. Values in the range [0 … 1] are mapped to the velocity range [0 … 127] in the Sample Map. Since this is an Audio input port, it can be used for ring modulation.

Output Ports

- **(Out)** "Out" is the Audio output port delivering the sample player's output signal.
- **(Len)** "Len" (length) is the polyphonic Event output port for the length of the current sample in milliseconds. Use this value to calculate parameters which depend on the sample length. The sample starting point (specified at the "St" input port), for example, is a parameter that should have a range from "0" to the length of the sample in milliseconds. The latter value is exactly what is supplied by the "Len" output port.

7.2.3 Example: FM Sampler

This example illustrates a sampler Instrument with frequency modulation and variable starting position. As can be seen in the figure below, the Sampler FM Module (↑7.2, Sampler FM) has been used. It receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its trigger signal from the Gate Module (↑2.3, Gate). Its amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Ampl", as shown in the Structure below. Note that you can create a Knob Module (or sometimes a Button or Fader Module) for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry.
A sine oscillator has been used as the frequency modulation signal. The pitch and amplitude of this signal can be controlled at the corresponding input ports of its source, the Sine Module (↑6.14, Sine Oscillator). The starting position of sample playback can also be controlled using the Knob Module labeled "Start", which has its range set to [0 … 1]. This range is multiplied by the length of the currently loaded sample, retrieved from the "Len" (length) output port and is then forwarded to the "St" (start) input port.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into a Sampler Module's Sample Map.

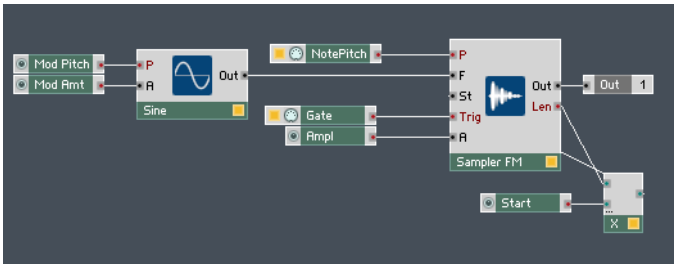


Fig. 7.4 The Structure for a frequency modulated sampler Instrument.

7.3 Sampler Loop

Fig. 7.5 Sampler Loop Module

7.3.1 Overview

The Sampler Loop Module is a "conventional" sampler, as described in subsection 10.2.1 in the Application Reference. The playback speed of the sample is coupled to the playback pitch. If you send a higher pitch value to the "P" (pitch) input port of the Module and trigger a playback, then the playback speed is increased. The spectral components of the sample are transposed uniformly, that is, all components are transposed by the same amount. This means that by transposing a sample too much, the perceived timbre of the sample is strongly altered. For example, an adult human voice that is transposed upwards an octave will sound like a chipmunk. Sample management is carried out in the Sample Map Editor. The "F" input port allows you to add frequency modulation to the sample playback.

With the Sampler Loop Module you can directly access the sample loop playback settings with the "St", "LS", and "LL" input ports to control the starting point of the sample playback, loop starting point and loop length, respectively. If the Loop checkbox in the Sampler Map editor is activated for a given sample, the sample will be continuously repeated with-

in the loop range as soon as playback enters this range. The loop range is preset using data from the sound file from which the sample was loaded. If the sound file does not contain any loop data, the beginning of the sample is taken to be the beginning of the loop and the sample length is taken as the loop length. The loop data from the sound file are the default settings. These defaults are always used if the "LS" (loop start) and "LL" (loop length) input ports are not connected to other Modules. If the Loop in Release feature is deactivated, loop playback will fade out upon a Gate Off Event occurring; depending on the direction set, the sample will run to its beginning or end and will then fade out. If the Loop in Release feature is activated or if loop playback is switched off, the Gate Events will only have a slight effect or none at all.

Application

The Sampler Loop Module is used for real-time polyphonic and transposed playback of mono or stereo samples in Sample Maps or Wavetables. You can build samplers that are triggered for a "one-shot" playback with the trigger signal coming from a button on the Instrument Panel or as an Event from a sequencer. Since the loop playback settings are accessible from input ports, you can use the Sampler Loop Module to build a sampler that lets you manipulate loop settings from the Instrument Panel.

Another application is to use the Sampler Loop Module as an oscillator where the oscillator waveform is given by the loaded sample. For this you must have the Oscillator Mode checkbox in the Module's Function page engaged. In Oscillator Mode the Sampler Loop Module interprets the "LL" (loop length) value as the length of one waveform cycle. The loop length is read from the sound file, but can be changed at the "LL" input of the Module. Therefore a sample can contain numerous waveforms. Such a sample is known as a Wavetable. Assuming a waveform contains 100 sample points, then 100 such waveforms fit into a sample comprising 10,000 sample points. The first waveform covers the sample points 0-99, the second 100-199 etc. If the "LL" (loop length) value sets the length of a waveform, then the "LS" (loop start) value logically sets the waveform to be played from the Wavetable. In Oscillator Mode the values at this input are quantized, so that glitch free playback between waveforms is achieved. The position in a Wavetable can be easily controlled by velocity, etc. Obviously such Wavetable samples need to be either created or generated from a sample. The REAKTOR Library contains many Ensembles that serve as examples for a Wavetable synthesizer. The Sampler Loop Module integrates Wavetable Synthesis with REAKTOR's existing synthesis capabilities. In this way, Wavetable Synthesis is available in combination with FM synthesis.

7.3.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for a Gate signal. A positive Event at this input starts the sample playback from the position that is set at the "St" input port. If a negative value is present at the "G" input port, the loop playback is interrupted unless the Loop in Release feature has been activated in the Sample Map Editor (see subsection 6.4.2 in the Application Reference for more information on this).
- **(P)** "P" (pitch) is the Event input port for controlling the playback rate (pitch) at which the sample is played. The "P" (pitch) value is given in the logarithmic pitch scale, in semitones belonging to the range [0 … 127], where "69" corresponds to "Concert A" or 440 Hz. If the pitch value is equal to the Root Key of the current sample, the sample will be played back at its original pitch. Furthermore, if the "Sel" input port is not connected, "P" determines the selection of samples from the Sample Map. In Oscillator Mode, the Sampler Loop functions as a digital oscillator. In the same way as in the Oscillator Modules in REAKTOR, "P" determines the fundamental pitch of the generated oscillation.
- **(Sel)** "Sel" (select) is the Audio input port for selecting a sample from the Sample Map. If this input is not connected, the values present at the "P" (pitch) input port are used instead.
- **(F)** "F" (frequency) is the linear Audio input port for modulating the sample playback rate in the linear frequency scale. When this input port is connected to another oscillator, the frequency modulation effect is achieved — the same as for the oscillator modules in REAKTOR. If a large negative value is present, the sample is played backwards.
- **(St)** "St" (start) is the Audio input port for the starting point to be used when the next positive Gate Event occurs. The position is set in milliseconds and counted from the start of the sample. The length of the loaded sample in milliseconds is delivered by the "Len" output port. This means that although the samples in your Sample Map might have different lengths, you can use the value from the "Len" output port and multiply it by a parameter in the range [0 … 1]. The result will yield you values in the full range of the current sample, in milliseconds, which you then feed to the "St" input port. If this input port is left unconnected, then the default value "0" is used.

- **(LS)** "LS" (loop start) is the Audio input port for the loop starting point in milliseconds, counted from the start of the sample. If this input is not connected, the loop data stored in the sound file will be used. If no loop data is stored in the sound file, the default is used: LS = 0. The values at this input port fall in the range [0 ... <length of sample in millisecons>]. Changes at this input take effect when samples are retriggered and when a loop limit is reached.
- **(LL)** "LL" (loop length) is the Audio input port for the loop length in milliseconds. If this input is not connected, the loop data stored in the sound file will be used. If no loop data are stored in the sound file, the default is used: LL = length of the whole sample. If LL = 0, the movement within the sample stops when the loop starting point is reached; the sound is frozen at this point. The typical range for the values at this input port range from "0" to the length of the sample in milliseconds. Changes at this input take effect when samples are retriggered and when a loop limit is reached.
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude. Since this is an Audio input port, it can be used for ring modulation. If this input port is disconnected, the default value is "0", that is, no audible output signal is generated.

Output Ports

- **(L)** "L" (left) is the Audio output port for the left stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "R" output port.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "L" output port.
- **(Len)** "Len" (length) is the polyphonic Event output port for the length of the current sample in milliseconds. Use this value to calculate parameters which depend on the sample length. The sample starting point (specified at the "St" input port), for example, is a parameter that should have a range from "0" to the length of the sample in milliseconds. The latter value is exactly what is supplied by the "Len" output port.

7.3.3 Example: Loop Sampler

This example illustrates a sampler Instrument with frequency modulation and variable (loop) starting position and loop length. As can be seen in the figure below, the Sampler Loop Module (↑7.3, Sampler Loop) has been used. It receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its Gate signal from the Gate Module (↑2.3, Gate). When a key on your computer or MIDI keyboard is pressed, a Gate On signal is sent from the Gate Module, carried by the Polyphonic Voice assigned to the key. As a result sample playback is started for this Voice until the key is released. When the key is released, a Gate Off signal is sent from the Gate Module (↑2.3, Gate) and stops the sample playback (at the "G" (gate) input port. Its amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Ampl", as shown in the Structure below. Note that you can create a Knob Module (↑1.1, Fader/Knob) (or sometimes a Button or Fader Module) for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry. Also, because the "Sel" input port is disconnected, the value at the "P" (pitch) input port also determines which sample is loaded from the Sample Map.

A sine oscillator has been used as the frequency modulation signal. The pitch and amplitude of this signal can be controlled at the corresponding input ports of its source, the Sine Module (↑6.14, Sine Oscillator). The starting position of sample playback, starting position of the loop and loop length can also be controlled using Knob Modules (↑1.1, Fader/Knob) labeled "St", "LS", and "LL", respectively. All three knobs have the range [0 … 1]. This range is multiplied by the length of the currently loaded sample, retrieved from the "Len" (length) output port, and is then forwarded to the corresponding input port. This way the three parameters at the input ports are always in the range of the currently loaded sample.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into a Sampler Module's Sample Map.

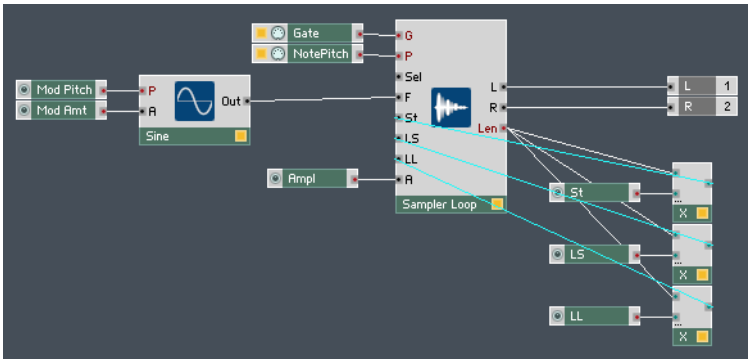


Fig. 7.6 The Structure for a frequency modulated loop sampler Instrument.

7.4 Grain Resynth

Fig. 7.7 Resynth Module

7.4.1 Overview

The Resynth Module is a "Grain Resynthesis" Module, as described in subsection 10.2.2 in the Application Reference. The playback speed of the sample is decoupled from the playback pitch. Sample management is carried out in the Sample Map Editor. In essence, the Resynth Module is a real-time resynthesizer for polyphonic, transposed playback of mono or stereo samples. In the same way as the "conventional" Sampler Modules, the Resynth Module uses a pointer at the current sample position. However, the signal at its Audio output ports is not simply the sample amplitude occurring at the pointer position. What actually happens is that the output signal is generated by a synthesizer inside the Module. This synthesizer resynthesizes the signal at the pointer position. The pitch of the

signal generated by this synthesizer is independent of the pointer's speed. Even if the pointer is not moving at all, the synthesizer continues to produce a sound. While the pitch of the output signal is determined, as is usual, over the "P" (pitch) input port, the value at the "Sp" (speed) input port determines the pointer speed.
With the Resynth Module you can directly access the sample loop playback settings with the "St", "LS", and "LL" input ports to control the starting point of the sample playback, loop starting point and loop length, respectively. If the Loop checkbox in the Sampler Map editor is activated for a given sample, the sample will be continuously repeated within the loop range as soon as playback enters this range. The loop range is preset using data from the sound file from which the sample was loaded. If the sound file does not contain any loop data, the beginning of the sample is taken to be the beginning of the loop and the sample length is taken as the loop length. The loop data from the sound file are the default settings. These defaults are always used if the "LS" (loop start) and "LL" (loop length) input ports are not connected to other Modules. If the Loop in Release feature is deactivated, loop playback will fade out upon a Gate Off Event occurring; depending on the direction set, the sample will run to its beginning or end and will then fade out. If the Loop in Release feature is activated or if loop playback is switched off, the Gate Events will only have a slight effect or none at all.
All of the Resynth's input ports, except "G" (gate), are designed as Audio input ports. The values at these inputs are applied when samples are (re-triggered (i.e. upon positive Gate On Events). Changes to values during sample playback take effect in the Granularity interval which is configured at the "Gr" input.

Application

The Resynth Module allows you to control pitch and playback speed independently and in real-time, and also lets you extensively manipulate samples. Of course, the slowed down or "frozen" signal does not always correspond to what you might have imagined it to be. What does a hammer hitting a nail sound like if it is slowed down to infinity? The resynthesis algorithm used by the Resynth Module is designed in such a way that a broad range of sounds can be processed in a (musically-speaking) sensible, subtle or drastic manner. The algorithm can be adjusted by configuring the "Gr" (granularity) and "Sm" (smoothness) parameters with the corresponding input ports. This means that you can manually adjust it to suit the sample and to produce drastic, strange sound effects. Use the Resynth Module to build Ensembles that let you get squeeze new and unheard sounds out of your existing samples!

7.4.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for a Gate signal. A positive Event at this input starts the sample playback from the position that is set at the "St" input port. If a negative value is present at the "G" input port, the loop playback is interrupted unless the Loop in Release feature has been activated in the Sample Map Editor (see subsection 6.4.2 in the Application Reference for more information on this).
- **(P)** "P" (pitch) is the Audio input port for controlling the playback rate (pitch) at which the sample is playe. The "P" (pitch) value is given in the logarithmic pitch scale, in semitones belonging to the range [0 … 127], where "69" corresponds to "Concert A" or 440 Hz. If the pitch value is equal to the Root Key of the current sample, the sample will be played back at its original pitch. Furthermore, if the "Sel" input port is not connected, "P" also determines the selection of samples from the Sample Map.
- **(Sel)** "Sel" (select) is the Audio input port for selecting a sample from the Sample Map. If this input is not connected, the values present at the "P" (pitch) input port are used instead.
- **(St)** "St" (start) is the Audio input port for the starting point to be used when the next positive Gate Event occurs. The position is set in milliseconds and counted from the start of the sample. The length of the loaded sample in milliseconds is delivered by the "Len" output port. This means that although the samples in your Sample Map might have different lengths, you can use the value from the "Len" output port and multiply it by a parameter in the range [0 … 1]. The result will yield you values in the full range of the current sample, in milliseconds, which you then feed to the "St" input port. If this input port is left unconnected, then the default value "0" is used.
- **(LS)** "LS" (loop start) is the Audio input port for the loop starting point in milliseconds, counted from the start of the sample. If this input is not connected, the loop data stored in the sound file will be used. If no loop data is stored in the sound file, the default is used: LS = 0. The values at this input port fall in the range [0 … <length of sample in millisecons>]. Changes at this input take effect when samples are retriggered and when a loop limit is reached.
- **(LL)** "LL" (loop length) is the Audio input port for the loop length in milliseconds. If this input is not connected, the loop data stored in the sound file will be used. If no loop data are stored in the sound file, the default is used: LL = length of the whole sample.

If LL = 0, the movement within the sample stops when the loop starting point is reached; the sound is frozen at this point. The typical range for the values at this input port range from "0" to the length of the sample in milliseconds. Changes at this input take effect when samples are retriggered and when a loop limit is reached.

- **(Sp)** "Sp" (speed) is the Audio input port to control the pitch-independent playback speed. Values that are present at the input port are interpreted as factors: when Sp = 1, playback occurs at the original speed; Sp = 2 corresponds to double the playback speed; Sp = 0 means stop (no playback). If this input is not connected, the transposed original speed is taken as the default so that – as in conventional samplers – the speed reduces as pitch decreases. The typical range of the values at this input port is [0 … 2].
- **(Gr)** "Gr" (granularity) is the Audio input port for the granularity of the resynthesis process, set in milliseconds. This parameter determines the size of the sound particles used for resynthesis. The typical range of the values at this input port is [10 … 100]. If this input port is left unconnected, the default value, 20 milliseconds, is used.
- **(SO)** "SO" (sample offset) is the Audio input port for modulation of the sample position (sample offset) in milliseconds. This input is used to modulate the position in the sample independent of pitch, for instance, via a noise generator.
- **(Sm)** "Sm (smoothing) is the Audio input port to control the Smoothness parameter of the resynthesis process. The sound particles are adjusted using this. Very small values generally lead to a rougher sound.
- **(Pan)** "Pan" is the Audio input port to control the position of the output signal in the stereo field (-1 = Left, 0 = Center, 1 = Right).
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude. Since this is an Audio input port, it can be used for ring modulation.

Output Ports

- **(L)** "L" (left) is the Audio output port for the left stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "R" output port.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "L" output port.

- **(Len)** "Len" (length) is the polyphonic Event output port for the length of the current sample in milliseconds. Use this value to calculate parameters which depend on the sample length. The sample starting point (specified at the "St" input port), for example, is a parameter that should have a range from "0" to the length of the sample in milliseconds. The latter value is exactly what is supplied by the "Len" output port.
- **(Pos)** "Pos" (position) is the polyphonic Event output for the current sample position in milliseconds. Events at this output port are generated at time intervals corresponding to the "Gr" (granularity) parameter.

7.4.3 Example: Grain Resynth Resampler

The Structure in the figure below shows how to achieve synchronization between playback of a Resynth Module and recording of a Tapedeck Module (↑14.2, Tapedeck 2 Channel). Controls have been created for the most relevant parameters of the Resynth Module by right-clicking the corresponding input port and choosing the *Create Control* menu entry. The "Rec" Button Module is set to Toggle Mode and sends a Gate On and Gate Off signal to the "G", "Rec", and "Play" input ports. This way recording and sample playback are synchronized. Additionally, since a Gate On signal also arrives at the "Play" input port, what is being recorded by the Tapedeck can be heard at its "L" and "R" output ports. Note that a Merge Module (↑13.16, Merge) has been used at the "Play" input port such that the recorded signal can be played back independently from the playback of the Resynth Module.

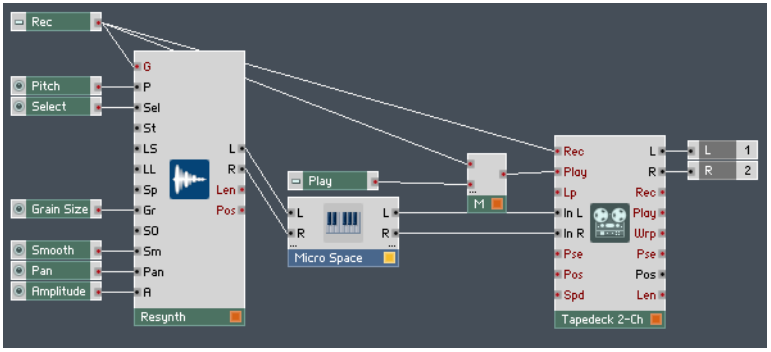


Fig. 7.8 The Structure for resampling the output of a Resynth Module with a Tapedeck 2 Channel Module.

7.5 Grain Pitch Former

Fig. 7.9 Pitch Former Module

7.5.1 Overview

The Pitch Former Module is a "Grain Resynthesis" Module, as described in subsection 10.2.2 in the Application Reference. The playback speed of the sample is decoupled from the playback pitch. Sample management is carried out in the Sample Map Editor. The Pitch Former Module is a real-time resynthesizer for polyphonic, transposed playback of mono or stereo samples. In particular, it is a Wavetable synthesizer in which not only Wavetables can be loaded but also any samples you like. The Pitch Former Module removes the pitch development from a sample and gives the sample any new pitch you wish. Besides independent real-time control over the playback speed, the Pitch Former Module also allows you to carry out a spectral transposition, that is, a pitch-independent transposition of the timbre.

While "conventional" Sampler Modules and the Resynth Module in REAKTOR achieve a relative pitch change by transposing a sample, the Pitch Former Module forces any absolute and definite pitch that you wish onto a sample. The Pitch Former Module can therefore be used like a REAKTOR oscillator. Depending on the type of processed sound material and the settings used, the result can sound “electronically” altered to a greater or lesser degree. The Pitch Former Module will also generate signals with a certain pitch if the processed sample has no unique pitch of its own (e.g. recordings of cymbals or gongs). The Pitch Former Module produces fewer sound alterations the more restricted the fundamental pitch of the processed material is to be defined, and the less the original pitch deviates from the generated pitch.

In conventional "conventional" Sampler Modules and the Resynth Module in REAKTOR, the transposing of the fundamental pitch goes hand in hand with the transposing of all the spectral components. This is often considered a limitation since the transposition also affects certain spectral components which the listener would not expect to hear changed. The “Mickey Mouse effect” that occurs when speech recordings are detuned can be traced back to the transposition of the formants whose position for a “real” human speaker would be largely independent of the fundamental pitch. Within certain limits, the Pitch Former Module decouples pitch and formant position from one another. The formant position can be shifted independent of pitch via the "FS" (formant shift) input port. Since the human ear uses all the spectral components to identify the fundamental pitch, you can manipulate this parameter to create interesting substitutions of fundamental pitch and timbre, especially at very low pitches. Similar sound effects are often created by synthesizer experts using oscillator synchronization.

In the same way as the "conventional" Sampler Modules, the Pitch Former Module uses a pointer at the current sample position. However, the signal at its Audio output ports is not simply the sample amplitude occurring at the pointer position. What actually happens is that the output signal is generated by a synthesizer inside the Module. This synthesizer resynthesizes the signal at the pointer position. The pitch of the signal generated by this synthesizer is independent of the pointer’s speed. Even if the pointer is not moving at all, the synthesizer continues to produce a sound. While the pitch of the output signal is determined, as is usual, over the "P" (pitch) input port, the value at the "Sp" (speed) input port determines the pointer speed.

With the Pitch Former Module you can directly access the sample loop playback settings with the "St", "LS", and "LL" input ports to control the starting point of the sample playback, loop starting point and loop length, respectively. If the Loop checkbox in the Sampler Map editor is activated for a given sample, the sample will be continuously repeated within the loop range as soon as playback enters this range. The loop range is preset using data from the sound file from which the sample was loaded. If the sound file does not contain any loop data, the beginning of the sample is taken to be the beginning of the loop and the sample length is taken as the loop length. The loop data from the sound file are the default settings. These defaults are always used if the "LS" (loop start) and "LL" (loop length) input ports are not connected to other Modules. If the Loop in Release feature is deactivated, loop playback will fade out upon a Gate Off Event occurring; depending on

the direction set, the sample will run to its beginning or end and will then fade out. If the Loop in Release feature is activated or if loop playback is switched off, the Gate Events will only have a slight effect or none at all.
All of the Pitch Former Module's input ports except "G" (gate) are designed as Audio input ports. The values at these inputs are applied when samples are (re-triggered (i.e. upon positive Gate On Events). Changes to values during sample playback take effect at intervals of the fundamental pitch period (which is set via the "P" (pitch) input port.

Application

With both the sample's playback speed and its spectral form decoupled from the pitch, the Pitch Former Module gives you maximum freedom in manipulating your samples. Use the Pitch Former Module to build Ensembles that let you get squeeze new and unheard sounds out of your existing samples!

7.5.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for a Gate signal. A positive Event at this input starts the sample playback from the position that is set at the "St" input port. If a negative value is present at the "G" input port, the loop playback is interrupted unless the Loop in Release feature has been activated in the Sample Map Editor (see subsection 6.4.2 in the Application Reference for more information on this).
- **(P)** "P" (pitch) is the Audio input port for controlling the playback rate (pitch) at which the sample is played back. The "P" (pitch) value is given in the logarithmic pitch scale, in semitones belonging to the range [0 … 127], where "69" corresponds to "Concert A" or 440 Hz. If the pitch value is equal to the Root Key of the current sample, the sample will be played back at its original pitch. Furthermore, if the "Sel" input port is not connected, "P" also determines the selection of samples from the Sample Map.
- **(Sel)** "Sel" (select) is the Audio input port for selecting a sample from the Sample Map. If this input is not connected, the values present at the "P" (pitch) input port are used instead.
- **(St)** "St" (start) is the Audio input port for the starting point to be used when the next positive Gate Event occurs. The position is set in milliseconds and counted from the start of the sample. The length of the loaded sample in milliseconds is delivered by the "Len" output port. This means that although the samples in your Sample Map might

have different lengths, you can use the value from the "Len" output port and multiply it by a parameter in the range [0 … 1]. The result will yield you values in the full range of the current sample, in milliseconds, which you then feed to the "St" input port. If this input port is left unconnected, then the default value "0" is used.

- **(LS)** "LS" (loop start) is the Audio input port for the loop starting point in milliseconds, counted from the start of the sample. If this input is not connected, the loop data stored in the sound file will be used. If no loop data is stored in the sound file, the default is used: LS = 0. The values at this input port fall in the range [0 … <length of sample in millisecons>]. Changes at this input take effect when samples are retriggered and when a loop limit is reached.
- **(LL)** "LL" (loop length) is the Audio input port for the loop length in milliseconds. If this input is not connected, the loop data stored in the sound file will be used. If no loop data are stored in the sound file, the default is used: LL = length of the whole sample. If LL = 0, the movement within the sample stops when the loop starting point is reached; the sound is frozen at this point. The typical range for the values at this input port range from "0" to the length of the sample in milliseconds. Changes at this input take effect when samples are retriggered and when a loop limit is reached.
- **(Sp)** "Sp" (speed) is the Audio input port to control the pitch-independent playback speed. Values that are present at the input port are interpreted as factors: when Sp = 1, playback occurs at the original speed; Sp = 2 corresponds to double the playback speed; Sp = 0 means stop (no playback). If this input is not connected, the value "1" is taken as the default. The typical range of the values at this input port is [0 … 2].
- **(FS)** "FS" (formant shift) is the Audio input port for pitch-independent transposition of the formant position. The units for the values at this input port are semitones. The typical range for the values at this input port is [-24 … 24]. When this input port is disconnected, the default value, "0" is used.
- **(SO)** "SO" (sample offset) is the Audio input port for modulation of the sample position (sample offset) in milliseconds. This input is used to modulate the position in the sample independent of pitch, for instance, via a noise generator.
- **(Sm)** "Sm (smoothing) is the Audio input port to control the Smoothness parameter of the resynthesis process. The sound particles are adjusted using this. Very small values generally lead to a rougher sound. The range for the values at this input port is [0 … 1]. The default value to be used when this input port is disconnected is "0.8".
- **(Pan)** "Pan" is the Audio input port to control the position of the output signal in the stereo field (-1 = Left, 0 = Center, 1 = Right).

- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude. Since this is an Audio input port, it can be used for ring modulation. The default value to be used when this input port is disconnected is "1".

Output Ports

- **(L)** "L" (left) is the Audio output port for the left stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "R" output port.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "L" output port.
- **(Len)** "Len" (length) is the polyphonic Event output port for the length of the current sample in milliseconds. Use this value to calculate parameters which depend on the sample length. The sample starting point (specified at the "St" input port), for example, is a parameter that should have a range from "0" to the length of the sample in milliseconds. The latter value is exactly what is supplied by the "Len" output port.
- **(Pos)** "Pos" (position) is the polyphonic Event output for the current sample position in milliseconds. Events at this output port are generated at time intervals corresponding to the current fundamental pitch period.

7.5.3 Example: Grain Pitch Former

This example illustrates how to connect a Pitch Former Module to create a sampler Instrument with independent traversal speed, formant position, and variable (loop) starting position and loop length. As can be seen in the figure below, the Pitch Former Module receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its Gate signal from the Gate Module (↑2.3, Gate). When a key on your computer or MIDI keyboard is pressed, a Gate On signal is sent from the Gate Module, carried by the Polyphonic Voice assigned to the key. As a result sample playback is started for this Voice until the key is released. When the key is released, a Gate Off signal is sent from the Gate Module (↑2.3, Gate) and stops the sample playback (at the "G" (gate) input port. The playback amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled

"Ampl", as shown in the Structure below. Also, because the "Sel" input port is disconnected, the value at the "P" (pitch) input port also determines which sample is loaded from the Sample Map.
The starting position of sample playback, starting position of the loop and loop length can also be controlled using Knob Modules labeled "St", "LS", and "LL", respectively. All three knobs have the range [0 … 1]. This range is multiplied by the length of the currently loaded sample, retrieved from the "Len" (length) output port, and is then forwarded to the corresponding input port. This way the three parameters at the input ports are always in the range of the currently loaded sample.
Knobs have been created for the traversal speed, formant shift, and grain smoothness parameters at the "Sp", "FS", and "Sm" input ports, respectively. Note that you can create controls for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into a Sampler Module's Sample Map.

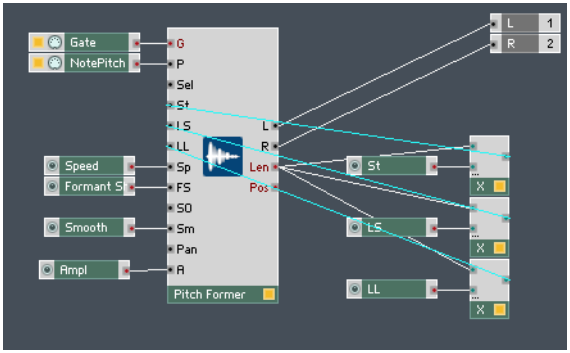


Fig. 7.10 The Structure for a grain pitch former Instrument.

7.6 Grain Cloud Sampler

Fig. 7.11 Grain Cloud Module

7.6.1 Overview

The Grain Cloud Module is a "Grain Resynthesis" Module, as described in subsection 10.2.2 in the Application Reference. The playback speed of the sample is decoupled from the playback pitch. Sample management is carried out in the Sample Map Editor. The Grain Cloud Module is a stereo multi-sample granular synthesizer with independent control over pitch, pitch shift, and sample selection. The envelope of each grain can be controlled at the "Att" (attack) and "Dec" (decay) input ports. Additionally, for each grain the time interval to the start of the next grain can be set at the "Dist" (distance) input port. The maximum number of simultaneous grains can be set in the Module's Function page. There are several "jitter" input ports which set the jitter amount for the respective parameter.

In the same way as the "conventional" Sampler Modules, the Grain Cloud Module uses a pointer at the current sample position. However, the signal at its Audio output ports is not simply the sample amplitude occurring at the pointer position. What actually happens is that the output signal is generated by a synthesizer inside the Module. This synthesizer re-synthesizes the signal at the pointer position. The pitch of the signal generated by this synthesizer is independent of the pointer's speed. Even if the pointer is not moving at all, the synthesizer continues to produce a sound. While the pitch of the output signal is determined, as is usual, over the "P" (pitch) input port, the value at the "Pos" (position) input port determines the pointer location.

Application

The Grain Cloud Module opens up a myriad of possibilities in the realm of granular synthesis. Typical applications for the Grain Cloud Module include granular synthesizers such as the Travelizer and Grainstates Ensembles from the REAKTOR Library.

7.6.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for a Gate signal. A positive Event at this input starts the sample playback from the position that is set at the "St" input port. If a negative value is present at the "G" input port, the loop playback is interrupted unless the Loop in Release feature has been activated in the Sample Map Editor (see subsection 6.4.2 in the Application Reference for more information on this).
- **(P)** "P" (pitch) is the Audio input port for controlling the pitch at which the sample is played back. The "P" (pitch) value is given in the logarithmic pitch scale, in semitones belonging to the range [0 ... 127], where "69" corresponds to "Concert A" or 440 Hz. If the pitch value is equal to the Root Key of the current sample, the sample will be played back at its original pitch. Furthermore, if the "Sel" input port is not connected, "P" also determines the selection of samples from the Sample Map. For the Grain Cloud Module the pitch is independent of the playback speed.
- **(D/F)** "D/F" (direction/frequency) is the Audio input port for direction control if the "P" (pitch) input port is connected. Otherwise this input port control the frequency. The sample plays back in the forward direction and at the original pitch when F = 1. Playback in reverse direction (and at original pitch) occurs when F = -1. The typical range for this input port is [-4 ... 4]. With the input port disconnected, the default value that is used is "1".
- **(PJ)** "PJ" (pitch jitter) is the Audio input port to control the amount of pitch jitter (in semitones). Typical values at this input port are in the range [0 ... 3].
- **(PS)** "PS" (pitch shift) is the Audio input port to control the pitch shift of the current grain. The values at this input port are given in the logarithmic pitch scale, in semitones, and typically fall in the range [-3 ... 3].
- **(Sel)** "Sel" (select) is the Audio input port for selecting a sample from the Sample Map. If this input is not connected, the values present at the "P" (pitch) input port are used instead.

- **(Pos)** "Pos" (position) is the Audio input port for setting the position in the sample from which the current grain is generated. The units for this input port are milliseconds and naturally fall in the range [0...<length of sample in ms>].
- **(PsJ)** "PsJ" (position jitter) is the Audio input port to control the jitter in the "Pos" (position) parameter. The units of this input port are milliseconds and the typical range is [0...<length of sample in ms>].
- **(Len)** "Len" (length) is the Audio input port for setting the grain length in milliseconds. The typical range for this parameter is [10 ... 100]. The default value that is used in the case of a disconnected "Len" input port is 20 milliseconds.
- **(LnJ)** "LnJ" (length jitter) is the Audio input port for the amount of jitter in the grain length parameter. The units of this input port are in milliseconds and these values typically fall in the range [10 ... 100]. The default value in the case of a disconnected "LnJ" input port is 0 milliseconds.
- **(Att)** "Att" (attack) is the Audio input port for setting the attack time of the grain envelope. The range of this input port is [0 ... 1]. The default value in the case that this input port is disconnected is "0.2".
- **(Dec)** "Dec" (decay) is the Audio input port for setting the decay time of the grain envelope. The range for this input port is [0 ... 1]. The default value in the case that this input port is disconnected is "0.2".
- **(Dist)** "Dist" (distance) is the Audio input port for setting the time duration before the next grain is started. The values at this input port are given in milliseconds and typically fall in the range [5 ... 100]. If this input port is disconnected, the default value that is used is "20".
- **(DisJ)** "DisJ" (distance jitter) is the Audio input port for the amount of jitter in the distance parameter. The units of the values at this input port are milliseconds and typically fall in the range [10 ... 100]. If this input port is disconnected, the default value that is used is 20 milliseconds.
- **(Pan)** "Pan" is the Audio input port to control the position of the output signal in the stereo field (-1 = Left, 0 = Center, 1 = Right).
- **(PnJ)** "PnJ" (pan jitter) is the Audio input port for the amount of jitter in the pan parameter. The range of the values at this input port is [0 ... 1]. The default value used in case of this input port being disconnected is "0".
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude. Since this is an Audio input port, it can be used for ring modulation. The default value to be used when this input port is disconnected is "1".

Output Ports

- **(L)** "L" (left) is the Audio output port for the left stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "R" output port.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "L" output port.
- **(Len)** "Len" (length) is the polyphonic Event output port for the length of the current sample in milliseconds. Use this value to calculate parameters which depend on the sample length. The sample starting point (specified at the "St" input port), for example, is a parameter that should have a range from "0" to the length of the sample in milliseconds. The latter value is exactly what is supplied by the "Len" output port.
- **(GTr)** "GTr" (grain trigger) is the Event output port for grain trigger signals. An Event with the value "1" is sent when a new grain starts, an Event with the value "0" is sent when the grain stops.

7.6.3 Example: Simple Grain Cloud Sampler

This example illustrates how to connect a Grain Cloud Module using the Ramp Module as a pointer to create a sampler Instrument with grain parameters which are independent from the traversal speed and position. As can be seen in the figure below, the Grain Cloud Module receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and the Gate signal from the Gate Module (↑2.3, Gate). When a key on your computer or MIDI keyboard is pressed, a Gate On signal is sent from the Gate Module, carried by the Polyphonic Voice assigned to the key. As a result sample playback is started for this Voice until the key is released. When the key is released, a Gate Off signal is sent from the Gate Module (↑2.3, Gate) and stops the sample playback (at the "G" (gate) input port. The playback amplitude can be controlled from the Instrument Panel using a Knob Module (↑1.1, Fader/Knob) labeled "Amplitude", as shown in the Structure below. Also, because the "Sel" input port is disconnected, the value at the "P" (pitch) input port also determines which sample is loaded from the Sample Map.

The pointer position in the loaded sample at which the played-back grain is synthesized, is determined by the value at the "Pos" input port. To have a pointer that linearly loops from the beginning of the sample to the end, you need a linear ramp signal in with the range [0

... <length of current sample in ms>]. The sample length is provided by the "Len" (length) output port and can therefore be used as the amplitude for the ramp generator, the Ramp Module (↑6.36, Ramp Oscillator) (see the Structure below).

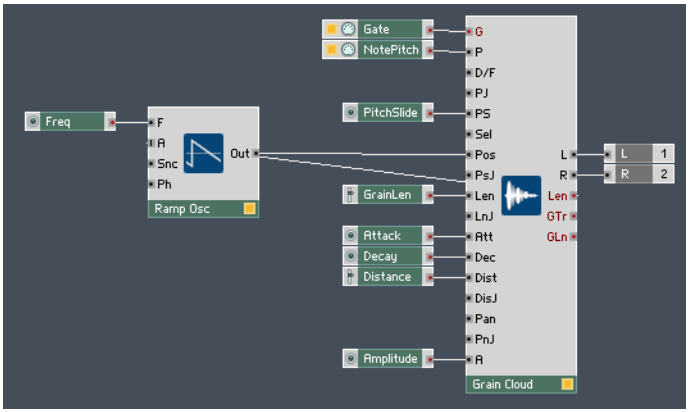


Fig. 7.12 The Structure for a simple grain cloud sampler Instrument.

Knobs have been created for the pitch slide, grain length, grain attack, grain decay, and grain distance parameters at the "PS", "Len", "Att", "Dec", and "Dist" input ports, respectively. Note that you can create controls for an input port with the right range and resolution simply by right-clicking the input port and choosing the *Create Control* menu entry.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into a Sampler Module's Sample Map.

7.7 Beat Loop

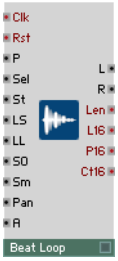


Fig. 7.13 Beat Loop Module

7.7.1 Overview

The Beat Loop Module is a "Grain Resynthesis" Module, as described in subsection 10.2.2 in the Application Reference. The transposing of beat-loops can be set via the "P" (pitch) input port independent of playback speed. Sample management is carried out in the Sample Map Editor. The Beat Loop Module is a real-time resynthesizer for the synchronized playback of beat-loop samples. Beat Loop synchronizes itself to a 96th note clock source (24 ppq) that is connected to the "Clk" (clock) input port or, if the "Clk" input port is not connected, it can sync with the global REAKTOR clock. Therefore, by default all Beat Loops in REAKTOR run in sync with one another independent of the sample's internal speed. Furthermore, Beat Loop also allows you to easily link internal REAKTOR sequencer modules and MIDI clocks to rhythmic sample material.

The Beat Loop Module requires samples that have been cut exactly and which contain an integer number beats. The speed of the beat loops used should be between 87 and 174 BPM. If the samples being used fulfill these conditions, the Beat Loop Module can output them in good quality even if the playback speed is changed.

With the Beat Loop Module you can directly configure the loop range of the sample with the "St", "LS", and "LL" input ports. These input ports control the starting point of playback, loop starting point and loop length, respectively. Sending a positive valued Event to the "Rst" input port causes the sample playback to be continued from the starting point (set at the "St" input port).

If the Loop checkbox in the Sampler Map editor is activated for a given sample, the sample will be continuously repeated within the loop range as soon as playback enters this range. The loop range is preset using data from the sound file from which the sample was loaded. If the sound file does not contain any loop data, the beginning of the sample is taken to be the beginning of the loop and the sample length is taken as the loop length. The loop data from the sound file are the default settings. These defaults are always used if the "LS" (loop start) and "LL" (loop length) input ports are not connected to other Modules. If the Loop in Release feature is deactivated, loop playback will fade out upon a Gate Off Event occurring; depending on the direction set, the sample will run to its beginning or end and will then fade out. If the Loop in Release feature is activated or if loop playback is switched off, the Gate Events will only have a slight effect or none at all.

All of the Beat Loop Module's input port, except "Clk" and "Rst", are designed as Audio input ports. During sample playback, changes to values take effect at intervals of sixteenths of a note value.

Application

As the name implies, the Beat Loop Module is ideal to playback rhythmic loop samples that are quantized to a grid. With this Module you could, for example, build a simple Ensemble that slices a drum loop and lets you play back these slices in a mixed-up order, in sync with the REAKTOR Clock.

7.7.2 Ports

Input Ports

- **(Clk)** "Clk" (clock) is the Event input port for clock signals. A clock signal is considered to be a positive Event at this input port. One such positive Event advances the Beat Loop Module's playback position by one 96th note value. If this input is not connected, the Module synchronizes itself with the global REAKTOR clock.
- **(Rst)** "Rst" (reset) is the Event input port for reset signals. A reset signal is considered to be a positive Event at this input port. One such positive Event resets playback to the position set at the "St" (start) input port.
- **(P)** "P" (pitch) is the Audio input port for controlling the playback rate (pitch) at which the sample is played back. The "P" (pitch) value is given in the logarithmic pitch scale, in semitones belonging to the range [0 … 127] where "69" corresponds to "Concert A" or 440 Hz. If the pitch value is equal to the Root Key of the current sample, the sample will be played back at its original pitch. Furthermore, if the "Sel" input port is not connected, "P" also determines the selection of samples from the Sample Map.
- **(Sel)** "Sel" (select) is the Audio input port for selecting a sample from the Sample Map. If this input is not connected, the values present at the "P" (pitch) input port are used instead.
- **(St) "St" (start)** is the Audio input port for the starting point to be used when the next clock signal is received (either at the "Clk" input port of from the REAKTOR Clock). The position is set in sixteenths of a note value from the start of the sample. . The length of the loaded sample in sixteenth notes is delivered by the "L16" output port. This means that although the samples in your Sample Map might have different lengths, you can use the value from the "L16" output port and multiply it by a parameter in the range [0 … 1]. The result will yield you values in the full range of the current sample, in sixteenth notes, which you then feed to the "St" input port. If this input port is left unconnected, then the default value "0" is used.

- **(LS)** "LS" (loop start) is the Audio input port for the loop starting point in sixteenths of a note value, counted from the start of the sample. If this input port is not connected, the default value that is used is LS = 0.
- **(LL)** "LL" (loop length) is the Audio input port for the loop length in sixteenths of a note value. If this input is not connected, the default value that is used is LL = length of the whole sample.
- **(SO)** "SO" (sample offset) is the Audio input port for modulation of the sample position (sample offset) in sixteenths of a note value. This input is used to reach steps in the sample deviating from the progression of clock signals. This input port, for instance, can be controlled by a sequencer module (↑8.1, 6-Step Sequencer).
- **(Sm)** "Sm" (smoothing) is the Audio input port to control the Smoothing parameter of the resynthesis process. The sound particles are adjusted using this. Very small values generally lead to a “crackling” sound at every sixteenth note interval. The range for the values at this input port is [0 … 1]. The default value to be used when this input port is disconnected is "0.1".
- **(Pan)** "Pan" is the Audio input port to control the position of the output signal in the stereo field (-1 = Left, 0 = Center, 1 = Right).
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude. Since this is an Audio input port, it can be used for ring modulation. The default value to be used when this input port is disconnected is "1".

Output ports

- **(L)** "L" (left) is the Audio output port for the left stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "R" output port.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the sample player. When mono samples are being processed or if the No Stereo checkbox in the Module's Function page has been engaged, the same signal is present here as at the "L" output port.
- **(Len)** "Len" (length) is the polyphonic Event output port for the length of the current sample in milliseconds. Use this value to calculate parameters which depend on the sample length. The sample starting point (specified at the "St" input port), for example, is a parameter that should have a range from "0" to the length of the sample in milliseconds. The latter value is exactly what is supplied by the "Len" output port.

- **(L16)** "L16" (length, sixteenths) is the polyphonic Event output port for the length of the current sample counted in sixteenths notes.
- **(P16)** "P16" (position, sixteenths) is the polyphonic Event output port for the current position in the sample counted in sixteenths notes.
- **(Ct16)** "Ct16" is the polyphonic Event output port for counting the sixteenths of a note value that have passed since the last start / reset Event.

7.7.3 Example: Simple Beat Loop Sampler

This example illustrates how to connect a Beat Loop Module to create a beat loop sampler Instrument with pitch-independent playback speed, and variable (loop) starting position and loop length. As can be seen in the figure below, the Beat Loop Module receives its pitch input from the Note Pitch Module (↑2.1, Note Pitch) and its playback is advanced by Events received from the Clock Module (↑2.14, Clock).

The MIDI Clock needs to be running in order for the Clock Module to send Events. Use the Router Module and a toggle button labeled "Run" to turn the playback of the Beat Loop Module on and off. Additionally, use a knob labeled "Amplitude" to control the playback amplitude. Because the "Sel" input port is disconnected, the value at the "P" (pitch) input port also determines which sample is loaded from the Sample Map. The knob at the "Sm" input port, labeled "Smooth", controls the granularity of the signal outgoing from the Beat Loop Module.

The starting position of sample playback, starting position of the loop and loop length can be controlled using Knob Modules (↑1.1, Fader/Knob) labeled "St", "LS", and "LL", respectively. All three knobs have the range [0 … 1]. This range is multiplied by the length of the currently loaded sample, retrieved from the "Len" (length) output port, and is then forwarded to the corresponding input port. This way the three parameters at the input ports are always in the range of the currently loaded sample.

Please refer to subsection 6.2.1 in the Application Reference for more information on loading samples into a Sampler Module's Sample Map.

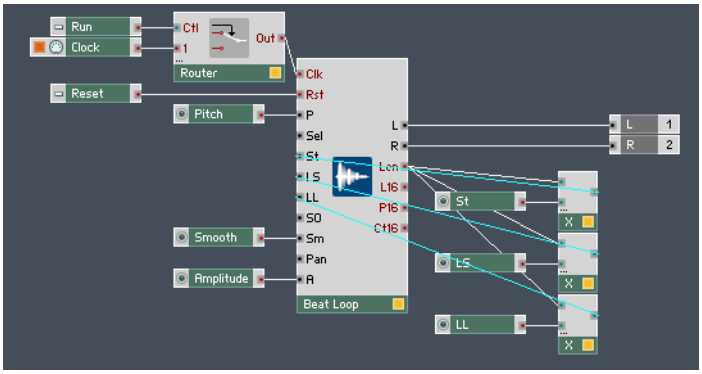


Fig. 7.14 The Structure for a beat loop sampler Instrument.

7.8 Sample Lookup

Fig. 7.15 Lookup Module

7.8.1 Overview

The Lookup Module makes samples available as function value look-up tables. A position within the sample in milliseconds is set via the "Pos" (position) input port. The value of the sample at this point is then sent to the output ports.
You can load sound files from the Instrument Panel by right-clicking the Module's Panel representation and choosing the *Load Sound...* menu entry. Sample management is carried out in the Sample Map Editor.

Application

Use the Sample Lookup Module as a replacement for a traditional oscillator. In that case you could connect the "Pos" input port to the output of a Ramp Module and use the "Len" (length) output port to normalize the amplitude (and its offset) of the Ramp Module.

7.8.2 Ports

Input Ports

- **(Pos)** is the Audio input port for controlling the playback position in the sample, in milliseconds. This input is usually connected to a Ramp Module, creating a Sample Lookup oscillator. Use the value from the "Len" (length) output port to calculate the desired position in the sample in milliseconds.
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude. Since this is an Audio input port, it can be used for ring modulation.

Output Ports

- **(L)** "L" (left) is the Audio output port for the left stereo channel of the sample player. When mono samples are being processed the same signal is present here as at the "R" output port.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the sample player. When mono samples are being processed the same signal is present here as at the "L" output port.
- **(Len)** "Len" (length) is the polyphonic Event output port for the length of the current sample in milliseconds. Use this value to calculate parameters which depend on the sample length.

7.8.3 Example: Sample Lookup Oscillator

This example shows how to build a Structure for a sample lookup oscillator. First, add a Lookup Module into the Structure and load the desired sample from the Lookup Module's Function page by clicking on the folder icon (shown in the figure below).

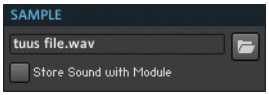


Fig. 7.16 Click on the folder icon the Lookup Module's Function page to load a sample.

Let's use the Sine Module (↑6.14, Sine Oscillator) as the pointer signal for the sample lookup position. Since we are using an oscillator for the lookup position, the outgoing signal will in effect also be an oscillator, but its waveform will in part be determined by the sample loaded into the Lookup Module. To choose the position in the sample around

which the pointer signal (stemming from the Sine Module) oscillates, add an "offset" value to the Sine Module's (↑6.14, Sine Oscillator) output signal. This "offset" value should be within the sample's range. Therefore the "offset" value is derived by multiplying the "Len" output port with a knob (with range [0 … 1]) and then added to the Sine Module's output signal.
The oscillator pitch is received from the Note Pitch Module (↑2.1, Note Pitch) and an envelope signal is applied to the "A" (amplitude) input port of the Lookup Module to have the Structure react to incoming MIDI Notes much like the oscillator-envelope combination of a conventional synthesizer. For the envelope, the ADSR Module (↑9.13, ADSR - Env) has been used. The Gate signal is received from the Gate Module (↑2.3, Gate).

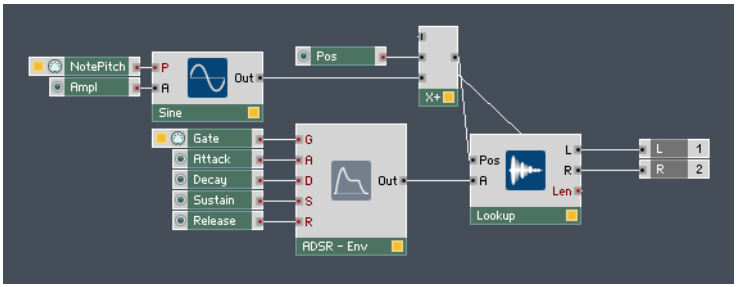


Fig. 7.17 The Structure for a simple sample lookup oscillator.

8 Sequencer

REAKTOR's sequencers include gated step-sequencers in four sizes as well as a selectable-position sequencer. You'll also find clocks to drive them here.
Sequencer Sequencer

8.1 6-Step Sequencer

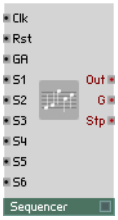


Fig. 8.1 6-Step Sequencer

8.1.1 Overview

The 6-Step Sequencer Module (↑8.1, 6-Step Sequencer) embodies a sequencer with 6 steps. The signal at the "Out" output port cycles through the sequencer step values which are set at the input ports labeled "S1", "S2", etc. These step values can be set independently for each step. When the sequencer is running, a Gate signal Event is output at the time the step advances. This Event is sent from the "G" (gate) output port with its value being specified at the "GA" (gate amplitude) input port.

Application

The sequencer is a key element in many Ensembles. It allows you to create repeating rhythmic and melodic patterns. For example, you could create a 6-step drum pattern using the 6-Step Sequencer Module by using it as a trigger for a Sampler Module that carries the desired sample. The step values you could use to carry note velocity information. Alternatively you could connect the 6-Step sequencer to the "P" (pitch) input port of an oscillator to have the oscillator playback a melodic sequence determined by the values at the input ports "S1", "S2", etc.

8.1.2 Ports

Input Ports

- **(Clk)** "Clk" (clock) is the Audio input port for the clock signal. A positive zero crossing causes the Module to advance to the next step. Typically you would connect a Pulse Oscillator or Sync Pulse Module (↑2.15, Sync Pulse) here.
- **(Rst)** "Rst" (reset) is the Audio input for port a reset signal. A positive zero crossing switches the sequencer back to the first step. Typically you would connect a Button or MIDI Start Module here.
- **(GA)** "GA" (gate amplitude) is the Audio input port for controlling the amplitude of the "G" (gate) output Events. When the value is zero or the input is not connected, no signal appears on the "G" (gate) output.
- **(S1)** "S1" (step 1) is the Audio input port for controlling the value of the first sequencer step.
- **(S2)** "S2" (step 2) is the Audio input port for controlling the value of the second sequencer step.
- **(S3)** "S3" (step 3) is the Audio input port for controlling the value of the third sequencer step.
- **(S4)** "S4" (step 4) is the Audio input for port controlling the value of the fourth sequencer step.
- **(S5)** "S5" (step 5) is the Audio input port for controlling the value of the fifth sequencer step.
- **(S6)** "S6" (step 6) is the Audio input port for controlling the value of the sixth sequencer step.

Output Ports

- **(Out)** "Out" is the Event output for the sequencer step signal. Its range is determined by the values at the input ports "S1", "S2", etc.
- **(G)** "G" (gate) is the Event output port for the Gate signal. Every time the sequencer advances one step, an Event with the value specified at the "GA" (gate amplitude) input port is sent from this output port. Therefore the range of this output port is determined by the values at the "GA" (gate amplitude) input port.
- **(St)** "St" (step) is the Event output port for the current step number. The range of this output port is [1 … 6].

8.1.3 Example: 6-Step Pitch Sequencer

This example shows how to use a 6-Step Sequencer Module to create a sequencer of pitch values along with a Gate signal of uniform velocity. To start, insert a 6-Step Sequencer Module and a Sync Clock Module (↑2.15, Sync Pulse) into the Structure, as shown in the figure below. For the Sequencer Module's input ports from "S1" to "S6", create controls simply by right-clicking the input ports and choosing the *Create Control* menu entry. Knobs with the standard MIDI range and resolution are created, perfectly suitable for sequencing pitch values to an oscillator, for example. Next, use the *Create Control* menu entry to create a Button Module (↑1.2, Button) at the "Rst" (reset) input port. This enables you to reset the sequencer position. Insert an On Velocity Module (↑2.6, On Velocity), set it to Monophonic Mode, and connect it to the "GA" (gate amplitude) input port. This lets you set the velocity of Gate On signals (outgoing at the "G" output port) to the velocity with which the last MIDI or computer keyboard key was pressed.

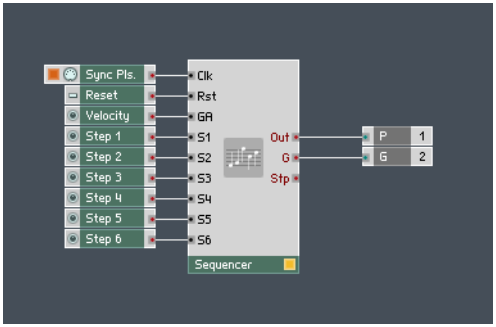


Fig. 8.2 The Structure for a 6-step pitch value sequencer.

Next, connect the Sync Pulse Module (↑2.15, Sync Pulse) to the "Clk" (clock) input port of the Sequencer Module (↑8.1, 6-Step Sequencer). Go to the Sync Pulse Module's Function page and choose the rate at which the sequencer runs from the Rate drop-down menu, shown in the figure below. The duration in this case does not make a difference. The rate will be synchronized to the MIDI Clock and clock Events will only be output to the Sequencer Module if the MIDI Clock is running. The setup shown in this example has the following functionality. When the MIDI Clock is running, the Sequencer Module will go through each step at the rate determined by the Sync Pulse Module (↑2.15, Sync Pulse).

At each step, the "Out" output port will send an Event carrying the value as determined by the knob connected to the input port of the corresponding step, as well as a Gate On signal that carries the velocity value of the last pressed MIDI or computer keyboard key.

Note that you could also use the Clock Module (↑2.14, Clock) and to get the desired rate using the Event Frequency Divider Module (↑13.4, Event Frequency Divider). The advantage of that method is that you can choose the frequency from the Instrument Panel.

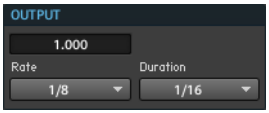


Fig. 8.3 Use the Rate drop-down menu in the Sync Pulse Module's Function page to choose the rate of clock Events sent to the Sequencer Module.

8.2 8-Step Sequencer

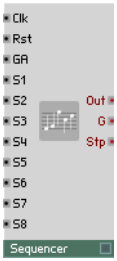


Fig. 8.4 8-Step Sequencer Module

The 8-Step Sequencer Module embodies a sequencer with 8 steps. The signal at the "Out" output port cycles through the sequencer step values which are set at the input ports labeled "S1", "S2", etc. These step values can be set independently for each step. When the sequencer is running, a Gate signal Event is output at the time the step advances. This Event is sent from the "G" (gate) output port with its value being specified at the "GA" (gate amplitude) input port.

Application

The sequencer is a key element in many Ensembles. It allows you to create repeating rhythmic and melodic patterns. For example, you could create an 8-step drum pattern using the 8-Step Sequencer Module by using it as a trigger for a Sampler Module that carries

the desired sample. The step values you could use to carry note velocity information. Alternatively you could connect the 8-Step sequencer to the "P" (pitch) input port of an oscillator to have the oscillator playback a melodic sequence determined by the values at the input ports "S1", "S2", etc.

8.2.1 Ports

Input Ports

- **(Clk)** "Clk" (clock) is the Audio input port for the clock signal. A positive zero crossing causes the Module to advance to the next step. Typically you would connect a Pulse Oscillator or Sync Pulse Module (↑2.15, Sync Pulse) here.
- **(Rst)** "Rst" (reset) is the Audio input for port a reset signal. A positive zero crossing switches the sequencer back to the first step. Typically you would connect a Button or MIDI Start Module here.
- **(GA)** "GA" (gate amplitude) is the Audio input port for controlling the amplitude of the "G" (gate) output Events. When the value is zero or the input is not connected, no signal appears on the "G" (gate) output.
- **(S1)** "S1" (step 1) is the Audio input port for controlling the value of the first sequencer step.
- **(S2)** "S2" (step 2) is the Audio input port for controlling the value of the second sequencer step.
- **(S3)** "S3" (step 3) is the Audio input port for controlling the value of the third sequencer step.
- **(S4)** "S4" (step 4) is the Audio input for port controlling the value of the fourth sequencer step.
- **(S5)** "S5" (step 5) is the Audio input port for controlling the value of the fifth sequencer step.
- **(S6)** "S6" (step 6) is the Audio input port for controlling the value of the sixth sequencer step.
- **(S7)** "S7" (step 7) is the Audio input port for controlling the value of the seventh sequencer step.
- **(S8)** "S8" (step 8) is the Audio input port for controlling the value of the eight sequencer step.

Output Ports

- **(Out)** "Out" is the Event output for the sequencer step signal. Its range is determined by the values at the input ports "S1", "S2", etc.
- **(G)** "G" (gate) is the Event output port for the Gate signal. Every time the sequencer advances one step, an Event with the value specified at the "GA" (gate amplitude) input port is sent from this output port. Therefore the range of this output port is determined by the values at the "GA" (gate amplitude) input port.
- **(St)** "St" (step) is the Event output port for the current step number. The range of this output port is [1 … 8].

8.2.2 Example: 8-Step Pitch Sequencer

This example shows how to use a 8-Step Sequencer Module to create a sequencer of pitch values along with a Gate signal of uniform velocity. To start, insert a 8-Step Sequencer Module and a Sync Clock Module (↑2.15, Sync Pulse) into the Structure, as shown in the figure below. For the Sequencer Module's input ports from "S1" to "S8", create controls simply by right-clicking the input ports and choosing the *Create Control* menu entry. Knobs with the standard MIDI range and resolution are created, perfectly suitable for sequencing pitch values to an oscillator, for example. Next, use the *Create Control* menu entry to create a Knob Module (↑1.1, Fader/Knob) at the "GA" (gate amplitude) input port and a Button Module at the "Rst" (reset) input port. The former control lets you set the velocity of the outgoing Gate signals at the "G" (gate) output port and the latter enables you to reset the sequencer position.

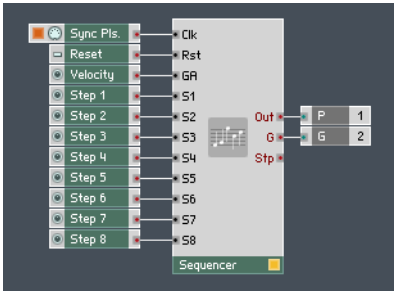


Fig. 8.5 The Structure for a 8-step pitch value sequencer.

Next, connect the Sync Pulse Module (↑2.15, Sync Pulse) to the "Clk" (clock) input port of the Sequencer Module. Go to the Sync Pulse Module's Function page and choose the rate at which the sequencer runs from the Rate drop-down menu, shown in the figure below. The duration in this case does not make a difference. The rate will be synchronized to the MIDI Clock and clock Events will only be output to the Sequencer Module if the MIDI Clock is running. The setup shown in this example has the following functionality. When the MIDI Clock is running, the Sequencer Module will go through each step at the rate determined by the Sync Pulse Module (↑2.15, Sync Pulse). At each step, the "Out" output port will send an Event carrying the value as determined by the knob connected to the input port of the corresponding step, as well as a Gate On signal that carries the velocity value set by the "Velocity" knob at the "GA" (gate amplitude) input port.

Note that you could also use the Clock Module (↑2.14, Clock) and to get the desired rate using the Event Frequency Divider Module (↑13.4, Event Frequency Divider). The advantage of that method is that you can choose the frequency from the Instrument Panel.

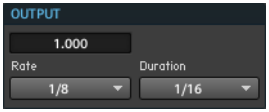


Fig. 8.6 Use the Rate drop-down menu in the Sync Pulse Module's Function page to choose the rate of clock Events sent to the Sequencer Module.

8.3 12-Step Sequencer

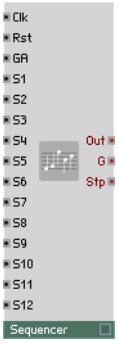


Fig. 8.7 12-Step Sequencer Module

The 12-Step Sequencer Module embodies a sequencer with 12 steps. The signal at the "Out" output port cycles through the sequencer step values which are set at the input ports labeled "S1", "S2", etc. These step values can be set independently for each step. When the sequencer is running, a Gate signal Event is output at the time the step advances. This Event is sent from the "G" (gate) output port with its value being specified at the "GA" (gate amplitude) input port.

Application

The sequencer is a key element in many Ensembles. It allows you to create repeating rhythmic and melodic patterns. For example, you could create a 12-step drum pattern using the 12-Step Sequencer Module by using it as a trigger for a Sampler Module that carries the desired sample. The step values you could use to carry note velocity information. Alternatively you could connect the 12-Step sequencer to the "P" (pitch) input port of an oscillator to have the oscillator playback a melodic sequence determined by the values at the input ports "S1", "S2", etc.

8.3.1 Ports

Input Ports

- **(Clk)** "Clk" (clock) is the Audio input port for the clock signal. A positive zero crossing causes the Module to advance to the next step. Typically you would connect a Pulse Oscillator or Sync Pulse Module (↑2.15, Sync Pulse) here.
- **(Rst)** "Rst" (reset) is the Audio input for port a reset signal. A positive zero crossing switches the sequencer back to the first step. Typically you would connect a Button or MIDI Start Module here.
- **(GA)** "GA" (gate amplitude) is the Audio input port for controlling the amplitude of the "G" (gate) output Events. When the value is zero or the input is not connected, no signal appears on the "G" (gate) output.
- **(S1)** "S1" (step 1) is the Audio input port for controlling the value of the first sequencer step.
- **(S2)** "S2" (step 2) is the Audio input port for controlling the value of the second sequencer step.
- **(S3)** "S3" (step 3) is the Audio input port for controlling the value of the third sequencer step.

- **(S4)** "S4" (step 4) is the Audio input for port controlling the value of the fourth sequencer step.
- **(S5)** "S5" (step 5) is the Audio input port for controlling the value of the fifth sequencer step.
- **(S6)** "S6" (step 6) is the Audio input port for controlling the value of the sixth sequencer step.
- **(S7)** "S7" (step 7) is the Audio input port for controlling the value of the seventh sequencer step.
- **(S8)** "S8" (step 8) is the Audio input port for controlling the value of the eight sequencer step.
- **(S9)** "S9" (step 9) is the Audio input port for controlling the value of the ninth sequencer step, and so on…

Output Ports

- **(Out)** "Out" is the Event output for the sequencer step signal. Its range is determined by the values at the input ports "S1", "S2", etc.
- **(G)** "G" (gate) is the Event output port for the Gate signal. Every time the sequencer advances one step, an Event with the value specified at the "GA" (gate amplitude) input port is sent from this output port. Therefore the range of this output port is determined by the values at the "GA" (gate amplitude) input port.
- **(St)** "St" (step) is the Event output port for the current step number. The range of this output port is [1 … 12].

8.3.2 Example: 12-Step Pitch Sequencer

This example shows how to use a 6-Step Sequencer Module to create a sequencer of pitch values along with a Gate signal of uniform velocity. To start, insert a 12-Step Sequencer Module and a Sync Clock Module (↑2.15, Sync Pulse) into the Structure, as shown in the figure below. For the Sequencer Module's input ports from "S1" to "S12", create controls simply by right-clicking the input ports and choosing the *Create Control* menu entry. Knobs with the standard MIDI range and resolution are created, perfectly suitable for sequencing pitch values to an oscillator, for example. Next, use the *Create Control* menu entry to create a Knob Module (↑1.1, Fader/Knob) at the "GA" (gate amplitude) input port and a Button Module at the "Rst" (reset) input port. The former control lets you set the velocity of the outgoing Gate signals at the "G" (gate) output port and the latter enables you to reset the sequencer position.

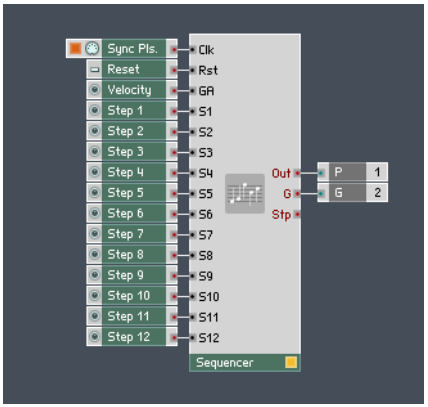


Fig. 8.8 The Structure for a 12-step pitch value sequencer.

Next, connect the Sync Pulse Module (↑2.15, Sync Pulse) to the "Clk" (clock) input port of the Sequencer Module. Go to the Sync Pulse Module's Function page and choose the rate at which the sequencer runs from the Rate drop-down menu, shown in the figure below. The duration in this case does not make a difference. The rate will be synchronized to the MIDI Clock and clock Events will only be output to the Sequencer Module if the MIDI Clock is running. The setup shown in this example has the following functionality. When the MIDI Clock is running, the Sequencer Module will go through each step at the rate determined by the Sync Pulse Module (↑2.15, Sync Pulse). At each step, the "Out" output port will send an Event carrying the value as determined by the knob connected to the input port of the corresponding step, as well as a Gate On signal that carries the velocity value set by the "Velocity" knob at the "GA" (gate amplitude) input port.

Note that you could also use the Clock Module (↑2.14, Clock) and to get the desired rate using the Event Frequency Divider Module (↑13.4, Event Frequency Divider). The advantage of that method is that you can choose the frequency from the Instrument Panel.

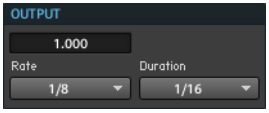


Fig. 8.9 Use the Rate drop-down menu in the Sync Pulse Module's Function page to choose the rate of clock Events sent to the Sequencer Module.

8.4 16-Step Sequencer

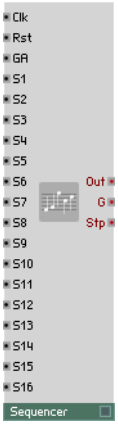


Fig. 8.10 16-Step Sequencer

The 16-Step Sequencer Module embodies a sequencer with 16 steps. The signal at the "Out" output port cycles through the sequencer step values which are set at the input ports labeled "S1", "S2", etc. These step values can be set independently for each step. When the sequencer is running, a Gate signal Event is output at the time the step advances. This Event is sent from the "G" (gate) output port with its value being specified at the "GA" (gate amplitude) input port.

Application

The sequencer is a key element in many Ensembles. It allows you to create repeating rhythmic and melodic patterns. For example, you could create a 16-step drum pattern using the 16-Step Sequencer Module by using it as a trigger for a Sampler Module that carries the desired sample. The step values you could use to carry note velocity information. Alternatively you could connect the 16-Step sequencer to the "P" (pitch) input port of an oscillator to have the oscillator playback a melodic sequence determined by the values at the input ports "S1", "S2", etc.

8.4.1 Ports

Input Ports

- **(Clk)** "Clk" (clock) is the Audio input port for the clock signal. A positive zero crossing causes the Module to advance to the next step. Typically you would connect a Pulse Oscillator or Sync Pulse Module (↑2.15, Sync Pulse) here.
- **(Rst)** "Rst" (reset) is the Audio input for port a reset signal. A positive zero crossing switches the sequencer back to the first step. Typically you would connect a Button or MIDI Start Module here.
- **(GA)** "GA" (gate amplitude) is the Audio input port for controlling the amplitude of the "G" (gate) output Events. When the value is zero or the input is not connected, no signal appears on the "G" (gate) output.
- **(S1)** "S1" (step 1) is the Audio input port for controlling the value of the first sequencer step.
- **(S2)** "S2" (step 2) is the Audio input port for controlling the value of the second sequencer step.
- **(S3)** "S3" (step 3) is the Audio input port for controlling the value of the third sequencer step.
- **(S4)** "S4" (step 4) is the Audio input for port controlling the value of the fourth sequencer step.
- **(S5)** "S5" (step 5) is the Audio input port for controlling the value of the fifth sequencer step.
- **(S6)** "S6" (step 6) is the Audio input port for controlling the value of the sixth sequencer step.
- **(S7)** "S7" (step 7) is the Audio input port for controlling the value of the seventh sequencer step.
- **(S8)** "S8" (step 8) is the Audio input port for controlling the value of the eight sequencer step.
- **(S9)** "S9" (step 9) is the Audio input port for controlling the value of the ninth sequencer step, and so on…

Output Ports

- **(Out)** "Out" is the Event output for the sequencer step signal. Its range is determined by the values at the input ports "S1", "S2", etc.

- **(G)** "G" (gate) is the Event output port for the Gate signal. Every time the sequencer advances one step, an Event with the value specified at the "GA" (gate amplitude) input port is sent from this output port. Therefore the range of this output port is determined by the values at the "GA" (gate amplitude) input port.
- **(St)** "St" (step) is the Event output port for the current step number. The range of this output port is [1 ... 16].

8.4.2 Example: 16-Step Pitch Sequencer

This example shows how to use a 16-Step Sequencer Module to create a sequencer of pitch values along with a Gate signal of uniform velocity. To start, insert a 16-Step Sequencer Module and a Sync Clock Module (↑2.15, Sync Pulse) into the Structure, as shown in the figure below. For the Sequencer Module's input ports from "S1" to "S16", create controls simply by right-clicking the input ports and choosing the *Create Control* menu entry. Knobs with the standard MIDI range and resolution are created, perfectly suitable for sequencing pitch values to an oscillator, for example. Next, use the *Create Control* menu entry to create a Knob Module (↑1.1, Fader/Knob) at the "GA" (gate amplitude) input port and a Button Module at the "Rst" (reset) input port. The former control lets you set the velocity of the outgoing Gate signals at the "G" (gate) output port and the latter enables you to reset the sequencer position.

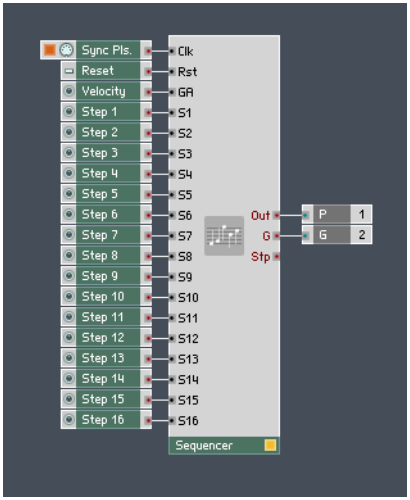


Fig. 8.11 The Structure for a 16-step pitch value sequencer.

Next, connect the Sync Pulse Module (↑2.15, Sync Pulse) to the "Clk" (clock) input port of the Sequencer Module. Go to the Sync Pulse Module's Function page and choose the rate at which the sequencer runs from the Rate drop-down menu, shown in the figure below. The duration in this case does not make a difference. The rate will be synchronized to the MIDI Clock and clock Events will only be output to the Sequencer Module if the MIDI Clock is running. The setup shown in this example has the following functionality. When the MIDI Clock is running, the Sequencer Module will go through each step at the rate determined by the Sync Pulse Module (↑2.15, Sync Pulse). At each step, the "Out" output port will send an Event carrying the value as determined by the knob connected to the input port of the corresponding step, as well as a Gate On signal that carries the velocity value set by the "Velocity" knob at the "GA" (gate amplitude) input port.

Note that you could also use the Clock Module (↑2.14, Clock) and to get the desired rate using the Event Frequency Divider Module (↑13.4, Event Frequency Divider). The advantage of that method is that you can choose the frequency from the Instrument Panel.

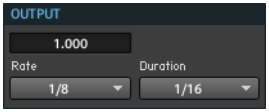


Fig. 8.12 Use the Rate drop-down menu in the Sync Pulse Module's Function page to choose the rate of clock Events sent to the Sequencer Module.

8.5 Multiplex 16 Sequencer

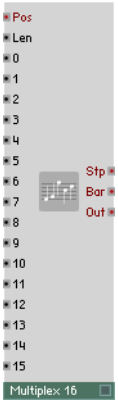


Fig. 8.13 Multiplex 16 Module

8.5.1 Overview

The Multiplex 16 Module is in essence a value selector that can also be used as a sequencer. The value at the "Pos" (position) input port addresses one of the 16 input ports and the current value at this input port is then forwarded to the "Out" output port. The values at the "Pos" input port are wrapped into the number range [0 ... (Len - 1)] to allow a variable sequence length, specified with the "Len" (length) input port.

Application

If the "Pos" (position) input port is driven by a signal which is incremented stepwise, the "Out" output port provides a sequence of the values at the inputs "0" to "1 - Len". The "Pos" input port can also be driven by a control element, a Beat Loop module, a random Event source, or by the REAKTOR Clock.

8.5.2 Ports

Input Ports

- **(Pos)** "Pos" (position) is the Event input port for addressing the input port whose value is forwarded to the "Out" output port. Each Event at this input results in an Event at all output ports. When running the Multiplex 16 Module as a sequencer, the "Pos" (position) value is set to the number of steps (for example, sixteenth notes) that have passed since the start or last reset Event.
- **(Len)** "Len" (length) Input to set the sequence length. Values at the "Pos" (position) input port will then be wrapped at "Len - 1". The range for the values at the "Len" (length) input port is [1 ... 16].
- **(0) "0"** is the Audio input port for controlling the value of the zeroth step. It is forwarded to the "Out" output port if Pos = 0 or a corresponding "wrapped" value.
- **(1) "1"** is the Audio input port for controlling the value of the first step. It is forwarded to the "Out" output port if Pos = 1 or a corresponding "wrapped" value.
- **(2) "2"** is the Audio input port for controlling the value of the second step, and so on…

Output Ports

- **(Stp)** "Stp" (step) is the Event output port for the current sequence step number. The number at this output is calculated as Pos modulo Len (Pos % Len). The range of values at this output port is [0 ... <sequence length>].
- **(Bar)** "Bar" is the Event output port for the current bar number. The number at this output is calculated as Bar = Pos / Len and rounded down to the closest integer.
- **(Out)** "Out" is the Event output port for the current sequence step value.

8.5.3 Example: Multiplex Sequencer

This example shows how to build a sequencer with variable length, but with a maximum of 16 steps. The first part of the Structure is shown in the figure below. There you see that the core of the sequencer is a Multiplex 16 Module and the values of the steps are received from Knob Modules (↑1.1, Fader/Knob) at the input ports "0" to "15". The sequence length of the step at which the sequencer "folds back" to the first step, is set with the knob labeled "Length" at the "Len" input port. The actual position of the sequencer is

received at the "Pos" (position) input port in the form of a regular Events with incremental values. The value of the Event determines the sequencer position where values greater than "Len" are "folded back" to the set sequencer range.

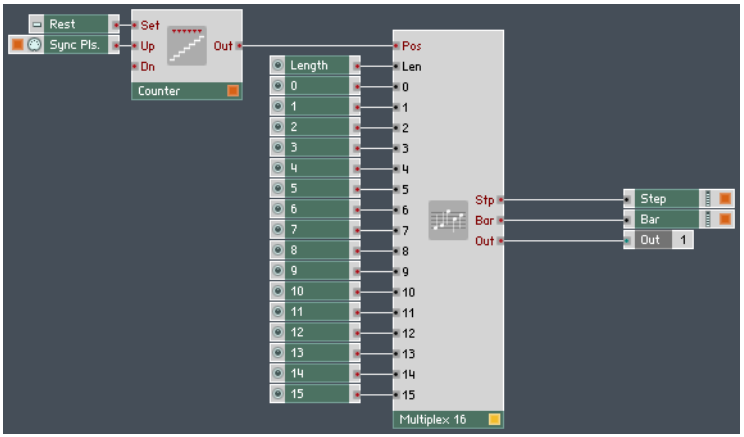


Fig. 8.14 Building a Multiplex Sequencer

In this example, the "Pos" values are ultimately received from the Sync Clock Module (↑2.15, Sync Pulse) which sends clock Events at regular time intervals, synchronized to the MIDI Clock. The rate of outgoing Events can be set at the Sync Clock Module's (↑2.15, Sync Pulse) Function page, with the Rate drop-down menu. In order to get regular Events carrying incremental values, the Counter Module (↑13.2, Counter) has been used. Each Event from the Sync Clock Module (↑2.15, Sync Pulse) increments the output of the Counter Module (↑13.2, Counter) by one, which then ends up being used as the "Pos" value for the Multiplex 16 Module. At the Counter Module's (↑13.2, Counter) "Set" input port you should have a Button Module (↑1.2, Button). In its Function page, set the Button Module (↑1.2, Button) to Trigger Mode and its "Max" value to zero. This enables you to reset the sequencer to its first position from the Instrument Panel.
As an additional feature, the XY Module (↑1.14, XY) has been used to graphically track the current sequencer position. The final Structure with this implemented is shown in the first figure at the end of the example and the final Panel representation of the sequencer is shown at the last figure at the end of the example. Please look at that figure to see where we are headed with the XY Module. The XY Module should be configured to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). This corresponds to going to its View

page and choosing the *Rectangle* menu entry from the Object Type drop-down menu (shown in the figure below). Also, you don't need a cursor and therefore choose the *None* menu entry from the Cursor Type drop-down menu (also shown in the figure below). While you are already in the XY Module's View page, disengage the Show Label and Show Value checkboxes and set the Height and Width edit fields so that the XY Module's Panel representation looks similar to what is depicted in the last figure of this example.

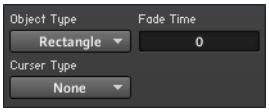


Fig. 8.15 To draw a rectangle from (X1, Y1) to (X2, Y2), choose the Rectangle menu entry from the Object Type drop-down menu.

The rectangle object type causes the XY Module's Panel representation to draw a rectangle from the coordinates (X1, Y1) to the coordinates (X2, Y2). We want to turn the XY Module (↑1.14, XY) into a row of 16 "imaginary" rectangles, each corresponding to one of the 16 steps in the sequencer. This is done the easiest when each box has the dimensions 1 by 1. For this reason, it is necessary to set the range of displayed values correctly in the XY Module's Function page. Set the "Min" and "Max" values for the X coordinate to "0" and "16", respectively. "Min" and "Max" for the Y coordinate should be "0" and "1", respectively. This is shown in the figure below.

Fig. 8.16 Use the Min and Max edit fields to set the range for the X and Y coordinates to [0 … 16] and [0 … 1], respectively.

Now that you have configured the XY Module (↑1.14, XY), you just need to feed the correct values from the Multiplex 16 Module to the input ports for the corresponding coordinates. Remember you wish to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). As shown in the Structure depicted below, connect the "Stp" (step) output port off the Multiplex 16 Module to the "X1" input port of the XY Module (↑1.14, XY), and the value Step + 1 (using an Add Module, see ↑4.2, Add, +) to the "X2" input port. Furthermore, connect the constant value "1" to the "Y2" input port. Your resulting Structure should look like the first figure below and its Instrument Panel counterpart should look similar to what

is shown in the second figure. Numeric Display Modules (↑1.8, Meter) have been added at the "Stp" and "Bar" output ports to give you quantitative visual access to the step and bar positions of the sequencer.

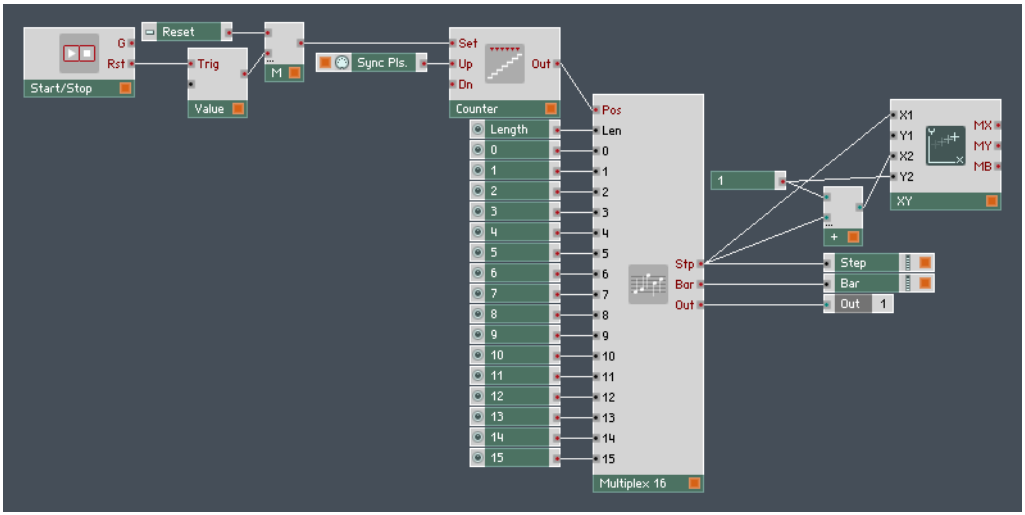


Fig. 8.17 The Structure for a simple sequencer with variable length and a simple step position display.

An additional feature to add would be that resetting the MIDI Clock also causes the sequencer position to be reset as well. Let's use the Start/Stop Module (↑2.13, Start/Stop) as that source of "reset" Events. For this, a reset operation should send the value "0" to the Counter Module's (↑13.2, Counter) "Set" input port. Since the "Rst" output port sends an Event with value "1", use a Value Module (↑13.15, Value) to change the Event's value to "0" and the a Merge Module to merge this reset signal with that received from the "Reset" button.

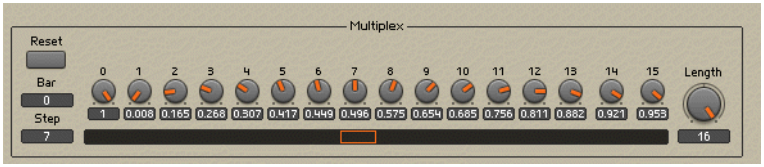


Fig. 8.18 The Instrument Panel of the simple sequencer.

9 LFO, Envelope

This category includes a fully modulatable, multiwaveform LFO, a random control generator, and envelope generators of just about any description including time/level ramp generators in three sizes.
All Envelopes can optionally display their curve in a panel graphic. This can be achieved with the Visible option on the Appearance page in the module Properties. The size of the display can be set in the Properties using the Size X and Size Y options. The timeline of the curve in the display is not scaled.

9.1 LFO

Fig. 9.1 LFO Module

9.1.1 Overview

The LFO Module is a low frequency oscillator (short: LFO) with sine, triangle and pulse waveforms available at the corresponding output ports. The output signal is a stream of Events at the Control Rate that has been selected in the *Settings > Control Rate* submenu. For each waveform you can control its frequency, amplitude, phase synchronization and phase offset at the corresponding input ports. To control the shape of the individual waveforms you can use the "W" (width) input port.

Application

It is typically used as a source for periodic modulation signals such as vibrato, tremolo, etc. Since the LFO operates at the Control Rate, it is much more CPU efficient than an audio oscillator which could do the same job. Therefore when working with control signals which are not meant to be audible by themselves, it is usually smart to use the LFO Module instead of an Oscillator Module

9.1.2 Ports

Input Ports

- **(F)** "F" (frequency) is the Audio input port for controlling of the oscillator frequency in Hz (cycles per second). If you wish to control the frequency in tempo-relevant units such as BPM (beats per minute), you can compute the required value in Hz as follows: F = oscillations-per-beat / 60 BPM. So, for example, for 3 oscillations per bar at 4 beats to the bar (that is ¾ oscillations per beat), you get F = (¾) / 60 BPM or 0.0125 BPM.
- **(A)** "A" (amplitude) is the Audio input port for controlling the LFO waveform amplitude. The output signal then moves in the range [-A … A].
- **(W)** "W' (width) is the Audio input port to control the shape of the individual waveforms. For the pulse waveform, the "W" (width) parameter is the ratio of the duration of the high state of the pulse signal to the low state of the pulse signal. For the other waveforms the "W" (width) parameter also controls the symmetry of the waveform. The values at this input port should be in the range [-1 … 1]. When the input port is left unconnected, the default value, "0" is used. This corresponds to a symmetric waveform.
- **(Snc)** "Sync" (synchronization) is the Audio input port for phase synchronization of the LFO waveform. A positive Event at this input port synchronizes the oscillator, resetting it to the phase given by the value at the "Ph" (phase) input port.
- **(Ph)** "Ph" (phase) is the Audio input port for specifying the phase (position in the waveform) to which the oscillator is reset when synchronization occurs. Ph = -1 corresponds to a phase of -180 degrees (start of rising slope), Ph = 0 corresponds to a phase of 0 degrees (middle of rising slope), and Ph = 1 corresponds to a phase of 180 degrees which is the same as -180 degrees. The default value, "0" is used if this port is left disconnected. Otherwise, the range [-1 … 1] should be used for the values at this input port.

Output Ports

- **(Sin)** "Sin" (sine) is the Event output port for the sine waveform signal.
- **(Tri)** "Tri" (triangle) is the Event output port for the triangle waveform signal.
- **(Pls)** "Pls" (pulse) is the Event output port for the pulse waveform signal.

9.1.3 Example: Tremolo

This example illustrates how to apply an LFO to the amplitude of an incoming signal (tremolo). The incoming signal can be any Audio signal, the simplest example being an oscillator signal. The LFO is applied to an incoming signal by multiplying the LFO Module's output with that signal. Here an LFO with a sine waveform has been used, as can be seen by the connected "Sin" (sinus) output port in the Structure below.

More often than not it is useful to have some control over the amount of tremolo effect that is applied to the incoming signal. The crossfading Structure, shown in the second example for the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+), is used here to crossfade between an unaltered input signal and an input signal with the tremolo effect. In the Structure below, the Knob labeled "Tremolo" has an output range [0 … 1] where "0" corresponds to an unaltered signal and "1" corresponds to fully applying the LFO to the signal amplitude. Note that for Tremolo = 0, the incoming signal is multiplied by "1" and for Tremolo = 1 the incoming signal is multiplied by the output of the LFO Module. In the second figure below, a screenshot shown how the waveform of a sine oscillator might look like after being treated by the tremolo effect.

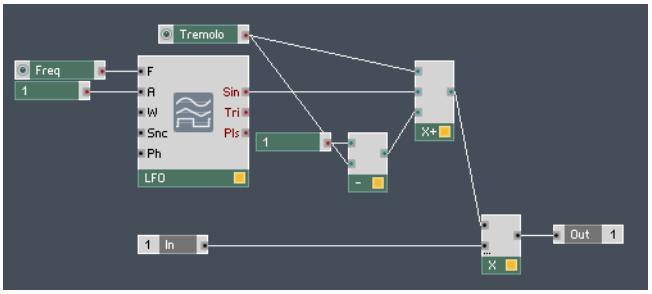


Fig. 9.2 The Structure for a simple tremolo effect, utilizing the LFO Module.

Fig. 9.3 The signal of a sine oscillator to which a tremolo effect has been applied.

9.2 Slow Random

Fig. 9.4 Slow Random Module

9.2.1 Overview

The Slow Random Module is a pseudorandom value generator. The output signal is a stepped waveform where the level of each step is pseudorandom in the amplitude interval. Each "level" has an equal probability of appearing. We say pseudorandom because the algorithm for generating the number is well-defined and therefore will always behave in a predictable manner. However, if you don't know that the sequence of numbers is generated by an algorithm, they appear to be perfectly random. The Slow Random Module works like a noise generator with uniform distribution followed by a sample & hold circuit clocked at a regular frequency.

Application

The Slow Random Module can be applied anywhere where you wish to make things a bit more interesting by adding an unpredictable component. You could use the Slow Random Module to give "human touch" to a sequencer by adding a slight random offset to the values it sends (such as velocity) and their timing. Adding a random waveform instead of a periodic LFO as a modulation of a filter cutoff or oscillator pitch also can have a very distinct effect.

9.2.2 Ports

Input Ports

- **(F)** "F" (frequency) is the Audio input port for the step frequency in Hz.
- **(A)** "A" (amplitude) is the Audio input port for controlling the amplitude range of the signal. The output values will be somewhere in the range [–A … A].

Output Ports

- **(Out)** "Out" is the Event output port for the pseudorandom signal.

9.2.3 Example: Random Pitch Deviation

In old analog synthesizers variations in the internal state of voltage controlled oscillators resulted in a random deviation from the intended oscillator pitch. This effect can be simulated in REAKTOR using the Slow Random Module. As shown in the Structure below, just add the output of a Slow Random Module to the intended pitch value (here from a Note Pitch Module, also see ↑2.1, Note Pitch). The extent of the random deviation can be controlled by a knob at the "A" (amplitude) input port of the Slow Random Module. The frequency has been set quite arbitrarily to 3 Hz, as indicated by the value "3" at the "F" (frequency) input port. The resulting pitch is then fed to the "P" (pitch) input port of a parabolic oscillator, for example. Note the Set Random Module (↑14.24, Set Random). With this Module the seed for the pseudorandom sequence created by the Slow Random Module is set to "24".

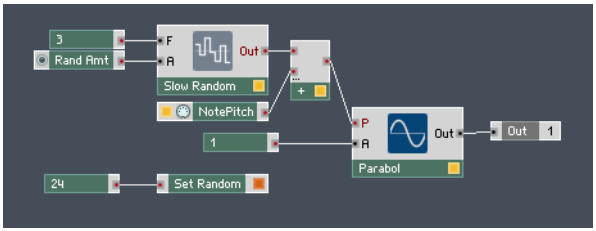


Fig. 9.5 A simple Structure to emulate random pitch deviations, as they appear in old analog synthesizers.

9.3 H - Env

Fig. 9.6 H Module

9.3.1 Overview

The H Module is an envelope generator with a hold characteristic. The envelope is triggered with a rising zero crossing at the "Trig" (trigger) input port. Upon a trigger the output value jumps to the current value of the "A" (amplitude) input port and stays there until the

hold time (set at the "H" (hold) input port) has passed, after which the output jumps back to zero. The envelope can be triggered at any time, including retriggering during the hold time.

Application

The H Module is the simplest envelope. It is commonly used for percussive sounds or as a control parameter in signal processing structures. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO. The "Trig" (trigger) input port can be connected to a Differentiator Module (↑10.22, Differentiator). This way when an incoming signal at the Differentiator Module (↑10.22, Differentiator) goes from a falling slope to a rising slope, the Envelope Module is triggered.

Please refer to the HR Envelope Module's "envelope follower" example for a demonstration of the Differentiator Module as a source for the envelope trigger signal.

9.3.2 Ports

Input Ports

- **(Trig)** "Trig" (Trigger) is the Audio input port for the trigger signal. A trigger signal is considered to be a positive zero crossing.
- **(A)** "A" (amplitude) is the Audio input port for the output amplitude of the envelope curve.
- **(H)** "H" (hold) is the logarithmic Event input port for controlling the hold time. H = 0 corresponds to 1 ms, H = 20 corresponds to 10 ms, and H = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

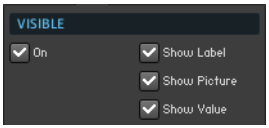
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.3.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.3.4 Example: Amplitude Envelope

This example shows how to apply a hold envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A trigger signal incoming at the "G" (gate) input port triggers the envelope at its "Trig" (trigger) input port. The trigger signal can, for example, come from a Gate Module (↑2.3, Gate) with its "Min" and "Max" values in the Function page set to "-1" and "1", respectively. Pressing a key on your MIDI keyboard then causes the signal at the "G" (and hence "Trig") input port to go

from "-1" to "1". This positive zero crossing constitutes a trigger signal for the H - Env Module. The amplitude is set to "1" with a Constant Module at the "A" input port. A Knob Module at the "H" (hold time) input port lets you set the envelope's hold time from the Instrument Panel.

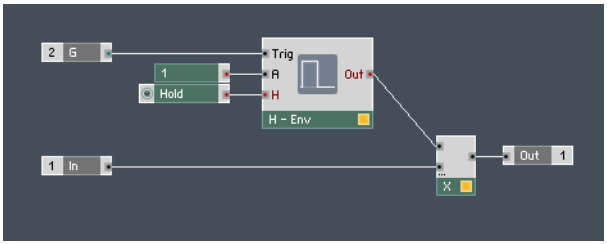


Fig. 9.7 A simple implementation of a hold envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.8 The Hold Envelope's Panel representation and knob for the hold time.

9.4 HR - Env

Fig. 9.9 HR - Env Module

9.4.1 Overview

The HR Module is an envelope generator with a hold-release characteristic. The envelope is triggered with a rising zero crossing at the "Trig" (trigger) input port. This causes the output value to jump to the value specified at the "A" (amplitude) Audio input port. The output signal is held at that value for the hold time (specified at the "H" (hold) input port) and then decays exponentially to zero with the release time, set at the "R" (release) input port.

Application

The HR Module is the simplest sustained envelope with a release curve. It is commonly used for percussive sounds. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO. The "Trig" (trigger) input port can be connected to a Differentiator Module (↑10.22, Differentiator), as shown in the second example. This way when an incoming signal at the Differentiator Module (↑10.22, Differentiator) goes from a falling slope to a rising slope, the Envelope Module is triggered.

9.4.2 Ports

Input Ports

- **(Trig)** "Trig" (Trigger) is the Audio input port for the trigger signal. A trigger signal is considered to be a positive zero crossing.
- **(A)** "A" (amplitude) is the Audio input port for the output amplitude of the envelope curve.
- **(H)** "H" (hold) is the logarithmic Event input port for controlling the hold time. H = 0 corresponds to 1 ms, H = 20 corresponds to 10 ms, and H = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to

10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 ... 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

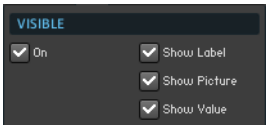
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.4.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.4.4 Example 1: Amplitude Envelope

This example shows how to apply a HR (hold-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A trigger signal incoming at the "G" (gate) input port triggers the envelope at its "Trig" (trigger) input port. The trigger signal can, for example, come from a Gate Module (↑2.3, Gate) with its "Min" and "Max" values in the Function page set to "-1" and "1", respectively. Pressing a key on your MIDI keyboard then causes the signal at the "G" (and hence "Trig") input port to go from "-1" to "1". This positive zero crossing constitutes a trigger signal for the HR - Env Module. The amplitude is set to "1" with a Constant Module at the "A" input port. Two Knob Modules at the "H" (hold time) and "R" (release time) input ports let you set the envelope's hold and release times from the Instrument Panel.

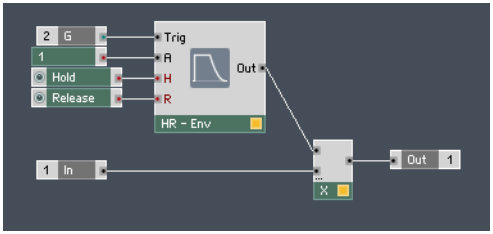


Fig. 9.10 A simple implementation of a hold-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.11 The HR Envelope's Panel representation and knobs for the hold and release times.

9.4.5 Example: Envelope Follower

A good way to modulate your effects, filters, envelopes, and oscillators, is using envelope followers. An envelope follower takes an input signal and uses its peaks to trigger an envelope whose amplitude matches that of the peak. The Structure in the figure below illustrates such an envelope follower Structure, utilizing the Peak Detector (↑12.14, Peak Detector), Differentiator (↑10.22, Differentiator), Sample and Hold (↑12.15, Sample & Hold), HR Envelope, and a 1-Pole Filter (↑10.2, HP/LP 1-Pole FM).

The incoming signal that is to trigger the envelope is first fed into the Peak Detector Module (↑12.14, Peak Detector). Depending on the frequency of peaks, the release time of the peaks in the Peak Detector's output signal is set with the "PkRel" knob at the "Rel" input port. Next, the Peak Detector Module's output is fed to the Sample and Hold Module (↑12.15, Sample & Hold). This signal is only sampled and held when a negative value at the Sample and Hold Module's "Trig" (trigger) input port goes from a negative value to a positive value. At the same time, the Differentiator Module (↑10.22, Differentiator) records if the Peak Detector Module's output is rising or falling. Therefore, if the Peak Detector signal goes from a falling state to a rising state, its output is sampled and forwarded to the "A" (amplitude) input port of the HR Envelope Module. At the same time, the HR Envelope Module is triggered (with this new amplitude, ultimately received from the Peak Detector Module). The envelope's output signal is then smoothed by a 1-Pole Filter (low-pass) where the value at its "P" (pitch) input port determines the amount of smoothing.

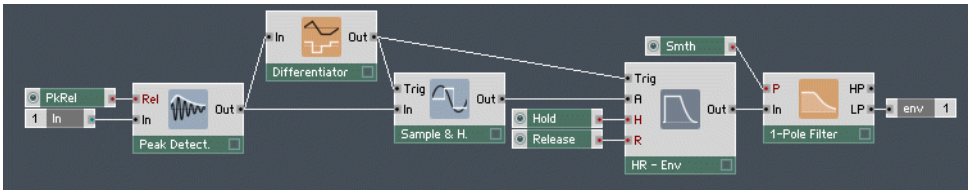


Fig. 9.12 The Structure for an envelope follower effect.

9.5 D - Env

Fig. 9.13 D- Env Module

9.5.1 Overview

The D Module is an envelope generator with a decay characteristic. The envelope is triggered with a rising zero crossing at the "Trig" (trigger) input port. This causes the output value to jump to the value specified at the "A" (amplitude) Audio input port. The output signal then decays exponentially to zero with the decay time, set at the "D" (decay) input port.

Application

The D Module is the simplest decay envelope. It is commonly used for percussive sounds. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO. The "Trig" (trigger) input port can be connected to a Differentiator Module (↑10.22, Differentiator). This way when an incoming signal at the Differentiator Module (↑10.22, Differentiator) goes from a falling slope to a rising slope, the Envelope Module is triggered.

Please refer to the HR Envelope Module's "envelope follower" example for a demonstration of the Differentiator Module as a source for the envelope trigger signal.

9.5.2 Ports

Input Ports

- **(Trig)** "Trig" (Trigger) is the Audio input port for the trigger signal. A trigger signal is considered to be a positive zero crossing.
- **(A)** "A" (amplitude) is the Audio input port for the output amplitude of the envelope curve.

- **(D)** "D" (decay is the logarithmic Event input port for controlling the decay time. This part of the curve is exponential. D = 0 corresponds to 1 ms, D = 20 corresponds to 10 ms, and D = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

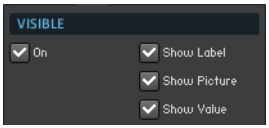
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.5.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.5.4 Example: Amplitude Envelope

This example shows how to apply a decay envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A trigger signal incoming at the "G" (gate) input port triggers the envelope at its "Trig" (trigger) input port. The trigger signal can, for example, come from a Gate Module (↑2.3, Gate) with its "Min" and "Max" values in the Function page set to "-1" and "1", respectively. Pressing a key on your MIDI keyboard then causes the signal at the "G" (and hence "Trig") input port to go from "-1" to "1". This positive zero crossing constitutes a trigger signal for the D - Env Module. The amplitude is set to "1" with a Constant Module at the "A" input port. A Knob Module at the "D" (decay time) input port lets you set the envelope's decay time from the Instrument Panel.

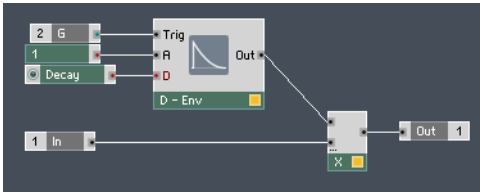


Fig. 9.14 A simple implementation of a decay envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

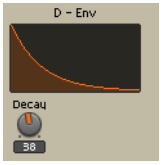


Fig. 9.15 The Decay Envelope's Panel representation and Knob for the decay time.

9.6 DR - Env

Fig. 9.16 DR Module

9.6.1 Overview

The DR Module is an envelope generator with a decay-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to jump to the amplitude value of the Gate signal (received at the "G" (gate) input port), after which the output decays linearly for the decay time set at the "D" (decay) input port. After that it continues to decay to zero, exponentially, with the time parameter set at the "R" (release) input port. If a gate Event with amplitude zero (Gate Off Event) arrives at the "G" (gate) input port, the exponential release decay curve is initiated immediately.

Application

The DR Module is useful in situations where you need a control parameter that instantly jumps to its maximum value and then decays along a curve with a linear and exponential part. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.6.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 ... 1]..

- **(D)** "D" (decay) is the logarithmic Event input port for controlling the decay time. This part of the curve is exponential. D = 0 corresponds to 1 ms, D = 20 corresponds to 10 ms, and D = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

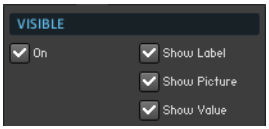
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.6.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.6.4 Example: Amplitude Envelope

This example shows how to apply a DR (decay-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A trigger signal incoming at the "G" (gate) input port triggers the envelope at its "Trig" (trigger) input port. The trigger signal can, for example, come from a Gate Module (↑2.3, Gate) with its "Min" and "Max" values in the Function page set to "-1" and "1", respectively. Pressing a key on your MIDI keyboard then causes the signal at the "G" (and hence "Trig") input port to go from "-1" to "1". This positive zero crossing constitutes a trigger signal for the DR - Env Module. The amplitude is set to "1" with a Constant Module (↑4.1, Constant) at the "A" input port. Two Knob Modules at the "D" (decay time) and "R" (release time) input ports let you set the envelope's decay and release times from the Instrument Panel.

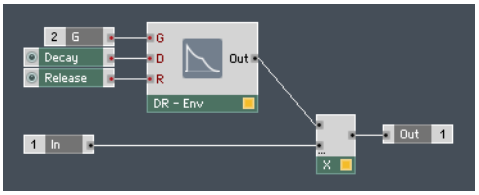


Fig. 9.17 A simple implementation of a decay-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant

controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.18 The DR Envelope's Panel representation and knobs for the decay and release times.

9.7 DSR - Env

Fig. 9.19 DSR Module

9.7.1 Overview

The DSR Module is an envelope generator with a decay-sustain-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to jump to the amplitude value of the Gate signal (received at the "G" (gate) input port), after which the output decays linearly with the decay time set at the "D" (decay) input port until it reaches the sustain level (multiplied by the amplitude) is reached. The sustain level is set at the "S" (sustain) input port. The output is held at the sustain level until a gate Event with amplitude zero (Gate Off Event) arrives at the "G" (gate) input port, after which the output decays exponentially with the release time (set at the "R" (release) input port) back to zero.

Application

The DSR Module is useful in situations where you need a control parameter that instantly jumps to its maximum value and then decays to the sustain level. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

Envelope generator with decay-sustain-release characteristic. When the envelope is triggered with a gate event the output value jumps to the amplitude value of the Gate signal, after which the output decays exponentially with the decay time to the sustain level (multiplied by the amplitude). After a gate event with amplitude zero (at note off) the output decays exponentially with the release time back to zero.

9.7.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(D)** "D" (decay) is the logarithmic Event input port for controlling the decay time. This part of the curve is exponential. D = 0 corresponds to 1 ms, D = 20 corresponds to 10 ms, and D = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(S)** "S" (sustain) is the Event input port for controlling the sustain level. Typical values at this input port are in the range [0 … 1]. However, values outside this range are allowed. S = 0 corresponds to a decay to zero whereas S = 1 corresponds to holding the value at the end of the attack phase (no decay curve). When this input port is disconnected, the default value, "0", is used.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

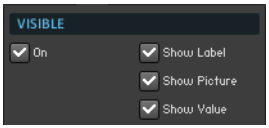
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.7.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.7.4 Example: Amplitude Envelope

This example shows how to apply a DSR (decay-sustain-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the envelope. Releasing the key sends a Gate

Off signal to the envelope, initiating the envelopes release curve. Knob Modules at the "D" (decay time) "S" (sustain level), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

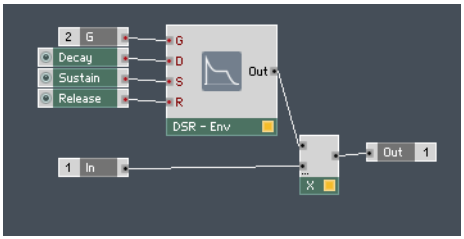


Fig. 9.20 A simple implementation of a decay-sustain-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.21 The DSR Envelope's Panel representation and knobs for its curve parameters.

9.8 DBDR - Env

Fig. 9.22 DBDR Module

9.8.1 Overview

The DBDR Module is an envelope generator with a decay-breakpoint-decay -release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to jump to the amplitude value of the Gate signal (received at the "G" (gate) input port), after which the output decays linearly with the decay time set at the "D1" (decay 1) input port until it reaches the breakpoint level (multiplied by the amplitude). The breakpoint level is set at the "B" (breakpoint) input port. Next it continues decaying exponentially to zero with the time parameter set at the "D2" (decay 2) input port for that part of the curve. If a gate Event with amplitude zero (Gate Off Event) arrives at the "G" (gate) input port, after which the output decays exponentially with the release time (set at the "R" (release) input port) back to zero. The "R" (release) value is normally set to be shorter than the "D2" (decay 2) value.

Application

The DBDR Module is useful in situations where you need a control parameter that instantly jumps to its maximum value and then decays along a more complex curve and eventually fades to zero. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.8.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(D1)** "D1" (decay 1) is the logarithmic Event input port for controlling the first decay time. This part of the curve is linear. D1 = 0 corresponds to 1 ms, D1 = 20 corresponds to 10 ms, and D1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

- **(B)** "B" (breakpoint) is the Event input port for controlling the breakpoint level. The range for the values at this input port is [0 … 1]. For B = 0, the "D2" (decay 2) time parameter is never used. For B = 1 the envelope curve immediately starts running through the "D2" (decay 2) curve when a positive Event arrives at the "G" (gate) input port (Gate On Event). When disconnected, the default value that is used for this input port is "0".
- **(D2)** "D2" (decay 2) is the logarithmic Event input port for controlling the second decay time. This part of the curve is exponential. D2 = 0 corresponds to 1 ms, D2 = 20 corresponds to 10 ms, and D2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

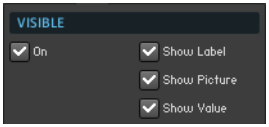
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.8.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.8.4 Example: Amplitude Envelope

This example shows how to apply a DBDR (decay-breakpoint-decay-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelopes release curve (if not already entered). Knob Modules at the "D1" (first decay time) "B" (breakpoint level), "D2" (second decay time), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

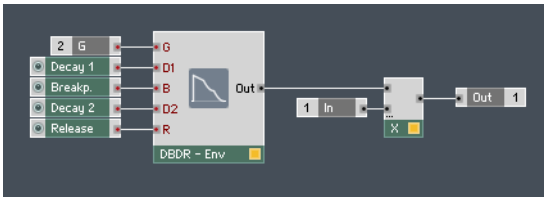


Fig. 9.23 A simple implementation of a decay-breakpoint-decay-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant

controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.24 The DBDR Envelope's Panel representation and knobs for its curve parameters.

9.9 DBDSR-Env

Fig. 9.25 DBDSR Module

9.9.1 Overview

The DBDSR Module is an envelope generator with a decay-breakpoint-decay-sustain-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to jump to the amplitude value of the Gate signal (received at the "G" (gate) input port), after which the output decays linearly with the decay time set at the "D1" (decay 1) input port until it reaches the breakpoint level (multiplied by the amplitude). The breakpoint level is set at the "B" (breakpoint) input port. Next it continues decaying exponentially with the time parameter set at the "D2" (decay 2) input port until the sustain level (multiplied by the amplitude) is reached. The sustain level is set at the "S" (sustain) input port. The output is held at the sustain level until a gate Event with amplitude zero (Gate Off Event) arrives at the "G" (gate) input port, after which the output decays exponentially with the release time (set at the "R" (release) input port) back to zero. The "R" (release) value is normally set to be shorter than the "D2" (decay 2) value.

Application

The DBDSR Module is useful in situations where you need a control parameter that instantly jumps to its maximum value and then decays along a more complex curve. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.9.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(D1)** "D1" (decay 1) is the logarithmic Event input port for controlling the first decay time. This part of the curve is linear. D1 = 0 corresponds to 1 ms, D1 = 20 corresponds to 10 ms, and D1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(B)** "B" (breakpoint) is the Event input port for controlling the breakpoint level. The range for the values at this input port is [0 … 1]. For B = 0, the "D2" (decay 2) time parameter is never used. For B = 1 the envelope curve immediately starts running through the "D2" (decay 2) curve when a positive Event arrives at the "G" (gate) input port (Gate On Event). When disconnected, the default value that is used for this input port is "0".
- **(D2)** "D2" (decay 2) is the logarithmic Event input port for controlling the second decay time. This part of the curve is exponential. D2 = 0 corresponds to 1 ms, D2 = 20 corresponds to 10 ms, and D2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(S)** "S" (sustain) is the Event input port for controlling the sustain level. Typical values at this input port are in the range [0 … 1]. However, values outside this range are allowed. When this input port is disconnected, the default value, "0", is used.

- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 ... 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Port

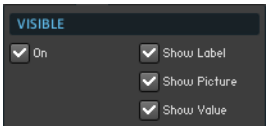
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.9.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.9.4 Example: Amplitude Envelope

This example shows how to apply a DBDSR (decay-breakpoint-decay-sustain-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelopes release curve. Knob Modules at the "D1" (first decay time) "B" (breakpoint level), "D2" (second decay time), "S" (sustain level), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

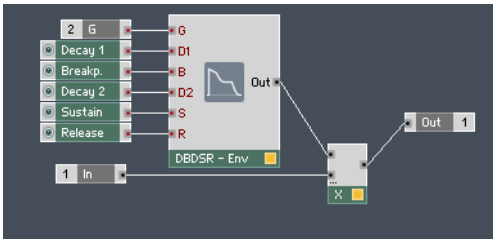


Fig. 9.26 A simple implementation of a decay-breakpoint-decay-sustain-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

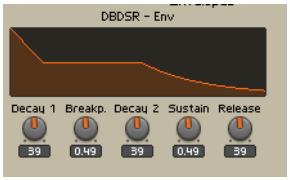


Fig. 9.27 The DBDSR Envelope's Panel representation and knobs for its curve parameters.

9.10 AD - Env

Fig. 9.28 AD Module

9.10.1 Overview

The AD Module is an envelope generator with an attack-decay characteristic. The envelope is triggered with a rising zero crossing at the "Trig" (trigger) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack input port) to the amplitude specified at the "A" (amplitude) Audio input port. The output signal then decays exponentially to zero with the decay time, set at the "D" (decay) input port.

Application

The AD Module is the simplest envelope with an attack slope. It is commonly used for percussive sounds. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO. The "Trig" (trigger) input port can be connected to a Differentiator Module (↑10.22, Differentiator). This way when an incoming signal at the Differentiator Module (↑10.22, Differentiator) goes from a falling slope to a rising slope, the Envelope Module is triggered.

Please refer to the HR Envelope Module's "envelope follower" example for a demonstration of the Differentiator Module as a source for the envelope trigger signal.

9.10.2 Ports

Input Ports

- **(Trig)** "Trig" (Trigger) is the Audio input port for the trigger signal. A trigger signal is considered to be a positive zero crossing.
- **(A)** "A" (amplitude) is the Audio input port for the output amplitude of the envelope curve.
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(D)** "D" (decay) is the logarithmic Event input port for controlling the decay time. This part of the curve is exponential. D = 0 corresponds to 1 ms, D = 20 corresponds to 10 ms, and D = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

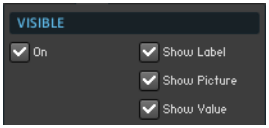
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.10.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.10.4 Example: Amplitude Envelope

This example shows how to apply an AD (attack-decay) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A trigger signal incoming at the "G" (gate) input port triggers the envelope at its "Trig" (trigger) input port. The trigger signal can, for example, come from a Gate Module (↑2.3, Gate) with its "Min" and "Max" values in the Function page set to "-1" and "1", respectively. Pressing a key on your MIDI keyboard then causes the signal at the "G" (and hence "Trig") input port to go from "-1" to "1". This positive zero crossing constitutes a trigger signal for the AD - Env Module. The amplitude is set to "1" with a Constant Module at the "A" input port. Two Knob Modules at the "A" (attack time) and "D" (decay time) input ports let you set the envelope's decay and release times from the Instrument Panel.

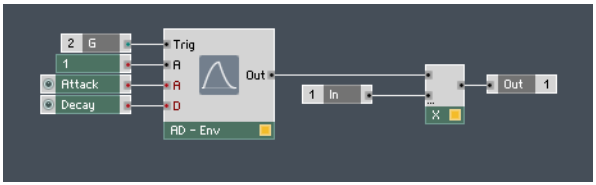


Fig. 9.29 A simple implementation of an attack-decay envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times,

Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

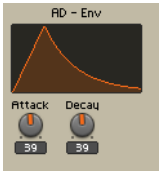


Fig. 9.30 The AD Envelope's Panel representation and knobs for the attack and decay times.

9.11 AR - Env

Fig. 9.31 AR Module

9.11.1 Overview

The AR Module is an envelope generator with an attack-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack) input port) to the amplitude value of the Gate signal at the "G" (gate) input port. The output signal stays at the maximum value until a zero valued Event arrives at the "G" (gate) input port (Gate Off Event). Then the output decays exponentially to zero with the release time, set at the "R" (release) input port.

Application

The AR Module is the simplest sustained envelope with an attack slope. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.11.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

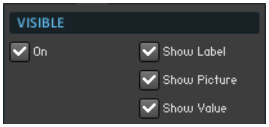
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.11.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.11.4 Example: Amplitude Envelope

This example shows how to apply an AR (attack-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release curve. Knob Modules at the "A" (attack time) and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

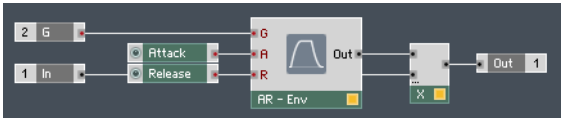


Fig. 9.32 A simple implementation of an attack-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.33 The AR Envelope's Panel representation and knobs for its curve parameters.

9.12 ADR-Env

Fig. 9.34 ADR Module

9.12.1 Overview

The ADR Module is an envelope generator with an attack-decay-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack) input port) to the amplitude value of the Gate signal at the "G" (gate) input port. After that amplitude value has been reached, the output signal decays to zero along an exponential slope according to the time parameter set at the "D" (decay) input port. If a zero valued Event arrives at the "G" (gate) input port (Gate Off Event), the output decays exponentially to zero, but with the release time, set at the "R" (release) input port.

Application

The ADR Module is a bread and butter envelope that eventually fades to zero. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.12.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(D)** "D" (decay) is the logarithmic Event input port for controlling the decay time. This part of the curve is exponential. D = 0 corresponds to 1 ms, D = 20 corresponds to 10 ms, and D = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

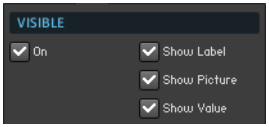
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.12.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.12.4 Example: Amplitude Envelope

This example shows how to apply an ADR (attack-decay-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the

key sends a Gate Off signal to the envelope, initiating the envelope's release curve (if not already reached). Knob Modules at the "A" (attack time), "D" (decay time), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

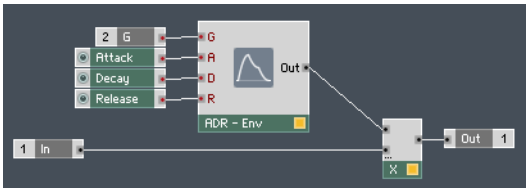


Fig. 9.35 A simple implementation of an attack-decay-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

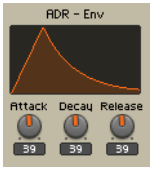


Fig. 9.36 The ADR Envelope's Panel representation and knobs for its curve parameters.

9.13 ADSR - Env

Fig. 9.37 ADSR Module

9.13.1 Overview

The ADSR Module is an envelope generator with an attack-decay-sustain-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack) input port) to the amplitude value of the Gate signal at the "G" (gate) input port. After that amplitude value has been reached, the output signal decays to the sustain level along an exponential slope according to the time parameter set at the "D" (decay) input port. The sustain level is specified at the "S" (sustain) input port. The output value is then held at the sustain level until a zero valued Event arrives at the "G" (gate) input port (Gate Off Event). After that the output decays exponentially back to zero. For this last decay the release time, set at the "R" (release) input port, is used.

Application

The ADSR Module is the bread and butter envelope that is used in most synths. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.13.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

- **(D)** "D" (decay) is the logarithmic Event input port for controlling the decay time. This part of the curve is exponential. D = 0 corresponds to 1 ms, D = 20 corresponds to 10 ms, and D = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(S)** "S" (sustain) is the Event input port for controlling the sustain level. Typical values at this input port are in the range [0 … 1]. However, values outside this range are allowed. S = 0 corresponds to a decay to zero whereas S = 1 corresponds to holding the value at the end of the attack phase (no decay curve). When this input port is disconnected, the default value, "0", is used.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

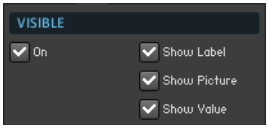
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.13.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.13.4 Example: Amplitude Envelope

This example shows how to apply an ADR (attack-decay-sustain-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release curve. Knob Modules at the "A" (attack time), "D" (decay time), "S" (sustain level), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

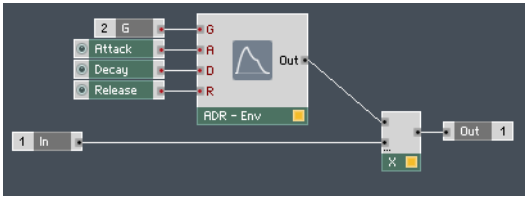


Fig. 9.38 A simple implementation of an attack-decay-sustain-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant

controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

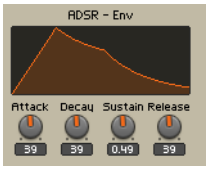


Fig. 9.39 The ADSR Envelope's Panel representation and knobs for its curve parameters.

9.14 ADBDR - Env

Fig. 9.40 ADBDR Module

9.14.1 Overview

The ADBDR Module is an envelope generator with an attack-decay-breakpoint-decay-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack) input port) to the amplitude value of the Gate signal (at the "G" input port. Then the output value then decays linearly over the course of the time set at the "D1" (decay 1) input port until it reaches the breakpoint level (multiplied by the amplitude received from the Gate signal). Next it continues decaying exponentially with the time parameter set at the "D2" (decay 2) input port back to zero. A gate Event with amplitude zero (a Note Off Event) causes the envelope to exponentially decay to zero. However, here the time parameter at the "R" (release) input port is used. The "R" (release) value is normally set to be shorter than the "D2" (decay 2) value.

Application

The ADBDR Module should be applied when a complex and eventually fading envelope curve is required. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.14.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(D1)** "D1" (decay 1) is the logarithmic Event input port for controlling the first decay time. This part of the curve is linear. D1 = 0 corresponds to 1 ms, D1 = 20 corresponds to 10 ms, and D1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.• B: Event input for controlling the breakpoint level. Range of values: 0 (never use decay-2 time) to 1 (immediately use decay-2 time).
- **(D2)** "D2" (decay 2) is the logarithmic Event input port for controlling the second decay time. This part of the curve is exponential. D2 = 0 corresponds to 1 ms, D2 = 20 corresponds to 10 ms, and D2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to

10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 ... 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

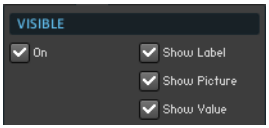
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.14.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.14.4 Example: Amplitude Envelope

This example shows how to apply an ADBDR (attack-decay-breakpoint-decay-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release curve (if not already reached). Knob Modules at the "A" (attack time), "D1" (first decay time), "B" (breakpoint level), "D2" (second decay time), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

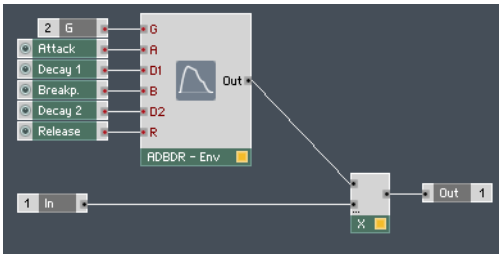


Fig. 9.41 A simple implementation of an attack-decay-breakpoint-decay-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

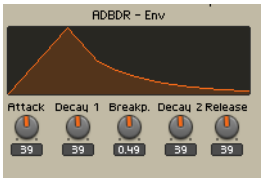


Fig. 9.42 The ADBDR Envelope's Panel representation and knobs for its curve parameters.

9.15 ADBDSR-Env

Fig. 9.43 ADBDSR Module

9.15.1 Overview

The ADBDSR Module is an envelope generator with an attack-hold-decay-sustain-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack) input port) to the amplitude value of the Gate signal at the "G" (gate) input port. After that amplitude value has been reached, the output signal decays to the break-point level (specified at the "B" input port) along a linear ramp with the decay time specified at the "D1" (decay 2) input port. Then the envelope signal decays to the sustain level along an exponential slope according to the time parameter set at the "D2" (decay 2) input port. The sustain level is specified at the "S" (sustain) input port. The output value is then held at the sustain level until a zero valued Event arrives at the "G" (gate) input port (Gate Off Event). After that the output decays exponentially back to zero. For this last decay the release time, set at the "R" (release) input port, is used.

Application

The ADBDSR Module should be applied when a more complex sustained envelope curve is required. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.15.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(D1)** "D1" (decay 1) is the logarithmic Event input port for controlling the first decay time. This part of the curve is linear. D1 = 0 corresponds to 1 ms, D1 = 20 corresponds to 10 ms, and D1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(B)** "B" (breakpoint) is the Event input port for controlling the breakpoint level. The range for the values at this input port is [0 … 1]. However, values outside this range are allowed. When disconnected, the default value that is used for this input port is "0".
- **(D2)** "D2" (decay 2) is the logarithmic Event input port for controlling the second decay time. This part of the curve is exponential. D2 = 0 corresponds to 1 ms, D2 = 20 corresponds to 10 ms, and D2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

- **(S)** "S" (sustain) is the Event input port for controlling the sustain level. Typical values at this input port are in the range [0 … 1]. However, values outside this range are allowed. When this input port is disconnected, the default value, "0", is used.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

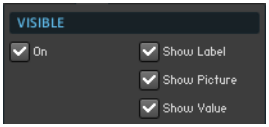
- **(Out)** Audio output for the envelope signal.

9.15.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.15.4 Example: Amplitude Envelope

This example shows how to apply an ADBDSR (attack-decay-breakpoint-decay-sustain-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release. Knob Modules at the "A" (attack time), "D1" (first decay time), "B" (breakpoint level), "D2" (second decay time), "S" (sustain level), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

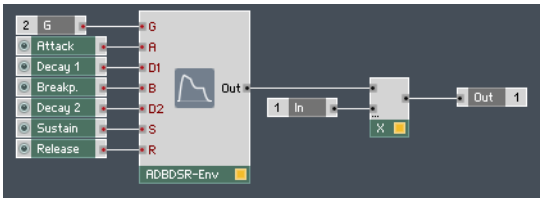


Fig. 9.44 A simple implementation of an attack-decay-breakpoint-decay-sustain-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

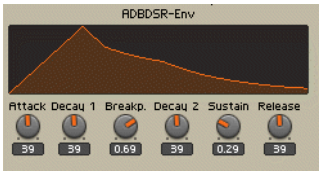


Fig. 9.45 The ADBDSR Envelope's Panel representation and knobs for its curve parameters.

9.16 AHDSR - Env

Fig. 9.46 AHDSR Module

9.16.1 Overview

The AHDSR Module is an envelope generator with an attack-hold-decay-sustain-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack) input port) to the amplitude value of the Gate signal at the "G" (gate) input port. After that amplitude value has been reached, the output signal stays there until the hold time (set at the "H" (hold) input port) has passed. Then it decays to the sustain level with a linear slope according to the time parameter set at the "D" (decay) input port. The sustain level is specified at the "S" (sustain) input port. The output value is then held at the sustain level until a zero valued Event arrives at the "G" (gate) input port (Gate Off Event). After that the output decays exponentially back to zero. For this last decay the release time, set at the "R" (release) input port, is used.

Application

The AHDSR Module should be applied when a more complex envelope curve with hold and sustain is required. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.16.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(H)** "H" (hold) is the logarithmic Event input port for controlling the hold time. H = 0 corresponds to 1 ms, H = 20 corresponds to 10 ms, and H = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(D)** "D" (decay) is the logarithmic Event input port for controlling the decay time. This part of the curve is exponential. D = 0 corresponds to 1 ms, D = 20 corresponds to 10 ms, and D = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(S)** "S" (sustain) is the Event input port for controlling the sustain level. Typical values at this input port are in the range [0 … 1]. However, values outside this range are allowed. S = 0 corresponds to a decay to zero whereas S = 1 corresponds to holding the value at the end of the attack phase (no decay curve). When this input port is disconnected, the default value, "0", is used.
- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

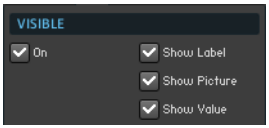
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.16.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.16.4 Example: Amplitude Envelope

This example shows how to apply an AHDSR (attack-hold-decay-sustain-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a

Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release. Knob Modules at the "A" (attack time), "H" (hold time), "D" (decay time), "S" (sustain level), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

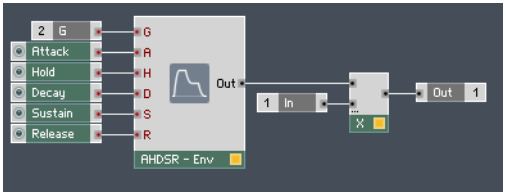


Fig. 9.47 A simple implementation of an attack-hold-decay-sustain-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.48 The AHDSR Envelope's Panel representation and knobs for its curve parameters.

9.17 AHDBDR - Env

Fig. 9.49 AHDBDR Module

9.17.1 Overview

The AHDBDR Module is an envelope generator with attack-hold-decay-breakpoint-decay-release characteristic. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise with a linear slope over the course of the attack time (set at the "A" (attack) input port) to the amplitude value of the Gate signal (at the "G" input port) and to stay there until the hold time (set at the "H" (hold) input port) has passed. Then the output value decays linearly over the course of the time set at the "D1" (decay 1) input port until it reaches the breakpoint level (multiplied by the amplitude received from the Gate signal). Next it continues decaying exponentially with the time parameter set at the "D2" (decay 2) input port back to zero. A gate Event with amplitude zero (a Note Off Event) causes the envelope to exponentially decay to zero. However, here the time parameter at the "R" (release) input port is used. The "R" (release) value is normally set to be shorter than the "D2" (decay 2) value.

Application

The AHDBDR Module should be applied when a more complex and eventually fading envelope curve is required. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.17.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers and releases the envelope. The envelope is triggered with a positive valued Event (a Note On Event) and released with a zero valued Event (a Note Off Event). The amplitude of the Gate signal determines the maximum output value. The typical range of values at this input port is [0 … 1].
- **(A)** "A" (attack) is the Event input port for controlling the attack time in the logarithmic scale. This part of the curve is linear. A = 0 corresponds to 1 ms, A = 20 corresponds to 10 ms, and A = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(H)** "H" (hold) is the logarithmic Event input port for controlling the hold time. H = 0 corresponds to 1 ms, H = 20 corresponds to 10 ms, and H = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(D1)** "D1" (decay 1) is the logarithmic Event input port for controlling the first decay time. This part of the curve is linear. D1 = 0 corresponds to 1 ms, D1 = 20 corresponds to 10 ms, and D1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(B)** "B" (breakpoint) is the Event input port for controlling the breakpoint level. The range for the values at this input port is [0 … 1]. However, values outside this range are allowed. When disconnected, the default value that is used for this input port is "0".
- **(D2)** "D2" (decay 2) is the logarithmic Event input port for controlling the second decay time. This part of the curve is exponential. D2 = 0 corresponds to 1 ms, D2 = 20 corresponds to 10 ms, and D2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

- **(R)** "R" (release) is the logarithmic Event input port for controlling the release time. This part of the curve is exponential and is adopted upon a zero valued Event at the "G" (gate) input port (Gate Off Event). T = 0 corresponds to 1 ms, T = 20 corresponds to 10 ms, and T = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 ... 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Output Ports

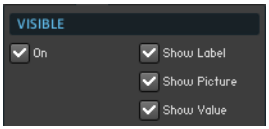
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.17.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.17.4 Example: Amplitude Envelope

This example shows how to apply an AHDBDR (attack-decay-breakpoint-decay-release) envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release curve (if not already reached). Knob Modules at the "A" (attack time), "H" (hold), "D1" (first decay time), "B" (breakpoint level), "D2" (second decay time), and "R" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

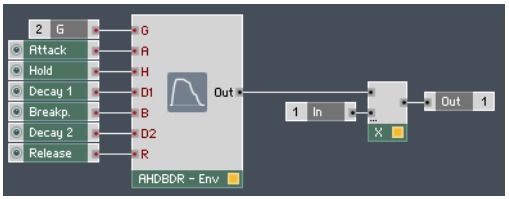


Fig. 9.50 A simple implementation of an attack-hold-decay-breakpoint-decay-release envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

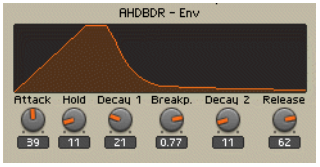


Fig. 9.51 The AHDBDR Envelope's Panel representation and knobs for its curve parameters.

9.18 4-Ramp

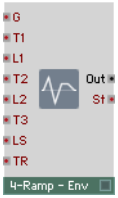


Fig. 9.52 4-Ramp Module

9.18.1 Overview

The 4-Ramp Module is a 4-stage envelope generator with linear slopes. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise to the first level, set at the "L1" (level 1) input port, which is reached within the time set for the first stage, set at the "T2" (time 2) input port. Then it the second level ("L2") is reached within the time set at the "T2" (time 2) input port for the second stage, and so on. The third level is the sustain level and is set at the "LS" (sustain level) input port. The envelope output is then held at that level until a Gate Off Event (an Event with value "0") arrives at the "G" (gate) input port. After that the envelope output returns to zero within the release time period, set at the "TR" (release time) input port.

Application

The 4-Ramp Module is useful for more complex modulations which are best defined by breakpoints connected by ramps. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.18.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers the envelope. The amplitude of the Gate signal combines with all the level values at the "L1", "L2", and "LS" input ports to determine the actual envelope levels reached.

- **(T1)** "T1" (time 1) is the Event input port for controlling the time parameter for the first stage in the logarithmic scale. T1 = 0 corresponds to 1 ms, T1 = 20 corresponds to 10 ms, and T1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L1)** "L1" (level1) is the Event input port for controlling the level which is reached at the end of the first stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T2)** "T2" (time 2) is the Event input port for controlling the time parameter for the second stage in the logarithmic scale. T2 = 0 corresponds to 1 ms, T2 = 20 corresponds to 10 ms, and T2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L2)** "L2" (level 2) is the Event input port for controlling the level which is reached at the end of the second stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T3)** "T3" (time 3) is the Event input port for controlling the time parameter for the third stage in the logarithmic scale. T3 = 0 corresponds to 1 ms, T3 = 20 corresponds to 10 ms, and T3 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(LS)** "LS" (sustain level) is the Event input port for controlling the level which is reached at the end of the third stage. This is the sustain level. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(TR)** "TR" (release time) is the Event input port for controlling the time parameter for the release in the logarithmic scale. TR = 0 corresponds to 1 ms, TR = 20 corresponds to 10 ms, and TR = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Please refer to section 9.4 in the Application Reference for more information on the logarithmic scales in REAKTOR.

Output Ports

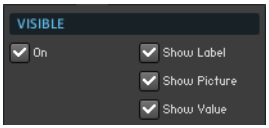
- **(St)** "St" (stage) is the Event output port delivering the envelope's current stage (1, 2 ...). After the end of the release stage and before a new trigger the value is "0". With appropriate processing this value in can be used to chain other envelopes, or for the envelope to retrigger itself.
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.18.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.18.4 Example: Amplitude Envelope

This example shows how to apply a 4-ramp envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release curve. Knob Modules at the "T1" (first transition time), "L1" (first breakpoint level), "T2" (second transition time), "L2" (second breakpoint level), "T3" (third transition time), "LS" (sustain level), and "TR" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

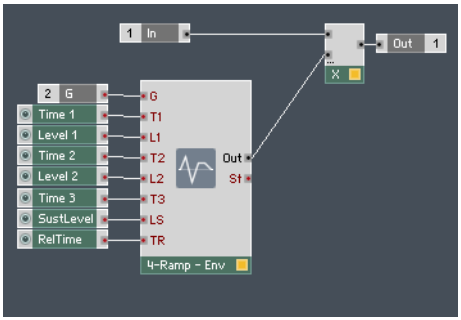


Fig. 9.53 A simple implementation of a 4-ramp envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

Fig. 9.54 The 4-Ramp Envelope Module's Panel representation and knobs for its curve parameters.

9.19 5-Ramp

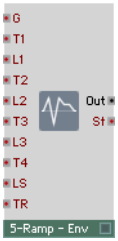


Fig. 9.55 5-Ramp Module

9.19.1 Overview

The 5-Ramp Module is a 5-stage envelope generator with linear slopes. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise to the first level, set at the "L1" (level 1) input port, which is reached within the time set for the first stage, set at the "T2" (time 2) input port. Then it the second level ("L2") is reached within the time set at the "T2" (time 2) input port for the second stage, and so on. The fourth level is the sustain level and is set at the "LS" (sustain level) input port. The envelope output is then held at that level until a Gate Off Event (an Event with value "0") arrives at the "G" (gate) input port. After that the envelope output returns to zero within the release time period, set at the "TR" (release time) input port.

Application

The 5-Ramp Module is useful for more complex modulations which are best defined by breakpoints connected by ramps. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.19.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers the envelope. The amplitude of the Gate signal combines with all the level values at the "L1", "L2", "L3", and "LS" input ports to determine the actual envelope levels reached.
- **(T1)** "T1" (time 1) is the Event input port for controlling the time parameter for the first stage in the logarithmic scale. T1 = 0 corresponds to 1 ms, T1 = 20 corresponds to 10 ms, and T1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L1)** "L1" (level1) is the Event input port for controlling the level which is reached at the end of the first stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T2)** "T2" (time 2) is the Event input port for controlling the time parameter for the second stage in the logarithmic scale. T2 = 0 corresponds to 1 ms, T2 = 20 corresponds to 10 ms, and T2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L2)** "L2" (level 2) is the Event input port for controlling the level which is reached at the end of the second stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T3)** "T3" (time 3) is the Event input port for controlling the time parameter for the third stage in the logarithmic scale. T3 = 0 corresponds to 1 ms, T3 = 20 corresponds to 10 ms, and T3 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

- **(L3)** "L3" (level 3) is the Event input port for controlling the level which is reached at the end of the third stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T4)** "T4" (time 4) is the Event input port for controlling the time parameter for the fourth stage in the logarithmic scale. T4 = 0 corresponds to 1 ms, T4 = 20 corresponds to 10 ms, and T4 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(LS)** "LS" (sustain level) is the Event input port for controlling the level which is reached at the end of the fourth stage. This is the sustain level. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(TR)** "TR" (release time) is the Event input port for controlling the time parameter for the release in the logarithmic scale. TR = 0 corresponds to 1 ms, TR = 20 corresponds to 10 ms, and TR = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Please refer to section 9.4 in the Application Reference for more information on the logarithmic scales in REAKTOR.

Output Ports

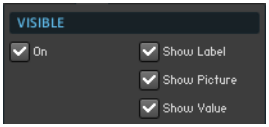
- **(St)** "St" (stage) is the Event output port delivering the envelope's current stage (1, 2 …). After the end of the release stage and before a new trigger the value is "0". With appropriate processing this value in can be used to chain other envelopes, or for the envelope to retrigger itself.
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.19.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.19.4 Example: Amplitude Envelope

This example shows how to apply a 5-ramp envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release curve. Knob Modules at the "T1" (first transition time), "L1" (first breakpoint level), "T2" (second transition time), "L2" (second breakpoint level), "T3" (third transition time), "L3" (third breakpoint level), "T4" (fourth transition time), "LS" (sustain level), and "TR" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

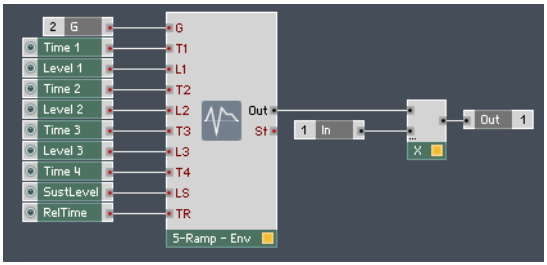


Fig. 9.56 A simple implementation of a 5-ramp envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

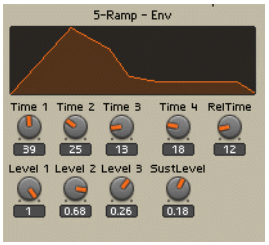


Fig. 9.57 The 5-Ramp Envelope Module's Panel representation and knobs for its curve parameters.

9.20 6-Ramp

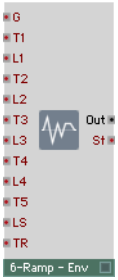


Fig. 9.58 6-Ramp Module

9.20.1 Overview

The 6-Ramp Module is a 6-stage envelope generator with linear slopes. The envelope is triggered with a Gate On Event at the "G" (gate) input. This causes the output value to rise to the first level, set at the "L1" (level 1) input port, which is reached within the time set for the first stage, set at the "T2" (time 2) input port. Then it the second level ("L2") is reached within the time set at the "T2" (time 2) input port for the second stage, and so on. The fifth level is the sustain level and is set at the "LS" (sustain level) input port. The envelope output is then held at that level until a Gate Off Event (an Event with value "0") arrives at the "G" (gate) input port. After that the envelope output returns to zero within the release time period, set at the "TR" (release time) input port.

Application

The 6-Ramp Module is useful for more complex modulations which are best defined by breakpoints connected by ramps. Parameters which are commonly modulated by envelopes include oscillator or sampler amplitude, filter cutoff and resonance, FM modulation index, and the modulation amplitude of an LFO.

9.20.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for the Gate signal that triggers the envelope. The amplitude of the Gate signal combines with all the level values at the "L1", "L2", "L3", "L4", and "LS" input ports to determine the actual envelope levels reached.
- **(T1)** "T1" (time 1) is the Event input port for controlling the time parameter for the first stage in the logarithmic scale. T1 = 0 corresponds to 1 ms, T1 = 20 corresponds to 10 ms, and T1 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L1)** "L1" (level1) is the Event input port for controlling the level which is reached at the end of the first stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T2)** "T2" (time 2) is the Event input port for controlling the time parameter for the second stage in the logarithmic scale. T2 = 0 corresponds to 1 ms, T2 = 20 corresponds to 10 ms, and T2 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L2)** "L2" (level 2) is the Event input port for controlling the level which is reached at the end of the second stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T3)** "T3" (time 3) is the Event input port for controlling the time parameter for the third stage in the logarithmic scale. T3 = 0 corresponds to 1 ms, T3 = 20 corresponds to 10 ms, and T3 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L3)** "L3" (level 3) is the Event input port for controlling the level which is reached at the end of the third stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".

- **(T4)** "T4" (time 4) is the Event input port for controlling the time parameter for the fourth stage in the logarithmic scale. T4 = 0 corresponds to 1 ms, T4 = 20 corresponds to 10 ms, and T4 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(L4)** "L4" (level 4) is the Event input port for controlling the level which is reached at the end of the fourth stage. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(T5)** "T5" (time 5) is the Event input port for controlling the time parameter for the fifth stage in the logarithmic scale. T5 = 0 corresponds to 1 ms, T5 = 20 corresponds to 10 ms, and T5 = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.
- **(LS)** "LS" (sustain level) is the Event input port for controlling the level which is reached at the end of the fifth stage. This is the sustain level. The typical range for the values at this input port is [-1 … 1]. When disconnected, this port receives the default value "0".
- **(TR)** "TR" (release time) is the Event input port for controlling the time parameter for the release in the logarithmic scale. TR = 0 corresponds to 1 ms, TR = 20 corresponds to 10 ms, and TR = 40 corresponds to 100 ms. The typical range for the values at this input port is [0 … 80]. When disconnected, this input port receives the default value "0" which corresponds to a transition time of 1 ms.

Please refer to section 9.4 in the Application Reference for more information on the logarithmic scales in REAKTOR.

Output Ports

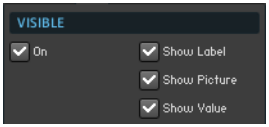
- **(St)** "St" (stage) is the Event output port delivering the envelope's current stage (1, 2 ...). After the end of the release stage and before a new trigger the value is "0". With appropriate processing this value in can be used to chain other envelopes, or for the envelope to retrigger itself.
- **(Out)** "Out" is the Audio output port for the envelope signal.

9.20.3 Properties: View Page

Making the Envelope Graph Visible on the Panel

Envelopes have a Panel representation which graphically depicts the time evolution of the envelope function. This can be a very useful component in your Instrument since you get direct visual feedback on how the time and level parameters that you have set for your envelope affect the overall envelope curve.

► To turn on the graphic display of your envelope in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Envelope's Panel Representation

Depending on the type of Envelope Module you are using, you might need a small or large graph of the envelope curve on your Instrument Panel.

► To change the width and height of the Envelope Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

9.20.4 Example: Amplitude Envelope

This example shows how to apply a 6-ramp envelope to an incoming signal, such as an oscillator output signal. The corresponding Structure is shown in the figure below. The signal to which the envelope is to be applied is incoming at the "In" input port. It is multiplied with the envelope output signal using a Multiply Module (↑4.5, Multiply, X). A gate signal incoming at the "G" (gate) input port usually comes from a Gate Module (↑2.3, Gate). Pressing a key on your MIDI keyboard then causes a Gate On signal to be sent to the "G" input port, triggering the attack phase of the envelope. Releasing the key sends a Gate Off signal to the envelope, initiating the envelope's release curve. Knob Modules at

the "T1" (first transition time), "L1" (first breakpoint level), "T2" (second transition time), "L2" (second breakpoint level), "T3" (third transition time), "L3" (third breakpoint level), "T4" (fourth transition time), "L4" (fourth breakpoint level), "T5" (fifth transition time), "LS" (sustain level), and "TR" (release time) input ports let you set the envelope's parameters from the Instrument Panel.

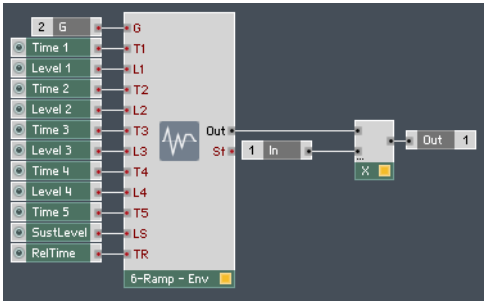


Fig. 9.59 A simple implementation of a 6-ramp envelope.

In the Rehearsal Ensemble you can get acquainted with the envelopes in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your envelope of choice. Since each envelope has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when an envelope has been selected from the "Envelopes" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the envelope is visible, as shown in the figure below.

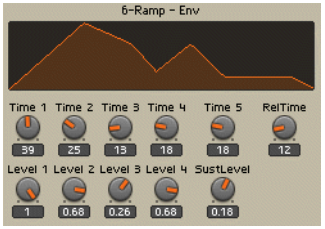


Fig. 9.60 The 6-Ramp Envelope Module's Panel representation and knobs for its curve parameters.

10 Filter

Filters make up REAKTOR's largest collection of signal processors. You'll find 22 of them here covering everything from standard low-, high-, and band-pass filters, to emulations of classic synth filters from Sequential and Moog. There's also an all-pass filter for reverb and dispersion circuits as well as integrating and differentiating filters.
All of REAKTOR's filters can operate at any frequency, from 0 Hz (constant signal) through the entire audio range right up to the limit set by the Sample Rate. This means they are all equally suited for audio processing or for smoothing control signals (e.g. portamento). When using a filter to process the input signal to an input port that only accepts Events (e.g. "P" as opposed to "F") you need to insert an A to E (perm) Module for conversion.

Please refer to the example of the Unit Delay Module (↑11.6, Unit Delay) for a brief discussion on filters (and how to build them, in principle, using the Unit Delay Module).

10.1 HP/LP 1-Pole

Fig. 10.1 1-Pole Module

10.1.1 Overview

The 1-Pole Module is a 1-pole filter with Audio output ports for the high-pass characteristic ("HP") and the low-pass characteristic ("LP"). A 1-pole filter corresponds to a signal attenuation of 6 dB per octave starting from the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (pitch) input port.

Application

One specific application for the 1-Pole Module is to create a brown noise source by applying the low-pass filter at 0 Hz cutoff frequency (a "P" value of "-60" suffices) to the output of a Noise Module (white noise). The 1-Pole Filter is not only useful for an audible effect. The 1-pole low-pass filter is also useful for smoothing Audio signals, as seen in the "envelope follower" example in this section. For a second example Structure, please refer to the 1-Pole FM Module's example.

10.1.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually In the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

- **(HP)** "HP" (high-pass) is the Audio output port for the high-pass filtered signal.
- **(LP)** "LP" (low-pass) Audio output port for the low-pass filtered signal.

10.1.3 Example: Envelope Follower

A good way to modulate your effects, filters, envelopes, and oscillators, is using envelope followers. An envelope follower takes an input signal and uses its peaks to trigger an envelope whose amplitude matches that of the peak. The Structure in the figure below illustrates such an envelope follower Structure, utilizing the Peak Detector (↑12.14, Peak Detector), Differentiator (↑12.14, Peak Detector), Sample and Hold (↑12.15, Sample & Hold), HR Envelope (↑9.4, HR - Env), and a 1-Pole Filter (↑10.2, HP/LP 1-Pole FM).
The incoming signal that is to trigger the envelope is first fed into the Peak Detector Module (↑12.14, Peak Detector) (↑12.14, Peak Detector). Depending on the frequency of peaks, the release time of the peaks in the Peak Detector's output signal is set with the "PkRel" knob at the "Rel" input port. Next, the Peak Detector Module's output is fed to the Sample and Hold Module (↑12.15, Sample & Hold). This signal is only sampled and held when a negative value at the Sample and Hold Module's "Trig" (trigger) input port goes from a negative value to a positive value. At the same time, the Differentiator Module (↑10.22, Differentiator) records if the Peak Detector Module's output is rising or falling. Therefore, if the Peak Detector signal goes from a falling state to a rising state, its output is sampled and forwarded to the "A" (amplitude) input port of the HR Envelope Module (↑9.4, HR - Env). At the same time, the HR Envelope Module is triggered (with this new

amplitude, ultimately received from the Peak Detector Module). The envelope's output signal is then smoothed by a 1-Pole Filter (low-pass) where the value at its "P" (pitch) input port determines the amount of smoothing.

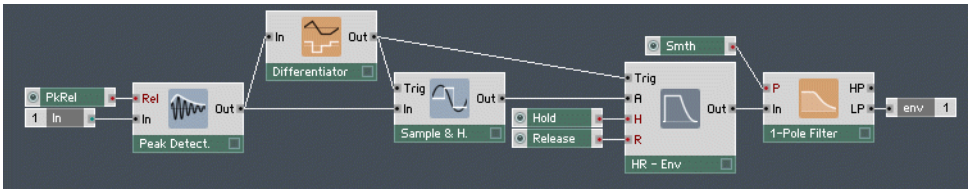


Fig. 10.2 The Structure for an envelope follower effect.

10.2 HP/LP 1-Pole FM

Fig. 10.3 1-Pole FM Module

10.2.1 Overview

The 1-Pole FM Module is a 1-pole filter with Audio output ports for the high-pass characteristic ("HP") and the low-pass characteristic ("LP"). A 1-pole filter corresponds to a signal attenuation of 6 dB per octave starting from the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.

Application

One specific application for the 1-Pole Module is to create a brown noise source by applying the low-pass filter at 0 Hz cutoff frequency to the output of a Noise Module (white noise). With the 1-Pole FM Module you can use both the "P" (cutoff pitch) and "F" (frequency) input ports to get an exact cutoff frequency of 0 Hz. For more exact (Audio Rate) modulations, use the "F" (cutoff frequency) input port. The 1-Pole Filter is not only useful for an audible effect. The 1-pole low-pass filter is also useful for smoothing Audio signals.

10.2.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

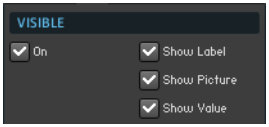
- **(HP)** "HP" (high-pass) is the Audio output port for the 1-pole high-pass filtered signal.
- **(LP)** "LP" (low-pass) Audio output port for the 1-pole low-pass filtered signal.

10.2.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency parameter affects the overall frequency response curve. Note that only one filter characteristic, the low-pass curve, is shown.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.2.4 Example: 1-Pole Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a 1-pole filter. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The cutoff pitch is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module. Since the 1-Pole FM Module comprises both a 1-pole low-pass and 1-pole high-pass filter, a List Module in combination with a Selector Module lets you select from the Instrument Panel which filtered signal is sent to the output.

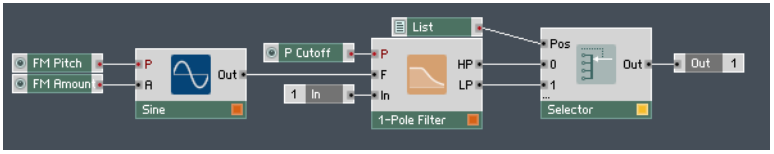


Fig. 10.4 A simple implementation of a 1-pole filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.3 1-Pole All-Pass

Fig. 10.5 1-Pole All-Pass Module

10.3.1 Overview

The 1-Pole All-Pass Module is a first order all-pass filter. This type of filter has a flat amplitude response, but the phase shift between input and output grows from 0 degrees at low frequencies to -180 degrees at high frequencies. At the cutoff frequency, controlled by the "P" (cutoff pitch) input port, the phase is shifted by -90 degrees.

Application

Common applications for all-pass filters include delay-based effects that "mix-up" the phases — reverbs and diffuser delays. All-pass filters have an approximately linear dispersion relation. This means that you can effectively delay a signal by fractions of a sample. Applications for this include the generation of plucked string sounds. This can be achieved by building a comb filter Structure but combining the Delay Module and the 1-Pole All-Pass Module to create a feedback path that can have delays of an arbitrary (not just integer) number of samples.

10.3.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency, where the phase shift is - 90 degrees, in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency

range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.

- **(In)** "In" is the Audio input port for the signal to be all-pass filtered (phase-shifted).

Output Ports

- **(Out)** "Out" is the Audio output port for the all-pass filtered (phased-shifted) signal.

10.3.3 Example: 1-Pole Allpass Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a 1-pole allpass filter. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The filter's phase response is controlled by the Knob Module labeled "Allpass P ", at the "P" (pitch) input port.

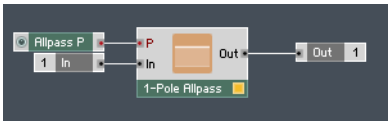


Fig. 10.6 A simple implementation of a 1-pole allpass filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On check-box in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.4 Multi 2-Pole

Fig. 10.7 Multi 2-Pole Module

10.4.1 Overview

The Multi 2-Pole Module is a 2-pole filter with Audio output ports for the high-pass characteristic ("HP"), the band-pass characteristic ("BP"), and the low-pass characteristic ("LP"). For the high-pass and low-pass characteristics, a 2-pole filter corresponds to a signal attenuation of 12 dB per octave starting from the cutoff frequency. For a band-pass characteristic a 2-pole filter corresponds to a signal attenuation of 6 db per octave both above and below the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port.
The Multi 2-Pole Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1" (corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency.

Very large amplitudes are generated when the "Res" parameter approaches "1".

Application

Filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape.
Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.
You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multiband compressor) and then later combine the signals again in the output signal. Learning

how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects. For an example Structure, please refer to the Multi 2-Pole FM Module's example.

10.4.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency range of [26 Hz … 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 … 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency[Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

- **(HP)** "HP" (high-pass) is the Audio output port for the 2-pole high-pass filtered signal.
- **(BP)** "BP" (band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP)** "LP" (low-pass) is the Audio output port for the 2-pole low-pass filtered signal.

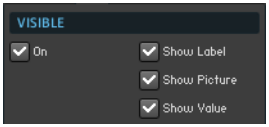
10.4.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequen-

cy range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the low-pass curve, is shown.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.5 Multi 2-Pole FM

Fig. 10.8 Multi 2-Pole FM Module

10.5.1 Overview

The Multi 2-Pole FM Module is a 2-pole filter with Audio output ports for the high-pass characteristic ("HP"), the band-pass characteristic ("BP"), and the low-pass characteristic ("LP"). For the high-pass and low-pass characteristics, a 2-pole filter corresponds to a signal attenuation of 12 dB per octave starting from the cutoff frequency. For a band-pass

characteristic a 2-pole filter corresponds to a signal attenuation of 6 db per octave both above and below the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.
The Multi 2-Pole FM Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1" (corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency.

Very large amplitudes are generated when the "Res" parameter approaches "1".

Application

Filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape. Additionally, you can use the "F" (cutoff frequency) input port for frequency modulation of the cutoff frequency. Modulating this input port at Audio Rate yields an FM'ish sound.
Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.
You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multiband compressor) and then later combine the signals again in the output signal. Learning how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects.

10.5.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 ... 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] $\cong$ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

- **(HP)** "HP" (high-pass) is the Audio output port for the 2-pole high-pass filtered signal.
- **(BP)** "BP" (band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP)** "LP" (low-pass) is the Audio output port for the 2-pole low-pass filtered signal.

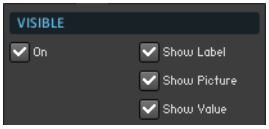
10.5.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integra-

tor and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the low-pass curve, is shown.

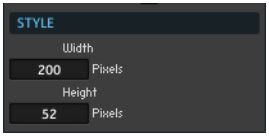
► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



10.5.4 Example: 2-Pole Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a 2-pole filter. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The cutoff pitch is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port and the resonance parameter is specified with the Knob Module labeled "Reson" at the "Res" input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module. Since the 2-Pole FM Module comprises a 2-pole high-pass, 2-pole band-pass, and 2-pole low-pass filter, a List Module in combination with a Selector Module lets you select from the Instrument Panel which filtered signal is sent to the output.

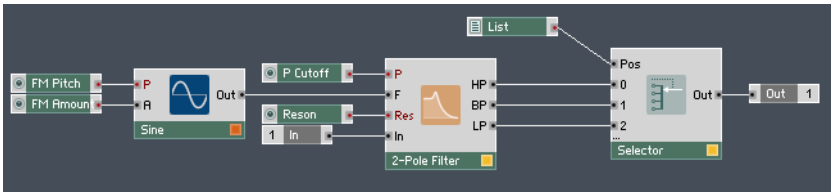


Fig. 10.9 A simple implementation of a 2-pole filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.6 2-Pole Notch

Fig. 10.10 2-Pole Notch

10.6.1 Overview

The 2-Pole Notch Module is a 2-pole filter with Audio output ports for the band -reject (notch) characteristic ("N"), the high-pass characteristic ("HP"), the band-pass characteristic ("BP"), and the low-pass characteristic ("LP"). If you are using the band-reject characteristic filter, then the frequency components at the cutoff frequency (set at the "P" (cutoff pitch) port) are completely removed. For the high-pass and low-pass characteristics, a 2-pole filter corresponds to a signal attenuation of 12 dB per octave starting from the cutoff frequency. For a band-pass characteristic a 2-pole filter corresponds to a signal attenuation of 6 db per octave both above and below the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port.

The 2-Pole Notch Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1" (corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency.

Very large amplitudes are generated for all characteristics except the notch characteristic when the "Res" parameter approaches "1".

Application

Sometimes you wish to remove unwanted frequency components (such as resonances from bad recordings) from a signal. The surest way to remove one frequency component is by using the notch filter in the Multi/Notch 2-Pole Module.

In general, filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape.

Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.

You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multiband compressor) and then later combine the signals again in the output signal. Learning how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects. For an example Structure, please refer to the 2-Pole Notch FM Module's example.

10.6.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 ... 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

- **(HP)** "HP" (high-pass) is the Audio output port for the 2-pole high-pass filtered signal.
- **(BP)** "BP" (band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP)** "LP" (low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(N)** "N" (notch) is the Audio output port for the band-reject (notch) filtered signal.

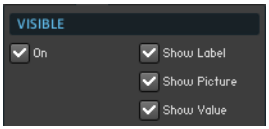
10.6.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful compo-

nent in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the low-pass curve, is shown.

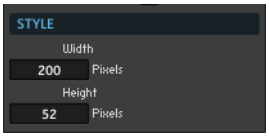
► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



10.7 2-Pole Notch FM

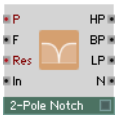


Fig. 10.11 2-Pole Notch Module

10.7.1 Overview

The 2-Pole Notch Module is a 2-pole filter with Audio output ports for the band -reject (notch) characteristic ("N"), the high-pass characteristic ("HP"), the band-pass characteristic ("BP"), and the low-pass characteristic ("LP"). If you are using the band-reject characteristic filter, then the frequency components at the cutoff frequency (set at the "P" (cutoff pitch) port) are completely removed. For the high-pass and low-pass characteristics, a 2-

pole filter corresponds to a signal attenuation of 12 dB per octave starting from the cutoff frequency. For a band-pass characteristic a 2-pole filter corresponds to a signal attenuation of 6 db per octave both above and below the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.
The 2-Pole Notch Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1" (corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency. Additionally, you can use the "F" (cutoff frequency) input port for frequency modulation of the cutoff frequency. Modulating this input port at Audio Rate yields an FM'ish sound.

Very large amplitudes are generated for all characteristics except the notch characteristic when the "Res" parameter approaches "1".

Application

Sometimes you wish to remove unwanted frequency components (such as resonances from bad recordings) from a signal. The surest way to remove one frequency component is by using the notch filter in the Multi/Notch 2-Pole Module.
In general, filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape.
Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.

You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multi-band compressor) and then later combine the signals again in the output signal. Learning how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects.

10.7.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency range of [26 Hz … 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 … 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

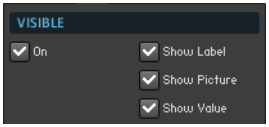
- **(HP)** "HP" (high-pass) is the Audio output port for the 2-pole high-pass filtered signal.
- **(BP)** "BP" (band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP)** "LP" (low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(N)** "N" (notch) is the Audio output port for the band-reject (notch) filtered signal.

10.7.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the low-pass curve, is shown.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.7.4 Example: 2-Pole Notch Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a 2-pole notch filter. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The cutoff pitch is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port and the resonance parameter is specified with the Knob Module labeled "Reson" at the "Res" input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module. Since the 2-Pole Notch FM Module comprises a 2-pole high-pass, 2-pole band-pass, 2-pole low-pass, and 2-pole notch filter, a List Module in combination with a Selector Module lets you select from the Instrument Panel which filtered signal is sent to the output.

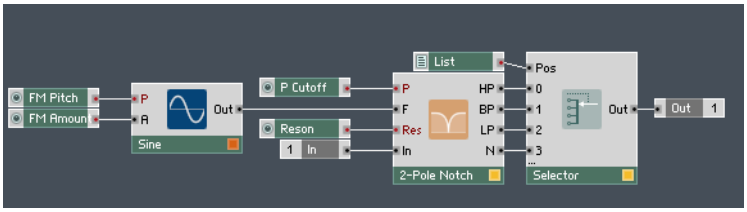


Fig. 10.12 A simple implementation of a 2-pole notch filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.8 Multi/LP 4-Pole

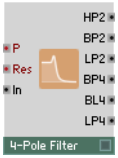


Fig. 10.13 Multi/LP 4-Pole Module

10.8.1 Overview

The Multi/LP 4-Pole Module comprises a 4-pole and a 2-pole filter with Audio output ports for the high-pass characteristic ("HP2"), the band-pass characteristic ("BP2" and "BP4"), the low-pass characteristic ("LP2" and "LP4"), and band/low-pass characteristic ("BL4"). For the high-pass and low-pass characteristics, a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 24 dB per octave (12 db per octave) starting from the cutoff frequency. For a band-pass characteristic a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 12 db per octave (6 db per octave) both above and below the cutoff frequency. For the band/low-pass characteristic, a 2-pole band pass filter is connected serially to a 2-pole low-pass filter. This corresponds to a 6 db per octave attenuation below the cutoff frequency and an 18 db per octave attenuation above the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port.
The Multi/LP 4-Pole Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1" (corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency.

Very large amplitudes are generated when the "Res" parameter approaches "1".

Application

Filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and

then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape.
Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.
You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multiband compressor) and then later combine the signals again in the output signal. Learning how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects. For an example Structure, please refer to the Multi/LP 4-Pole FM Module's example.

10.8.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency range of [26 Hz … 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 … 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

- **(HP2)** "HP2" (2-pole high-pass) is the Audio output port for the 2-pole high-pass filtered signal.

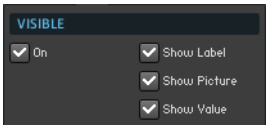
- **(BP2)** "BP2" (2-pole band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP2)** "LP2" (2-pole low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(BP4)** "BP4" (4-pole band-pass) is the Audio output port for the 4-pole band-pass filtered signal.
- **(BL4)** "BL4" (4-pole band/low-pass) is the Audio output port for the 4-pole band/low-pass filtered signal.
- **(LP4)** "LP4" (4-pole low-pass) is the Audio output port for the 4-pole low-pass filtered signal.

10.8.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the low-pass curve, is shown.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.9 Multi/LP 4-Pole FM

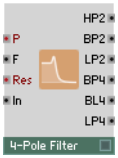


Fig. 10.14 Multi/LP 4-Pole Fm Module

10.9.1 Overview

The Multi/LP 4-Pole FM Module comprises a 4-pole and a 2-pole filter with Audio output ports for the high-pass characteristic ("HP2"), the band-pass characteristic ("BP2" and "BP4"), the low-pass characteristic ("LP2" and "LP4"), and band/low-pass characteristic ("BL4"). For the high-pass and low-pass characteristics, a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 24 dB per octave (12 db per octave) starting from the cutoff frequency. For a band-pass characteristic a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 12 db per octave (6 db per octave) both above and below the cutoff frequency. For the band/low-pass characteristic, a 2-pole band pass filter is connected serially to a 2-pole low-pass filter. This corresponds to a 6 db per octave attenuation below the cutoff frequency and an 18 db per octave attenuation above the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.
The Multi/LP 4-Pole FM Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1"

(corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency. Additionally, you can use the "F" (cutoff frequency) input port for frequency modulation of the cutoff frequency. Modulating this input port at Audio Rate yields an FM'ish sound.

Very large amplitudes are generated when the "Res" parameter approaches "1".

Application

Filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape.
Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.
You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multiband compressor) and then later combine the signals again in the output signal. Learning how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects.

10.9.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.

- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 ... 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

- **(HP2)** "HP2" (2-pole high-pass) is the Audio output port for the 2-pole high-pass filtered signal.
- **(BP2)** "BP2" (2-pole band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP2)** "LP2" (2-pole low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(BP4)** "BP4" (4-pole band-pass) is the Audio output port for the 4-pole band-pass filtered signal.
- **(BL4)** "BL4" (4-pole band/low-pass) is the Audio output port for the 4-pole band/low-pass filtered signal.
- **(LP4)** "LP4" (4-pole low-pass) is the Audio output port for the 4-pole low-pass filtered signal.

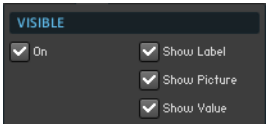
10.9.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequen-

cy range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the low-pass curve, is shown.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.9.4 Example: 4-Pole Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a 4-pole filter. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The cutoff pitch is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port and the resonance parameter is specified with the Knob Module labeled "Reson" at the "Res" input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module. Since the Multi/LP 4-Pole FM Module comprises a 2-pole high-pass, 2-pole band-pass, 2-pole low-pass, 4-pole band-pass, 4-pole band- and low-pass, and 4-pole low-pass filter, a List Module in combination with a Selector Module lets you select from the Instrument Panel which filtered signal is sent to the output.

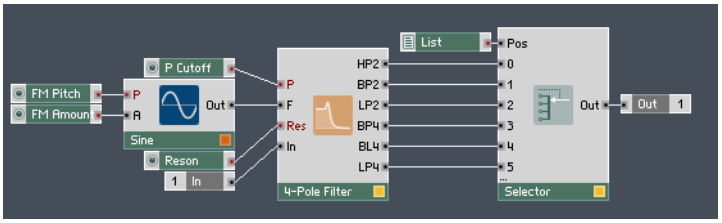


Fig. 10.15 A simple implementation of a 4-pole filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.10 Multi/HP 4-Pole

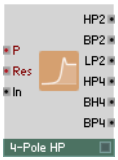


Fig. 10.16 Multi/HP 4-Pole Module

10.10.1 Overview

The Multi/HP 4-Pole Module comprises a 4-pole and a 2-pole filter with Audio output ports for the high-pass characteristic ("HP2" and "HP4"), the band-pass characteristic ("BP2" and "BP4"), the low-pass characteristic ("LP2"), and band/high-pass characteristic ("BH4"). For the high-pass and low-pass characteristics, a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 24 dB per octave (12 db per octave) starting from the cutoff frequency. For a band-pass characteristic a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 12 db per octave (6 db per octave) both above and below the cutoff frequency. For the band/high-pass characteristic, a 2-pole band pass filter is connected serially to a 2-pole high-pass filter. This corresponds to an 18 db per octave at-

tenuation below the cutoff frequency and a 6 db per octave attenuation above the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port.
The Multi/HP 4-Pole Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1" (corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency.
Very large amplitudes are generated when the "Res" parameter approaches "1".

Application

Filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape.
Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.
You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multiband compressor) and then later combine the signals again in the output signal. Learning how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects. For an example Structure, please refer to the Multi/HP 4-Pole FM Module's example.

10.10.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 ... 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

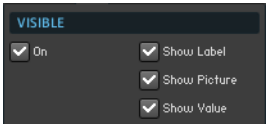
- **(HP2)** "HP2" (2-pole high-pass) is the Audio output port for the 2-pole high-pass filtered signal.
- **(BP2)** "BP2" (2-pole band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP2)** "LP2" (2-pole low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(HP4)** "HP4" (4-pole high-pass) is the Audio output port for the 4-pole high-pass filtered signal.
- **(BH4)** "BP4" (4-pole band/high-pass) is the Audio output port for the 4-pole band/high-pass filtered signal.
- **(BP4)** "BP4" (4-pole band-pass) is the Audio output port for the 4-pole band-pass filtered signal.

10.10.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the high-pass curve, is shown.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.11 Multi/HP 4-Pole FM

Fig. 10.17 Multi/HP 4-Pole FM Module

10.11.1 Overview

The Multi/HP 4-Pole FM Module comprises a 4-pole and a 2-pole filter with Audio output ports for the high-pass characteristic ("HP2" and "HP4"), the band-pass characteristic ("BP2" and "BP4"), the low-pass characteristic ("LP2"), and band/high-pass characteristic ("BH4"). For the high-pass and low-pass characteristics, a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 24 dB per octave (12 db per octave) starting from the cutoff frequency. For a band-pass characteristic a 4-pole filter (2-pole filter) corresponds to a signal attenuation of 12 db per octave (6 db per octave) both above and below the cutoff frequency. For the band/high-pass characteristic, a 2-pole band pass filter is connected serially to a 2-pole high-pass filter. This corresponds to an 18 db per octave attenuation below the cutoff frequency and a 6 db per octave attenuation above the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.

The Multi/HP 4-Pole FM Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak. The pass band gain is always "1" (corresponding to 0 dB), whereas the gain at the cutoff frequency increases as the "Res" (resonance) parameter approaches "1". At high resonance settings the filter output exhibits a strong frequency component (nearly in the form of a sine tone) at the cutoff frequency. Additionally, you can use the "F" (cutoff frequency) input port for frequency modulation of the cutoff frequency. Modulating this input port at Audio Rate yields an FM'ish sound.

Very large amplitudes are generated when the "Res" parameter approaches "1".

Application

Filters are used to variably change the magnitude (and phase) of the frequency components in a signal. This is the key aspect to subtractive synthesis where one starts out with a signal that has many harmonic components (such as a saw wave or white noise) and then starts removing or reducing unwanted frequency components, finally ending up with the desired sound. This is not unlike the work of a sculptor who chisels away "unwanted" components of a piece of rock to (hopefully) end up with the envisioned shape.
Of course, filters can also be applied to Audio signals stemming from samplers and to external Audio signals. With the resonance parameter you can enhance certain components. For example, you can enhance a bass kick sound by applying a high-pass filter to it with moderate resonance. The cutoff frequency should be set at the main fundamental frequency component of the kick sound.
You could also use filters to split a signal up into different frequency components (low, middle, and high) which then receive different treatment by effects (such as in a multiband compressor) and then later combine the signals again in the output signal. Learning how the different filters affect your Audio signal and using filters creatively is important to mastering signal processing for audio instruments and effects.

10.11.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.

- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 ... 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

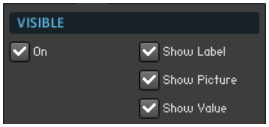
- **(HP2)** "HP2" (2-pole high-pass) is the Audio output port for the 2-pole high-pass filtered signal.
- **(BP2)** "BP2" (2-pole band-pass) is the Audio output port for the 2-pole band-pass filtered signal.
- **(LP2)** "LP2" (2-pole low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(HP4)** "HP4" (4-pole high-pass) is the Audio output port for the 4-pole high-pass filtered signal.
- **(BH4)** "BP4" (4-pole band/high-pass) is the Audio output port for the 4-pole band/high-pass filtered signal.
- **(BP4)** "BP4" (4-pole band-pass) is the Audio output port for the 4-pole band-pass filtered signal.

10.11.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve. Note that only one filter characteristic, the high-pass curve, is shown.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.11.4 Example: 4-Pole Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a 4-pole filter. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The cutoff pitch is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port and the resonance parameter is specified with the Knob Module labeled "Reson" at the "Res" input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module. Since the Multi/HP 4-Pole FM Module comprises a 2-pole high-pass, 2-pole band-pass, 2-pole low-pass, 4-pole high-pass, 4-pole band- and high-pass, and 4-pole band-pass filter, a List Module in combination with a Selector Module lets you select from the Instrument Panel which filtered signal is sent to the output.

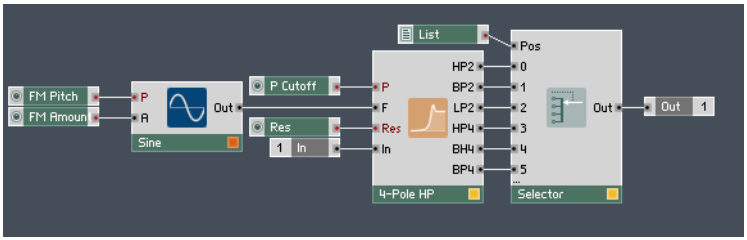


Fig. 10.18 A simple implementation of a 4-pole filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.12 Pro-52 Filter

Fig. 10.19 Pro-52 Module

10.12.1 Overview

The Pro-52 Module embodies the filter taken from the Pro-52 virtual analog synthesizer. It is a 4-pole low-pass filter corresponding to a 24 dB per octave attenuation above the cutoff frequency. The cutoff frequency is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.

The Pro-52 Module also enables you to vary the filter resonance. Resonance corresponds to a peak in the frequency response at the cutoff frequency and the resonance parameter lets you control the width / height of this peak in respect to the pass band. At high resonance settings the filter output exhibits a strong frequency component (nearly in the form

of a sine tone) at the cutoff frequency. The filter goes into self-oscillation when "Res" (resonance) approaches the value "1". The amplitude of the self-oscillation is approximately "1".

Application

The distinct sound of a subtractive synthesizer is in large part due to the type of filters used. The Pro-52 Module helps you reproduce the sound of the Pro-52 virtual analog synthesizer by emulating the filter component used in that synthesizer. Additionally, you can use the "F" (cutoff frequency) input port for frequency modulation of the cutoff frequency. Modulating this input port at Audio Rate yields an FM'ish sound.

10.12.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency range of [26 Hz … 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 … 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

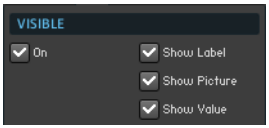
- **(Out)** "Out" is the Audio output port for the 4-pole low-pass filtered signal.

10.12.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.12.4 Example: Pro-52 4-Pole Low-Pass Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a Pro-52 4-pole low-pass filter. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter

Module. The cutoff pitch is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port and the resonance parameter is specified with the Knob Module labeled "Reson" at the "Res" input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module.

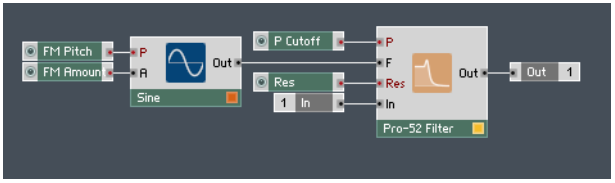


Fig. 10.20 A simple implementation of a Pro-52 4-pole low-pass filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.13 Ladder Filter

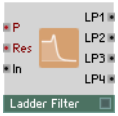


Fig. 10.21 Ladder Filter Module

10.13.1 Overview

The Ladder Filter Module embodies the filter modeled after the classic ladder circuit patented by Bob Moog. It is a 4-pole filter with low-pass outputs for the 1-pole, 2-pole, 3-pole, and 4-pole characteristic. Each additional pole in the filter contributes to an additional attenuation of 6 dB per octave above the cutoff frequency. The saturation characteristic of the analog circuit can optionally be simulated by activating the Distortion feature in the Module's Function page. When the Distortion is enabled, the filter goes into self-

oscillation when the value of "Res" (resonance) is "1" or more. The amplitude of the self-oscillation is approximately "1" when "Res" (resonance) is just above "1", but can be much larger when driving "Res" (resonance) even higher. The filter also has options in the Module's Function page for selecting the quality of the simulation. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input.

Application

The distinct sound of a synthesizer is in large part due to the type of filters used. The Ladder Filter Module helps you reproduce the sound of the Moog analog synthesizer by emulating the filter component used in that synthesizer. For an example Structure, please refer to the Ladder Filter FM Module's example.

10.13.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency range of [26 Hz … 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 … 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] $\cong$ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

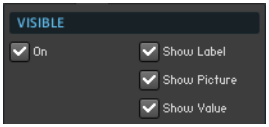
- **(LP1)** "LP1"(1-pole low-pass) is the Audio output port for the 1-pole low-pass filtered signal.
- **(LP2)** "LP2"(2-pole low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(LP3)** "LP3"(3-pole low-pass) is the Audio output port for the 3-pole low-pass filtered signal.
- **(LP4)** "LP4"(4-pole low-pass) is the Audio output port for the 4-pole low-pass filtered signal.

10.13.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

Setting the Simulation Quality

The filter also has options in the Module's Function page for selecting the quality of the simulation: Standard, High and Excellent. The effect is particularly noticeable when distortion is turned on. Of course, the higher quality settings come at the price of increased CPU usage.

► To set the quality of filter simulation, go to the Module's Function page and choose the desired quality from the Quality drop-down menu, shown in the picture below.

Enabling the Distortion Feature

The saturation characteristic of the analog circuit can be simulated by activating the Distortion feature in the Module's Function page.

► To activate the Distortion feature, go to the Module's Function page and engage the Distortion and Self-Oscillation checkbox, shown in the figure above.

10.14 Ladder Filter FM

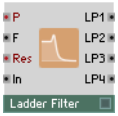


Fig. 10.22 Ladder Filter FM Module

10.14.1 Overview

The Ladder Filter FM Module embodies the filter modeled after the classic ladder circuit patented by Bob Moog. It is a 4-pole filter with low-pass outputs for the 1-pole, 2-pole, 3-pole, and 4-pole characteristic. Each additional pole in the filter contributes to an additional attenuation of 6 dB per octave above the cutoff frequency. The saturation characteristic of the analog circuit can optionally be simulated by activating the Distortion feature in the Module's Function page. When the Distortion is enabled, the filter goes into self-oscillation when the value of "Res" (resonance) is "1" or more. The amplitude of the self-oscillation is approximately "1" when "Res" (resonance) is just above "1", but can be much larger when driving "Res" (resonance) even higher. The filter also has options in the Module's Function page for selecting the quality of the simulation. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.

Application

The distinct sound of a synthesizer is in large part due to the type of filters used. The Ladder Filter FM Module helps you reproduce the sound of the Moog analog synthesizer by emulating the filter component used in that synthesizer. Additionally, you can use the "F" (cutoff frequency) input port for frequency modulation of the cutoff frequency. Modulating this input port at Audio Rate yields an FM'ish sound.

10.14.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.

- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 ... 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(In)** "In" is the Audio input port for the signal to be filtered.

Output Ports

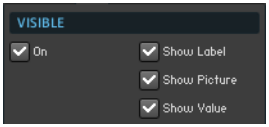
- **(LP1)** "LP1"(1-pole low-pass) is the Audio output port for the 1-pole low-pass filtered signal.
- **(LP2)** "LP2"(2-pole low-pass) is the Audio output port for the 2-pole low-pass filtered signal.
- **(LP3)** "LP3"(3-pole low-pass) is the Audio output port for the 3-pole low-pass filtered signal.
- **(LP4)** "LP4"(4-pole low-pass) is the Audio output port for the 4-pole low-pass filtered signal.

10.14.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency and resonance parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

Setting the Simulation Quality

The filter also has options in the Module's Function page for selecting the quality of the simulation: Standard, High and Excellent. The effect is particularly noticeable when distortion is turned on. Of course, the higher quality settings come at the price of increased CPU usage.

► To set the quality of filter simulation, go to the Module's Function page and choose the desired quality from the Quality drop-down menu, shown in the picture below.

Enabling the Distortion Feature

The saturation characteristic of the analog circuit can be simulated by activating the Distortion feature in the Module's Function page.

► To activate the Distortion feature, go to the Module's Function page and engage the Distortion and Self-Oscillation checkbox, shown in the figure above.

10.14.4 Example: Ladder Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a ladder filter that can be "tapped" at the output of each pole with low-pass characteristic. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The cutoff pitch is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port and the resonance parameter is specified with the Knob Module labeled "Reson" at the "Res" input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module. Since the Ladder Filter Module comprises four 1-pole low-pass "stages", there are outputs for 1-pole, 2-pole, 3-pole, and 4-pole low-pass filter characteristics. A List Module in combination with a Selector Module lets you select from the Instrument Panel which filtered signal is sent to the output.

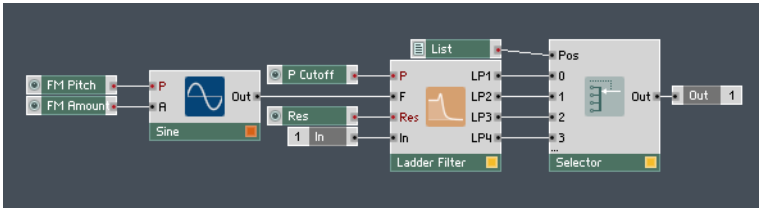


Fig. 10.23 A simple implementation of a ladder filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible.

10.15 Modal Bank

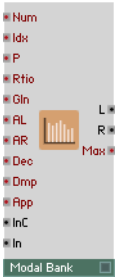


Fig. 10.24 Modal Bank Module

10.15.1 Overview

The Modal Bank Module is a bank of parallel resonant filters, called partials. This Module enables you to perform modal synthesis with REAKTOR. For each partial, its frequency ratio to the fundamental, its input and output amplitudes, and its decay time can be specified at the “Rtio”, “GIn”, “AL” and “AR”, and “Dec” input ports, respectively. Setting these parameters for each partial is done serially and with the use of the “Idx” input port to specify to which partial the incoming parameters should be sent. The fundamental pitch of the Modal Bank is specified at the “P” input; it corresponds to the frequency in respect to which the frequencies of all the partials are calculated. Also, the number of partials that the Modal Bank Module generates has to be specified, either in the Function page of the Module's Properties or at the “Num” input port. The “Dmp” input port is used for the damping value which represents a global value that is subtracted from the decay times of all partials. After sending all the parameters to the corresponding input ports, an “apply” Event has to be sent to the “App” input port to effectively apply these changes to the partials. Before the exciter signal reaches a partial, it is multiplied by the “GIn” factor specified at that partials index. After the exciter signal has passed through the filter of the corresponding partial, the resulting stereo Audio signal is sent to the “L” and “R” output ports for the left and right stereo channels, respectively. Also, in the Function page of the Module's Properties, one can specify the smoothing time and smoothing quality. The smoothing quality setting has an effect on the Module's CPU usage.

The default parameters for the partials are such that the “L” output sends a pulse wave with the pitch of the fundamental (set at the “P” input port) and the “R” output sends a saw wave with the pitch one octave above the fundamental. By adding both signals together you get a saw wave at the pitch of the fundamental.

Application

The Modal Bank Module is meant for modal synthesis which is a synthesis method that basically consists of an exciter signal and a resonator. This system can be great for synthesizing percussive sounds ranging from mallets to vibraphones, but also for sustain sounds like woodwinds. The Modal Bank Module can also be used in a feedback circuit, extending its sonic possibilities even further. As with the Sine Bank Module, some sounds require a lot of data to be sent to the partials of the Modal Bank Module. However, this is not the case for all sounds. The strength of the Modal Module lies in the fact that depending on the exciter signal, it is possible to create complex output signals with limited use of real-time modulation of the partial parameters. Often a large number of partials are used; the maximum number of partials is 10'000. Usually much less partials than this maximum number are used, mainly because of CPU limitations. In a case where there are a lot of partials it is most effective to make use of the Iterator Module (found in the *Built-In Module > Event Processing* submenu) to supply all partials with the necessary parameters between the processing of subsequent audio samples.

Please refer to section 12.3 in the Application Reference for a short tutorial on creating a simple modal synthesizer.

10.15.2 Ports

Input Ports

- **(Num)** “Num” (number of partials) is the Event input port for setting the number of active partials in the Modal Bank Module. This value should be a positive integer in the range [0 … Max] where “Max” corresponds to the number set in the Max. Number edit field in the Function page of the Module's Properties. If this input port is left disconnected, the Modal Bank Module automatically sets the number of partials to the value to “Max”. A higher number of active partials requires more CPU resources. Often the CPU load is increased for multiples of 4 partials.

- **(Idx)** “Idx” (index) is the Event input port for specifying the partial number to which the subsequent parameters arriving at the “Rtio”, “AL”, “AR”, and “Ph” input ports should be routed. This value should be a positive integer in the range [1 … Num] where “Num” denotes the number of active partials (see “Num” input port).
- **(P)** “P” (pitch) is the Event input port for controlling the fundamental pitch of the Modal Bank Module. The frequency corresponding to the fundamental pitch is the frequency in respect to which the frequencies of all the partials are calculated. The values at this input port are set in the logarithmic pitch scale and are usually in the range [0 … 127].
- **(Rtio)** “Rtio” (frequency ratio) is the Event input port for setting the ratio of the partial's frequency to the fundamental frequency determined at the “P” input port. If the “Rtio” value of a partial is “1”, it means that the ratio between the partial's frequency and the fundamental frequency is “1”, that is, the partial has the fundamental frequency. If the “Rtio” value of a partial is “2”, then the partial has double the frequency of the fundamental. In general, the frequency of the partial (“f”) can be calculated from the fundamental frequency (“F”) and the frequency ratio (“Rtio”) with the following formula: f = F * Rtio.
- **(GIn)** “GIn” (gain in) is the Event input port for specifying the gain factor with which the exciter signal (at the “In” or “InP” input ports) is multiplied before it is sent to the corresponding partial's filter. This means that each partial has its own “GIn” value associated with it. Using “GIn” values you can consecutively excite different partials without suppressing the output signals of the already excited partials.
- **(AL)** “AL” (amplitude, left channel) is the Event input port for specifying the partial amplitude in the left stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “L” output port. The typical range is [0 … 1], but negative values are also allowed; they cause the waveform of the corresponding partial to be inverted.
- **(AR)** “AR” (amplitude, right channel) is the Event input port for specifying the partial amplitude in the right stereo channel. It is the value that the unit amplitude oscillator waveform of the partial is multiplied by before being mixed together with the other partials and forwarded to the “R” output port. The typical range is [0 … 1], but negative values are also allowed; they cause the waveform of the corresponding partial to be inverted.

- **(Dec)** “Dec” (decay) is the Event input port for specifying the decay time (resonance) of the filter associated with the addressed partial. The decay time is specified in the logarithmic scale, just like the envelope time inputs of the envelope Modules in the Primary Level. It is the time (in milliseconds) that the decay curve requires to reach the factor “1 / e” in respect to its initial (maximum) value. Here “e” denotes the mathematical constant known as Euler's number. It is approximately “2.72”. The typical range for the “Dec” input of the Modal Bank Module is [0 ... 100] where a value of “0” corresponds to a decay time of 1 ms, a value of “20” corresponds to a decay time of 10 ms, a value of “200” corresponds to a decay time of 100 ms, and so on. The formula for calculating the decay time (“T”) from the value at the “Dec” input port is T = 10 ^ (Dec / 20) ms.
- **(Dmp)** “Dmp” (damping) is the Event input port for specifying the global damping parameter. The damping value is subtracted from the “Dec” values of all partials. Positive damping values reduce the decay times. Usually the damping parameter is zero at the beginning of the attack of the exciter signal and then is set to a positive value to “damp” the envelopes of the outgoing signal at some later point. Of course, after sending the new value to the “Dmp” input port, an “apply” Event needs to be sent to the “App” input port in order for the damping value to take effect in the behavior of the Module.
- **(App)** “App” (apply) is the Event input port for applying the latest parameter values that have arrived at the various input ports to the actual filters inside the Modal Bank Module. The parameter values sent to the Modal Bank Module are stored in a temporary buffer and only applied to the partials once an Event arrives at the “App” input port. If the value of the Event is greater than zero, the apply operation sets the state variables of all the filters to zero, thus working like a “reset”. If the “apply” Event is less or equal to zero, all new parameters are applied but the filters retain their state variable.
- **(InP)** “InP” (amplitude compensated audio input) is the Audio input port that is usually used for sustained exciter signals that would be too loud when passed through partial filters with low decay times (resonance). The amplitude compensation was calibrated with a white noise signal of constant amplitude which has equal power at all frequencies. Feeding oscillator signals (like a saw wave) into this input port will not necessarily yield good amplitude compensation since the spectrum of such an oscillator is very different from that of white noise.

- **(In)** “In” (audio input) is the Audio input port for that is usually used for impulse-like exciter signals. This input was calibrated using one cycle of a sine wave as an impulse signal.

Output Ports

- **(L)** “L” (left stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AL” parameter.
- **(R)** “R” (right stereo channel) is the output port for the Audio signal that is a mix of all the partials with amplitudes set by the “AR” parameter.
- **(Max)** “Max” (maximum number of partials) is the Event output port for the value set in the Max. Number edit field of the Module's Function page.

10.15.3 Properties: Function Page

Setting the Maximum Number of Partials

The maximum number of partials that the Modal Bank Module generates has to be specified in the Max. Number edit field in the Module's Function page, shown below. The actual number of active partials is specified at the “Num” input port. “Num” values larger than the value set in the Max. Number edit field are clipped. If the “Num” input port is left disconnected, the Modal Bank Module automatically sets the number of partials to the value specified by the Max. Number edit field. The default value for this edit field is “128”. Also, the value in the Max. Number edit field is sent to the “Max” output port. A higher number of active partials requires more CPU resources.

Due to internal optimization, for many computers the CPU load is increased when the number of partials reaches a multiple of 4 partials.

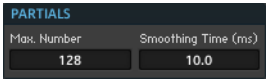


Fig. 10.25 The maximum number of partials and the smoothing time are set in the Partials area of the Modal Bank Module's Function page.

Setting the Smoothing Time

The smoothing time defines the time-interval over which the partial amplitudes are faded between different values. This parameter can be set in the Smoothing Time edit field. The units for this edit field are milliseconds and the default value is 10 ms.

Setting the Smoothing Quality

You can specify the quality of the smoothing curves with the Smoothing Quality drop-down menu. Higher smoothing quality causes the Module to respond better to fast modulations of the partial amplitudes. However, increasing the smoothing quality also increases the CPU load.

Fig. 10.26 The Smoothing Quality drop-down menu lets you increase the quality of the Module's sound in response to fast modulations of the partial amplitudes.

10.16 Peak EQ

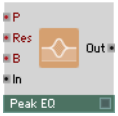


Fig. 10.27 Peak EQ Module

10.16.1 Overview

The Peak EQ Module is a parametric equalizer with adjustable boost and band width (resonance) parameters. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.

Application

With the Peak EQ Module, a specific frequency in the signal and a narrow or wide band of frequencies around it can be amplified or attenuated. More distant frequencies are not affected. For negative "B" (boost) values, the Peak EQ Module acts similar to a notch filter. Since the "B" (boost) value can be varied (and even made positive), along with the "Res"

(resonance) parameter, the Peak EQ Module gives more flexibility in attenuating or amplifying certain frequency components in an incoming signal. For an example Structure, please refer to the Peak EQ FM Module's example.

10.16.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency range of [26 Hz … 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 … 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(B)** "B" (boost) is the Event input port for controlling boost / attenuation of the peak, in dB. At the (default) value B = 0 the signal remains unchanged. The typical range for values at the "B" (boost) input port is [-20 … 20].
- **(In)** "In" is the Audio input port for the signal to be equalized.

Output Ports

- **(Out)** "Out" is the Audio output port for the equalized signal.

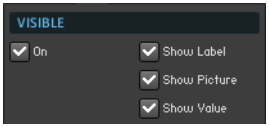
10.16.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integra-

tor and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency, resonance, and boost parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.17 Peak EQ FM

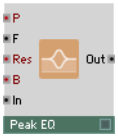


Fig. 10.28 Peak EQ FM Module

10.17.1 Overview

The Peak EQ Module is a parametric equalizer with adjustable boost and band width (resonance) parameters. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input.

Application

With the Peak EQ Module, a specific frequency in the signal and a narrow or wide band of frequencies around it can be amplified or attenuated. More distant frequencies are not affected. For negative "B" (boost) values, the Peak EQ Module acts similar to a notch filter. Since the "B" (boost) value can be varied (and even made positive), along with the "Res" (resonance) parameter, the Peak EQ Module gives more flexibility in attenuating or amplifying certain frequency components in an incoming signal.

10.17.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 … 120] which roughly corresponds to a frequency range of [26 Hz … 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(Res)** "Res" (resonance) is the Event input port for controlling the filter's resonance / damping parameter. The range of values at this input port is [0 … 1] where "0" corresponds to maximal damping, no resonance, and "1" to no damping, maximal resonance. At Res = 1 the filter begins to self-oscillate at the cutoff frequency, resulting in a sine tone at that frequency. For high resonance values the filter's Q-factor = frequency [Hz] / bandwidth [Hz] ≅ 1 / (2 – 2*Res).
- **(B)** "B" (boost) is the Event input port for controlling boost / attenuation of the peak, in dB. At the (default) value B = 0 the signal remains unchanged. The typical range for values at the "B" (boost) input port is [-20 … 20].
- **(In)** "In" is the Audio input port for the signal to be equalized.

Output Ports

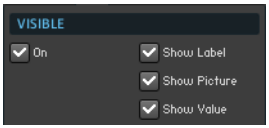
- **(Out)** "Out" is the Audio output port for the equalized signal.

10.17.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency, resonance, and boost parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.17.4 Example: Peak Equalizer

This example shows how to filter an incoming signal, such as an oscillator output signal, with a peak equalizer. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The cen-

ter frequency of the peak is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port, the resonance parameter (peak width) is specified with the Knob Module labeled "Reson" at the "Res" input port, and the amplification or attenuation at the peak frequency is set with the "Boost" Knob Module, at the "B" (boost) input port. Simple frequency modulation of the filter's peak frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module.

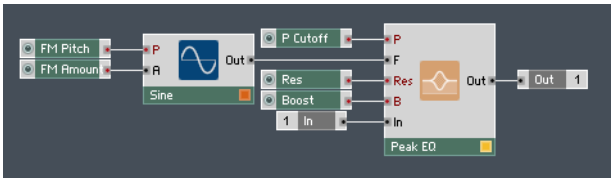


Fig. 10.29 A simple implementation of a peak equalizer.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible, shown in the figure below.

Fig. 10.30 The Peak EQ FM Module's Panel representation with a positive "Boost" value exhibits a positive peak.

10.18 High Shelf EQ

Fig. 10.31 Hi Shelf EQ Module

10.18.1 Overview

The Hi Shelf EQ Module is a parametric equalizer with a high shelving characteristic. The "B" (boost) parameter determines the amplification or attenuation of the frequencies above the cutoff frequency. Frequencies below the cutoff frequency are not affected. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input port.

Application

With the Hi Shelf EQ Module you can amplify or attenuate high frequency components without affecting the lower frequencies. This Module is generally useful in making a signal "brighter" or "duller". For an example Structure, please refer to the High Shelf EQ FM Module's example.

10.18.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(B)** "B" (boost) is the Event input port for controlling boost / attenuation of the "shelf", in dB. At the (default) value B = 0 the signal remains unchanged. The typical range for values at the "B" (boost) input port is [-20 ... 20].
- **(In)** "In" is the Audio input port for the signal to be equalized.

Output Ports

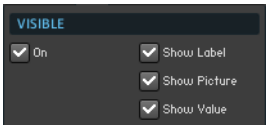
- **(Out)** "Out" is the Audio output port for the equalized signal.

10.18.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency, resonance, and boost parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.19 High Shelf EQ FM

Fig. 10.32 Hi Shelf EQ FM Module

10.19.1 Overview

The Hi Shelf EQ FM Module is a parametric equalizer with a high shelving characteristic. The "B" (boost) parameter determines the amplification or attenuation of the frequencies above the cutoff frequency. Frequencies below the cutoff frequency are not affected. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.

Application

With the Hi Shelf EQ FM Module you can amplify or attenuate high frequency components without affecting the lower frequencies. This Module is generally useful in making a signal "brighter" or "duller".

10.19.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.

- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(B)** "B" (boost) is the Event input port for controlling boost / attenuation of the "shelf", in dB. At the (default) value B = 0 the signal remains unchanged. The typical range for values at the "B" (boost) input port is [-20 ... 20].
- **(In)** "In" is the Audio input port for the signal to be equalized.

Output Ports

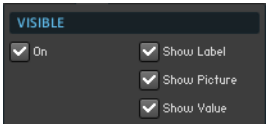
- **(Out)** "Out" is the Audio output port for the equalized signal.

10.19.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency, resonance, and boost parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.19.4 Example: High Shelf Equalizer

This example shows how to filter an incoming signal, such as an oscillator output signal, with a high shelf equalizer. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The "starting" (cutoff) frequency of the high shelf is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port, and the amplification or attenuation at the shelf is set with the "Boost" Knob Module, at the "B" (boost) input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module.

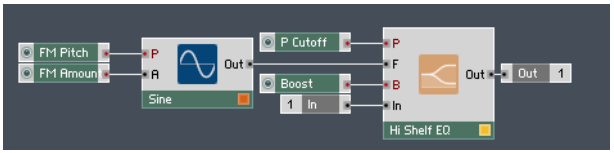


Fig. 10.33 A simple implementation of a high shelving filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On check-box in the View page has been engaged such that the graphic Panel representation of the frequency response is visible, shown in the figure below.

Fig. 10.34 The High Shelf EQ FM Module's Panel representation with a positive "Boost" value.

10.20 Low Shelf EQ

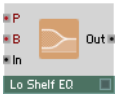


Fig. 10.35 Low Shelf EQ Module

10.20.1 Overview

The Lo Shelf EQ Module is a parametric equalizer with a low shelving characteristic. The "B" (boost) parameter determines the amplification or attenuation of the frequencies below the cutoff frequency. Frequencies above the cutoff frequency are not affected. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input.

Application

With the Lo Shelf EQ Module you can amplify or attenuate low frequency components without affecting the higher frequencies. This Module is generally useful in reducing low frequencies in or adding a "bass boost" to an incoming signal. For an example Structure, please refer to the Low Shelf EQ FM Module's example.

10.20.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The

pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.

- **(B)** "B" (boost) is the Event input port for controlling boost / attenuation of the "shelf", in dB. At the (default) value B = 0 the signal remains unchanged. The typical range for values at the "B" (boost) input port is [-20 ... 20].
- **(In)** "In" is the Audio input port for the signal to be equalizer.

Output Ports

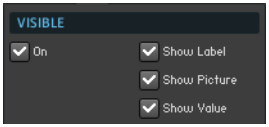
- **(Out)** "Out" is the Audio output port for the equalized signal.

10.20.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency, resonance, and boost parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.21 Low Shelf EQ FM

Fig. 10.36 Lo Shelf EQ FM Module

10.21.1 Overview

The Lo Shelf EQ FM Module is a parametric equalizer with a low shelving characteristic. The "B" (boost) parameter determines the amplification or attenuation of the frequencies below the cutoff frequency. Frequencies above the cutoff frequency are not affected. The cutoff frequency for this Module is determined in the logarithmic scale at the "P" (cutoff pitch) input port and can be modulated in the linear frequency scale using the "F" (cutoff frequency) input port. Only the graphic display stemming from the "P" (cutoff pitch) value is shown in the Panel representation.

Application

With the Lo Shelf EQ FM Module you can amplify or attenuate low frequency components without affecting the higher frequencies. This Module is generally useful in reducing low frequencies in or adding a "bass boost" to an incoming signal.

10.21.2 Ports

Input Ports

- **(P)** "P" (cutoff pitch) is the Event input port for controlling the cutoff frequency in the logarithmic scale. The cutoff pitch is given in semitones, usually in the range [20 ... 120] which roughly corresponds to a frequency range of [26 Hz ... 8370 Hz]. The pitch value "69" corresponds to "Concert A" or 440 Hz. If left disconnected, the default value of "0" is used for this input port. This default value corresponds to 8 Hz in the frequency scale.
- **(F)** "F" (cutoff frequency) is the Audio input port for linear modulation of the cutoff frequency in Hz. The value at the "F" (frequency) input port is added to the frequency determined by the "P" (cutoff pitch) input port. The sum is then the final cutoff frequency.
- **(B)** "B" (boost) is the Event input port for controlling boost / attenuation of the "shelf", in dB. At the (default) value B = 0 the signal remains unchanged. The typical range for values at the "B" (boost) input port is [-20 ... 20].
- **(In)** "In" is the Audio input port for the signal to be equalized.

Output Ports

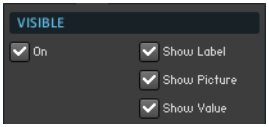
- **(Out)** "Out" is the Audio output port for the equalized signal.

10.21.3 Properties: View Page

Making the Frequency Response Graph Visible on the Panel

One way to graphically characterize a filter is to plot its frequency response. On the horizontal axis you have the frequencies of the signal, in the logarithmic scale. On the vertical axis you have the corresponding frequency component's magnitude in respect to the incoming magnitude (also in the logarithmic, decibel scale). All of REAKTOR's filters (except the all-pass filter, which has a flat frequency (magnitude) response, and the Integrator and Differentiator) have this characteristic plot as a Panel representation. The frequency range of this plot in REAKTOR is 10 Hz to 20 kHz. This can be a very useful component in your Instrument since you get direct visual feedback on how the cutoff frequency, resonance, and boost parameters affect the overall frequency response curve.

► To turn on the graphic display of your filter's frequency response in the Instrument Panel, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Filter's Panel Representation

Depending on your Panel layout, you want a small or large graph of the frequency response on your Instrument Panel.

► To change the width and height of the filter's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

10.21.4 Example: Low Shelf Equalizer

This example shows how to filter an incoming signal, such as an oscillator output signal, with a low shelf equalizer. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. The "upper" (cutoff) frequency of the low shelf is specified with the Knob Module labeled "P Cutoff" at the "P" (pitch) input port, and the amplification or attenuation at the shelf is set with the "Boost" Knob Module, at the "B" (boost) input port. Simple frequency modulation of the filter's cutoff frequency is achieved by connecting the output of a sine oscillator to the "F" (frequency) input port of the Module.

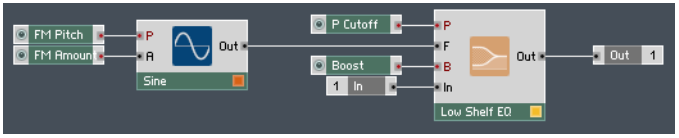


Fig. 10.37 A simple implementation of a low shelving filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list. Also, the On checkbox in the View page has been engaged such that the graphic Panel representation of the frequency response is visible, shown in the figure below.

Fig. 10.38 The Low Shelf EQ FM Module's Panel representation with a positive "Boost" value.

10.22 Differentiator

Fig. 10.39 Differentiator Module

10.22.1 Overview

The Differentiator Module gives the slope of the input signal in change units per millisecond.

Application

The effect of the Differentiator Module is like a high-pass filter where the amplification is proportional to the frequency. Unity gain (no amplification or attenuation) happens at the frequency 159 Hz. It is usually used to determine if the incoming signal is rising or falling.

10.22.2 Ports

Input Ports

- **(In)** "In" is the Audio input port for the signal to be differentiated.

Output Ports

- **(Out)** "Out" is the Audio output port for the differentiated signal.

10.22.3 Example 1: Differentiator Audio Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with a differentiator. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. A Mixer Module has been placed between after Differentiator output to safeguard from abrupt jumps in the output signal by attenuating the signal.

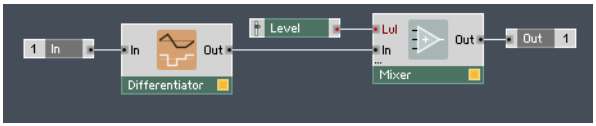


Fig. 10.40 A simple implementation of a Differentiator Module as a high-pass filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list.

10.22.4 Example 2: Envelope Follower

A good way to modulate your effects, filters, envelopes, and oscillators, is using envelope followers. An envelope follower takes an input signal and uses its peaks to trigger an envelope whose amplitude matches that of the peak. The Structure in the figure below illustrates such an envelope follower Structure, utilizing the Peak Detector (↑12.14, Peak Detector), Differentiator (↑12.14, Peak Detector), Sample and Hold (↑12.15, Sample & Hold), HR Envelope (↑9.4, HR - Env), and a 1-Pole Filter (↑10.2, HP/LP 1-Pole FM).

The incoming signal that is to trigger the envelope is first fed into the Peak Detector Module (↑12.14, Peak Detector). Depending on the frequency of peaks, the release time of the peaks in the Peak Detector's output signal is set with the "PkRel" knob at the "Rel" input port. Next, the Peak Detector Module's output is fed to the Sample and Hold Module (↑12.15, Sample & Hold). This signal is only sampled and held when a negative value at the Sample and Hold Module's "Trig" (trigger) input port goes from a negative value to a positive value. At the same time, the Differentiator Module records if the Peak Detector Module's output is rising or falling. Therefore, if the Peak Detector signal goes from a falling state to a rising state, its output is sampled and forwarded to the "A" (amplitude) input port of the HR Envelope Module (↑9.4, HR - Env). At the same time, the HR Envelope Module is triggered (with this new amplitude, ultimately received from the Peak Detector Module). The envelope's output signal is then smoothed by a 1-Pole Filter (low-pass) where the value at its "P" (pitch) input port determines the amount of smoothing.

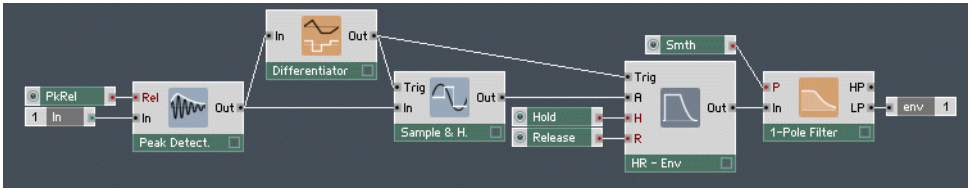


Fig. 10.41 The Structure for an envelope follower effect.

10.23 Integrator

Fig. 10.42 Integrator Module

10.23.1 Overview

The Integrator Module embodies an integration function, including an input port to reset the state of the function ("Set"). The outgoing signal is the sum of the values at the "In" Audio input port, sampled at every millisecond.

Amplification

Integrator circuits are common components in analog circuits. The effect of an integrator on an incoming signal is like a low-pass filter where amplification is proportional to 1 / frequency. Unity gain (no amplification or attenuation) happens at 159 Hz. It is usually used when the values of subsequent Audio signal values continuously need to be summed.

Note that this Module outputs the sum of the incoming values. This means that if only positive or only negative values are present at the input port, the output signal of the Integrator Module "explodes"!

10.23.2 Ports

Input Pors

- **(Set)** "Set" is the Event input for the reset action. When an Event is received, the internal state and output signal of the Module is set to the value of the Event.
- **(In)** "In" is the Audio input port for the signal to be integrated.

Output Ports

- **(Out)** "Out" is the Audio output port for the integrated signal.

10.23.3 Example: Integrator Audio Filter

This example shows how to filter an incoming signal, such as an oscillator output signal, with an integrator. The corresponding Structure is shown in the figure below. The signal which is to be filtered is incoming at the "In" input port of the Filter Module. A Mixer Module has been placed between after Integrator output to safeguard from an "exploding" output signal by attenuating that signal.

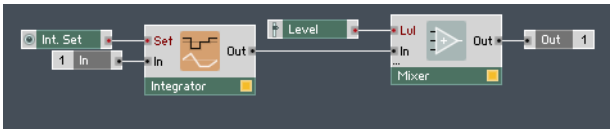


Fig. 10.43 A simple implementation of an Integrator Module as a low-pass filter.

In the Rehearsal Ensemble you can get acquainted with the various filters in REAKTOR. The Ensemble consists of an easy to use interface to select and tweak your filter of choice. Since each filter has its own set of controls for the signal levels and transition times, Stacked Macro Modules (↑1.19, Stacked Macro) have been used to make only the relevant controls appear when a filter has been selected from the "Filters" list.

11 Delay

The delay is a fundamental entity in digital signal processing. Even filters use delays in their internal structures. REAKTOR provides six types of Delay Modules to accomplish most delay functions. Although for tasks like building your own filters we suggest that you use REAKTOR Core, most common tasks such as building comb filters can be done using REAKTOR Primary Level Delay Modules. Those include two tap delays, two granular delays, a diffuser delay (handy for reverb circuits), and a unit delay useful in signal processing with feedback loops and in physical modeling.
All Modules in this section have hybrid input port, meaning they can be used as either Audio or Event input ports, depending on the type of signal connected to them. You can identify hybrid input and output ports by their green color.

11.1 Single Delay

Fig. 11.1 Single Delay Module

11.1.1 Overview

The Single Delay Module is a polyphonic delay line for Audio and Event signals. The input signal (from the "In" input port) appears at the output port with a delay corresponding to the set time. The delay time is controlled at the "Dly" (delay) input port.
The upper limit for the delay time can be adjusted in the Module's Function page. How big this value can be set depends on the amount of available RAM (random access memory) storage in your computer. At a sample rate of 44.1 kHz you need 172 kB of RAM for each second of buffer length and per Polyphonic Voice. For a one minute delay you need 10 MB of RAM per Voice.

Application

This module replaces several Modules of former REAKTOR version:

- It behaves like a REAKTOR 3 Static Delay Module if you connect an Audio signal to the "In" input port and an Event signal to the "Dly" (delay) input port. If the delay time does not correspond to an integer number of samples, the output signal is generated by interpolation. A setup where you would use this type of functionality would be a delay effect with a delay time that is set via a knob on the Instrument Panel.
- It behaves like a REAKTOR 3 Modulation Delay Module if you connect an Audio signal to the "In" input port and an Audio signal to the "Dly" (delay) input port. The delay time can be continuously modulated by an Audio signal. If the delay time does not correspond to an integer number of samples, the output signal is generated by interpolation. An exemplary structure where a modulation of the delay time is necessary is that of a chorus effect.
- It behaves like a REAKTOR 3 Event Delay Module if you connect an Event signal to the "In" input port. Use the Delay Module for Events when, for example, you wish to create a MIDI effect that creates echoes of incoming MIDI Events.

11.1.2 Ports

Input Ports

- **(Dly)** "Dly" (delay) is the hybrid input port for controlling the delay time. The values at this input port are set in milliseconds and should be in the range from "0" to the maximum delay time set in the Module's Function page. Negative values are not allowed, since a negative delay time would require a time machine, not to mention that using a future sample to calculate that same future sample would get you stuck in an infinite cause-and-effect-loop.
- **(In)** "In" is the hybrid input port for the signal to be delayed.

Output ports

- **(Out)** "Out" is the hybrid output port for the delayed signal.

11.1.3 Properties: Function Page

Setting the Maximum Delay Time

The maximum delay time that can be set at the "Dly" (delay) input port is set in the Module's Function page. The default value for this edit field is 1 second. The greater the maximum delay time is, the more of the Module's input signal is buffered in your computer's RAM. Very large maximum delay time values can have a detrimental effect on your computer's memory usage.

► To set the maximum delay time for the Single Delay Module, go to the Module's Function page and enter the desired buffer value into the Buffer edit field, shown in the figure below.

If the Delay is used as an Event Delay (in this case the "In" input port will be red, indicating that the Module is in Event processing mode) the Buffer edit field will be set in units of Events (see the figure below). Effectively you will then set the maximum number of buffered Events with this edit field.

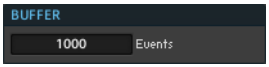


Fig. 11.2 If your Single Delay Module is working as an Event Module, the Buffer edit field applies to the maximum number of buffered Events.

Setting the Interpolation Quality

If the delay time at the "Dly" (delay) input port does not correspond to an integer number of samples, the output signal is generated by interpolation. The interpolation method can be chosen in the Module's Function page.

► To set the interpolation used by your Single Delay Module, go to the Module's Function page and choose the desired interpolation mode from the Quality drop-down menu.

Linear interpolation may reduce high frequency components in the sound somewhat.

11.1.4 Example: Comb Filter

A common application for the Single Delay Module is in a comb filter Structure. A comb filter has a frequency response that looks similar to a comb. This means that certain frequencies at regular intervals are amplified, whereas other frequencies are attenuated. The amplification peaks are multiples of the "base frequency" of the comb filter, determined by the delay time set at the Single Delay Module's "Dly" (delay time) input port.

The Structure of a comb filter is simple. As can be seen in the figure below, the incoming "dry" signal is combined with the feedback signal using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) and fed into the Single Delay Module. With the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) you multiply the feedback signal with a value in the range [0 … 1] (received from the "Feedback" knob) and so determine the amount of feedback signal that is sent to the Single Delay Module. Since feedback Structures can cause the signal to "explode", a Saturator Module (↑12.1, Saturator) has been placed between the Single Delay Module's output and the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+). This way the feedback signal never exceeds the bounds [-2 … 2].

The "base frequency" is calculated from a pitch value which is received from the knob labeled "pitch" (range [1 … 128]) and converted to the linear frequency scale using the Exp P-to-F Module (↑4.16, Exp (P-to-F)). The output of the Exp P-to-F Module delivers the pitch in Hertz (oscillations per second) but the we need the corresponding delay time in milliseconds. Therefore we use the Divide Module to divide 1000 by the pitch value in Hertz, giving us an Event signal that gives the delay time in milliseconds. Subsequently we convert this Event signal to an Audio Signal using the Audio Smoother Module (↑14.14, Audio Smoother).

Last, the "wet" signal, derived from the Single Delay Module's output, is added to the "dry" incoming signal. This is done again using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+), where the amount of "wet" signal mixed to the "dry" signal can be controlled with the "Mix" knob (range [0 … 1]).

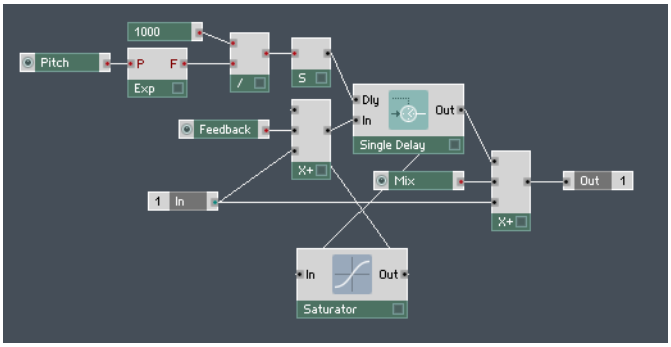


Fig. 11.3 The Structure of a comb filter.

Don't forget to choose the interpolation quality with the Interpolation drop-down menu in the Delay Module's Function page!

11.2 Multi-Tap Delay

Fig. 11.4 Multi-Tap Module

11.2.1 Overview

The Multi-Tap Module is a multi-tap delay line for Audio signals. The input signal (from the "In" input port) appears at the output ports, each with a delay set at the corresponding input port. The delay time is controlled at the "Dly1" to "Dly8" (delay1 to delay 8) input ports.

The upper limit for the delay time can be adjusted in the Module's Function page. How big this value can be set depends on the amount of available RAM (random access memory) storage in your computer. At a sample rate of 44.1 kHz you need 172 kB of RAM for each second of buffer length and per Polyphonic Voice. For a one minute delay you need 10 MB of RAM per Voice.
If the set delay time corresponds to a non-integer number of sample, interpolation occurs. The interpolation method can be set in the properties.

Application

This Module helps you recreate the multi-tap delay effect. You could connect the Multi-Tap Module's output ports to a Mixer Module or, as shown in the example, connect them to a Selector Module and use an LFO to "scan" through the different delay taps.
Interesting effects can also be achieved by mixing the delayed signal from one or more delay taps back into the input signal. This way, dense delay clusters can be produced. Combining such a Structure with LFOs or envelopes that modulate they delay times also yields interesting results.

Note that when using feedback, it is smart to attenuate the feedback signal so that your signal doesn't explode!

Input Ports

- **(In)** "In" is the Audio input port for the signal to be delayed.
- **(Dly1)** "Dly1" (delay 1) is the Audio input port for controlling the delay time of the first delay tap in milliseconds. The values at this input port are set in milliseconds and should be in the range from "0" to the maximum delay time set in the Module's Function page. Negative values are not allowed at this input port.
- **(Dly2)** "Dly2" (delay 2) is the Audio input port for controlling the delay time of the second delay tap in milliseconds. The values at this input port are set in milliseconds and should be in the range from "0" to the maximum delay time set in the Module's Function page, and so on…

Output Ports

- **(1)** "1" (output 1) is the Audio output port for the delayed input signal from the first tap. It is delayed by the delay time set at the "Dly1" (delay 1) input port.
- **(2)** "2" (output 2) is the Audio output port for the delayed input signal from the second tap. It is delayed by the delay time set at the "Dly2" (delay 2) input port, and so on…

11.2.2 Properties: Function Page

Setting the Maximum Delay Time

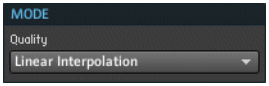
The maximum delay time that can be set at the "Dly1" to "Dly8" (delay 1 to delay 8) input ports is set in the Module's Function page. The default value for this edit field is 1 second. The greater the maximum delay time is the more of the Module's input signal is buffered in your computer's RAM. Very large maximum delay time values can have a detrimental effect on your computer's memory usage.

► To set the maximum delay time for the Multi-Tap Module, go to the Module's Function page and enter the desired buffer value into the Buffer edit field, shown in the figure below.

Setting the Interpolation Quality

If the delay times at one or more of the "Dly" (delay) input ports does not correspond to an integer number of samples, the output signal is generated by interpolation. The interpolation method can be chosen in the Module's Function page.

► To set the interpolation used by your Multi-Tap Module, go to the Module's Function page and choose the desired interpolation mode from the Quality drop-down menu.



Linear interpolation may reduce high frequency components in the sound somewhat.

11.2.3 Example: Scanning Delay

This example shows how to create a delay multi-tap delay effect where delay taps are connected to a Selector Module (↑5.1, Selector) and an LFO Module (↑9.1, LFO) is used to "scan" through these outputs, where each delivers a signal that is the incoming signal delayed by a different amount.

First, insert a Multi-Tap Delay Module into the Structure. Feed the "dry" incoming signal to the "In" input port. To control the delay time of each delay tap from the Instrument Panel, create a knob at each "Dly" input port by right-clicking the input port and choosing the *Create Control* menu entry. This is shown in the figure below.
Next, insert a Selector Module (↑5.1, Selector) and feed to its "channel" input ports the outputs of the Multi-Tap Delay Module. To create additional "channel" input ports (when all existing ones are connected), hold down Ctrl in Windows (Cmd in Mac OS X) when dragging a wire to the input port area of the Selector Module.
Once all delay taps have been connected to the Selector Module (↑5.1, Selector), you need to create an LFO that causes the Selector Module to "scan" through its input ports. The values at the "Pos" (position) input ports should oscillate within the range [0 … 7], corresponding to the eight delay taps. For this, insert an LFO Module (↑9.1, LFO) into the Structure and connect a Constant (↑4.1, Constant) with the value "3.5" to its "A" (amplitude) input port. Use the *Create Control* menu entry to create a knob for the LFO frequency at the "F" (frequency) input port. Now the LFO Module outputs values in the range [- 3.5 … 3.5] but you need values in the range [0 … 7]. For this, use an Add Module (↑4.2, Add, +) to add 3.5 to the LFO signal. Now the signal is in the range [0 … 7] and can be fed to the "Pos" (position) input port of the Selector Module.

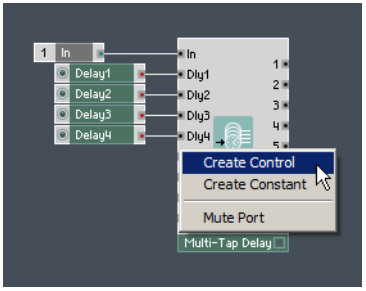


Fig. 11.5 Use the Create Control menu entry (by right-clicking on each of the "Dly" input ports) to quickly create a knob for each delay time.

Finally, add a Crossfade Module (↑5.3, Crossfade) to the Structure. Connect the "dry" incoming signal to the first "channel" input port and the "wet" output signal from the Selector Module (↑5.1, Selector) to the second "channel" input port. Create a control labeled "Dry - Wet" to be able to crossfade between the "dry" and "wet" signals. The output of the Crossfade Module is then the output of your multi-tap scanning delay effect, shown in the figure below.

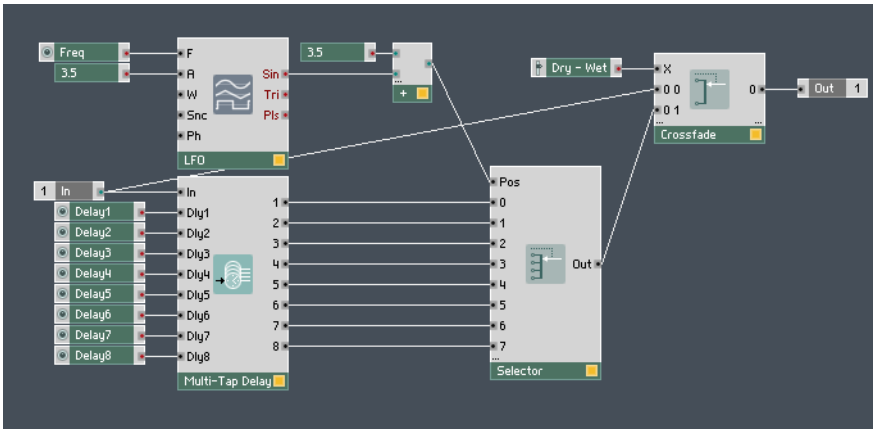


Fig. 11.6 The Structure for a scanning delay effect

Don't forget to choose the interpolation quality with the Interpolation drop-down menu in the Delay Module's Function page!

11.3 Diffuser Delay

Fig. 11.7 Diffuser Module

11.3.1 Overview Module

The Diffuser Module is an all-pass filter containing a delay line with feedback. Its function is to smear (decorrelate) the input signal without emphasizing any particular frequency component. The delay time is controlled at the "Dly" (delay) input port and can be continuously modulated by an Audio signal. The amount of internal feedback is controlled at the "Dffs" input port. The upper limit for the delay time can be adjusted in the module's properties dialog window (default: 200 ms) and affects the memory usage.

Application

The typical application for the Diffuser Delay Module is as a building block for reverb type effects, where you would connect several of these Modules in series and give them different delay times of a few milliseconds.
When Dly = 0, the Diffuser Delay Module functions as a 1-pole all-pass filter. Like this you can construct a phaser, for example, by connecting several Diffuser Modules in series and modulating the "Dffs" (diffusion) parameter.

11.3.2 Ports

Input Ports

- **(Dly)** "Dly" (delay) is the hybrid input port for controlling the delay time. The values at this input port are set in milliseconds and should be in the range from "0" to the maximum delay time set in the Module's Function page. Negative values are not allowed.
- **(Dffs)** "Dffs" (diffusion) is the Event input port for controlling the diffusion coefficient. The range of values at this input port is [-1… 1]. When Dffs = 0, the Module is a pure delay, when Dffs = 1, the output is equal to the input input, when Dffs = -1 the output is the inverted input. The most useful values for "Dffs" (diffusion) are near "0.5" and "-0.5".
- **(In)** "In" is the Audio input port for the signal to be diffused.

Output Ports

- **(Out)** "Out" is the Audio output port for the diffused signal.

11.3.3 Properties: Function Page

Setting the Maximum Delay Time

The maximum delay time that can be set at the "Dly" (delay) input port is set in the Module's Function page. The default value for this edit field is 1 second. The greater the maximum delay time is, the more of the Module's input signal is buffered in your computer's RAM. Very large maximum delay time values can have a detrimental effect on your computer's memory usage.

► To set the maximum delay time for the Diffuser Delay Module, go to the Module's Function page and enter the desired buffer value into the Buffer edit field, shown in the figure below.

If the Delay is used as an Event Delay (in this case the "In" input port will be red, indicating that the Module is in Event processing mode) the Buffer edit field will be set in units of Events (see the figure below). Effectively you will then set the maximum number of buffered Events with this edit field.

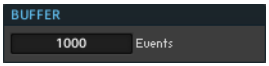


Fig. 11.8 If your Diffuser Delay Module is working as an Event Module, the Buffer edit field applies to the maximum number of buffered Events.

Setting the Interpolation Quality

If the delay time at the "Dly" (delay) input port does not correspond to an integer number of samples, the output signal is generated by interpolation. The interpolation method can be chosen in the Module's Function page.

► To set the interpolation used by your Diffuser Delay Module, go to the Module's Function page and choose the desired interpolation mode from the Quality drop-down menu.

Linear interpolation may reduce high frequency components in the sound somewhat.

11.3.4 Example: Diffuser Delay Effect

This example illustrates how to build a simple diffuser delay effect. The diffuser delay effect is more than just a simple delay, it also "smears" the phases of the incoming signal, giving it a sound characteristic of reverb. As can be seen in the figure below, the incoming "dry" signal is combined with the feedback signal using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) and fed into the Diffuser Delay Module. With the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) you multiply the feedback signal with a value in the range

[0 ... 1] (received from the "Feedback" knob) and so determine the amount of feedback signal that is sent to the Diffuser Delay Module. Since feedback Structures can cause the signal to "explode", a Saturator Module (↑12.1, Saturator) has been placed between the Diffuser Delay Module's output and the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+). This way the feedback signal never exceeds the bounds [-2 ... 2].
The delay time and diffusion parameters are controlled with knobs at the "Dly" and "Dffs" input ports, respectively. The most "diffuse" sound is achieved for "Dffs" values around "0.5" or "-0.5".
Last, the "wet" signal, derived from the Diffuser Delay Module's output, is added to the "dry" incoming signal. This is done again using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+), where the amount of "wet" signal mixed to the "dry" signal can be controlled with the "Dry - Wet" knob (range [0 ... 1]).

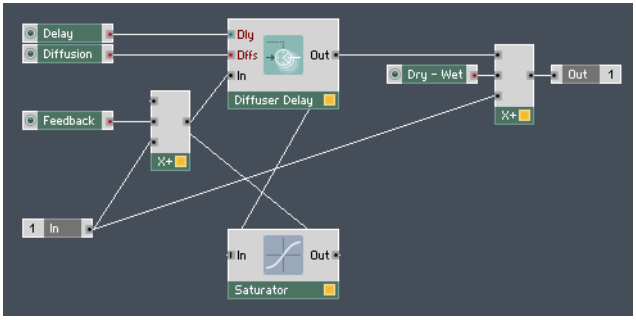


Fig. 11.9 The Structure of a simple diffuser delay effect.

Don't forget to choose the interpolation quality with the Interpolation drop-down menu in the Delay Module's Function page!

11.4 Grain Delay

Fig. 11.10 Grain Delay Module

11.4.1 Overview

The Grain Delay Module comprises a delay line and pitch-shifter effect to be used for Audio signals. The input signal appears at the outputs with a delay corresponding to the set time at the "Dly" (delay) input, transposed by the number of semitones set at the "P" (pitch) input port. The working principle of the Grain Delay Module is that the input signal is “chopped-up” into sound particles (called grains), whose size is controlled by the "Gr" (granularity) input port. The “roughness” of the resulting sound can be increased or reduced via the "Sm" (smoothness) input port. Additionally, the position of the sound particles in the stereo field can be controlled via the "Pan" input port.

Application

The Grain Delay Module is useful in creating more complex delay effects. For the Grain Delay Module, the delay time can be varied without affecting the grain pitch. You can create interesting effects when controlling the different grain parameters such as "P" (grain pitch), "Dly" (delay), "Gr" (granularity), and "Sm" (smoothness) with random value or noise generators. For distinctive delay effects, be sure to mix the signal from the "Out" output port with the incoming signal at the "In" input port. It is also worth experimenting by combining the different types of Delay Modules and using the Module's output signal to control some of the grain parameters.

11.4.2 Ports

Input Ports

- **(P)** "P" (grain pitch) is the Audio input port for controlling the grain transposition in the logarithmic pitch scale. The pitch value is given in semitones and typically is in the range [-20 … 20].
- **(Dly)** "Dly" (delay) is the Audio input port for controlling the delay time. The values at this input port are set in milliseconds and should be in the range from "0" to the maximum delay time set in the Module's Function page. Negative values are not allowed.
- **(Gr)** "Gr" (granularity) is the Audio input port to control the granularity of the re-synthesis process, set in milliseconds. This parameter sets the duration of the sound particles (size of the grains) used for the re-synthesis. Values at this input port are typically in the range [10 … 100]. When this input port is disconnected, the default value of 20 ms is used.

- **(Sm)** "Sm" (smoothness) is the Audio input port to control the smoothness of the re-synthesis process. This parameter affects the formation of the sound particles (grains). Low values at this input port generally result in a rougher sound. Values at this input port should be in the range [0 … 1]. When left disconnected, this input port uses the default value "0.1".
- **(Pan)** "Pan" is the Audio input port to control the position of the output signal in the stereo field (-1 = Left, 0 = Center, 1 = Right).
- **(A)** "A" (amplitude) is the Audio input port to control the amplitude of the Grain Delay Module's output signal.
- **(In)** "In" is the input port for the Audio signal to be grain delayed.

Output Ports

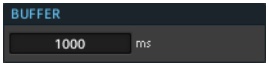
- **(L)** "L" (left) is the Audio output port for the left stereo channel of grain delayed input signal.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the grain delayed input signal.
- **(Dly)** "Dly" (delay) is the polyphonic Event output port which delivers the value of the current delay time, in milliseconds. This output port always produces an Event whenever a new sound particle (grain) is synthesized. Use the Events from this output port to couple the Grain Delay Module with other Delay Modules.

11.4.3 Properties: Function Page

Setting the Maximum Delay Time

The maximum delay time that can be set at the "Dly" (delay) input port is set in the Module's Function page. The default value for this edit field is 1 second. The greater the maximum delay time is the more of the Module's input signal is buffered in your computer's RAM. Very large maximum delay time values can have a detrimental effect on your computer's memory usage.

► To set the maximum delay time for the Grain Delay Module, go to the Module's Function page and enter the desired buffer value into the Buffer edit field, shown in the figure below.



Setting the Interpolation Quality

If the delay times at the "Dly" (delay) input port does not correspond to an integer number of samples, the output signal is generated by interpolation. The interpolation method can be chosen in the Module's Function page.

► To set the interpolation used by your Grain Delay Module, go to the Module's Function page and choose the desired interpolation mode from the Quality drop-down menu.

Linear interpolation may reduce high frequency components in the sound somewhat.

11.4.4 Example: Pitch Shifter Delay Effect

This example illustrates how to build a pitch shifter delay effect. This delay allows you to shift the pitch of the delayed signal in real time, which can create an interesting combination of effects. As can be seen in the figure below, the incoming "dry" signal is combined with the feedback signal using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) and fed into the Grain Delay Module. With the Mult/Add Module you multiply the feedback signal with a value in the range [0 … 1] (received from the "Feedback" knob) and so determine the amount of feedback signal that is sent to the Grain Delay Module. Since feedback Structures can cause the signal to "explode", a Saturator Module (↑12.1, Saturator) has been placed between the Grain Delay Module's output and the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+). This way the feedback signal never exceeds the bounds [-2 … 2].

The pitch transposition, delay time, granularity, smoothness, and amplitude of the output signal are controlled with knobs at the "P", "Dly", "Gr", "Sm", and "A" input ports, respectively. Since the pitch of the output signal can be transposed in real-time, an LFO signal (from the LFO Module, see also ↑9.1, LFO) has been added to the transposition value received from the "Pitch" knob. This way you can create interesting detuning and chorus effects using the Grain Delay Module. The "Pan" value is set to "-1", such that the whole effect output signal is sent to the "L" (left) output port.

Last, the "wet" signal, derived from the Grain Delay Module's output, is added to the "dry" incoming signal. This is done using the Add Module (↑4.2, Add, +), where the amount of "wet" signal mixed to the "dry" signal can be controlled at the "A" (amplitude) input port of the Grain Delay Module.

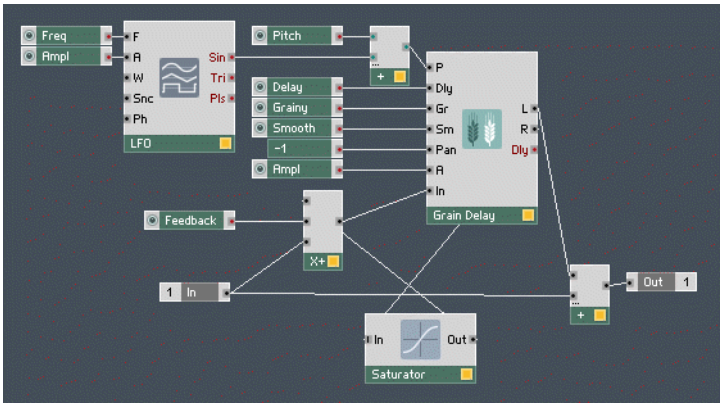


Fig. 11.11 The Structure of a simple pitch shifter delay effect.

Don't forget to choose the interpolation quality with the Interpolation drop-down menu in the Delay Module's Function page!

11.5 Grain Cloud Delay

Fig. 11.12 Grain Cloud Delay Module

11.5.1 Overview

The Grain Cloud Delay Module is similar to the Grain Cloud Module (↑7.6, Grain Cloud Sampler) in the Samplers section, except that instead of operating on audio files loaded into its memory, it processes a constantly changing audio buffer that is fed by the "In" input port. The Grain Cloud Delay is a stereo granular delay with independent control over delay time at the "Dly" (delay) input port and the transposition of the grain pitch at the "P" (grain pitch) input port, among a number of grain parameters. The envelope of each grain can be controlled at the "Att" (attac) and "Dec" (decay) input ports. Additionally, for each grain the time interval to the start of the next grain can be set at the "Dist" (distance) input port. The maximum number of simultaneous grains can be set in the Module's Function page. There are several "jitter" input ports which set the jitter amount for the respective parameter.

Application

The Grain Cloud Delay Module opens up a myriad of possibilities in the realm of real-time transposed delay effects. The grain pitch can be varied independent of the delay time. This means that you can create interesting delay and chorus effects with integrated pitch-shifting of the delayed (wet) signal. For distinctive delay effects, be sure to mix the signal

from the "Out" output port with the incoming signal at the "In" input port. Also, try controlling the different grain parameters with random value or noise generators or even by modulating them with the Grain Cloud Delay Module's output signal. Try using the Event output ports to modulate other Delay Modules that are serially connected to the Grain Cloud Delay Module.

11.5.2 Ports

Input Ports

- **(G)** "G" (gate) is the Event input port for turning the output the next grain. When G > 0, the next grain will be output. When G = -1, the next grai will be suppressed (see the "Dist" (distance) input port).
- **(Frz)** "Frz" (freeze) is the Event input port to freeze the Grain Cloud Delay Module's audio buffer. For values where Frz > 0, internal buffer will be frozen. This means that the incoming signal will not overwrite the internal buffer as new grains are synthesized still from the "old" buffer.
- **(P)** "P" (pitch) is the Audio input port for controlling the grain transposition in the logarithmic pitch scale. The pitch value is given in semitones and typically is in the range [-20 ... 20].
- **(D/F)** "D/F" (direction/frequency) is the Audio input port for direction control if the "P" (pitch) input port is connected. Otherwise this input port control the frequency. The grain plays back in the forward direction and at the original pitch when F = 1. Playback in reverse direction (and at original pitch) occurs when F = -1. The typical range for this input port is [-4 ... 4]. With the input port disconnected, the default value that is used is "1".
- **(PJ)** "PJ" (pitch jitter) is the Audio input port to control the amount of pitch jitter (in semitones). Typical values at this input port are in the range [0 ... 3].
- **(PS)** "PS" (pitch shift) is the Audio input port to control the pitch shift of the current grain. The values at this input port are given in the logarithmic pitch scale, in semitones, and typically fall in the range [-3 ... 3].
- **(Dly)** "Dly" (delay) is the Audio input port for controlling the delay time. The values at this input port are set in milliseconds and should be in the range from "0" to the maximum delay time set in the Module's Function page. Negative values are not allowed.

- **(DlJ)** "DlJ" (delay jitter) is the Audio input port for the amount of jitter in the delay time. The units of this input port are in milliseconds and these values should be in the range [0 ... <maximum delay time>]. Negative values are not allowed. The default valut at this input port is 0 milliseconds.
- **(Len)** "Len" (length) is the Audio input port for setting the grain length in milliseconds. The typical range for this parameter is [10 ... 100]. The default value that is used in the case of a disconnected "Len" input port is 20 milliseconds.
- **(LnJ)** "LnJ" (length jitter) is the Audio input port for the amount of jitter in the grain length parameter. The units of this input port are in milliseconds and these values typically fall in the range [10 ... 100]. The default value in the case of a disconnected "LnJ" input port is 0 milliseconds.
- **(Att)** "Att" (attack) is the Audio input port for setting the attack time of the grain envelope. The range of this input port is [0 ... 1]. The default value in the case that this input port is disconnected is "0.2".
- **(Dec)** "Dec" (decay) is the Audio input port for setting the decay time of the grain envelope. The range for this input port is [0 ... 1]. The default value in the case that this input port is disconnected is "0.2".
- **(Dist)** "Dist" (distance) is the Audio input port for setting the time duration before the next grain is started. The values at this input port are given in milliseconds and typically fall in the range [5 ... 100]. If this input port is disconnected, the default value that is used is "20".
- **(DisJ)** "DisJ" (distance jitter) is the Audio input port for the amount of jitter in the distance parameter. The units of the values at this input port are milliseconds and typically fall in the range [10 ... 100]. If this input port is disconnected, the default value that is used is 20 milliseconds.
- **(Pan)** "Pan" is the Audio input port to control the position of the output signal in the stereo field (-1 = Left, 0 = Center, 1 = Right).
- **(PnJ)** "PnJ" (pan jitter) is the Audio input port for the amount of jitter in the pan parameter. The range of the values at this input port is [0 ... 1]. The default value used in case of this input port being disconnected is "0".
- **(A)** "A" (amplitude) is the Audio input port for controlling the playback amplitude. Since this is an Audio input port, it can be used for ring modulation. The default value to be used when this input port is disconnected is "1".
- **(In)** "In" is the Audio input port for the Audio signal to be treated with the grain cloud delay effect.

Output Ports

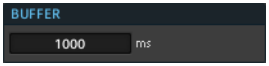
- **(L)** "L" (left) is the Audio output port for the left stereo channel of grain delayed input signal.
- **(R)** "R" (right) is the Audio output port for the right stereo channel of the grain delayed input signal.
- **(Dly)** "Dly" (delay) is the polyphonic Event output port which delivers the value of the current delay time, in milliseconds. This output port always produces an Event whenever a new sound particle (grain) is synthesized. Use the Events from this output port to couple the Grain Delay Module with other Delay Modules.
- **(GTr)** "GTr" (grain trigger) is the Event output port for grain trigger signals. An Event with the value "1" is sent when a new grain starts, an Event with the value "0" is sent when the grain stops.

11.5.3 Properties: Function Page

Setting the Maximum Delay Time

The maximum delay time that can be set at the "Dly" (delay) input port is set in the Module's Function page. The default value for this edit field is 1 second. The greater the maximum delay time is the more of the Module's input signal is buffered in your computer's RAM. Very large maximum delay time values can have a detrimental effect on your computer's memory usage.

► To set the maximum delay time for the Grain Cloud Delay Module, go to the Module's Function page and enter the desired buffer value into the Buffer edit field, shown in the figure below.



Setting the Interpolation Quality

If the delay times at the "Dly" (delay) input port does not correspond to an integer number of samples, the output signal is generated by interpolation. The interpolation method can be chosen in the Module's Function page.

► To set the interpolation used by your Grain Cloud Delay Module, go to the Module's Function page and choose the desired interpolation mode from the Quality drop-down menu.

Linear interpolation may reduce high frequency components in the sound somewhat.

Specifying Grain Settings for the Grain Cloud Module

The Grain Cloud Delay Module has some unique settings in its Function page. First, you can specify the shape of the envelope of the grain using the Attack and Decay edit fields. Second, you can control the maximum number of grains that can be played simultaneously. This becomes relevant when the length of the grains is considerably greater than the distance between the starting points of the individual grains (these values can be specified using the corresponding input ports). Since more grains use up more CPU resources, you can save your CPU resources by reducing the number of overlapping grains with the Overlap edit field.

Another thing to consider when you have several grains playing simultaneously is the amplitude of the output signal. More overlapping grains cause a greater amplitude. The Grain Cloud Delay Module has an amplitude compensation algorithm built into it. You can activate it by engaging the Gain Correction checkbox.

The Pitch Jitter effect causes the pitch to change in a random manner from one grain to the next. The change in pitch fluctuates in the range [-PJ, PJ] where PJ is the Pitch Jitter value at the “PJ” (pitch jitter) input port. Sometimes you want this pitch fluctuation to be more orderly. By engaging the PJ Static checkbox you cause the change in pitch to alternate between the explicit values “-PJ” and “PJ”, instead of the whole range between them.

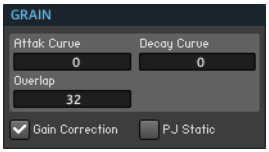


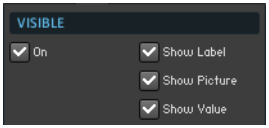
Fig. 11.13 The Grain Cloud Delay Module has specific settings in its Function page. Using that area you can change the envelope and overlap of the grain window, activate gain correction, and make the Pitch Jitter effect non-random.

11.5.4 Properties: View Page

Making the Grain Cloud Delay Panel Representation Visible on the Panel

The Grain Cloud Delay's Panel representation is a graph of the incoming signal which also reflects the state of the internal buffer. That same graph also has the grain start and end points plotted, as reference. Changes in the "Dst" (distance) parameter is reflected by that plot. This Panel representation gives you visual information on the state of the internal buffer and can be useful when mixing the "Out" output signal back into the incoming signal.

► To turn on the Grain Cloud Delay Module's Panel representation, go to the Module's View page and engage the On checkbox, shown in the figure below.



Changing the Size of the Audio Buffer Graph

Depending on your Panel layout, you want a small or large graph of the Grain Cloud Delay Module's audio buffer.

► To change the width and height of the Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.

11.5.5 Example: Grain Cloud Delay Effect

This example illustrates how to build a grain cloud delay effect. This granular delay allows you to shift the pitch of the delayed signal in real-time, freeze the grain buffer, and control other parameters which create a variety of granular effects. As can be seen in the figure below, the incoming "dry" signal is combined with the feedback signal using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) and fed into the Grain Cloud Delay Module. With the Mult/Add Module you multiply the feedback signal with a value in the range [0 … 1]

(received from the "Feedback" knob) and so determine the amount of feedback signal that is sent to the Grain Cloud Delay Module. Since feedback Structures can cause the signal to "explode", a Saturator Module (↑12.1, Saturator) has been placed between the Grain Cloud Delay Module's output and the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+). This way the feedback signal never exceeds the bounds [-2 … 2].
The pitch transposition, pitch slide, delay time, grain length, grain distance, and amplitude of the output signal are controlled with knobs at the "P", "PS", "Dly", "Len", and "A" input ports, respectively. A toggle button between "0" and "1" and with the label "Freeze" is connected "Frz" input port lets you freeze the grain buffer by toggling the button on the Instrument Panel. Since the pitch of the output signal can be transposed in real-time, an LFO signal (from the LFO Module, see also ↑9.1, LFO) has been added to the transposition value received from the "Pitch" knob. This way you can create interesting detuning and chorus effects using the Grain Cloud Delay Module. The "Pan" value is set to "-1", such that the whole effect output signal is sent to the "L" (left) output port.
Last, the "wet" signal, derived from the Grain Cloud Delay Module's output, is added to the "dry" incoming signal. This is done using the Add Module (↑4.2, Add, +), where the amount of "wet" signal mixed to the "dry" signal can be controlled at the "A" (amplitude) input port of the Grain Delay Module.

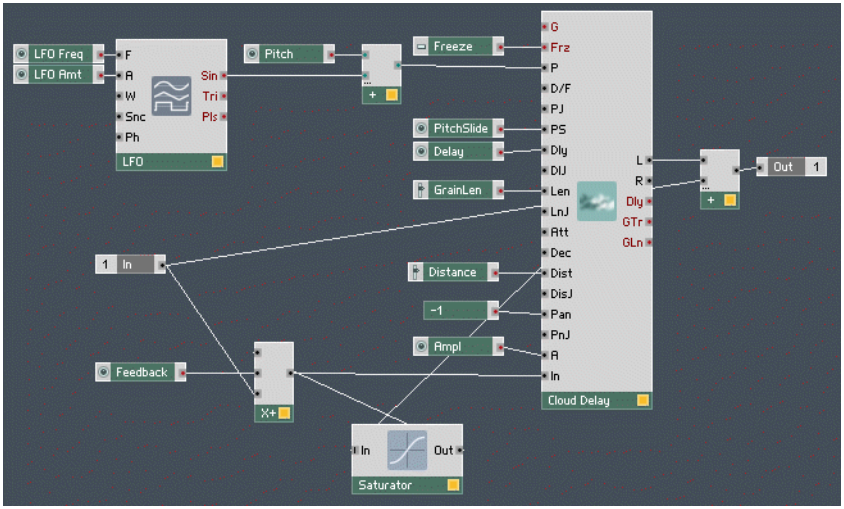


Fig. 11.14 The Structure of a simple version of the grain cloud delay effect.

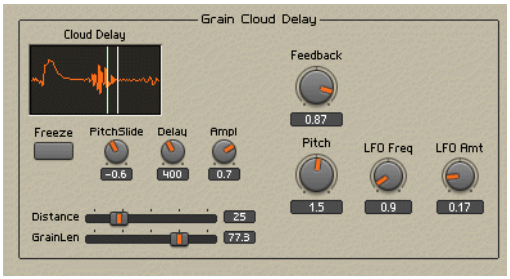


Fig. 11.15 The Instrument Panel of a grain cloud delay effect.

Don't forget to choose the interpolation quality with the Interpolation drop-down menu in the Delay Module's Function page!

11.6 Unit Delay

Fig. 11.16 Unit Delay Module

11.6.1 Overview

The Unit Delay Module delays the incoming Audio signal by one sample period (1 / Sample Rate).

Application

Every structure that uses some kind of feedback must have a Unit Delay Module in the loop. If no explicit unit delay is there, the program will insert an invisible Unit Delay Module somewhere in the loop. The unit delay is a fundamental component in low level DSP structures. For example, filters are built using delayed input and output signals (of the filter) and adding it to the current input signal. Such tasks, however, are best left for REAKTOR Core.

11.6.2 Ports

Input Ports

- **(In)** "In" is the input port for the signal to be delayed by one sample period.

Output Ports

- **(Out)** "Out" is the output port for the incoming signal, delayed by one sample.

11.6.3 Example: Simple Non-Recursive Filter

In this example you will learn how to build a basic non-recursive filter using the Unit Delay Module. Filters alter signals by amplifying or attenuating certain frequency components of said signal. Only the amplitude and/or phase of each frequency component are modified; the frequency remains unchanged.

One central characteristic of a filter is its frequency response. This quantifies the relative amplitude and phase of a steady-state sine signal that has been sent through a filter. These parameters are plotted in an amplitude response and a phase response graph. The amplitude response graph is what you see on the Panel representation of a Filter Module.

A digital filter is a computational algorithm that uses past input and/or output samples to calculate the current output sample. The two principal types of filters are non-recursive and recursive filters. A non-recursive filter calculates the current value of the output only from the present and past values of the input. A recursive filter also uses past output values to calculate the current output value of the filter.

Let us look at a simple non-recursive filter given by the following equation:

$y(n) = 1/2*x(n) + 1/2*x(n-1)$

where y(n) denotes the nth output sample; x(n) and x(n-1) denote the nth and (n-1)th input samples, respectively. In the Primary Level we use the Unit Delay Module to delay the filter's input signal by one sample, thus yielding the x(n-1) input sample.

Notice that this y(n) is the arithmetic average between the current and last input sample. Averaging has the effect of smoothing the waveform, which suggests that this process reduces the amount of high-frequency content in the signal. In fact, this equation is that of a first order (albeit very crude) low-pass filter. The Structure below shows how to realize this very simple filter using the Unit Delay Module.

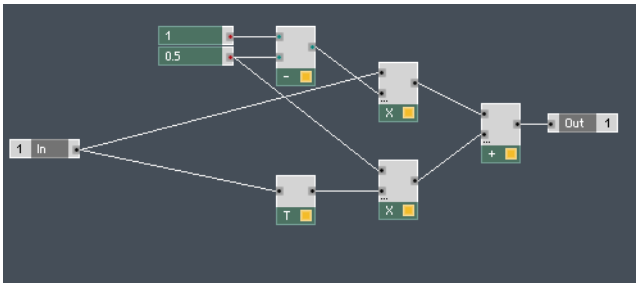


Fig. 11.17 The Structure of a very crude low-pass filter.

The effects of this filter are barely audible and in order to see how this is a low-pass filter, we need to look at what is happening, sample per sample. Remember, the filter equation is

$y(n) = 1/2*x(n) + 1/2*x(n-1)$

where y(n) denotes the nth output sample; x(n) and x(n-1) denote the nth and (n-1)th input samples, respectively.

Let us try to characterize this filter: assuming that the input signal prior to the application of our hypothetical test signal has been zero, say we apply a unit-amplitude signal at 0 Hz given by the sequence of samples {1, 1, 1, ...} to the input. The output, according to our equation is {0.5, 1, 1, 1, ...}. The first sample (0.5) results from calculating the average between "0" (the input sample prior to our test signal) and "1" (the first sample of our test signal). This is called the transient response of the filter. We see that the output of the filter upon applying a steady-state 0 Hz signal at the input is also a steady-state 0 Hz signal. We assert that the filter completely passes 0 Hz signals.

How does our filter react to input signals at higher frequencies? Let us apply a steady state sine wave at half the Nyquist frequency as a hypothetical test signal to the input. As a sample sequence our test sequence would be {1, -1, 1, -1, ...}, which corresponds to a sine wave with the highest possible frequency (half the Sample Rate). Using the filter equation we calculate the output signal {0.5, 0, 0, 0, ...}. The filter doesn´t pass high frequencies. Assuming that nothing strange happens with the amplitude response as we increase the frequency from 0 Hz, we realize that we are dealing with a simple low-pass filter.

Usually filter equations hold variables which then change depending on the resonance setting and cutoff frequency parameters that you can choose on the fly. Our crude filter could, for example, have one parameter, "a", and the equation (usually called the "difference equation") would look as follows:
y(n) = a*x(n) + (1 - a)*x(n-1)
The Structure shown above is just a special case of this equation, namely with a = 0.5. Here's a quick thought experiment: what sort of filter do you get when "a" is negative, say a = - 0.5? Answer: a high-pass filter! As the saying goes, filter design is not only a science but also an art and whole books have been written on this subject. We do hope, however, that this little tutorial has given a little bit of new insight on what is going on inside REAKTOR's filters (especially the ones you can peek inside, in REAKTOR Core).

12 Audio Modifier

The Audio Modifier category offers Modules for audio processing. There are Modules for various forms of distortion (shaping, clipping, and mirroring, among other). When building a synthesizer framework, often the control signals stemming from knobs and faders are shaped to achieve a better "feel" when tweaking them. Use the Shaper Modules to control the shape of control curves. For other audio processing applications such as building effects based on envelope followers or compression you will find the Slew Limiter, Peak Detector, Sample and Hold, and Frequency Divider Modules.

12.1 Saturator

Fig. 12.1 Saturator Module

12.1.1 Overview

The Saturator Module is a simple distortion unit with a smoothly rounded symmetric input-output curve for soft transition to saturation. The output value is limited to ±2 (reached for input values bigger than ±4). Very small input values are transmitted linearly (remain unchanged). When the incoming signal has values greater/less than ±4, the signal is distorted less than in the case of the Clipper Module. However, in contrast to the hard clipping, the harmonic content of the saturated signal differs from that of the incoming signal already for moderate amplitudes.

Application

The Saturator Module is a bread and butter component in audio signal processing Structures. Use it to limit signals that might otherwise become too large like in a feedback path. For example, it is worth experimenting with your own filter or delay characteristics (including comb filters) where you combine a feedback path with a filter or delay and place a Saturator Module in between not only to keep the signal from exploding but also to add more color to the signal.

12.1.2 Ports

Input Ports

- **(In)** "In" is the Audio input port for signal to be saturated.

Output Ports

- **(Out)** "Out" is the Audio output port for the saturated signal.

12.1.3 Example: Comb Filter

A place where the Saturator Module comes in handy is in the feedback path of a comb filter. A comb filter has a frequency response that looks similar to a comb. This means that certain frequencies at regular intervals are amplified, whereas other frequencies are attenuated. The amplification peaks are multiples of the "base frequency" of the comb filter, determined by the delay time set at the Single Delay Module's (↑11.1, Single Delay) "Dly" (delay time) input port.

The Structure of a comb filter is simple. As can be seen in the figure below, the incoming "dry" signal is combined with the feedback signal using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) and fed into the Single Delay Module (↑11.1, Single Delay). With the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) you multiply the feedback signal with a value in the range [0 … 1] (received from the "Feedback" knob) and so determine the amount of feedback signal that is sent to the Single Delay Module. Since feedback Structures can cause the signal to "explode", a Saturator Module has been placed between the Single Delay Module's output and the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+). This way the feedback signal never exceeds the bounds [-2 … 2], which is reached for input values outside the range [-4 … 4].

The "base frequency" is calculated from a pitch value which is received from the knob labeled "pitch" (range [1 … 128]) and converted to the linear frequency scale using the Exp P-to-F Module (↑4.16, Exp (P-to-F)). The output of the Exp P-to-F Module delivers the pitch in Hertz (oscillations per second) but the we need the corresponding delay time in milliseconds. Therefore we use the Divide Module to divide 1000 by the pitch value in Hertz, giving us an Event signal that gives the delay time in milliseconds. Subsequently we convert this Event signal to an Audio Signal using the Audio Smoother Module (↑14.14, Audio Smoother).

Last, the "wet" signal, derived from the Single Delay Module's (↑11.1, Single Delay) output, is added to the "dry" incoming signal. This is done again using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+), where the amount of "wet" signal mixed to the "dry" signal can be controlled with the "Mix" knob (range [0 … 1]).

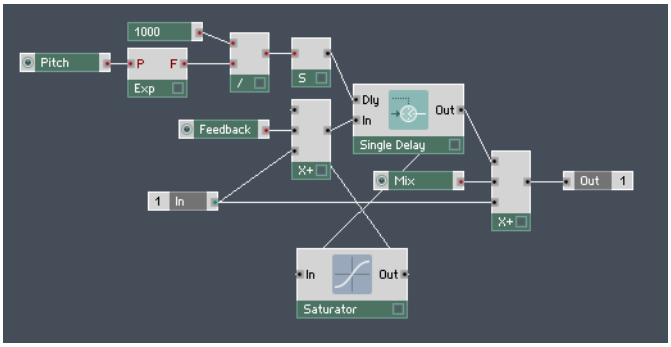


Fig. 12.2 The Structure of a comb filter.

12.2 Saturator 2

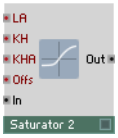


Fig. 12.3 Saturator 2 Module

12.2.1 Overview

The Saturator 2 Module is a fourth-order asymmetric parabolic saturator and unlike the Saturator Module, offers control over the saturation curve. The output value is limited to ±2 (reached for input values bigger than ±4). Please refer to the description of the input ports for more information on controlling the saturation curve.

Application

The Saturator 2 Module is useful for saturation based audio effects. A common application is to filter and equalize the audio signal before feeding it to the "In" input port of the Saturator 2 Module and then again at the Module's output.

12.2.2 Ports

Input Ports

- **(LA)** "LA" (level asymmetry) is the Event input port for the vertical asymmetry of the shaping polynomial. At LA = 0 the saturation levels for positive and negative input signals are equal. For LA > 0 the positive values of the signal are attenuated and at LA = 1 all positive input values become zero. For LA < 0 the negative values of the signal are attenuated and at LA = -1 all negative input values become zero. The range for typical values at this input port is [-1 … 1] and the default value is "0".
- **(KH)** "KH" (KH) (knee hardness) is the Event input port for the "hardness" of the saturation curve. At KH = 0 the saturation rises as soft as possible. The full range between zero and the saturation level is used for the rounded curve. With growing values of "KH" the curve range is reduced to (1 - KH) of the saturation level. At KH = 1 the signal is clipped hard at the saturation level. Use this input port for a parametrized control of the amount of higher harmonics generated by the Module, for saturated incoming signals. The range for typical values at this input port is [0 … 1] and the default value is "0".
- **(KHA)** "KHA" (knee hardness asymmetry) is the Event input port to set the vertical symmetry of the "knee hardness" parameter. At KHA = 0 the knee hardness is equal for positive and negative incoming signals. For KHA > 0 the knee hardness for positive parts of the incoming signal is reduced and at KHA = 1 it becomes zero. For KHA < 0 the knee hardness is reduced for negative values of the incoming signal and at KHA = -1 it becomes zero. The range for typical values at this input port is [-1 … 1] and the default value is "0".
- **(Offs)** "Offs" (offset) is the Event input port to add an offset (DC offset) to the input signal, shifting it relative to the saturator curve. For a zero input signal the offset is fully compensated at the output. The range for typical values at this input port is [-2 … 2] and the default value is "0".
- **(In)** "In" is the Audio input port for signal to be distorted.

Output Ports

- **(Out)** "Out" is the Audio output port for the distorted signal.

12.2.3 Example: Parabolic Saturator as a Distortion Effect

When using the Saturator 2 Module to change the harmonic content of the incoming Audio signal, it is best to hear the effect with your own ears. For example, you could build a simple Structure, shown in the figure below, to test out the sonic possibilities of the Saturator 2 Module with an oscillator signal to which an amplitude envelope has been applied. Note that since the harmonic content changes non-linearly, the effect of the Saturator 2 Module for different oscillator waveforms will be different. Also, since the saturator's transfer function is by nature a function of input amplitude, the signal amplitude and hence the envelope form also have an effect on the way the saturator effect affects the incoming signal.

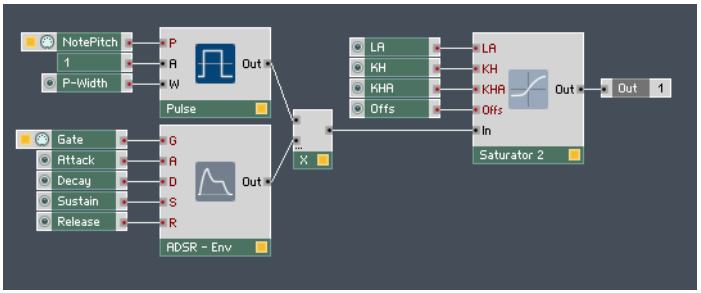


Fig. 12.4 The Structure to test out the effect of the Saturator 2 Module with an oscillator that as an amplitude envelope.

To quickly create knobs (with the right resolution and range) for input ports such as the envelope transition time input ports or the "LA", "KH", "KHA"', and "Offs" input ports of the Saturator 2 Module, remember to right-click the input port and choose the *Create Control* menu entry.

12.3 Clipper

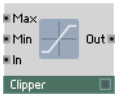


Fig. 12.5 Clipper Module

12.3.1 Overview

The Clipper Module is a distortion unit with a hard clipping characteristic and adjustable upper and lower limit. When a signal is "clipped" at an upper limit, for example, all values of the incoming signal which are greater than the limit value are replaced by the limit value. Similar behavior applies to incoming values which are less than the lower limit value. Signal values between the limits are passed unchanged.

Application

The Clipper Module is useful for applying hard distortion to oversaturated signals. It can also be used as a hard limiter in a feedback path. An interesting effect can be achieved when ring modulating an oscillator with its own (perhaps frequency shifted) output signal. Run the feedback (and modulator) signal through the Clipper Module, adjusting the feedback from the Instrument Panel and if you like, modulating the feedback amount with an envelope.

12.3.2 Ports

Input Ports

- **(Max)** "Max" (maximum) is the Audio input port for controlling the upper limit at which the signal is clipped. The typical range of values at this input port is [0 … 2]. When disconnected, the default value used for this input port is "0".
- **(Min)** "Min" (minimum) is the Audio input port for controlling the lower limit at which the signal is clipped. The typical range of values at this input port is [-2 … 0]. When disconnected, the default value used for this input port is "0".
- **(In)** "In" is the Audio input port for the signal to be clipped.

Output Ports

- **(Out)** "Out is the Audio output port for the clipped signal.

12.3.3 Example: Conventional Hard Clipper

The most straight-forward example of the Clipper Module is the conventional hard clipper effect which equally clips the input signal both above and below zero. Although it is even easier to create this effect using the Mod Clipper Module (↑12.4, Mod Clipper), the Structure for the hard clipper using the Clipper Module is shown in the figure below. The knob

labeled "Thres" is used to determine the absolute value of the signal at which the signal is clipped. It should only send nonnegative values, such as in the range [0 … 1], for example. Since the Knob Module's output values are positive, connect it directly to the "Max" input port of the Clipper Module. To use the same (but negative) value for the "Min" input port, first feed the Knob Module's output signal through an Invert Module (↑4.4, Invert, -x).

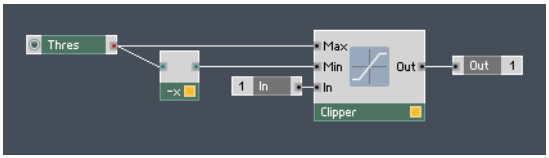


Fig. 12.6 The Structure for the hard clipper effect.

12.4 Mod Clipper

Fig. 12.7 Mod Clipper Module

12.4.1 Overview

The Mod Clipper Module comprises a hard clipper effect unit with a variable clipping limit. When the absolute value of the incoming signal exceeds the limit set at the "M" (modulation) input port, the signal is "clipped" at that value. This means that all incoming values exceeding the limit are replaced by the limit value.
Signal values between the limits remain unaffected.

Application

The Mod Clipper Module is a special case of the Clipper Module (↑12.3, Clipper) when for the latter Min = Max at its input ports. Use the Mod Clipper Module as a quick way to insert a hard clipper effect into your synthesizer. Try connecting the "M" input port to a modulating Audio signal from a sampler or oscillator. The Mod Clipper Module can also be used as a hard limiter in a feedback path. An interesting effect can be achieved when ring modulating an oscillator with its own (perhaps frequency shifted) output signal. Run the

feedback (and modulator) signal through the Mod Clipper Module, adjusting the feedback from the Instrument Panel and if you like, modulating the feedback amount with an envelope.

12.4.2 Ports

Input Ports

- **(M)** "M" (modulator) is the Audio input port for controlling the limit at which the absolute value of the incoming signal is clipped. For a value "M" at this input port it means that the output range of the Module is [- M … M]. Typical values at the "M" (modulation) input port are in the range [-1 … 1]. Negative values invert the incoming signal. When unconnected, the default value zero is used.
- **(In)** "In" is the Audio input port for the signal to be clipped.

Output Ports

- **(Out)** "Out" is the Audio output port for clipped signal.

12.4.3 Example: Hard Clipper

The most straight-forward example of the Mod Clipper Module is the conventional hard clipper effect which equally clips the input signal both above and below zero. The Structure for the hard clipper is shown in the figure below. The knob labeled "Thres" is used to determine the absolute value of the signal at which the signal is clipped. It should only send nonnegative values to the "M" input port, such as in the range [0 … 1]. Negative values, however, are also allowed but in that case their absolute value is used.

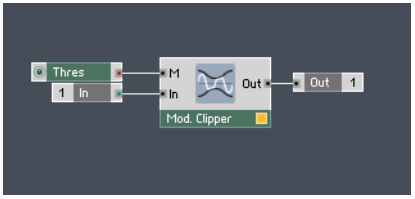


Fig. 12.8 The Structure for the hard clipper effect.

12.5 Mirror 1

Fig. 12.9 Mirror 1 Module

12.5.1 Overview

The Mirror 1 Module is a distortion effect that works by mirroring incoming signal values which are beyond an adjustable mirror level. By "mirroring" it is meant that signal values above the mirror level are folded back at the mirror level. To calculate the value of an outgoing "mirrored" signal value, subtract the amount that the incoming value exceeds the mirror level from the mirror level value. Signal values below the mirror level, however, remain unchanged.

Application

The Mirror 1 Module has an effect that is similar to the Clipper Module (with asymmetric settings) but more pronounced. Folding back the signal at the mirror level creates sharper edges in the outgoing signal and therefore creates more higher harmonics. Try also applying this Module to an LFO signal, for example, to create more complex LFO waveforms from the three offered by the LFO Module (↑9.1, LFO)

12.5.2 Ports

Input Ports

- **(Max)** "Max" (maximum) is the Audio input for controlling the mirror level.
- **(In)** "In" is the Audio input port for the signal to be modified.

Output Ports

- **(Out)** "Out" is the Audio output port for the modified signal.

12.5.3 Example: Mirroring Distortion Effect

The Mirror 1 Module allows you to create a variety of distortion effects, mainly adding higher harmonic content to the incoming signal. To explore its sound, you could build a simple Structure, shown in the figure below. This way you can test out the Mirror 1 Mod-

ule with an oscillator signal to which an amplitude envelope has been applied. Note that since the harmonic content changes non-linearly, the effect of the Mirror 1 Module for different oscillator waveforms will be different. Also, since the mirror effect is by nature a function of input amplitude, the input signal's amplitude and hence the envelope form also have an effect on the way the mirroring affects its harmonic content.
In essence the by just adding the Mirror 1 Module after an oscillator with an amplitude envelope, you can have a signal whose harmonic content varies along with the amplitude levels determined by the envelope. With the right envelope curves and mirroring values you can efficiently create quite dynamic waveforms! The second figure below shows the effect of the Mirror 1 Module on a parabolic oscillator waveform.

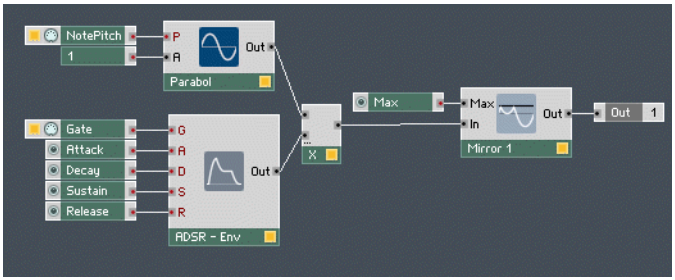


Fig. 12.10 The Structure for a single mirroring distortion effect.

To quickly create knobs (with the right resolution and range) for input ports such as the envelope transition time input ports or the "Max" input port of the Mirror 1 Module, remember to right-click the input port and choose the *Create Control* menu entry.

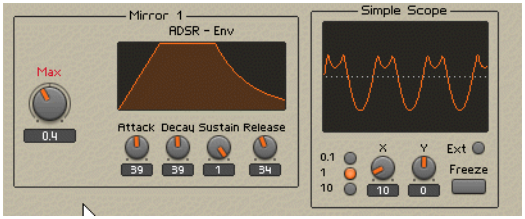


Fig. 12.11 The resulting waveform of a parabolic oscillator sent through a single level mirroring effect.

12.6 Mirror 2

Fig. 12.12 Mirror 2 Module

12.6.1 Overview

The Mirror 2 Module, like its "little brother" the Mirror 1 Module (↑12.5, Mirror 1), is a distortion effect that works by mirroring incoming signal values which are beyond an adjustable mirror levels. By "mirroring" it is meant that signal values beyond a mirror level are folded back at that mirror level. To calculate the value of an outgoing signal value that has been "mirrored" at a positive mirror level, subtract the amount that the incoming value exceeds the mirror level from the mirror level value. For "mirroring" at a negative mirror level you just need to add the amount that the incoming value exceeds the mirror level to the mirror level. Signal values less than the positive mirror level and greater than the negative mirror level, however, remain unchanged.

Application

The Mirror 2 Module has an effect that is similar to the Clipper Module but more pronounced. Folding back the signal at the mirror levels creates sharper edges in the outgoing signal and therefore creates more higher harmonics. Try also applying this Module to an LFO signal, for example, to create more complex LFO waveforms from the three offered by the LFO Module (↑9.1, LFO).

12.6.2 Ports

Input Ports

- **(Max)** "Max" (maximum) is the Audio input port for controlling the upper mirror level.
- **(Min)** "Min" (minimum) is the Audio input for port controlling the lower mirror level.
- **(In)** "In" is the Audio input port for the signal to be modified.

Output Ports

- **(Out)** "Out" is the Audio output for the modified signal.

12.6.3 Example: Mirroring Distortion Effect

The Mirror 2 Module, like the Mirror 1 Module (↑12.5, Mirror 1), allows you to create a variety of distortion effects (albeit more flexibly), mainly adding higher harmonic content to the incoming signal. To explore its sound, you could build a simple Structure, shown in the figure below. This way you can test out the Mirror 2 Module with an oscillator signal to which an amplitude envelope has been applied. Note that since the harmonic content changes non-linearly, the effect of the Mirror 2 Module for different oscillator waveforms will be different. Also, since the mirror effect is by nature a function of input amplitude, the input signal's amplitude and hence the envelope form also have an effect on the way the mirroring affects its harmonic content.

In essence the by just adding the Mirror 2 Module after an oscillator with an amplitude envelope, you can have a signal whose harmonic content varies along with the amplitude levels determined by the envelope. With the right envelope curves and mirroring values you can efficiently create quite dynamic waveforms! The second figure below shows the effect of the Mirror 2 Module on a parabolic oscillator waveform.

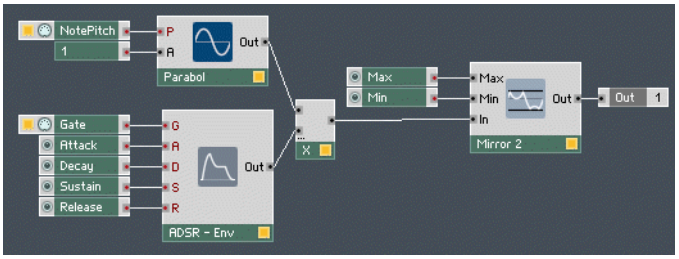


Fig. 12.13 The Structure for a double mirroring distortion effect.

To quickly create knobs (with the right resolution and range) for input ports such as the envelope transition time input ports or the "Max" and "Min" input ports of the Mirror 2 Module, remember to right-click the input port and choose the *Create Control* menu entry.

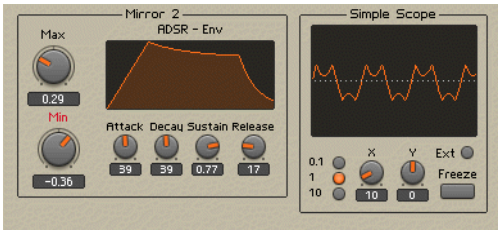


Fig. 12.14 The resulting waveform of a parabolic oscillator sent through a double level mirroring effect.

12.7 Chopper

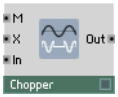


Fig. 12.15 Chopper Module

12.7.1 Overview

The Chopper Module "chops" out parts of the signal incoming at the "In" input port and amplifies them by a factor set at the "X" input port. "Unchopped" signal values are passed to the output port unchanged. "Chopping" happens when the modulation signal at the "M" (modulator) input port is positive, When the modulation signal at the "M" (modulator) input port is negative, the incoming signal at the "In" input port remains unchanged.

Application

The Chopper Module is the most general of the clipper and mirroring type of distortion Modules. With proper signal processing for the values at the "M" and "X" input ports you could recreate the Clipper and Mirror 1 and Mirror 2 Modules with the Chopper Module. However, with more general settings the Chopper Module can create very strong transients and can therefore be the "harshest" sounding of these distortion Modules. If "X" is set to "0", the modulation signal at the "M" input port switches the outgoing signal on and off. Make sure that the signal at the "M" (modulator) input port is an Audio signal that switches between positive and negative values fast enough (e.g. an oscillator) to create Audio rate effects. For the example given above, turning the signal on and off at LFO-rate will "just" cause clicks in your outgoing signal. Of course, this might exactly be what you want.

12.7.2 Ports

Input Ports

- **(M)** "M" is the (modulator) Audio input for the modulation signal (only the sign is relevant). The range of typical values at this input port is [-1 … 1].
- **(X)** "X" is the Audio input for controlling the amplification factor used when M is positive. The range of values at this input port typically is [-2 … 2].
- **(In)** "In" is the Audio input port for the signal to be chopped.

Output Ports

- **(Out)** is the Audio output port for the chopped signal.

12.7.3 Example: Chopper Distortion Effect

The Chopper Module lets you create heavy lo-fi distortion effects. To explore its sound, you could build a simple Structure, shown in the figure below. This way you can test out the Chopper Module with an oscillator signal to which an amplitude envelope has been applied.

Here the Sine Module (↑6.14, Sine Oscillator) has been used as the periodic modulator signal which oscillates between positive and negative values and is fed to the "M" (modulator) input port of the Chopper Module. The Knob Module labeled "Mod" controls the amplification factor when "M" is positive. Here too you could connect an oscillating (or otherwise dynamic) signal to create interesting effects. Note that the chopping is not determined by the signal values at the "In" input port but by the sign of the current signal value at the "M" (modulator) input port. This means that the chopper distortion effect inherently lacks the dynamic properties that arise from the amplitude dependency of the distortion of the Saturation, Mirroring, Clipping, and Breakpoint Shaper Modules. The second figure below shows the effect of the Chopper Module on a parabolic oscillator waveform.

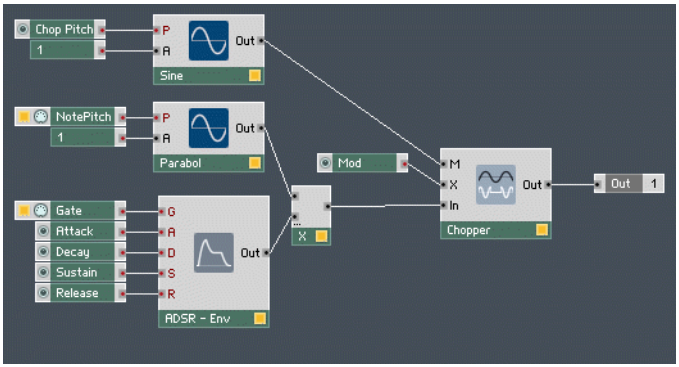


Fig. 12.16 The Structure for a chopper distortion effect.

To quickly create knobs (with the right resolution and range) for input ports such as the envelope transition time input ports or the "X" input port of the Chopper Module, remember to right-click the input port and choose the *Create Control* menu entry.

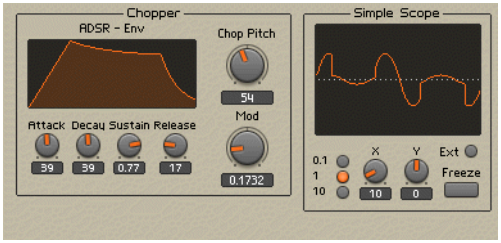


Fig. 12.17 The resulting waveform of a parabolic oscillator sent through a chopper effect.

12.8 Shaper 1 BP

Fig. 12.18 Shaper 1 BP Module

12.8.1 Overview

The Shaper 1 BP is a signal shaper consisting of a piecewise linear transfer function with one breakpoint. The input signal is fed to the "In" input port. Incoming signal values below the "Bp" (breakpoint) value are transmitted linearly (remain unchanged). Incoming values above the "Bp" value are scaled by a linear transfer function that has a slope set with the "Sl" (slope) input port.

Application

The Shaper Modules can be used for waveshaping type of effects on Audio signals. The sharper the edges are in a transfer function, the more higher harmonics are created in the output signal. Waveshaping is an efficient way of creating signals with a variable harmonic distribution (and higher harmonic content) from simple oscillator signals such as the sine wave. Experiment with different transfer functions on different signals to see which "colorings" the different transfer functions add to incoming signals. You can feed envelope signals through Audio Shaper Modules to create interesting envelope curves which can be modulated at Audio Rate.

12.8.2 Ports

Input Ports

- **(Bp)** "Bp" (breakpoint) is the Audio input port for controlling the level of the breakpoint. The typical range of values at this input port is [0 … 1]. When disconnected, the default value "0" is used for this input port.
- **(Sl)** "Sl" (slope) is the Audio input port for controlling the slope of the upper part of the transfer function (input/output curve). The typical range of values at this input port is [-1 … 4]. Sl = 0 corresponds to clipping, Sl = 1 corresponds to no change in the incoming signal, and Sl > 1 corresponds to peak emphasis. Negative values cause negative slopes which "fold back" the signal values which are greater than the "Bp" (breakpoint) value. This works the same as "mirroring" in the Mirror 1 (↑12.5, Mirror 1) and Mirror 2 Modules (↑12.6, Mirror 2).
- **(In)** "In" is the Audio input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Audio output port for the shaped signal.

12.8.3 Example: Simple Single Breakpoint Shaper

The Shaper 1 BP Module is similar to the Mirror 1 Module (↑12.5, Mirror 1), but with the added possibility to gradually control how sharply the input signal is mirrored. Sharper corners in the output signal mean more high harmonics. To explore how this effect sounds with this added degree of freedom, you could build a simple Structure, shown in the figure below. This way you can test out the Shaper 1 BP Module with an oscillator signal to which an amplitude envelope has been applied. Note that since the harmonic content changes non-linearly, the effect of the Shaper 1 BP Module for different oscillator waveforms will be different. Also, since waveshaping is by nature done as a function of input amplitude, the signal amplitude and hence the envelope form also have an effect on the way the shaping affects the harmonic content of the signal.

In essence the by just adding the Shaper 1 BP Module after an oscillator with an amplitude envelope, you can have a signal whose harmonic content varies along with the amplitude levels determined by the envelope. You could even add an additional LFO or envelope to the "Slp" and "BP" input ports to create very dynamic waveforms! This is the basis for waveshaping synthesis. The second figure below shows the effect of the Shaper 1 BP Module on a parabolic oscillator waveform. Note the similarity to the effect of the Mirror 1 Module (↑12.5, Mirror 1).

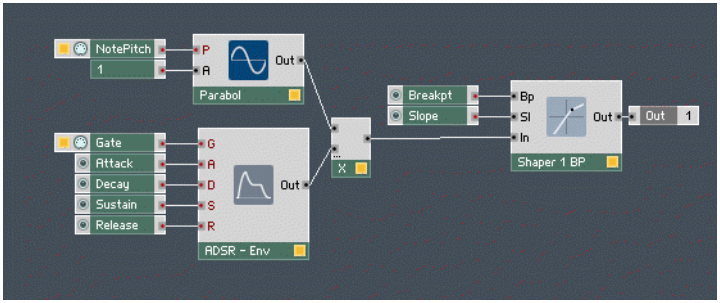


Fig. 12.19 The Structure for a waveshaping effect with one breakpoint.

To quickly create knobs (with the right resolution and range) for input ports such as the envelope transition time input ports or the "BP" and "Sl" input ports of the Shaper 1 BP Module, remember to right-click the input port and choose the *Create Control* menu entry.

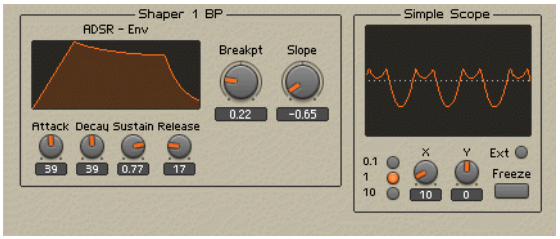


Fig. 12.20 The resulting waveform of a parabolic oscillator sent through a waveshaper with one breakpoint. Note the similarity to the Mirror 1 Module's effect on the same input signal.

12.9 Shaper 2 BP

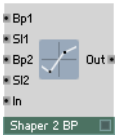


Fig. 12.21 Shaper 2 BP Module

12.9.1 Overview

The Shaper 2 BP is a signal shaper consisting of a piecewise linear transfer function with two breakpoints. The input signal is fed to the "In" input port. Incoming signal values below the "Bp2" (lower breakpoint) value are transmitted by a linear transfer function with the slope set by the "Sl2" (lower slope) input port. Independent of the "Sl2" value, for In = Bp2 the output is also "Bp2" (Out = Bp2). Values between "Bp2" and "Bp1" remain unchanged. Incoming values subsequently above the "Bp1" (higher breakpoint) are scaled by a linear transfer function with the slope set by the "Sl1" (higher) input port. Again, the transfer function is such that for In = Bp1 also Out = Bp1.

Application

The Shaper Modules can be used for waveshaping type of effects on Audio signals. The sharper the edges are in a transfer function, the more higher harmonics are created in the output signal. Waveshaping is an efficient way of creating signals with a variable harmonic distribution (and higher harmonic content) from simple oscillator signals such as the sine wave. Experiment with different transfer functions on different signals to see which "color-

ings" the different transfer functions add to incoming signals. You can feed envelope signals through Audio Shaper Modules to create interesting envelope curves which can be modulated at Audio Rate.

12.9.2 Ports

Input Ports

- **(Bp1)** "Bp1" (higher breakpoint) is the Audio input port for controlling the level of the upper breakpoint. The typical range for values at this input port is [0 … 1] and when this input port is disconnected, the default value of "0" is used.
- **(Sl1)** "Sl1" (higher slope) is the Audio input port for controlling the slope of the transfer function (input/output curve) above the "Bp1" (higher breakpoint) value. The typical range of values at this input port is [-1 … 4]. The default value is "0". Sl1 = 0 corresponds to clipping, Sl1 = 1 corresponds to no change in the incoming signal, and Sl1 > 1 corresponds to peak emphasis. Negative values cause negative slopes which "fold back" the signal values which are greater than the "Bp1" value. This works the same as "mirroring" in the Mirror 1 (↑12.5, Mirror 1) and Mirror 2 Modules (↑12.6, Mirror 2).
- **(Bp2)** "Bp2" (lower breakpoint) is the Audio input port for controlling the level of the lower breakpoint. The typical range for values at this input port is [-1 … 0] and when this input port is disconnected, the default value of "0" is used.
- **(Sl2)** "Sl2" (lower slope) is the Audio input port for controlling the slope of the transfer function (input/output curve) below the "Bp2" (lower breakpoint) value. The typical range of values at this input port is [-1 … 4]. The default value is "0". Sl2 = 0 corresponds to clipping, Sl2 = 1 corresponds to no change in the incoming signal, and Sl2 > 1 corresponds to peak emphasis. Negative values cause negative slopes which "fold back" the signal values which are greater than the "Bp2" value. This works the same as "mirroring" in the Mirror 1 (↑12.5, Mirror 1) and Mirror 2 Modules (↑12.6, Mirror 2).
- **(In)** "In" is the Audio input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Audio output port for the shaped signal.

12.9.3 Example: Simple Two Breakpoint Shaper

The Shaper 2 BP Module is similar to the Mirror 2 Module (↑12.6, Mirror 2), but with the added possibility to gradually control how sharply the input signal is mirrored at each breakpoint. Sharper corners in the output signal mean more high harmonics. To explore how this effect sounds with these two added degrees of freedom, you could build a simple Structure, shown in the figure below. This way you can test out the Shaper 2 BP Module with an oscillator signal to which an amplitude envelope has been applied. Note that since the harmonic content changes non-linearly, the effect of the Shaper 2 BP Module for different oscillator waveforms will be different. Also, since waveshaping is by nature done as a function of input amplitude, the signal amplitude and hence the envelope form also have an effect on the way the shaping affects the harmonic content of the signal.

In essence the by just adding the Shaper 2 BP Module after an oscillator with an amplitude envelope, you can have a signal whose harmonic content varies along with the amplitude levels determined by the envelope. You could even add an additional LFO or envelope to the input ports for the breakpoint and slope parameters to create waveforms with truly dynamic harmonic content! This is the basis for waveshaping synthesis. The second figure below shows the effect of the Shaper 2 BP Module on a parabolic oscillator waveform. Note the similarity to the effect of the Mirror 2 Module (↑12.6, Mirror 2).

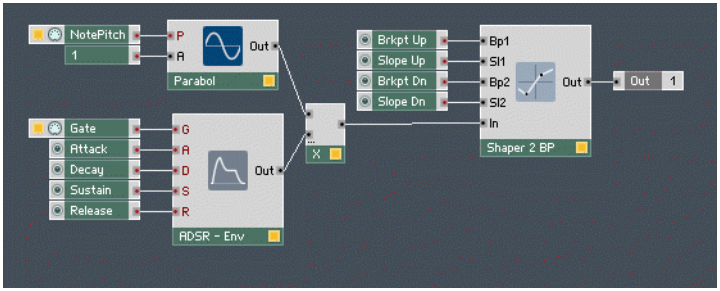


Fig. 12.22 The Structure for a waveshaping effect with two breakpoints.

To quickly create knobs (with the right resolution and range) for input ports such as the envelope transition time input ports or the "BP1","Sl1", "Bp2", and "Sl2" input ports of the Shaper 2 BP Module, remember to right-click the input port and choose the *Create Control* menu entry.

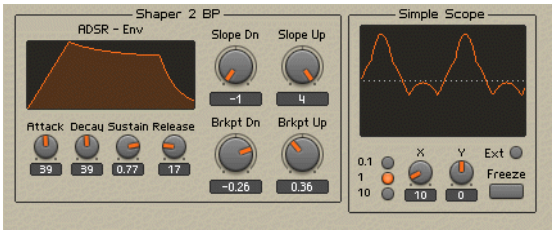


Fig. 12.23 The resulting waveform of a parabolic oscillator sent through a waveshaper with two breakpoints. Note the similarity to the Mirror 2 Module's effect on the same input signal. Here a positive slope value, "4", was used for the segment after the second breakpoint. That is why the upper part of the waveform is not mirrored.

12.10 Shaper 3 BP

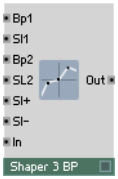


Fig. 12.24 Shaper 3 BP Module

12.10.1 Overview

The Shaper 3 BP is a signal shaper consisting of a piecewise linear transfer function with three breakpoints, two of which are adjustable. The input signal is fed to the "In" input port. Incoming signal values below the "Bp2" (lower breakpoint) value are transmitted by a linear transfer function with the slope set by the "Sl2" (lower slope) input port. Independent of the "Sl2" value, for In = Bp2 the output is also "Bp2" (Out = Bp2). Values between "Bp2" and "0" are transmitted according to a linear transfer function with the slope set by the "Sl-" (slope to zero) input port. Values between "0" and "Bp1" are transmitted according to a linear transfer function with the slope set by the "Sl+" (slope from zero) input port. Incoming values subsequently above the "Bp1" (higher breakpoint) are scaled by a linear transfer function with the slope set by the "Sl1" (higher) input port. Again, the transfer function is such that for In = Bp1 also Out = Bp1.

Application

The Shaper Modules can be used for waveshaping type of effects on Audio signals. The sharper the edges are in a transfer function, the more higher harmonics are created in the output signal. Waveshaping is an efficient way of creating signals with a variable harmonic distribution (and higher harmonic content) from simple oscillator signals such as the sine wave. Experiment with different transfer functions on different signals to see which "colorings" the different transfer functions add to incoming signals. You can feed envelope signals through Audio Shaper Modules to create interesting envelope curves which can be modulated at Audio Rate.

12.10.2 Ports

Input Ports

- **(Bp1)** "Bp1" (higher breakpoint) is the Audio input port for controlling the level of the upper breakpoint. The typical range for values at this input port is [0 … 1] and when this input port is disconnected, the default value of "0" is used.
- **(Sl1)** "Sl1" (higher slope) is the Audio input port for controlling the slope of the transfer function (input/output curve) above the "Bp1" (higher breakpoint) value. The actual higher slope is determined by the value at this input port and the value at the "Sl+" (slope from zero) input port. The typical range of values at this input port is [-1 … 4]. The default value is "0". Sl1 = 0 corresponds to clipping, Sl1 = 1 corresponds to no change in the incoming signal, and Sl1 > 1 corresponds to peak emphasis. Negative values cause negative slopes which "fold back" the signal values which are greater than the "Bp1" value. This works the same as "mirroring" in the Mirror 1 (↑12.5, Mirror 1) and Mirror 2 Modules (↑12.6, Mirror 2).
- **(Bp2)** "Bp2" (lower breakpoint) is the Audio input port for controlling the level of the lower breakpoint. The typical range for values at this input port is [-1 … 0] and when this input port is disconnected, the default value of "0" is used.
- **(Sl2)** "Sl2" (lower slope) is the Audio input port for controlling the slope of the transfer function (input/output curve) below the "Bp2" (lower breakpoint) value. The actual lower slope is determined by the value at this input port and the value at the "Sl-" (slope to zero) input port. The typical range of values at this input port is [-1 … 4]. The default value is "0". Sl2 = 0 corresponds to clipping, Sl2 = 1 corresponds to no change in the incoming signal, and Sl2 > 1 corresponds to peak emphasis. Negative values cause

negative slopes which "fold back" the signal values which are greater than the "Bp2" value. This works the same as "mirroring" in the Mirror 1 (↑12.5, Mirror 1) and Mirror 2 Modules (↑12.6, Mirror 2).

- **(Sl+)** "Sl+" (slope from zero) is the Audio input port for the slope of the transfer function from zero to "Bp1". Note that this slope value also affects the lower slope parameter. The typical range for values at this input port is [0.25 ... 4] When disconnected, the default value of "0" is used. The input values are interpreted the same as for the "Sl1" (higher slope) and "Sl2" (lower slope) input ports.
- **(Sl-)** "Sl-" (slope to zero) is the Audio input port for the slope of the transfer function from "Bp2" to zero. Note that this slope value also affects the higher slope parameter. The typical range for values at this input port is [0.25 ... 4] When disconnected, the default value of "0" is used. The input values are interpreted the same as for the "Sl1" (higher slope) and "Sl2" (lower slope) input ports.
- **(In)** "In" is the Audio input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Audio output port for the shaped signal.

12.10.3 Example: Modulated Three Breakpoint Shaper

The Shaper 3 BP Module is the tool for waveshaping synthesis with the most flexibility. Sharper corners in the output signal mean more high harmonics. To explore how this effect sounds, you could build a simple Structure, shown in the figure below. This way you can test out the Shaper 3 BP Module with an oscillator signal to which an amplitude envelope has been applied.

To add a strong inharmonic components resulting from the waveshaping, the Sine Module has been used as a source for Audio Rate modulation of the transfer function parameters "S+" (slope from zero) and "S-" (slope to zero). Since the Sine Module (↑6.14, Sine Oscillator) oscillates at audible frequencies, its audible effect to the incoming signal is similar to ring modulation.

Note that since the harmonic content changes non-linearly, the effect of the Shaper 3 BP Module for different oscillator waveforms will be different. Also, since waveshaping is by nature done as a function of input amplitude, the signal amplitude and hence the envelope form also have an effect on the way the shaping affects the harmonic content of the signal

In essence the by just adding the Shaper 3 BP Module after an oscillator with an amplitude envelope, you can have a signal whose harmonic content varies along with the amplitude levels determined by the envelope. You could even add an additional LFO or envelope to the input ports for the breakpoint and slope parameters to create waveforms with truly dynamic harmonic content! This is the basis for waveshaping synthesis. The second figure below shows the effect of the Shaper 3 BP Module with Audio Rate modulation on a parabolic oscillator waveform.

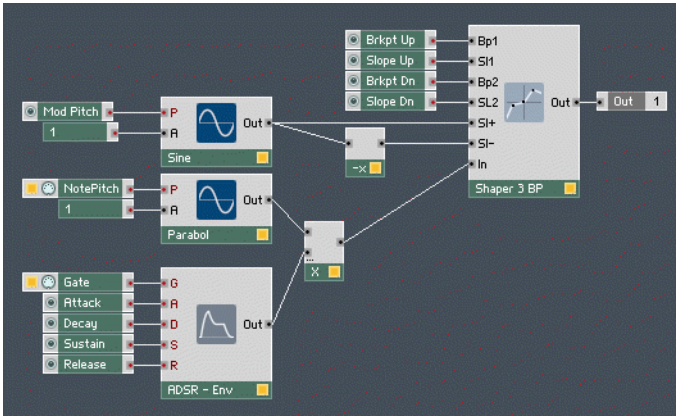


Fig. 12.25 The Structure for a waveshaping effect with three breakpoints and Audio Rate modulation for two of the four slope parameters.

To quickly create knobs (with the right resolution and range) for input ports such as the envelope transition time input ports or the "BP1","SI1", "Bp2", "SI2", "S+", and "S-" input ports of the Shaper 3 BP Module, remember to right-click the input port and choose the *Create Control* menu entry.

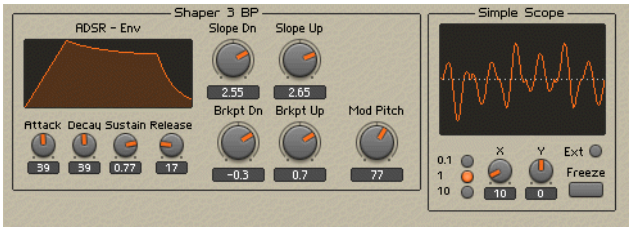


Fig. 12.26 The resulting waveform of a parabolic oscillator sent through a waveshaper with three breakpoints where two slopes are modulated by a sine oscillator with pitch set by the "Mod Pitch" knob.

12.11 Parabolic Shaper

Fig. 12.27 Parabolic Shaper Module

12.11.1 Overview

The Parabolic Shaper Module is a signal shaper that has a parabolic (2nd order polynomial) transfer function (input/output curve). In general, the transfer function can be described as follows: f(x) = a*x + b*x^2 where "x" denotes the incoming signal at the "In" input port and f(x) the outgoing signal at the "Out" output port. The parameters "a" and "b" can be set at the "Lin" (linear) and "Par" (parabolic) input ports, respectively.

Application

The Parabolic Shaper Module is useful for audio waveshaping type effects where smooth transfer functions are desired. Special for the Parabolic Shaper Module is that increasing the "Lin" parameter simply amplifies the incoming signal and increasing the "Par" parameter gradually adds second and higher even harmonics to the output signal. However, the Parabolic Shaper Module is also useful for shaping control signals (see example).

12.11.2 Ports

Input Ports

- **(Lin)** "Lin" (linear) is the Audio input port for controlling the level of the parameter "a" in the transfer function f(x) = a*x + b*x^2 .
- **(Par)** "Par" (parabolic) is the Audio input port for controlling the level of parameter "b" in the transfer function f(x) = a*x + b*x^2.
- **(In)** "In" is the Audio input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Audio output port for the shaped signal.

12.11.3 Example: Parabolic Control Shaper

If you have a control parameter (that you control with a knob) in the range [0 … 1] and for which you want more precision for small values, then the Parabolic Shaper Module is what you need since the parabolic part of the transfer function adds more "weight" (in the form of a smaller slope) to the incoming values that are close to zero. Naturally, you would need to connect a non-zero value to the "Par" (parabolic) input port to observe the described shaping effect. The scope in the figure shows the output signal corresponding to an input ramp from "0" to "1".

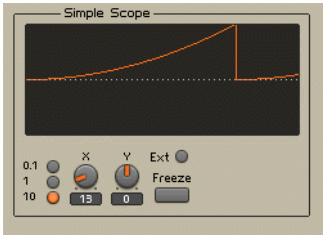


Fig. 12.28 The resulting signal of an incoming ramp from "0" to "1" to a Parabolic Shaper Module with Lin = 0 and Par = 1.

12.12 Cubic Shaper

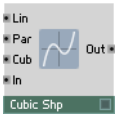


Fig. 12.29 Cubic Shaper Module

12.12.1 Overview

The Cubic Shaper Module is a signal shaper that has a cubic (3rd order polynomial) transfer function (input/output curve). In general, the transfer function can be described as follows: f(x) = a*x + b*x^2 + c*x^3 where "x" denotes the incoming signal at the "In" input port and f(x) the outgoing signal at the "Out" output port. The parameters "a", "b", and "c" can be set at the "Lin" (linear), "Par" (parabolic), and "Cub" (cubic) input ports, respectively.

Application

The Cubic Shaper Module is useful not only for audio waveshaping type effects, but also for shaping control signals. An easy way to create an Audio signal with a variable harmonic distribution is to feed a simple oscillator signal into the shaper Module. In the case of a sine signal you can vary the harmonic distribution from just consisting of the sine's frequency component to a distribution with strong high components.

12.12.2 Ports

Input Ports

- **(Lin)** "Lin" (linear) is the Audio input port for controlling the level of the parameter "a" in the transfer function f(x) = a*x + b*x^2 + c*x^3.
- **(Par)** "Par" (parabolic) is the Audio input port for controlling the level of parameter "b" in the transfer function f(x) = a*x + b*x^2 + c*x^3.
- **(Cub)** "Cub" (cubic) is the Audio input port for controlling the level of the parameter "c" in the transfer function f(x) = a*x + b*x^2 + c*x^3.
- **(In)** "In" is the Audio input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Audio output port for the shaped signal.

12.12.3 Example: Simple Cubic Shaper

The Cubic Shaper Module has a smooth transfer function as determined by the polynomial coefficients set at its "Lin", "Par", and "Cub" input ports. This means that although giving your three degrees of waveshaping freedom, the Cubic Shaper Module only relatively softly adds higher harmonics to the output signal. To explore how the cubic shaper sounds you could build a simple Structure, shown in the figure below. A good way to test it is with a Sine oscillator, since the sine signal represents one frequency component and any components added by the Cubic Shaper Module can easily be distinguished. Note however, that since the harmonic content changes non-linearly, the effect of the Cubic Shaper Module for different oscillator waveforms will be different. Also, since waveshaping is by nature done as a function of input amplitude, the signal amplitude and hence the envelope form also have an effect on the way the shaping affects the harmonic content of the signal.

In essence the by just adding the Cubic Shaper Module after an oscillator with an amplitude envelope, you can have a signal whose harmonic content varies along with the amplitude levels determined by the envelope. You could even add an additional LFO or envelope to the input ports for the polynomial's parameters to create waveforms with softly changing harmonic content! This is the basis for waveshaping synthesis. The second figure below shows the effect of Cubic Shaper Module on a sine oscillator waveform.

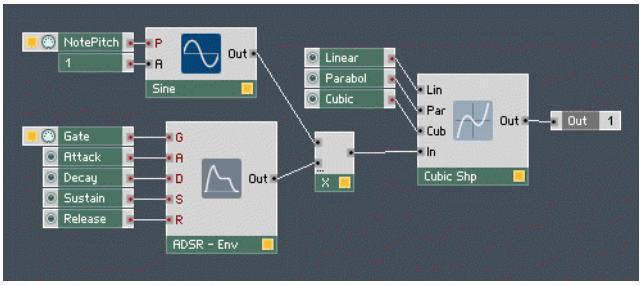


Fig. 12.30 The Structure for a simple cubic waveshaper.

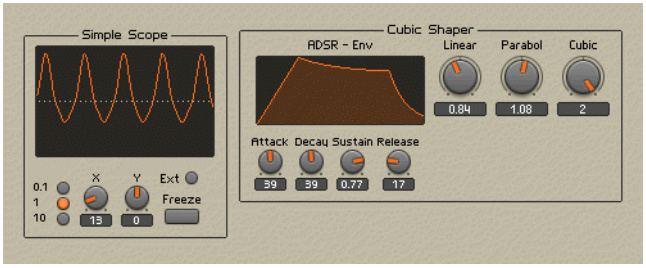


Fig. 12.31 For a sine input signal the Cubic Shaper Module does not add any sharp transients to the output signal.

12.13 Slew Limiter

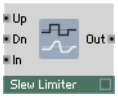


Fig. 12.32 Slew Limiter Module

12.13.1 Overview

The Slew Limiter Module comprises a slew rate limiter with separately adjustable maximum rates for rising and falling signals. The output signal tracks the input signal, but for fast movement and jumps at the input, the output follows with a limited rate of change (ramp) until its value reaches that of the input signal.

Application

Use the Slew Limiter Module to create effects based on envelope followers but which don't need a fast attack time. Feed an Audio signal to the "In" input port and use the output signal as a control signal for an effect parameter such as a filter cutoff, delay time, or the drive of a distortion input. Depending on your input signal, you need to make sure that the slew rates are set about an order of magnitude lower than the lowest audible frequencies.

12.13.2 Ports

Input Ports

- **(Up)** "Up" is the Audio input port for controlling the maximum slew rate for rising signals. The rate of the rising signal is limited to the "Up" value in units of Hertz. The typical range for values at this input port is [20 … 2000]. If left disconnected, the default value used for this input port is 0 Hz, meaning that the Slew Limiter Module does not follow rising signals.
- **(Dn)** "Dn" (down) is the Audio input port for controlling the maximum slew rate for falling signals. The rate of the rising signal is limited to the "Dn" value in units of Hertz. The typical range for values at this input port is [20 … 2000]. If left disconnected, the default value used for this input port is 0 Hz, meaning that the Slew Limiter Module does not follow falling signals.
- **(In)** "In" is the Audio input port for signal to be slew rate limited

Output Ports

- **(Out)** "Out" is the Audio output port for slew rate limited signal.

12.13.3 Example: Beat Tracing

In this example the Slew Limiter Module is used to extract the rough rhythmic pattern of a drum loop sample. This Structure is displayed in the figure below. A Sampler Module (↑7.1, Sampler) plays back the drum loop sample on which to demonstrate the Slew Limiter Module's capabilities. The "Pitch" and "Ampl" controls send values for the "P" (pitch) and "A" (amplitude) input ports of the Sampler Module, respectively. Sample playback is triggered by pressing a MIDI (or computer keyboard) key, as is evident from the Gate Module (↑2.3, Gate) that has been connected to the Sampler Module's (↑7.1, Sampler) "Trig" (trigger) input port.

The played back sample's Audio signal is sent to the Slew Limiter Module's "In" input port. The upwards and downwards slew rates are controlled by Knob Modules at the "Up" and "Dn" input ports, respectively. The second figure below shows how the Sampler output signal (left Audio Table, labeled "Beat Loop") is processed by the Slew Limiter Module (right Audio Table, labeled "Slew Limiter"). Note that the upward slew rate is "425", meaning that it quickly follows sharp increases in incoming signal values but the downward slew rate is "1", causing the output signal to fall slowly. The result is a trace of the sample's rhythmic grid.

The Sampler Module's (↑7.1, Sampler) output signal as well as the Slew Limiter Module's output signal were recorded using Audio Tables. Please refer to section 11.2 in the Application Reference for information on recording signals with Table modules.

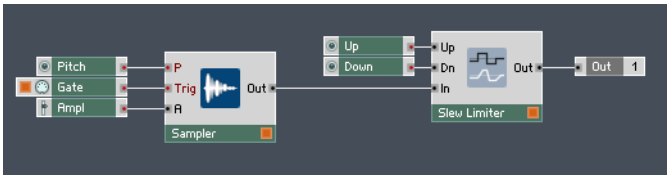


Fig. 12.33 The Slew Limiter traces the signal incoming from the Sampler Module, depending on the slew rates set at the "Up" and "Down" input ports.

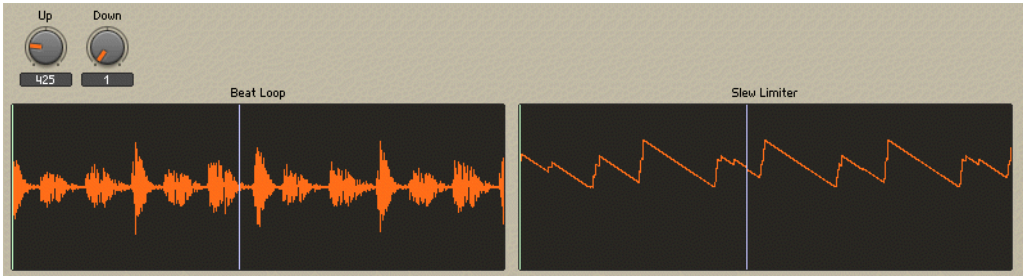


Fig. 12.34 The left Audio Table (labeled "Beat Loop") shows the signal going into the Slew Limiter Module and the right Audio Table (labeled "Slew Limiter") shows the signal coming out of the Slew Limiter whose slew rates are set with the knobs on the top left.

12.14 Peak Detector

Fig. 12.35 Peak Detector Module

12.14.1 Overview

The Peak Detector Module detects the peak amplitude of the incoming Audio signal. Technically, the input signal is rectified and smoothed with an adjustable release time. The result is fed to the output. The attack time of peak detection is zero.

Application

The result of the Peak Detector's action is that the output value follows the amplitude envelope of the input. Use the Peak Detector Module as an envelope follower. Since the attack time of the peak detection is zero, you can use the Peak Detector Module when building a simple compressor in the Primary Level.

12.14.2 Ports

Input Ports

- **(Rel)** "Rel" (release) is the logarithmic Event input port for controlling the release time of the detected peak level. The release time is considered the time that it takes for a detected peak level to fall to 1 / 10th of its value (given that no other peaks are detect-

ed). Rel = 0 corresponds to 2.3 ms, Rel = 20 corresponds to 23 ms, Rel = 40 corresponds to 230 ms, and Rel = 60 corresponds to 2300 ms. It takes twice as long for the peak level to fall to 1 / 100th of its original level.

- **(In)** "In" Audio input port for the signal whose peaks are to be detected.

Output Ports

- **(Out)** "Out" Audio output port for the peak level of the incoming signal. When a nonzero signal is present at the "In" input port the values at this output port are always positive.

12.14.3 Example: Envelope Follower

A good way to modulate your effects, filters, envelopes, and oscillators, is using envelope followers. An envelope follower takes an input signal and uses its peaks to trigger an envelope whose amplitude matches that of the peak. The Structure in the figure below illustrates such an envelope follower Structure, utilizing the Peak Detector (↑12.14, Peak Detector), Differentiator (↑12.14, Peak Detector), Sample and Hold (↑12.15, Sample & Hold), HR Envelope (↑9.4, HR - Env), and a 1-Pole Filter (↑10.2, HP/LP 1-Pole FM).
The incoming signal that is to trigger the envelope is first fed into the Peak Detector Module (↑12.14, Peak Detector). Depending on the frequency of peaks, the release time of the peaks in the Peak Detector's output signal is set with the "PkRel" knob at the "Rel" input port. Next, the Peak Detector Module's output is fed to the Sample and Hold Module (↑12.15, Sample & Hold). This signal is only sampled and held when a negative value at the Sample and Hold Module's "Trig" (trigger) input port goes from a negative value to a positive value. At the same time, the Differentiator Module (↑10.22, Differentiator) records if the Peak Detector Module's output is rising or falling. Therefore, if the Peak Detector signal goes from a falling state to a rising state, its output is sampled and forwarded to the "A" (amplitude) input port of the HR Envelope Module (↑9.4, HR - Env). At the same time, the HR Envelope Module is triggered (with this new amplitude, ultimately received from the Peak Detector Module). The envelope's output signal is then smoothed by a 1-Pole Filter (low-pass) where the value at its "P" (pitch) input port determines the amount of smoothing.

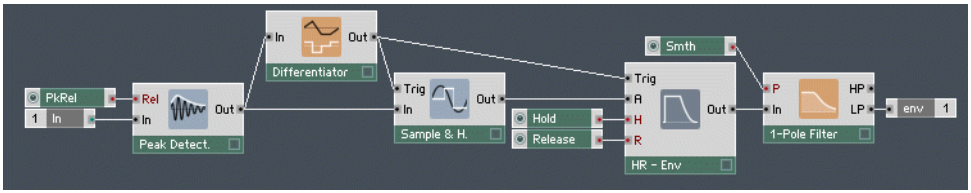


Fig. 12.36 The Structure for an envelope follower effect.

12.15 Sample & Hold

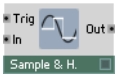


Fig. 12.37 Sample and Hold Module

12.15.1 Overview

The Sample and Hold Module comprises a sample & hold circuit with an input port for the trigger signal. When a trigger signal arrives at the "Trig" (trigger) input port the current value at the "In" input port is passed to the output and held there until the next trigger arrives, upon which a new value is fetched from the "In" input port. The result is a step waveform at the output. The signal at the "Trig" input port sends a sample and hold trigger when it rises from a negative value to a positive value (positive zero crossing).

Application

You can use the Sample and Hold Module to create a sample reduction effect. Make sure that you have a periodic oscillating signal with precise frequency control at the "Trig" (trigger) input port. A Ramp Module would suffice for this. You can then control the sample and hold frequency via the frequency of the Ramp Module (at its "F" (frequency) input port). Note that the maximum sample reduction frequency is half the Nyquist frequency. You can fetch this value from the System Info Module (↑14.20, System Info).

12.15.2 Ports

Input ports

- **(Trig)** "Trig" (trigger) is the Audio input port for the trigger signal. When the signal at this input port rises from a negative value to a positive value, it triggers the sample and hold process.
- **(In)** "In" is the Audio input port for the signal to be sampled.

Output Port

- **(Out)** "Out" is the Audio output port for the sampled signal.

12.15.3 Example: Envelope Follower

A good way to modulate your effects, filters, envelopes, and oscillators, is using envelope followers. An envelope follower takes an input signal and uses its peaks to trigger an envelope whose amplitude matches that of the peak. The Structure in the figure below illustrates such an envelope follower Structure, utilizing the Peak Detector (↑12.14, Peak Detector), Differentiator (↑12.14, Peak Detector), Sample and Hold, HR Envelope (↑9.4, HR - Env), and a 1-Pole Filter (↑10.2, HP/LP 1-Pole FM).

The incoming signal that is to trigger the envelope is first fed into the Peak Detector Module (↑12.14, Peak Detector). Depending on the frequency of peaks, the release time of the peaks in the Peak Detector's output signal is set with the "PkRel" knob at the "Rel" input port. Next, the Peak Detector Module's output is fed to the Sample and Hold Module. This signal is only sampled and held when a negative value at the Sample and Hold Module's "Trig" (trigger) input port goes from a negative value to a positive value. At the same time, the Differentiator Module (↑10.22, Differentiator) records if the Peak Detector Module's output is rising or falling. Therefore, if the Peak Detector signal goes from a falling state to a rising state, its output is sampled and forwarded to the "A" (amplitude) input port of the HR Envelope Module (↑9.4, HR - Env). At the same time, the HR Envelope Module is triggered (with this new amplitude, ultimately received from the Peak Detector Module). The envelope's output signal is then smoothed by a 1-Pole Filter (low-pass) where the value at its "P" (pitch) input port determines the amount of smoothing.

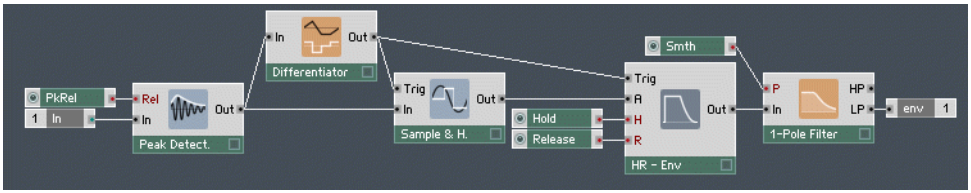


Fig. 12.38 The Structure for an envelope follower effect.

12.16 Audio Frequency Divider

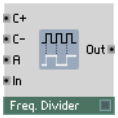


Fig. 12.39 Audio Frequency Divider Module

12.16.1 Overview

The Audio Frequency Divider Module is a pulse sub-oscillator with independently controllable duration of the pulse's high and low states. The outgoing pulse wave is generated by counting the positive zero crossings of the input signal. A positive zero crossing happens then the signal rises from a negative value to a positive value. The frequency of the output waveform will be a fraction of the frequency of the input waveform. The frequency ratio between the output signal and the incoming zero crossings can be adjusted by the following formula: fout = 2 * fin / (C+ + C–) where "fout" and "fin" correspond to the frequencies of the outgoing signal and incoming zero crossings, respectively. "C+" and "C-" are adjustable parameters at the corresponding input ports, as described below. An asymmetric output waveform results when "C+" and "C-" have different values.

Application

The Audio Frequency Divider Module is the Audio counterpart of the Event Frequency Divider Module (↑13.4, Event Frequency Divider). You can use it create a regular pulsed clock signal from an incoming period signal with zero crossings. The frequency stays the same if both "C+" and "C-" are equal to "1". You can also creatively use the Audio Frequency Divider Module as a pulse oscillator with a special type of Audio Rate pulse width modulation at the "C+" and "C-" input ports.

12.16.2 Ports

Input Ports

- **(C+)** "C+" (positive clock) is the Audio input port for controlling the duration of the high states of the output, in number of positive zero crossings of the input signal.
- **(C-)** "C-" (negative clock) is the Audio input port for controlling the duration of the low states of the output, in number of positive zero crossings of the input signal.
- **(A)** "A" (amplitude) is the Audio input port for controlling the output amplitude. The output signal switches between the high state (Out = +A) and the low state (Out = –A).
- **(In)** "In" is the Audio input port for the signal to be frequency-divided

Output Ports

- **(Out)** "Out" is the Audio output port for the frequency divided signal.

12.16.3 Example: Envelope Follower with Frequency Divider

The Frequency Divider Module takes an Audio clock signal which is based on positive zero crossings and turns it into an Audio clock signal which is based on pulses, including positive zero crossings (when the output value goes from the low to the high pulse state). Additionally, you can reduce the frequency of these pulses (and zero crossings) in respect to the positive zero crossings of the incoming signal.

Looking at the "envelope follower" example in the previous section (see also Sample and Hold Module in ↑12.15, Sample & Hold), you might get the wish to be able to reduce how many peaks detected by the Peak Detector Module (↑12.14, Peak Detector) actually go on to trigger the HR Envelope Module (↑9.4, HR - Env). For this, use the Frequency Divider Module, as shown in the figure below, to create an Audio clock signal with reduced frequency from the positive zero crossings sent from the Differentiator Module (↑10.22, Differentiator). For this application the amplitude of the outgoing Audio clock signal from the Frequency Divider Module can be an arbitrary nonzero value, so we choose "1" (as indicated by the constant connected to that Module's "A" (amplitude input port). With the Frequency Divider Module connected to the HR Envelope Module's (↑9.4, HR - Env)"Trig" (trigger) input port, the envelope is triggered each time the Frequency Divider's output goes from the low pulse state to the high pulse state (positive zero crossing). One this is set up, you only have to define the frequency division factor by connecting values or knobs to the "C+" and "C-" input ports of the Frequency Divider Module. For example, to have on-

ly one in every four positive zero crossings at the Frequency Divider Module's input port send a trigger signal to the HR Envelope Module, set both the "C+" and "C-" values to "2" (as explained in the Ports subsection). Of course, since the Frequency Divider's output pulse width does not matter here, you could also use values "1" and "3" for "C+" and "C-", respectively, or even the other way around. In any case you will get a positive zero crossing at the Frequency Module's output for every four positive zero crossings at its "In" input. This way every fourth peak that is detected by the Peak Detector Module (↑12.14, Peak Detector) actually triggers the HR Envelope Module.

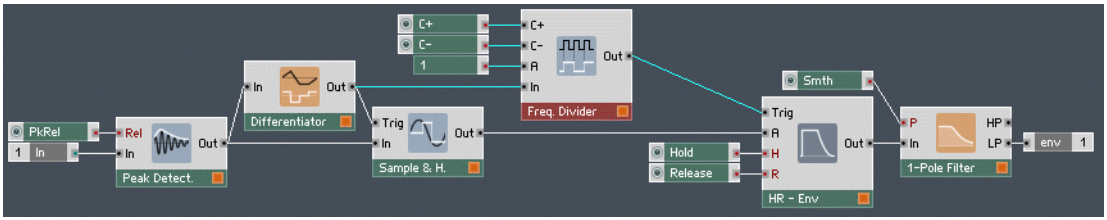


Fig. 12.40 In this modified envelope follower Structure, the Frequency Divider Module lets you filter out positive zero crossings which trigger the HR Envelope Module.

12.17 Audio Table

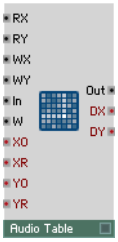


Fig. 12.41 Audio Table Module

12.17.1 Overview

The Audio Table Module holds a table of data values and offers read and write access to these values. The values can be read out as an Audio signal and a Monophonic Audio signal can be loaded from a file or written into the table via the "In" input port. Supported file

types are the native table file (*.ntf), text file (*.txt), and the *.aif and *.wav audio files. Stored values can be displayed and edited graphically with draw and data manipulation functions on the Module's Panel representation.
The internal table can be 1-dimensional (a row of values) where each cell is addressed by the "RX" (read X) and "WX" (write X) input port for read and write operations, respectively. It can also be 2-dimensional (a matrix of rows and columns, or a set of independent rows). In the 2-dimensional case each cell is addressed by the "RX" (read X) and "RY" (read Y) input ports for read operations and by the "WX" (write X) and "WY" (write Y) input ports for write operations.
The value at the "Out" output port is taken from the table by a read operation at the position given by the "RX" and "RY" input ports. With a "read pointer" to the table cells moving at Audio Rate you can get an audible signal at the Audio Table's "Out" port.
Incoming values at the "In" input port are written to individual cells according to the position determined by the "WX" and "WY" input ports, and only when a positive signal lies at the "W" (write) input port.
In the Panel representation's graphic display, "X" is the horizontal table cell position, counted from left to right and "Y" is the vertical table cell position, counted from top to bottom. The table cells are always counted starting with "0" for the first table cell. The Audio Table Module's Panel representation can show all the data or a limited region of it. You can control the range of displayed table cells in the horizontal dimension by specifying the "X" coordinate of the first displayed table cell at the "XO" (X origin) input port and the number of displayed table cells in the "X" dimension at the "XR" (X range) input port. If you have chosen a graph style that shows 2-dimensional tables, you can analogously specify the coordinate of the first table cell in the "Y" dimension at the "YO" (Y origin) input port and the number of displayed table cells in the "Y" dimension at the "YR" input port. Many options in the Properties pages allow customizing the behavior.

For full details on the Properties pages of the Audio Table Module and a tutorial on recording and playing back an Audio signal, please refer to chapter 11 in the Application Reference.

Application

Use the Audio Table Module to record Audio signals within your Structure in a live setting. With extensive possibilities for specifying and warping the read pointer, you can then playback the recorded or loaded table data in a multitude of ways. You can also use the Audio Table Module as an oscillator, as described in the tutorial in chapter 11 of the Application

Reference. The read-out pointer then should be a ramp signal, as supplied by the Ramp Module. Use the "DX" (dimension X) and "DY" (dimension Y) output ports to forward the information about the table dimensions to the ramp signal generator such that the read pointer fits the number of table cells.

12.17.2 Ports

Input Ports

- **(RX)** "RX" (read X) is the Audio input port for the "X" position of the table cell from which the data is read.
- **(RY)** "RY" (read Y) is the Audio input port for the "Y" position of the table cell from which the data is read. This is used in 2-dimensional tables for addressing the row number if more than one row exists.
- **(WX)** "WX" (write X) is the Audio input port for the "X" position of the table cell to which the value at the "In" input port is written. Write operations are executed when a positive value lies at the "W" (write) input port.
- **(WY)** "WY" (write Y) is the Audio input port for the "Y" position of the table cell to which the value at the "In" input port is written. Write operations are executed when a positive value lies at the "W" (write) input port.
- **(In)** "In" is the Audio input port for the signal to be written into the table. When the value at the "W" (write) input port is greater than 0, the value at the "In" input port is written into the table cell that is at the position given by the "WX" and "WY" input ports.
- **(W)** "W" (write) is the Audio input port for activating table write operation.
- **(XO)** "XO" (X origin) is the Event input port for the horizontal offset of the displayed data region. "XO" controls which table cell appears in the left edge of the Panel representation's graphic display. The values at this input port should be specified in the units set with the X Units drop-down menu in the Function page.
- **(XR)** "XR" (X range) is the Event input port for the horizontal range of the displayed data region. "XR" controls how many units of data fit in the display, i.e. it lets you graphically zoom into the data. The values at this input port should be specified in the units set with the X Units drop-down menu in the Function page.

- **(YO)** "YO" (Y origin) is the Event input port for the vertical offset of the displayed data region. "YO" controls which table cell appears in the bottom edge of the Panel representation's graphic display. The values at this input port should be specified in the units set with the Y Units drop-down menu in the Function page.
- **(YR)** "YR" (Y range) is the Event input port for the vertical range of the displayed data region in a 2-dimensional graphic display. "YR" controls how many units of data fit in the display, i.e. it lets you graphically zoom into the data. The values at this input port should be specified in the units set with the Y Units drop-down menu in the Function page.

Output Ports

- **(Out)** "Out" is the Audio output port for the signal read from the table cell at the position controlled by the "RX" and "RY" input ports.
- **(DX)** "DX" (dimension X) is the Event output port for the horizontal size of the table in units set with the X Units drop-down menu in the Function page. Use this output port to forward the information about the table dimension to the Structures for the read and write pointers such that they agree with the number of table cells.
- **(DY)** "DY" (dimension Y) is the Event output port for the vertical size of the vertical table in the units set with the Y Units drop-down menu in the Function page. Use this output port to forward the information about the table dimension to the Structures for the read and write pointers such that they agree with the number of table cells.

13 Event Processing

Event Processing Modules are used for counting Events and keeping track of their values, for logical operations, for splitting and merging control signals, and for timing. You'll also find Modules here for shaping and randomizing control signals. Additionally, there's a full-featured lookup table in the form of the Event Table Module which can be put to various uses such as sequencers or even for buffering Event data.

13.1 Accumulator

Fig. 13.1 Accumulator Module

13.1.1 Overview

The Accumulator Module adds up the values of subsequent Events that arrive at the "In" input port. The value of each Event at the input is added to the total value stored in the Module. Each time an Event arrives at either input port, an Event carrying the new sum as its value is sent from the output port. Use the "Set" input port to set the internal state of the Module to any desired value.

Application

The Accumulator Module can be used to monitor the sum of accumulated Events which then can be clipped, or trigger a new Event, if a threshold has been released. For example, use it for Structures that implement a "stopwatch", as shown in the example below.

13.1.2 Ports

Input Ports

- **(In)** "In" is the input port for the Events whose values are to be summed up by the Module. An Event at this input port creates an Event at the output port.

- **(Set)** "Set" is the Event input port for (re)setting the internal state of the Module. The value of an Event at this input port determines the new value to which values from subsequent Events at the "In" input port are added. An Event at this input port triggers an Event at the output port.

Output Ports

- **(Out)** "Out" Event output port for the accumulated total value.

13.1.3 Example: Stopwatch

In this example (shown in the Structure below) you will build a stopwatch that runs at Control Rate. The output of this Structure is a constant stream of Events at the Control Rate, where each Event carries the total time that has passed since the last stopwatch reset Event.

First, you need a constant stream of Events at the Control Rate. The "CR" (Control Rate) output port of the System Info Module (↑14.20, System Info) is the perfect solution, since each Event additionally carries the Control Rate as its value. After converting the value carried by each Event to the time interval between them in milliseconds, you will use the Accumulator Module to sum up these time increments, resulting in real clock that runs at Control Rate.

First, you need to convert the Control Rate value (in Hz) carried by the Events stemming from the "CR" output port to the corresponding time interval (in milliseconds). As can be seen in the Structure below, you just need to divide 1000 by the Event signal from the "CR" output port). However, this calculation needs to be done only once after the Control Rate has been set. Therefore you can conserve CPU resources by making the 1/ CR calculation only once for every new Control Rate by using the Step Filter Module (↑13.17, Step Filter). Connect the Step Filter Module between the System Info (↑14.20, System Info) and Divide Modules (↑4.8, Divide, /) (shown in the Structure below) and leave the "Tol" (tolerance) input port disconnected (zero). This way you make sure that a new "CR" Event is sent to the Divide Module only if the Control Rate has been changed. This is a handy trick for conserving CPU resources.

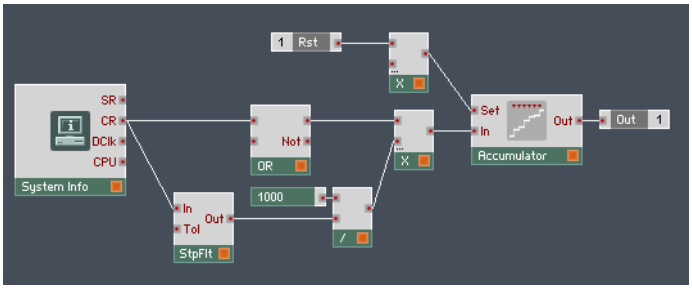


Fig. 13.2 The Structure for a simple Control Rate stopwatch.

Now that you have calculated the time increment in milliseconds between two Events at Control Rate, you need to make sure that Events carrying this value are sent at the Control Rate to the Accumulator Module. You could use a Value Module (↑13.15, Value) to do this, or alternatively, just connect the "CR" output port to an input port of a Logic OR Module (↑13.9, Logic OR). Since the "CR" value is always positive, the Logic OR Module will sent an Event with the value "1" (true) from its output port. Multiply the output of the Divide Module (↑4.8, Divide, /) with the "1" sent from the Logic OR Module and feed it into the "In" input port of the Accumulator Module (↑13.1, Accumulator). The Divide Module multiplies 1 * T at the Control Rate, where "T" is the time interval in milliseconds that you calculated above. Now the Accumulator Module adds up the time interval values at a constant rate and the resulting output is a stopwatch that counts time at the Control Rate, in unit milliseconds. Additionally, add an input for the reset signal for the Accumulator. Again, you could use the Value Module, or just connect the "reset" input port to a Multiply Module (↑4.5, Multiply, X) with one input port disconnected. Any incoming Event at the "reset" input will then cause the Multiply Module to send the value "0" to the "Set" input port of the Accumulator Module (↑13.1, Accumulator) and in turning resetting it to "0", as would be the case for a simple stopwatch.

13.2 Counter

Fig. 13.3 Counter Module

13.2.1 Overview

The Counter Module keeps track of the number of positive incoming Events. A positive Event at the "Up" input port adds an increment of "1" to the internal state of the Counter Module. A positive Event at the "Down" input port reduced the internal state of the Counter Module by "1". Use the "Set" input port to set the internal state to any desired value. Each time an Event that changes the internal state of the Module arrives at an input port, an Event with the new value of the internal state is sent from the output port.

Application

A common use for the Counter Module is to track the positions of sequencers. Another example where you can use the Counter Module is when keeping track of the number of active MIDI notes at an Instrument's MIDI In. This might be useful when creating specialized unison spread or chord effects, where one played MIDI note causes the synthesizer to play several notes at once. Because you have a limited number of Polyphonic Voices in the Instrument (and limited CPU resources), keeping track of the number of active Voices helps you distribute the synthesizer's notes to the Polyphonic Voices in the most optimal manner. For such an application you would use Gate On and Gate Off signals from the Gate Module (↑2.3, Gate) to increment and decrement the internal state of the Counter Module. The internal state would then correspond to the number of active MIDI notes. Make sure that the Event stemming from the Gate Off event is positive valued.

13.2.2 Ports

Input Ports

- **(Set)** "Set" is the Event input port for (re)setting the counter value. The value of an Event at this input port determines the new value of the internal state of the Counter Module. An Event at this input port creates an Event at the output port.
- **(Up)** "Up" is the Event input port for counting up. Only Events with a positive, nonzero value have an effect here, incrementing the internal state of the Module and its output value by one. A positive Event at this input port creates an Event at the output port.
- **(Dn)** "Dn" (down) is the Event input port for counting down. Only Events with a positive, nonzero value have an effect here, decrementing the internal state of the Module and its output value by one. A positive Event at this input port creates an Event at the output port.

Output Ports

- **(Out)** "Out" is the Event output port for the counter value (internal state of the Module).

13.2.3 Example 1: Logic Event Counter

The Counter Module in combination with Compare (↑4.12, Compare) and Logic Modules can be a powerful tool for algorithmic sequencing. The example shown in the figure below utilizes the comparison of three incoming values: "A", "B", and "C". Depending on the relation of these values to each other at any given point in time, the Counter Module will either increment or decrement its internal state. The Logic EXOR Module sends a "1" valued Event from its upper output port if only one (but not both) of its input ports receive the value "1". Such an Event would increment the Counter Module. If both input ports of the Logic EXOR Module receive the value "1" or both "0", its "Not" output port will send a "1" valued Event. This will cause the Counter Module to decrement its internal state.

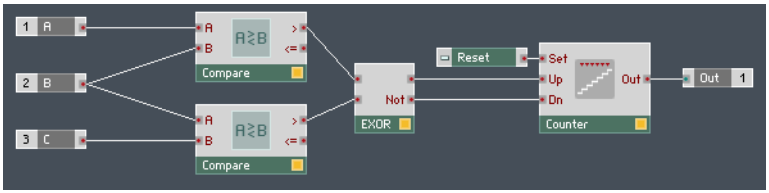


Fig. 13.4 Compare, Logic, and Counter Modules make a useful combination for algorithmic sequencing.

13.2.4 Example 2: Multiplex Sequencer

This example shows how to build a sequencer with variable length, but with a maximum of 16 steps. The first part of the Structure is shown in the figure below. There you see that the core of the sequencer is a Multiplex 16 Module (↑8.5, Multiplex 16 Sequencer) and the values of the steps are received from Knob Modules (↑1.1, Fader/Knob) at the input ports "0" to "15". The sequence length of the step at which the sequencer "folds back" to the first step, is set with the knob labeled "Length" at the "Len" input port. The actual position of the sequencer is received at the "Pos" (position) input port in the form of a regular Events with incremental values. The value of the Event determines the sequencer position where values greater than "Len" are "folded back" to the set sequencer range.

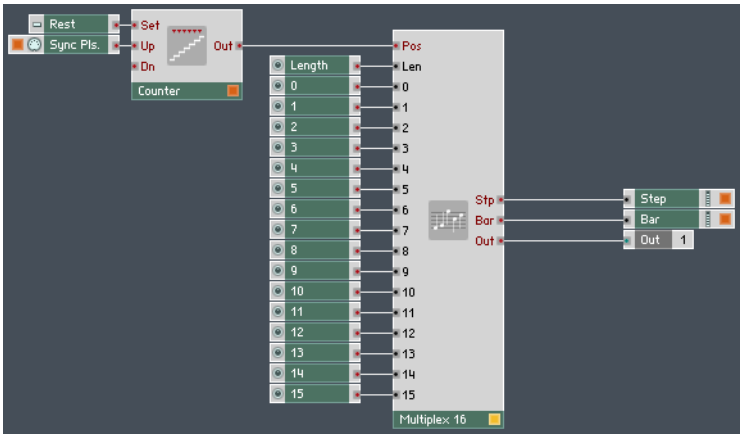


Fig. 13.5 Multiplex Sequencer example 2

In this example, the "Pos" values are ultimately received from the Sync Clock Module (↑2.15, Sync Pulse) which sends clock Events at regular time intervals, synchronized to the MIDI Clock. The rate of outgoing Events can be set at the Sync Clock Module's (↑2.15, Sync Pulse) Function page, with the Rate drop-down menu. In order to get regular Events carrying incremental values, the Counter Module has been used. Each Event from the Sync Clock Module (↑2.15, Sync Pulse) increments the output of the Counter Module by one, which then ends up being used as the "Pos" value for the Multiplex 16 Module (↑8.5, Multiplex 16 Sequencer). At the Counter Module's "Set" input port you should have a Button Module (↑1.2, Button). In its Function page, set the Button Module (↑1.2, Button) to Trigger Mode and its "Max" value to zero. This enables you to reset the sequencer to its first position from the Instrument Panel.

As an additional feature, the XY Module (↑1.14, XY) has been used to graphically track the current sequencer position. The final Structure with this implemented is shown in the first figure at the end of the example and the final Panel representation of the sequencer is shown at the last figure at the end of the example. Please look at that figure to see where we are headed with the XY Module. The XY Module should be configured to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). This corresponds to going to its View page and choosing the *Rectangle* menu entry from the Object Type drop-down menu (shown in the figure below). Also, you don't need a cursor and therefore choose the *None* menu entry from the Cursor Type drop-down menu (also shown in the figure below). While you are

already in the XY Module's View page, disengage the Show Label and Show Value checkboxes and set the Height and Width edit fields so that the XY Module's Panel representation looks similar to what is depicted in the last figure of this example.

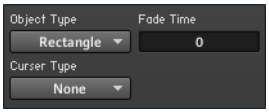


Fig. 13.6 To draw a rectangle from (X1, Y1) to (X2, Y2), choose the Rectangle menu entry from the Object Type drop-down menu.

The rectangle object type causes the XY Module's Panel representation to draw a rectangle from the coordinates (X1, Y1) to the coordinates (X2, Y2). We want to turn the XY Module (↑1.14, XY) into a row of 16 "imaginary" rectangles, each corresponding to one of the 16 steps in the sequencer. This is done the easiest when each box has the dimensions 1 by 1. For this reason, it is necessary to set the range of displayed values correctly in the XY Module's Function page. Set the "Min" and "Max" values for the X coordinate to "0" and "16", respectively. "Min" and "Max" for the Y coordinate should be "0" and "1", respectively. This is shown in the figure below.

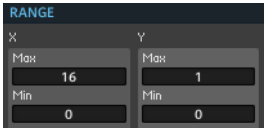


Fig. 13.7 Use the Min and Max edit fields to set the range for the X and Y coordinates to [0 ... 16] and [0 ... 1], respectively.

Now that you have configured the XY Module (↑1.14, XY), you just need to feed the correct values from the Multiplex 16 Module (↑8.5, Multiplex 16 Sequencer) to the input ports for the corresponding coordinates. Remember you wish to draw a rectangle from the coordinates (Step, 0) to (Step + 1, 1). As shown in the Structure depicted below, connect the "Stp" (step) output port off the Multiplex 16 Module to the "X1" input port of the XY Module, and the value Step + 1 (using an Add Module, see ↑4.2, Add, +) to the "X2" input port. Furthermore, connect the constant value "1" to the "Y2" input port. Your resulting Structure should look like the first figure below and its Instrument Panel counterpart should look similar to what is shown in the second figure. Numeric Display Modules (↑1.8, Meter) have been added at the "Stp" and "Bar" output ports to give you quantitative visual access to the step and bar positions of the sequencer.

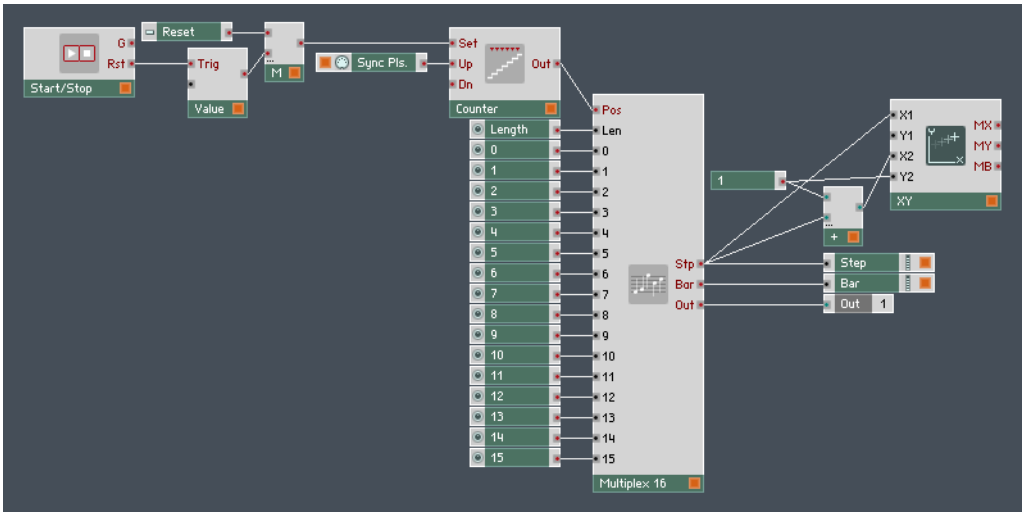


Fig. 13.8 The Structure for a simple sequencer with variable length and a simple step position display.

An additional feature to add would be that resetting the MIDI Clock also causes the sequencer position to be reset as well. For this, a reset operation should send the value "0" to the Counter Module's "Set" input port. Use a Start/Stop Module (↑2.13, Start/Stop) for this "reset" Event source. Since the "Rst" output port sends an Event with value "1", use a Value Module to change the Event's value to "0" and the a Merge Module to merge this reset signal with that received from the "Reset" button.

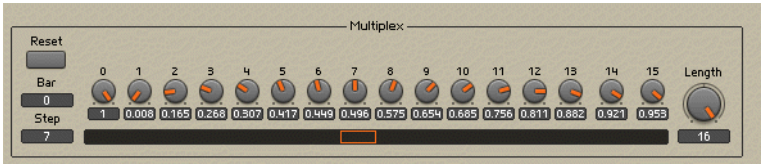


Fig. 13.9 The Instrument Panel of the simple sequencer.

13.3 Randomize

Fig. 13.10 Randomize Module

13.3.1 Overview

The Randomize Module randomizes the incoming signal with adjustable spread. The values of the Events arriving at the "In" input port are modified with a pseudorandom positive or negative offset in the range defined by the "Rng" (range) input port: Out = In + X, where –Rng ≤ X ≤ Rng. That is, X is a random value in the range [-Rng … Rng]. The seed for creating different pseudorandom sequences used by the Randomize Module is set using the Set Random Module (↑14.24, Set Random).

Application

Use the Randomize Module to program your own random unison spread effect. For this you would need to allocate several Polyphonic Voices to one pitch value where each Voice then has a small offset from the original pitch to create a unison spread effect. Of course, you can use the Randomize Module to add a bit of variation to all sorts of parameters, starting from MIDI velocity values to filter cutoff frequency values. In the latter case you would make sure that the original cutoff frequency value is sent as an Event each time a new note is played. This way each time you play a note, the (perhaps otherwise static) cutoff frequency is varied by a random offset in the range [- Rng … Rng].

13.3.2 Ports

Input Ports

- **(Rng)** "Rng" (range) is the Audio input port for controlling the range of the random signal modification.
- **(In)** "In" is the Event input port for the signal to be randomized.

Output Ports

- **(Out)** "Out" is the Event output port for the randomized signal.

13.3.3 Example: Voice Spread

For unison spread effects, the Randomize Module (↑13.3, Randomize) gives a random value offset for each incoming Voice. This example, shown in the figure below, demonstrates this effect by making the offset for each Voice visible from the Instrument Panel using a From Voice Module (↑14.12, From Voice) and a Numeric Display Module (↑1.8, Meter). Use a Button Module (↑1.2, Button) for the incoming signal. In its Function page, set it to

Trigger Mode and both the "Min" and "Max" values to "0", using the corresponding edit fields. This way every time you press the button on the Instrument Panel, an Event with the value "0" (at each Voice) is sent to the "In" input port of the Randomize Module. Now create a control for the "Rng" (range) input port by right-clicking the input port and choosing the *Create Control* menu entry.
In the example, the number of Polyphonic Voices of the parent Instrument has been set to "3" or higher. To extract the value carried by a Polyphonic Voice, use the From Voice Module (↑14.12, From Voice). Insert three From Voice Modules and connect the lower input of each to the Randomize Module's output. Connect Constants with the values "1", "2", "3", each to one From Voice Module's "V" (Voice) input port. As shown in the Structure below, the first From Voice Module outputs the value (including random offset) carried by the first Voice, the second From Voice Module outputs the value carried by the second Voice, and so on. Insert three Numeric Display Modules (↑1.8, Meter) into the Structure to display these values on the Instrument Panel. Remember to engage the Always Active checkbox for at least one of the Numeric Display Modules to activate the Structure. The second figure below shows the Instrument Panel where the button has sent a value "0" and the "Rng" value is "0.52". Sending another zero valued Event to the "In" input port creates a new random offset for each Voice.

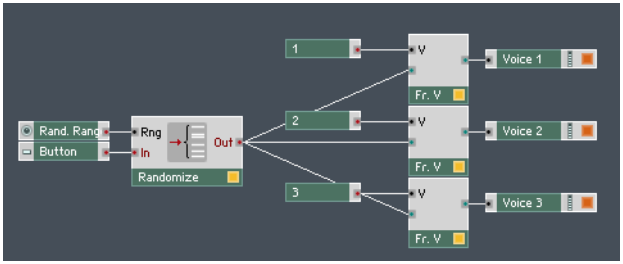


Fig. 13.11 The Structure to test the Voice spreading function of the Randomize Module.

Fig. 13.12 With the From Voice Modules you can see how the Randomize Module spreads out the values for the different Voices.

13.4 Event Frequency Divider

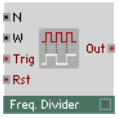


Fig. 13.13 Event Frequency Divider Module

13.4.1 Overview

The Event Frequency Divider Module creates an Event pulse signal at its output port based on the frequency of positive Events at the "Trig" (trigger) input port. If the signal at the "Trig" input port is a train of positive Events at constant frequency, then the effect of the Frequency Divider Module is to divide the frequency of the input Events by a number controlled by the input "N" and send a pulse signal with the resulting frequency from the output port.
Technically, the first positive Event at the "Trig" input port (trigger Event) causes the output signal to be set to "1". Depending on the value at the "W" (width) input port, the output will return to "0" after a number of an trigger Events and return again to "1" after "N" trigger Events.

Application

The Frequency Divider Module is the Event counterpart of the Frequency Divider Module. You can use it to get Clock Events at arbitrary note rates by connecting the output of the Clock Module (↑2.14, Clock) to the "Trig" (trigger) input port. The initial clock rate is equal to 96th notes, but by specifying N = 24 at the "N" input port, for example, you can get a clock signal that runs at the rate of quarter notes. This is useful when choosing the rate at which a sequencer runs, for example.

13.4.2 Ports

Input Ports

- **(N)** "N" (division factor) is the Audio input port for controlling the number of incoming trigger Events that comprise one period of the output pulse signal. N < 2 corresponds to no frequency division, 2 ≤ N < 2.99 corresponds to division by 2, 3 ≤ N < 3.99 corresponds to division by 3, etc.
- **(W)** "W" is the Audio input port controlling the pulse width of the output signal. The range of values at this input port should be [0 ... 1] which corresponds to [0% ... 100%] of the output pulse period. After an initial positive Event at the output port, W = 0 causes the output signal to return zero already at the next trigger Event at the "Trig" input port. In the same situation W = 0.5 causes the output signal to return to zero after half the period determined by "N". And for W = 1 the output signal permanently keeps the value "1".
- **(Trig)** "Trig" (trigger) is the input port for trigger (clock) Events whose frequency is to be divided (e.g. MIDI Clock pulses). An Event is only considered a trigger Event if it has a positive, nonzero value.
- **(Rst)** "Rst" (reset) is the Event input port for resetting the internal counter in order to force a synchronous start of the outgoing pulse signal. Connect this input port to a MIDI Start signal from a Start/Stop Module (↑2.13, Start/Stop) if using the Event Frequency Divider for MIDI synchronization. A reset happens only if an Event with a positive, nonzero value reaches this input port.

Output Ports

- **(Out)** "Out" is the Event output port for the pulse signal that has a frequency determined by the frequency of the trigger Events at the "Trig" input port and the division factor at the "N" input port.

13.4.3 Example: 1/16th Notes Event Source

The Clock Module (↑2.14, Clock) is a source of Events that run at a constant rate of 96th notes, synced to the MIDI Clock. To get arbitrary note divisions, combine the Clock Module with the Frequency Divider Module. As shown in the Structure below, connect the Clock Module to the Frequency Divider Module's "Trig" input port. To get a Gate On Events (followed by Gate Off Events) running at the rate of sixteenth notes, send the value "6" to the

"N" input port. This is because 96 / 6 = 16. To allocate an equal time interval for the Gate On values as to the Gate Off values at the Frequency Divider Module's output, send the value "0.5" to the "W" (width) input port.

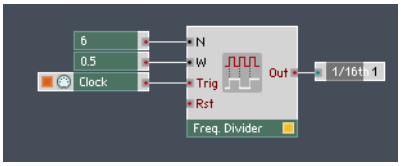


Fig. 13.14 Use the Frequency Divider Module to get arbitrary note divisions from the Clock Module.

13.5 Control Shaper 1

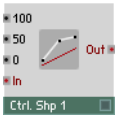


Fig. 13.15 Control Shaper 1 Module

13.5.1 Overview

The Control Shaper 1Module is a signal shaper consisting of a piecewise linear transfer function with three breakpoints. The input signal should be in the range [0 … 1]. You can specify the Shaper's output value for In = 0, In = 0.5, and In = 1 at the "0", "50", and "100" input ports, respectively. The shaping function is linearly interpolated for input values between these breakpoints.

Application

The Control Shaper Modules are used to shape control signals. When building a synthesizer framework, the control signals stemming from knobs and faders are often shaped to achieve a better "feel" when tweaking them. Use the Control Shaper Modules to control the shape of these control curves. You can also try sending the output signal of the LFO Module (↑9.1, LFO) trough a Shaper Module to get more interesting LFO shapes for modulation.

13.5.2 Ports

Input Ports

- **(100)** "100" is the Audio input port for the output value when In = 1 (100%). When disconnected, the value "0" is used for this input port.
- **(50)** "50" is the Audio input port for the output value when In = 0.5 (50%). When disconnected, the value "0" is used for this input port.
- **(0)** "0" is the Audio input port for the output value when In = 0 (0%). When disconnected, the value "0" is used for this input port.
- **(In)** "In" is the Event input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Event output port for the shaped signal.

13.5.3 Example: Simple Shaped Resonance Parameter

You might want to control the resonance parameter of a filter. For filter resonance values close to "1", small changes in the resonance parameter can have a great effect on the filter's output signal. For lower filter resonance values the audible effect of such changes is comparatively small. To shape the signal accordingly, send the "resonance" knob's output signal through the Control Shaper 1 BP Module, as shown in the figure below. Send the constant value "1" to the "100" input port (the default value for the "0" input port is already zero). Since we want relatively precise control for resonance values above 0.98 at the cost of precision for the lower resonance values, connect a Constant with the value "0.98" to the "50" input port. This means that when you turn the "resonance" knob from "0" to "0.5", the Shaper Module's output signal goes from "0" to "0.98". Turning the knob from "0.5" to "1" corresponds to Shaper output values "0.98" to "1". Hence, you have half of your knob's movement range dedicated to control resonance values which correspond nearly bringing the filter to self-oscillation.

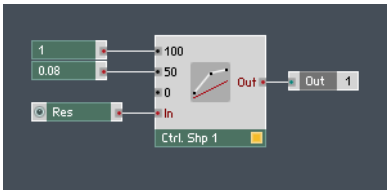


Fig. 13.16 The Structure for an implementation of the Control Shaper 1 BP Module.

13.6 Control Shaper 2

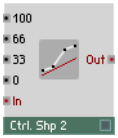


Fig. 13.17 Control Shaper 2 Module

13.6.1 Overview

The Control Shaper 2 Module is a signal shaper consisting of a piecewise linear transfer function with four breakpoints. The input signal should be in the range [0 … 1]. You can specify the Shaper's output value for In = 0, In = 0.33, In = 0.66, and In = 1 at the "0", "33","66", and "100" input ports, respectively. The shaping function is linearly interpolated for input values between these breakpoints.

Application

The Control Shaper Modules are used to shape control signals. When building a synthesizer framework, the control signals stemming from knobs and faders are often shaped to achieve a better "feel" when tweaking them. Use the Control Shaper Modules to control the shape of these control curves. You can also try sending the output signal of the LFO Module (↑9.1, LFO) trough a Shaper Module to get more interesting LFO shapes for modulation.

13.6.2 Ports

Input Ports

- **(100)** "100" is the Audio input port for the output value when the value In = 1 (100%). When disconnected, the value "0" is used for this input port.
- **(66)** "66" is the Audio input port for the output value when In = 0.66 (66%). When disconnected, the value "0" is used for this input port.
- **(33)** "33" is the Audio input port for the output value when In = 0.33 (33%). When disconnected, the value "0" is used for this input port.
- **(0)** "0" is the Audio input port for the output value when In = 0 (0%). When disconnected, the value "0" is used for this input port.
- **(In)** "In" is the Event input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Event output port for the shaped signal.

13.6.3 Example: Shaped Resonance Parameter

You might want to control the resonance parameter of a filter. For filter resonance values close to "1", small changes in the resonance parameter can have a great effect on the filter's output signal. When the resonance parameter is equal to "1", the filter goes into self-oscillation, and for lower filter resonance values the audible effect of such changes is comparatively small. To shape the signal accordingly and to keep the resonance parameter from reaching "1", send the "resonance" knob's output signal through the Control Shaper 2 BP Module (shown in the figure below). Send the constant value "0.99999" to the "100" input port to keep the shaped resonance parameter from reaching "1". Leave the "0" input port disconnected, since its default value is already "0". Since we want relatively precise control for resonance values above 0.98 at the cost of precision for the lower resonance values, connect a Constant with the value "0.98" to the "66" input port. We can further slowly decrease the precision as the incoming values go towards "0" by connecting the "33" input port with the value "0.5". This means that when you turn the "resonance" knob from "0" to "0.33", the Shaper Module's output signal goes from "0" to "0.5". Turning the knob from "0.33" to "0.66" corresponds to Shaper output values "0.5" to "0.98" and turning the knob from "0.66" to "1" sends the values "0.98" to "0.99999" from the Shaper's

Module's output. This way you have a smoother transition for the precision of the resonance parameter from low values to high values and in most cases will keep the filter from going into self-oscillation.

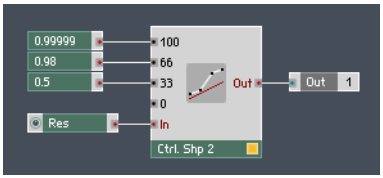


Fig. 13.18 The Structure for an implementation of the Control Shaper 2 BP Module.

13.7 Control Shaper 3

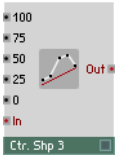


Fig. 13.19 Control Shaper 3 Module

13.7.1 Overview

The Control Shaper 3 Module is a signal shaper consisting of a piecewise linear transfer function with five breakpoints. The input signal should be in the range [0 ... 1]. You can specify the Shaper's output value for In = 0, In = 0.25, In = 0.50, In = 0.75, and In = 1 at the "0", "25","50", "75", and "100" input ports, respectively. The shaping function is linearly interpolated for input values between these breakpoints.

Application

The Control Shaper Modules are used to shape control signals. When building a synthesizer framework, the control signals stemming from knobs and faders are often shaped to achieve a better "feel" when tweaking them. Use the Control Shaper Modules to control the shape of these control curves. You can also try sending the output signal of the LFO Module (↑9.1, LFO) trough a Shaper Module to get more interesting LFO shapes for modulation.

13.7.2 Ports

Input Ports

- **(100)** "100" is the Audio input port for the output value when In = 1 (100%). When disconnected, the value "0" is used for this input port.
- **(75)** "75" is the Audio input port for the output value when In = 0.75 (75%). When disconnected, the value "0" is used for this input port.
- **(50)** "50" is the Audio input port for the output value when In = 0.5 (50%). When disconnected, the value "0" is used for this input port.
- **(25)** "25" is the Audio input port for the output value when In = 0.25 (25%). When disconnected, the value "0" is used for this input port.
- **(0)** "0" is the Audio input port for the output value when In = 0 (0%). When disconnected, the value "0" is used for this input port.
- **(In)** "In" is the Event input port for the signal to be shaped.

Output Ports

- **(Out)** "Out" is the Event output port for the shaped signal.

13.7.3 Example: S-Shaped Control Parameter

Let's say you wish to control a parameter with a control curve in the range [0 … 1] that has an S-shape. This would entail high precision for values close to zero and values close to one at the cost of precision in between. To shape the signal accordingly, send the knob's output signal through the Control Shaper 3 BP Module (shown in the figure below). Send the constant value "1" to the "100" input port to keep the shaped resonance parameter from reaching "1" and the value "0" to the "0" input port. Since we want relatively precise control for values close to the value "1", then connect a Constant with the value "0.98" to the "75" input port. To increase precision for inputs close to the value "0", connect a Constant with the value "0.02" to the "25" input port. To make the S-curved shaping symmetric around the value "0.5", connect the value "0.5" to the "50" input port. Now you have the following shaping behavior for a knob with the range [0 … 1]. Turning the knob from "0" to "0.25", the Shaper Module's output signal goes from "0" to "0.02". Turning the knob from "0.25" to "0.5" corresponds to Shaper output values "0.02" to "0.5". Similarly, turning the knob from "0.5" to "0.75" corresponds to output values from "0.5" to

"0.08" and turning the knob from "0.75" to "1" sends the values "0.98" to "1" from the Shaper's Module's output. This shaping curve could be interesting for knobs that control mixing parameters for effects or between different filters.

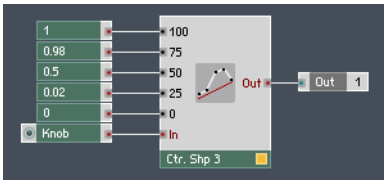


Fig. 13.20 The Structure for an implementation of the Control Shaper 3 BP Module.

13.8 Logic AND

Fig. 13.21 Logic AND Module

13.8.1 Overview

The Logic AND Module is a logic gate for Event signals. The upper output value is the result of the AND logic operation between the two input states The upper output value is "1" (True) when both input values are positive, otherwise the output value is "0" (False).

Application

Logic operations are usually used when conditionally routing signals to and from different parts of your Structure. The input signals can be results of value comparisons as performed by the Compare and Compare/Equal Modules. Another source for logic input ports can be Gate signals.

13.8.2 Ports

Input Ports

- **(1)** "1" is the Event input port for the first signal to be used in the AND logic operation. Negative and zero input values are treated as the logic False state, positive values are treated as the logic True state.

- **(2)** "2" is the Event input port for the second signal to be used in the AND logic operation. Negative and zero input values are treated as the logic False state, positive values are treated as the logic True state.

Output Ports

- **(Out)** "Out" is the Event output port for the result of the AND logic operation. If both input values are positive, the output value is "1" (True). Otherwise it is "0" (False).
- **(Not)** "Not" is the Event output port for the result of the NOT AND logic operation. If both input values are positive, the output value is "0" (False). Otherwise it is "1" (True).

13.8.3 Example: Logic AND Router

The Router Module (↑13.19, Router) can effectively be combined with Logic and Compare Modules (↑4.12, Compare) to selectively route Event signals. In the example shown in the figure below, the Events from the "B" input port are only forwarded by the router, if the results of both (wired) operations done by the Compare Modules are "true", as tested by the Logic AND Module. In other words, the Event at the "B" input port is forwarded only when C > B > A. The Order Module (↑13.12, Order) is used to ensure that the comparison takes place before the Event is sent to the Router Module.

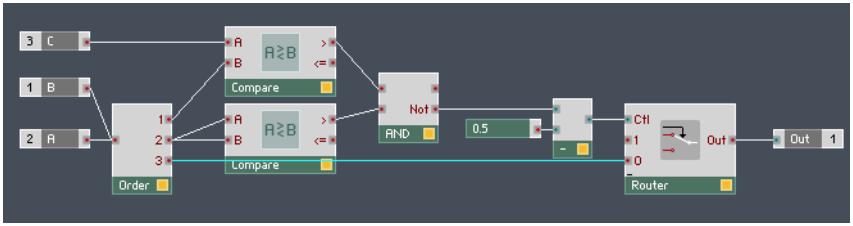


Fig. 13.22 Compare and Logic Modules can be used to determine whether a signal is to be forwarded by a Router Module or not.

13.9 Logic OR

Fig. 13.23 Logic OR Module

13.9.1 Overview

The Logic OR Module is a logic gate for Event signals. The upper output value is the result of the OR logic operation between the two input states. The upper output value is "1" (True) when at least one of the two input values is positive. If both input values are either zero or negative, the output value is "0" (False).

Application

Logic operations are usually used when conditionally routing signals to and from different parts of your Structure. The input signals can be results of value comparisons as performed by the Compare and Compare/Equal Modules. Another source for logic input ports can be Gate signals.

13.9.2 Ports

Input Ports

- **(1)** "1" is the Event input port for the first signal to be used in the OR logic operation. Negative and zero input values are treated as the logic False state, positive values are treated as the logic True state.
- **(2)** "2" is the Event input port for the second signal to be used in the OR logic operation. Negative and zero input values are treated as the logic False state, positive values are treated as the logic True state.

Output Ports

- **(Out)** "Out" is the Event output port for the result of the OR logic operation. If at least one of the two input values is positive, the output value is "1" (True). Otherwise it is "0" (False).
- **(Not)** "Not" is the Event output port for the result of the NOT OR logic operation. If both input values are either zero or negative, the output value is "1" (True). Otherwise it is "0" (False).

13.9.3 Example: Stopwatch

In this example (shown in the Structure below) you will build a stopwatch that runs at Control Rate. The output of this Structure is a constant stream of Events at the Control Rate, where each Event carries the total time that has passed since the last stopwatch reset Event.

First, you need a constant stream of Events at the Control Rate. The "CR" (Control Rate) output port of the System Info Module (↑14.20, System Info) is the perfect solution, since each Event additionally carries the Control Rate as its value. After converting the value carried by each Event to the time interval between them in milliseconds, you will use the Accumulator Module to sum up these time increments, resulting in real clock that runs at Control Rate.

First, you need to convert the Control Rate value (in Hz) carried by the Events stemming from the "CR" output port to the corresponding time interval (in milliseconds). As can be seen in the Structure below, you just need to divide 1000 by the Event signal from the "CR" output port). However, this calculation needs to be done only once after the Control Rate has been set. Therefore you can conserve CPU resources by making the 1/ CR calculation only once for every new Control Rate by using the Step Filter Module (↑13.17, Step Filter). Connect the Step Filter Module between the System Info (↑14.20, System Info) and Divide Modules (↑4.8, Divide, /) (shown in the Structure below) and leave the "Tol" (tolerance) input port disconnected (zero). This way you make sure that a new "CR" Event is sent to the Divide Module only if the Control Rate has been changed. This is a handy trick for conserving CPU resources.

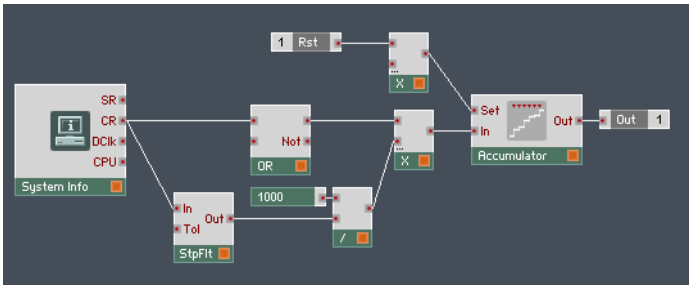


Fig. 13.24 The Structure for a simple Control Rate stopwatch.

Now that you have calculated the time increment in milliseconds between two Events at Control Rate, you need to make sure that Events carrying this value are sent at the Control Rate to the Accumulator Module (↑13.1, Accumulator). You could use a Value Module (↑13.15, Value) to do this, or alternatively, just connect the "CR" output port to an input port of a Logic OR Module. Since the "CR" value is always positive, the Logic OR Module will sent an Event with the value "1" (true) from its output port. Multiply the output of the Divide Module (↑4.8, Divide, /) with the "1" sent from the Logic OR Module and feed it into the "In" input port of the Accumulator Module. The Divide Module multiplies 1 * T at the Control Rate, where "T" is the time interval in milliseconds that you calculated above. Now the Accumulator Module adds up the time interval values at a constant rate and the resulting output is a stopwatch that counts time at the Control Rate, in unit milliseconds. Additionally, add an input for the reset signal for the Accumulator. Again, you could use the Value Module, or just connect the "reset" input port to a Multiply Module (↑4.5, Multiply, X) with one input port disconnected. Any incoming Event at the "reset" input will then cause the Multiply Module to send the value "0" to the "Set" input port of the Accumulator Module (↑13.1, Accumulator) and in turning resetting it to "0", as would be the case for a simple stopwatch.

13.10 Logic EXOR

Fig. 13.25 Logic EXOR Module

13.10.1 Overview

The Logic EXOR Module is a logic gate for Event signals. The upper output value is the result of the EXOR logic operation between the two input states. The upper output value is "1" (True) when exactly one of the two inputs is positive. Otherwise the output value is "0" (False).

Application

Logic operations are usually used when conditionally routing signals to and from different parts of your Structure. The input signals can be results of value comparisons as performed by the Compare and Compare/Equal Modules. Another source for logic input ports can be Gate signals.

13.10.2 Ports

Input Ports

- **(1)** "1" is the Event input port for the first signal to be used in the EXOR logic operation. Negative and zero input values are treated as the logic False state, positive values are treated as the logic True state.
- **(2)** "2" is the Event input port for the second signal to be used in the NOT EXOR logic operation. Negative and zero input values are treated as the logic False state, positive values are treated as the logic True state.

Output Ports

- **(Out)** "Out" is the Event output port for the result of the EXOR logic operation. If one but not both input values are positive, the output value is "1" (True). Otherwise it is "0" (False).
- **(Not)** "Not" is the Event output port for the result of the NOT EXOR logic operation. If both input values are positive, or both input values are zero or negative, the output value is "1" (True). Otherwise it is "0" (False).

13.10.3 Example: Logic Event Counter

The Counter Module (↑13.2, Counter) in combination with Compare (↑4.12, Compare) and Logic Modules can be a powerful tool for algorithmic sequencing. The example shown in the figure below utilizes the comparison of three incoming values: "A", "B", and "C". Depending on the relation of these values to each other at any given point in time, the Counter Module (↑13.2, Counter) will either increment or decrement its internal state. The Logic EXOR Module sends a "1" valued Event from its upper output port if only one (but not both) of its input ports receive the value "1". Such an Event would increment the Counter Module (↑13.2, Counter). If both input ports of the Logic EXOR Module receive the value "1" or both "0", its "Not" output port will send a "1" valued Event. This will cause the Counter Module to decrement its internal state.

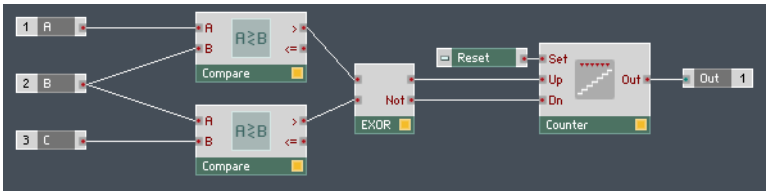


Fig. 13.26 Compare, Logic, and Counter Modules make a useful combination for algorithmic sequencing.

13.11 Logic NOT

Fig. 13.27 Logic NOT Module

13.11.1 Overview

The Logic NOT Module is a logic gate for Event signals. The upper output value is the result of the NOT logic operation on the input states. The upper output value is "1" (True) when the input value is zero or negative, and "0" (False) when the input value is positive.

Application

Logic operations are usually used when conditionally routing signals to and from different parts of your Structure. The input signals can be results of value comparisons as performed by the Compare and Compare/Equal Modules. Another source for logic input ports can be Gate signals.

13.11.2 Ports

Input Ports

- **(In)** "In" is the Event input port for the signal to be used in the NOT logic operation. Negative and zero input values are treated as the logic False state, positive values are treated as the logic True state.

Output Ports

- **(Out)** "Out" is the Event output port for the result of the NOT logic operation. If the input value is positive, the output value is "0" (False). Otherwise it is "0" (False).

- **(Not)** "Not" is the Event output port for the result of the NOT NOT logic operation. If the input value is positive, the output value is "1" (True). Otherwise it is "0" (False).

13.11.3 Example: Gate Off Hold

Use the Logic NOT Module to turn Gate Off signals into "1" valued, Events, for example. This is shown in the Structure below. Further, you can use it to trigger a Hold Module (↑13.22, Hold) (at the "Trig" input port). The Hold Module then sends an Event with the value "1" upon a Gate Off Event and another Event with the value "0" after the hold time has passed.

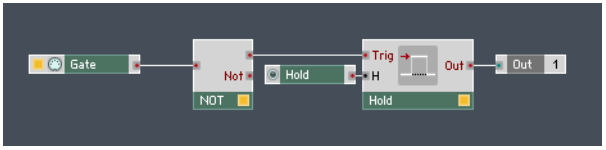


Fig. 13.28 Gate Off signals sent through the Logic NOT Module can be used to trigger Event processing, such as that of a Hold Module .

13.12 Order

Fig. 13.29 Order Module

13.12.1 Overview

The Order Module takes an Event and sends it to its output ports in the following order: “1”, “2”, and then “3”. There is one additional aspect to the timing of the output Events. After the Event at the input port is sent from the “1” output port, the Module “waits” until the signal stemming from the “1” output port has been fully processed by all the Modules downstream before the next Event is sent from the “2” output port. Similarly, the Order Module sends the final Event from the “3” output port only when the signal from the “2” output port has reached all terminuses.

Application

The Order Module is an essential Module in the Primary Level. The functionality of many Modules, such as the Multi Display (↑1.16, Multi Display), Snap Array (↑14.27, Snap Value Array), and Iteration Module depends on the order in which Events arrive at their input ports. For this reason it is important to learn how and when to apply the Order Module to process Events in the correct order and to get the desired functionality from your Structures.

13.12.2 Ports

Input Ports

- **(In)** "In" is the input port for Events to be re-transmitted in a defined order. An Event at this input port triggers an Event at each output port in their order from top to bottom.

Output Ports

- **(1)** "1" is the output port for the Event to be transmitted first to the Structure.
- **(2)** "2" is the output port for the Event to be transmitted second to the Structure.
- **(3)** "3" is the output port for the Event to be transmitted third to the Structure.

13.12.3 Example: Logic AND Router

The Router Module (↑13.19, Router) can effectively be combined with Logic and Compare Modules (↑4.12, Compare) to selectively route Event signals. In the example shown in the figure below, the Events from the "B" input port are only forwarded by the router, if the results of both (wired) operations done by the Compare Modules are "true", as tested by the Logic AND Module (↑13.8, Logic AND). In other words, the Event at the "B" input port is forwarded only when C > B > A. The Order Module is used to ensure that the comparison takes place before the Event is sent to the Router Module.

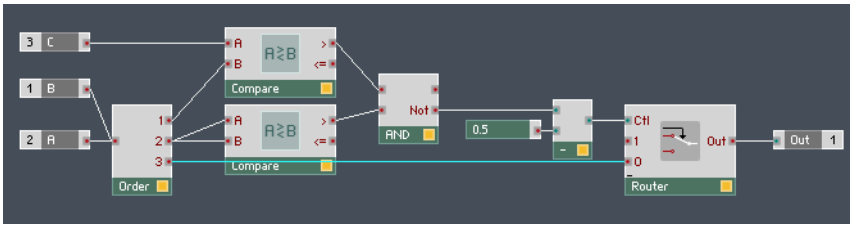


Fig. 13.30 Compare and Logic Modules can be used to determine whether a signal is to be forwarded by a Router Module or not.

13.13 Iteration

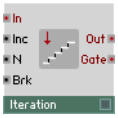


Fig. 13.31 Iteration Module

13.13.1 Overview

An Event arriving at the "In" input port is passed to the "Out" output port and triggers a series of (up to) "N" additional output Events. Each subsequent Event has the same value as its predecessor but incremented by the value at the "Inc" (increment) input port. The iteration can be stopped before all of the "N" subsequent Events have been sent if a positive value is read from the "Brk" (break) input port.

Per default, all Events are sent before the next audio sample is processed. If the Limited Speed checkbox in the Function page is engaged, the Event generation rate can be set limited to the rate set with the Iteration Speed edit field in the Function page.

Application

The Iteration Module is used where iterative Event calculation is needed. IT helps to avoid the construction of Event loops which can lead to unstable operation of REAKTOR. With the Limited Speed checkbox disengaged, the Iteration Module can trigger Event processing that completely takes place before the next audio sample is processed. This not only means that the Iteration Module lets you do things faster than the Audio Rate, but also that you can "instantaneously" populate Multi Display Modules, Event Table Modules, and other entities with Event values that they require for correct functionality.

When using the Iteration Module to populate another Module with data during Initialization, make sure that the correct values have arrived at the "Inc", "N", and "Brk" input ports before the trigger Event arrives at the "In" input port.

13.13.2 Ports

Input Ports

- **(In)** "In" is the Event input port for the trigger signal. An Event at this input triggers N +1 Events at the output.
- **(Inc)** "Inc" (increment) is the Audio input port for the increment for the value of each subsequent Event. The first Event has the value "In", the second "In + Inc", and so on.
- **(N)** "N" (number of Events) is the Audio input port for the number of additional output Events after the Event from the "In" input port has been output. The value at the "N" input port should be a positive integer. Fractional values are rounded down to the nearest integer.
- **(Brk)** "Brk" (break) is the Audio input port for the break signal. A positive value at the "Brk" input port stops the iteration.

Output Ports

- **(Out)** "Out" is the output port for the iterated Events, each incremented from the last by the value "Inc".
- **(Gate)** "Gate" is the Event output port for the iteration Gate signal. Before the first Event leaves the "Out" output port, a Gate On Event (with value "1") is sent from the "Gate" output port. After the last iteration Event, the "Gate" output port sends a Gate Off Event (with value "0").

13.13.3 Properties: Function Page

Limiting the Iteration Speed

Some processes triggered by the Iteration Module might need more CPU resources than are available between the calculations of two audio samples. In such a case, to avoid audio drop-outs, activate the Limited Speed feature.

1. To do this, go to the Iteration Module's Function page and engage the Limited Speed checkbox (shown in the figure below).

2. Then enter the iteration rate (in iterations per second) into the Iteration Speed edit field, shown in the screenshot below.

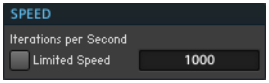


Fig. 13.32 To limit the iteration speed, engage the Limited Speed checkbox and enter the desired number of iterations per second into the Iteration Speed edit field.

13.13.4 Example: Simple Multi Display

Usually instantaneously updating a Multi Display Module's (↑1.16, Multi Display) internal state is done with a Structure that makes use of the Iteration Module. This example illustrates how to plot a graph consisting of three segments on the Multi Display with the help of an Iteration Module. The corresponding Structure is shown in the figure below.

First let's begin by choosing the type of graphical object for plotting our lines. This is specified at the "Obj" input port of the Multi Display Module (↑1.16, Multi Display). For Obj = -3 the Multi Display Module draws a line from the coordinates "X1" and "Y1" of an object to the coordinates "X1" and "Y1" of the next object of the same type. Let's use that, since this way you can draw three line segments by specifying the "X1" and "Y1" coordinates of four objects. Connect a Constant with the value "-3" to the "Obj" input port and make sure that the Ignore Object checkbox in the Multi Display Module's Function page has been engaged. This way all graphical objects have the same object type of "-3".

Now let's turn to making use of the Iteration Module. For each object you need it to trigger the sending of the object's "Idx", "X1", and "Y1" values to the corresponding input ports. Note that the "Idx" value has to arrive first. All in all, since we decided to work with four objects (to draw three segments), we need the Iteration Module to send four Events. To make things easier, let's agree that the "X1" coordinates of the objects are the following, in increasing order of "Idx" value: "0", "1", "2", and "3". These values come in handy, since they can directly be generated by the Iteration Module. Connect a button (labeled "Iterate") in to the "In" input port of the Iteration Module. It should be in Trigger Mode and have its "Max" value in the Function page set to "0". Pressing the button on the Instrument Panel sends an Event with the value "0" to the Iteration Module, which then in turn should output Events carrying the values "0", "0 + 1", "0 + 2", and "0 + 3". You see already that the increment value, as specified at the "Inc" input port should be "1" and that "N" should be "3" (for three additional output Events). This is shown in the Structure below.

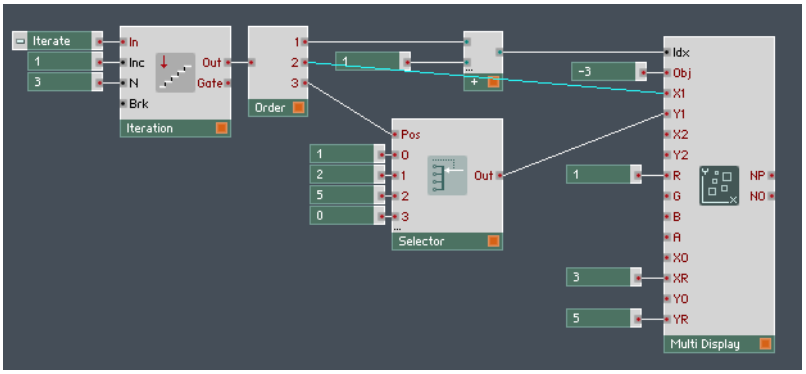


Fig. 13.33 The Structure for drawing three line segments using the Multi Display Module.

Next, use an Order Module (↑13.12, Order) at the Iteration Module's output to cause the "Idx" value to be sent first for each object and only then the "X1" and "Y1" values. Use the "1" output port of the Order Module for the "Idx" value. Since the "Idx" value starts with "1" for the first object and the values from your Iteration Module start with "0", you need to add "1" to the signal from the Order Module's "1" output port before routing it to the "Idx" input port of the Multi Display Module (↑1.16, Multi Display). Now you can connect the "2" output port of the Order Module directly to the "X1" input port of the Multi Display Module since that is how we chose the segment intervals for our plot to be.

Last, you need to specify the "Y1" value for each object. This is most easily done by using the "3" output port of the Order Module (↑13.12, Order) as the "Pos" value for a Selector Module (↑5.1, Selector) where each of the four "channel" input ports has the value corresponding to the "Y2" value of the corresponding graphical object. Such an implementation is shown in the figure above. In that example, the resulting plot would be segments between the points (0, 1), (1, 2), (2, 5), and (3, 0). Connect the output of the Selector Module to the "Y1" input port of the Multi Display Module (↑1.16, Multi Display).

To be able to see all of the plot, set the X Range parameter of the Multi Display Module (↑1.16, Multi Display) at its "XR" input port to "3" and the Y Range parameter at the "YR" input port to "5" (since the greatest "Y1" value used here is "5" and the Y Origin parameter is "0"). To make the lines appear in red, connect the Constant (↑4.1, Constant) "1" to the "R" input port and make sure that the Ignore RGB checkbox in the Multi Display Module's Function page has been engaged. The resulting plot done by the Multi Display Module's (↑1.16, Multi Display) Panel representation is shown in the figure below.

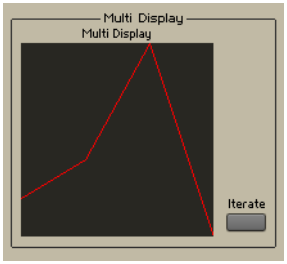


Fig. 13.34 A plot consisting of three line segments done by the Multi Display Module .

13.14 Separator

Fig. 13.35 Separator Module

13.14.1 Overview

The Separator Module compares the values of all incoming Events at the "In" input port to the threshold value (set at the "Thld" input port). These Events are forwarded unaltered to either the "Hi" or "Lo" output port, depending if their value is greater or less than the threshold value. If the value of the incoming Event is greater than "Thld" then it gets sent from the "Hi" output port. If the value of the incoming Event is less than or equal to "Thld" then it gets sent from the "Lo" output port.

Application

The Separator Module is a less general and often more handy way of sorting out Events according to their values (as opposed to the Compare (↑4.12, Compare) and Compare/Equal Modules (↑4.13, Compare/Equal)). One application for the Separator Module would be to convert a Gate signal to a trigger signal by filtering out the Gate Off Events which usually have the value "0". Just set the threshold "Thld" to "0" and use the "Hi" (high) output port to trigger one-shot events such as sample playback or certain envelopes. Also, if you were to build a portamento (pitch glide) Structure, you would probably use the Separator Module to test if the pitch glide has reached the new note and then subsequently use the corresponding output port to turn off any further pitch gliding .

13.14.2 Ports

Input Ports

- **(Thld)** "Thld" (threshold) is the Audio input port to control the threshold value.
- **(In)** "In" is the input port for the Events to be separated and sent to the two outputs according to their values in respect to the threshold value.

Output Ports

- **(Hi)** "Hi" (high) is the Event output port for those incoming Events whose value is greater than the threshold level.
- **(Lo)** "Lo" (low) is the Event output port for those Events whose value is less than or equal to the threshold level.

13.14.3 Example: Event Clipper

A very common Structure in Event processing is the Event clipper. Events which are beyond a certain "Min" or "Max" value are clipped, that is, replaced by the value they surpassed. Events in the allowed range are forwarded unchanged. The Structure shown in the figure below implements this feature. First, the Separator Module tests if the incoming signal is greater than "Max" or not. If not, it is simply forwarded from its "Lo" output port to the next Separator Module. If yes, it is forwarded from its "Hi" output port, triggering the Value Module (↑13.15, Value) to send an Event with the value "Max" to the Merge Module (↑13.16, Merge) at the output. In the case that the first Separator sends the Event on to the next Separator, the analogous procedure happens, only now the incoming Event's value is tested if it is less than "Min". All in all, there are three ways an Event arrives at the output: unchanged, clipped at "Max", or clipped at "Min". All three "sources" are merged using the Merge Module (↑13.16, Merge) and sent to the "Out" output port.

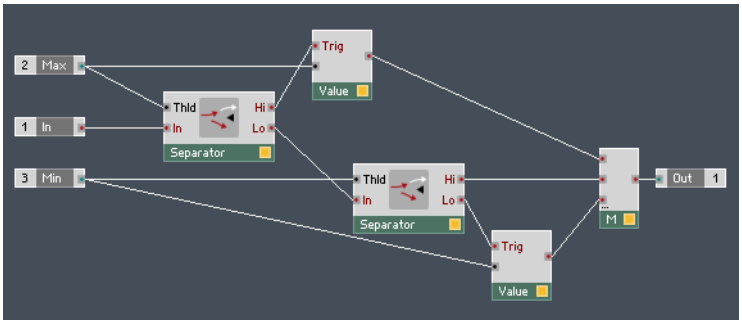


Fig. 13.36 The Structure for an Event clipper.

13.15 Value

Fig. 13.37 Value Module

13.15.1 Overview

The Value Module lets you use an Event at the “Trig” (trigger) input port to trigger an Event from the output port. The output Event carries the value that lies at the bottom input port. You could also look at it this way: Events arriving at the "Trig" input port have their value replaced with the current value at the bottom input port, and are then sent with the new value from the output.

Application

The Value Module is often used in combination with the Order Module (↑13.12, Order). For example, use the Value Module to set the index of an object in the Multi Display Module (↑1.16, Multi Display) before sending it its coordinates and other parameters. The order of the index Event and the parameter Events is controlled by the Order Module (↑13.12, Order).The Value Module can also be used as an Event-controlled sample & hold unit.

Please refer to subsection 9.2.4 in the Application Reference for more information on using the Value Module with the Order Module (↑13.12, Order) to control the flow of Events.

13.15.2 Ports

Input Ports

- **(In)** "In" is the input port for Events that trigger new Events at the output port, but with the value set at the bottom ("Val") input port.
- **(Val)** "Val" (value) is the Audio input port for specifying the value of the outgoing Events.

Output Ports

- **(Out)** "Out" is the Event output port for the Events carrying the value specified by the bottom, "Val", input port.

13.15.3 Example: Event Clipper

A very common Structure in Event processing is the Event clipper. Events which are beyond a certain "Min" or "Max" value are clipped, that is, replaced by the value they surpassed. Events in the allowed range are forwarded unchanged. The Structure shown in the figure below implements this feature. First, the Separator Module (↑13.14, Separator) tests if the incoming signal is greater than "Max" or not. If not, it is simply forwarded from its "Lo" output port to the next Separator Module. If yes, it is forwarded from its "Hi" output port, triggering the Value Module to send an Event with the value "Max" to the Merge Module (↑13.16, Merge) at the output. In the case that the first Separator sends the Event on to the next Separator, the analogous procedure happens, only now the incoming Event's value is tested if it is less than "Min". All in all, there are three ways an Event arrives at the output: unchanged, clipped at "Max", or clipped at "Min". All three "sources" are merged using the Merge Module (↑13.16, Merge) and sent to the "Out" output port.

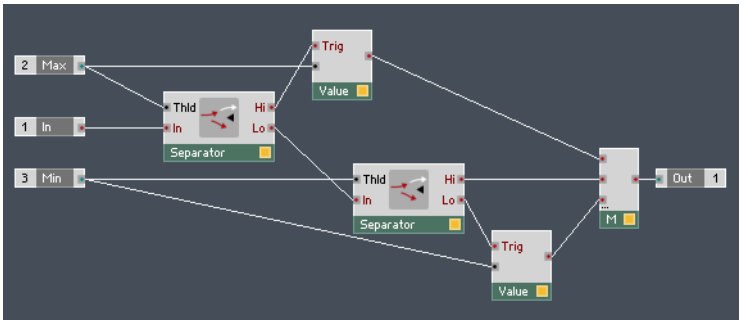


Fig. 13.38 The Structure for an Event clipper.

13.16 Merge

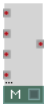


Fig. 13.39 Merge Module

13.16.1 Overview

The Merge Module merges Event signals. When more than one input port is connected, the output value is that of the last received input Event, irrespective of which wire it came from. During Initialization, only the Event at the lowest input port is forwarded to the output.

The module has a dynamic in-port management. Additional input ports can be created (if all existing input ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Merge Module. The minimum number of input ports can be defined with the Min Num Port Groups edit field in the Function page.

An Event at any input port triggers an Event at the output port. Unlike in previous REAKTOR versions, subsequent Events with the same value will not be filtered out anymore. For that purpose use the Step Filter Module (↑13.17, Step Filter).

Application

The Merge Module is used funnel several Event wires to one input port. You might want an envelope or sample to be triggered by Events from a number of sources. In that case you would connect all the wires that carry potential trigger Events to a Merge Module and connect the Merge Module to the Envelope or Sampler Module's gate or trigger input port.

Please refer to subsection 9.2.5 in the Application Reference on the Initialization behavior of the Merge Module.

13.16.2 Example: Event Clipper

A very common Structure in Event processing is the Event clipper. Events which are beyond a certain "Min" or "Max" value are clipped, that is, replaced by the value they surpassed. Events in the allowed range are forwarded unchanged. The Structure shown in the figure below implements this feature. First, the Separator Module (↑13.14, Separator) tests if the incoming signal is greater than "Max" or not. If not, it is simply forwarded from its "Lo" output port to the next Separator Module. If yes, it is forwarded from its "Hi" output port, triggering the Value Module (↑13.15, Value) to send an Event with the value "Max" to the Merge Module at the output. In the case that the first Separator sends the Event on to the next Separator, the analogous procedure happens, only now the incoming Event's value is tested if it is less than "Min". All in all, there are three ways an Event arrives at the output: unchanged, clipped at "Max", or clipped at "Min". All three "sources" are merged using the Merge Module and sent to the "Out" output port.

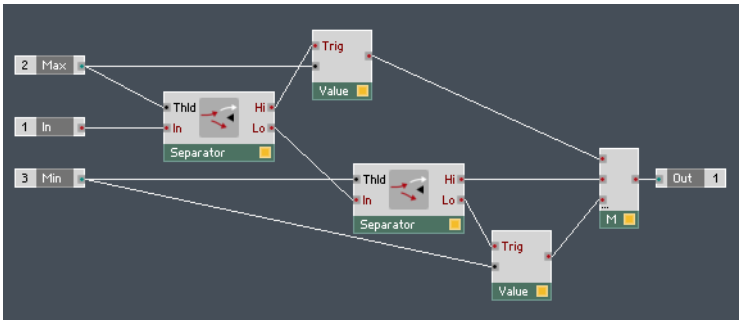


Fig. 13.40 The Structure for an Event clipper.

13.17 Step Filter

Fig. 13.41 Step Filter Module

13.17.1 Overview

The Step Filter Module filters out (stops) Events that have values equal to the previous Event's value or within an adjustable tolerance level thereof. If the previous Event's value is "x", then an Event at the "In" input port is only passed to the output if its value is either larger than x + Tol or smaller than x - Tol where "Tol" is the tolerance parameter set at the corresponding input port.

Application

Use the Step Filter Module to filter out duplicate Events. Since each Event can trigger a number operations down the signal stream which all require CPU operations, it is sometimes optimal to filter out "duplicate" Events that carry the same value. This way you avoid unnecessary Event processing in your Structure.

13.17.2 Ports

Input Ports

- **(In)** "In" is the input port for the Events to be filtered.
- **(Tol)** "Tol" (tolerance) is the Event input port for the tolerance level.

Output Ports

- **(Out)** "Out" is the output port for the Events with values beyond the preceding Event's value and the tolerance level.

13.17.3 Example: Stopwatch

In this example (shown in the Structure below) you will build a stopwatch that runs at Control Rate. The output of this Structure is a constant stream of Events at the Control Rate, where each Event carries the total time that has passed since the last stopwatch reset Event. If the Step Filter were not in the structure then the Accumulator would receive double the number of events - and would accumulate a double the proper rate.

If you change the CR while the stopwatch is running, then the Accumulator will receive one extra Event and will not have the proper output.

First, you need a constant stream of Events at the Control Rate. The "CR" (Control Rate) output port of the System Info Module (↑14.20, System Info) is the perfect solution, since each Event additionally carries the Control Rate as its value. After converting the value carried by each Event to the time interval between them in milliseconds, you will use the Accumulator Module to sum up these time increments, resulting in real clock that runs at Control Rate.

You need to convert the Control Rate value (in Hz) carried by the Events stemming from the "CR" output port to the corresponding time interval (in milliseconds). As can be seen in the Structure below, you just need to divide 1000 by the Event signal from the "CR" output port). However, this calculation needs to be done only once after the Control Rate has been set. Therefore you can conserve CPU resources by making the 1/CR calculation only once for every new Control Rate by using the Step Filter Module. Connect the Step Filter Module between the System Info (↑14.20, System Info) and Divide Modules (↑4.8, Divide, /) (shown in the Structure below) and leave the "Tol" (tolerance) input port disconnected (zero). This way you make sure that a new "CR" Event is sent to the Divide Module only if the Control Rate has been changed. This is a handy trick for conserving CPU resources.

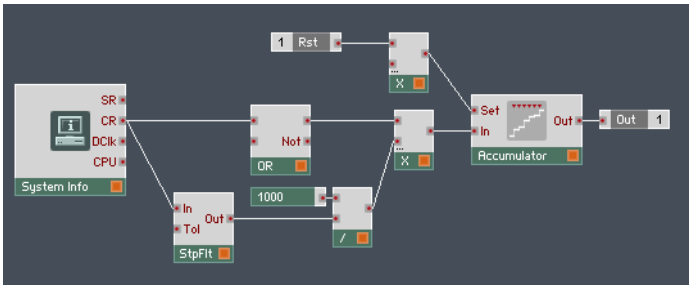


Fig. 13.42 The Structure for a simple Control Rate stopwatch.

Now that you have calculated the time increment in milliseconds between two Events at Control Rate, you need to make sure that Events carrying this value are sent at the Control Rate to the Accumulator Module (↑13.1, Accumulator). You could use a Value Module (↑13.15, Value) to do this, or alternatively, just connect the "CR" output port to an input port of a Logic OR Module (↑13.9, Logic OR). Since the "CR" value is always positive, the Logic OR Module will sent an Event with the value "1" (true) from its output port. Multiply the output of the Divide Module (↑4.8, Divide, /) with the "1" sent from the Logic OR Module and feed it into the "In" input port of the Accumulator Module. The Divide Module multiplies 1 * T at the Control Rate, where "T" is the time interval in milliseconds that you calculated above. Now the Accumulator Module adds up the time interval values at a constant rate and the resulting output is a stopwatch that counts time at the Control Rate, in unit milliseconds. Additionally, add an input for the reset signal for the Accumulator. Again, you could use the Value Module, or just connect the "reset" input port to a Multiply Module (↑4.5, Multiply, X) with one input port disconnected. Any incoming Event at the "reset" input will then cause the Multiply Module to send the value "0" to the "Set" input port of the Accumulator Module (↑13.1, Accumulator) and in turning resetting it to "0", as would be the case for a simple stopwatch.

13.18 Router M to 1

Fig. 13.43 Router M to 1

13.18.1 Overview

The Router M to 1 Module only lets Events through which arrive at the "channel" input port selected by the "Pos" (position) input port. Events at other "channel" input ports are filtered out. The input signals are connected to the input ports below the "Pos" input port. These dynamic input ports are called "channel" input ports. Additional "channel" input ports can be created (if all existing "channel" input ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Router M to 1 Module. The minimum number of "channel" input ports can be defined with the Min Num Port Groups edit field in the Function page. An Event at the selected "channel" input port triggers an Event at the output port.

Controlling the Routing

One can distinguish between three cases for the functionality of the Module depending on the value at the "Pos" input port:

- When the value at the "Pos" input port is an integer "n", the signal at the "n"-th "channel" input port, counting from the top, is forwarded to the output port.
- When the "Pos" value lies between two integers, it is rounded down. This means that for Pos = 0.6 the effective "Pos" value is "0".
- If the value at the "Pos" input port is greater than the number of "channel" input ports, the Module's behavior depends on the setting of the Wrap checkbox in the Function page of the Properties Tab. If the Wrap checkbox is disengaged, the signal at the lowest lying "channel" input port is forwarded to the output port. Otherwise, the "Pos" value wraps around the number of "channel" input ports, "Max", so that Max + 1 becomes "0", Max + 2 becomes "1", and so on.

Application

An application of the Router M to 1 Module is using it in conjunction with a Compare Module or a Compare/Equal Module to selectively route Event to a destination. If in your synthesizer you have several modulation sources such as LFOs and envelopes, then you might wish to be able to choose in the Instrument Panel which of these sources modulates a certain parameter, like the cutoff frequency, for example. Connect all modulation sour-

ces to the Router M to 1 Module and connect the output to the destination parameter. Then you can control the "Pos" value from the Instrument Panel to select which source (LFO, envelope, etc.) modulates said parameter.

Please refer to the header "Switch vs. Router vs. Selector" in the subsection 8.2.3 in the Application Reference for more detailed information on the differences when applying the List and Selector Module's versus just using a Switch Module.

13.18.2 Ports

Input Ports:

- **(Pos)** "Pos" (position) is the hybrid input port for selecting the "channel" input port which is forwarded to the output port. Pos = 0 selects In0 (the first "channel"), Pos = 1 selects In1 (the second "channel"), and so on. The typical range for the "Pos" value is [0 ... Max] where "Max" is the number of "channel" input ports.
- **(In0)** "In0" is the first Event "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Pos" input port.
- **(In1)** "In1" is the second Event "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Pos" input port.
- **(...)** "In2, In3, ..." are the dynamic "channel" input ports. Additional "channel" input ports can be created (if all existing "channel" input ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Router M to 1 Module. The maximum number of "channel" input ports for the Router M to 1 Module is 16.

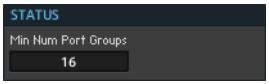
Output Ports:

- **(Out)** "Out" is the output port for the Events from the "channel" input port selected by the "Pos" (position) input port.

13.18.3 Properties: Function Page

Changing the Minimum Number of Input Ports

► To change the minimum number of "channel" input ports, type the desired number into the Min Num Port Groups edit field in the Function page (shown in the figure below). The maximum number of "channel" input ports for the Router M to 1 Module is 16.



Wrapping the "Pos" Value

► If you want the value at the "Pos" input port to "wrap" around the number of "channel" input ports, engage the Wrap checkbox (shown in the figure below).

Activating the Wrap feature means that for "Pos" greater than the number of "channel" input ports, "Max", the new "Pos" value is equal to the old value minus a multiple of "Max" so that the new "Pos" value is in the range [0 ...Max]. For example, Max + 1, becomes 0, Max + 2 becomes 2, and so on. For "Pos" less than zero the new "Pos" value is equal to the negative value plus a multiple of "Max". If the Wrap checkbox is disengaged and Pos > Max, the signal at the "channel" input port with the greatest index is forwarded to the output port and if Pos < 0, the signal at the smallest index is forwarded to the output port.

13.18.4 Example: LFO Waveform Router

The Router M to 1 Module lets you select which "channel" input port's Events should be forwarded. A useful combination is shown in the figure below, where the Router M to 1 Module has been used with the List Module to select which LFO (↑9.1, LFO) waveform to forward.

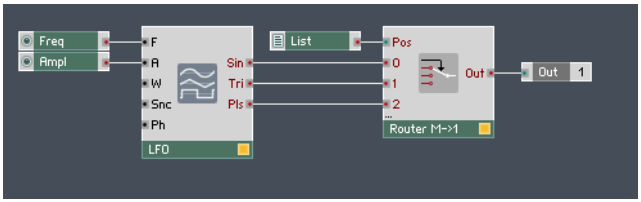


Fig. 13.44 Use the Router M to 1 Module to choose which LFO waveform to use.

13.19 Router

Fig. 13.45 Router Module

13.19.1 Overview

The Router Module is basically the same as a Router M to 1 Module, however it only has a maximum of two "channel" input ports (M = 2). The Router Module only lets Events through which arrive at the "channel" input port selected by the "Ctl" (control) input port. Events at "channel" input ports that have not been selected by the "Ctl" value are filtered out. The Router Module holds the value of the last Event passed through until a new Event arrives at the selected "channel" input port — even if the value at the "Ctl" input port is changed in the meantime.

Events at the upper input port are let through to the output if Ctl > 0. Otherwise the Events at the lower input port are passed to the output. If the lower input port is not connected to anything (or not present), no Events are sent from the Module's output port.

Up to one additional "channel" input port can be created (if the existing "channel" input port is already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an output port of another Module to the three dots on the Router Module. The number of "channel" input ports can be set to "1" or "2" with the Min Num Port Groups edit field in the Function page. An Event at the selected "channel" input port triggers an Event at the output port.

Application

The Router Module can be used as an on/off switch for one Event signal or a toggle switch for two Event signals. It can be substituted by the Router M to 1 Module with one or two inputs. The Router Module is available only for backward compatibility because the "Ctl" signal is interpreted differently.

Please refer to the header "Switch vs. Router vs. Selector" in the subsection 8.2.3 in the Application Reference for more detailed information on the differences when applying the List and Selector Module's versus just using a Switch Module.

13.19.2 Ports

Input Ports

- **(Ctl)** "Ctl" (control) is the Hybrid input for to choose from which "channel" input port Events are forwarded to the Module's output port. If Ctl > 0, Events at the upper input port are let through. Otherwise Events only from the lower input port are passed on. . If the lower input port is not connected to anything (or not present), no Events are sent from the Module's output port.
- **(1)** "1" is the first Event "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Ctl" input port.
- **(0)** "0" is the second Event "channel" input port for the signal that may be forwarded to the output port depending on the value at the "Ctl" input port.

Output Ports

- **(Out)** "Out" is the output port for the Events from the "channel" input port selected by the "Ctl" (control) input port.

13.19.3 Properties: Function Page

Changing the Minimum Number of Input Ports

To choose if one or two "channel" input ports appear on the Module, type the desired number into the Min Num Port Groups edit field in the Function page (shown in the figure below). The maximum number of "channel" input ports for the Router Module is 2.

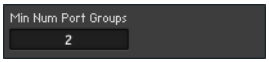


Fig. 13.46 Enter the desired number of minimum input ports into the Min Num Port Groups edit field. These ports show up even if they are unconnected.

13.19.4 Example: Logic AND Router

The Router Module can effectively be combined with Logic and Compare Modules (↑4.12, Compare) to selectively route Event signals. In the example shown in the figure below, the Events from the "B" input port are only forwarded by the router, if the results of both (wired) operations done by the Compare Modules are "true", as tested by the Logic AND Module (↑13.8, Logic AND). In other words, the Event at the "B" input port is forwarded only when C > B > A. The Order Module (↑13.12, Order) is used to ensure that the comparison takes place before the Event is sent to the Router Module.

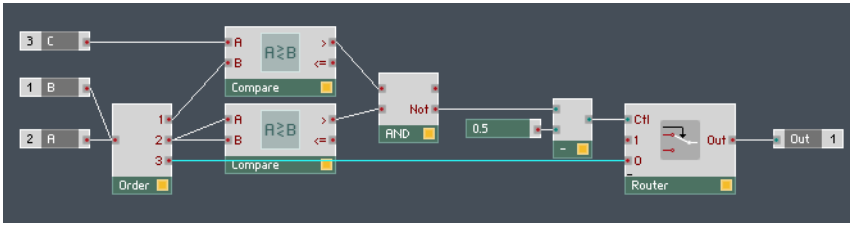


Fig. 13.47 Compare and Logic Modules can be used to determine whether a signal is to be forwarded by a Router Module or not.

13.20 Router 1 to M

Fig. 13.48 Router 1 to M Module

13.20.1 Overview

The Router 1 to M Module forwards incoming Events only to the "channel" output port selected by the "Pos" (position) input port. Other "channel" output ports send no Events. The output ports are dynamic output ports. Additional "channel" output ports can be created (if all existing "channel" output ports are already connected) by holding down the Ctrl key in

Windows (Cmd key in Mac OS X) while dragging a wire from an input port of another Module to the three dots on the Router 1 to M Module. The minimum number of "channel" output ports can be defined with the Min Num Port Groups edit field in the Function page. An Event at the "In" input port triggers an Event at the selected output port.

Controlling the Routing

One can distinguish between three cases for the functionality of the Module depending on the value at the "Pos" input port:

- When the value at the "Pos" input port is an integer "n", the signal at the "In" input port is forwarded to the "n"-th "channel" output port, counting from the top.
- When the "Pos" value lies between two integers, it is rounded down. This means that for Pos = 0.6 the effective "Pos" value is "0".
- If the value at the "Pos" input port is greater than the number of "channel" output ports, the Module's behavior depends on the setting of the Wrap checkbox in the Function page of the Properties Tab. If the Wrap checkbox is disengaged, the signal is forwarded to the lowest lying "channel" output port. Otherwise, the "Pos" value wraps around the number of "channel" output ports, "Max", so that Max + 1 becomes "0", Max + 2 becomes "1", and so on.

Application

An application of the Router 1 to M Module is using it in conjunction with a Compare Module or a Compare/Equal Module to selectively route Event signals in a more complex manner than the Separator Module (↑13.14, Separator) can offer. Also, if in your synthesizer you have one LFO Module and wish to be able to choose from the Instrument Panel which modulation destination it reaches, using the Router 1 to M Module is an option.

Please refer to the header "Switch vs. Router vs. Selector" in the subsection 8.2.3 in the Application Reference for more detailed information on the differences when applying the List and Selector Module's versus just using a Switch Module.

13.20.2 Ports

Input Ports:

- **(Pos)** "Pos" (position) is the hybrid input port for selecting the "channel" output port to which the Events at the "In" input port forwarded. Pos = 0 selects In0 (the first "channel"), Pos = 1 selects In1 (the second "channel"), and so on. The typical range for the "Pos" value is [0 ... Max] where "Max" is the number of "channel" output ports.
- **(In)** "In" is the Event input port whose Events are forwarded to one of the "channel" output ports, depending on the value at the "Pos" (position) input port.

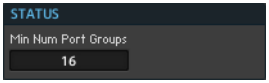
Output Ports:

- **(Out0)** "Out0" is the first Event "channel" output port for the signal from the "In" input port, depending on the value at the "Pos" input port.
- **(Out1)** "Out1" is the second Event "channel" output port for the signal from the "In" input port, depending on the value at the "Pos" input port.
- **(...)** "Out2, Out3, ..." are the dynamic "channel" output ports. Additional "channel" output ports can be created (if all existing "channel" output ports are already connected) by holding down the Ctrl key in Windows (Cmd key in Mac OS X) while dragging a wire from an intput port of another Module to the three dots on the Router 1 to M Module. The maximum number of "channel" output ports for the Router 1 to M Module is 16.

13.20.3 Properties: Function Page

Changing the Minimum Number of Output Ports

► To change the minimum number of "channel" output ports, type the desired number into the Min Num Port Groups edit field in the Function page (shown in the figure below). The maximum number of "channel" input ports for the Router 1 to M Module is 16.



Wrapping the "Pos" Value

► If you want the value at the "Pos" input port to "wrap" around the number of "channel" output ports, engage the Wrap checkbox (shown in the figure below).

Activating the Wrap feature means that for "Pos" greater than the number of "channel" output ports, "Max", the new "Pos" value is equal to the old value minus a multiple of "Max" so that the new "Pos" value is in the range [0 ...Max]. For example, Max + 1, becomes 0, Max + 2 becomes 2, and so on. For "Pos" less than zero the new "Pos" value is equal to the negative value plus a multiple of "Max". If the Wrap checkbox is disengaged and Pos > Max, the signal at the "In" input port is forwarded to the "channel" output port with the greatest index and if Pos < 0, it is forwarded to the "channel" output port with the smallest index.

13.20.4 Example: Choosing Modulation Destination

The Router 1 to M Module lets you select to which "channel" output port incoming Events should be forwarded. A useful combination is shown in the figure below, where the Router 1 to M Module has been used with the List Module to selectively route a modulation signal, in this case an LFO (↑9.1, LFO), to three different destinations: a filter cutoff, a filter resonance, and an envelope sustain parameter.

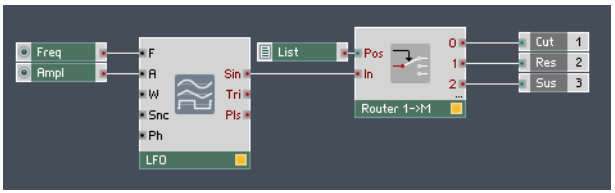


Fig. 13.49 Use the Router 1 to M Module to choose a destination for an Event modulation signal such as an LFO.

13.21 Timer

Fig. 13.50 Timer Module

13.21.1 Overview

The Timer Module measures the elapsed time between the two last received Events at the "Trig" (trigger) input port and outputs the time period and corresponding frequency at the "T" (time) and "F" (frequency) output ports, respectively.

Application

You could build a device allows you to "tap" a heard tempo on a button on the Instrument Panel and have the Timer Module pick up the frequency of the taps and calculate the corresponding BPM. For this you need to make sure that the connected Button Module (↑1.2, Button) been set to Trigger Mode.

13.21.2 Ports

Input Ports

- **(Trig)** "Trig" (trigger) is the Polyphonic input port for the Events between which the time duration is to be measured.

Output Ports

- **(F)** "F" (frequency) is the Polyphonic Event output port for the frequency of the incoming Events, in Hz (Events per second).
- **(T)** "T" (time) is the Polyphonic Event output port for the time between the Events, in milliseconds.

13.21.3 Example: BPM Tap

A simplified example for the Timer Module is shown in the figure below. The Structure lets you tap a button on the Instrument Panel and has the Timer Module calculate the BPM corresponding to the tap frequency of the last few Events. For this functionality, connect a Button Module (in Trigger Mode) to the "Trig" input port and multiply the value from the "F" (frequency) output port by "60". This last operation converts the frequency units Events per second (Hz) to Events per minute (since there are 60 seconds in a minute). Display the result using a Numeric Display Module (↑1.8, Meter).

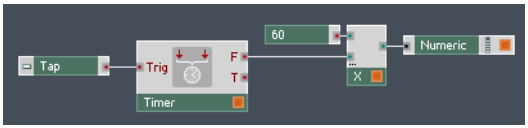


Fig. 13.51 The Structure for a simple BPM timer.

13.22 Hold

Fig. 13.52 Hold Module

13.22.1 Overview

The Hold Module is a hold function triggered by a positive Event at the "Trg" (trigger) input port, upon which an Event with the value "1" (Gate On) is sent from the output port. After the hold period (set at the "H" (hold) input port), an Event with the value "0" (Gate Off) is sent from the output. The Module can be triggered at any time, including retriggering during the hold time.

Application

You can use the Hold Module together with the Separator Module (↑13.14, Separator) to separate the Gate On and Gate Off Events. For example, you can cause an incoming positive Event to trigger a process which is then stopped or modified by the Gate Off Event after the hold time has passed.
In order to save CPU load, the Hold Module's hold time is based on the Control Rate clock and can be very coarse. For precisely timed pulses please use the H Envelope Module (↑9.3, H - Env).

13.22.2 Ports

Input Ports

- **(Trig)** "Trig" (trigger) is the input port for Events to trigger the hold function. Only events with positive values (not zero) have an effect here.

- **(H)** "H" (hold) is the Audio input port for controlling the hold time in milliseconds. Typical values at this input port are in the range [10 ... 1000]. When this input port is disconnected, the default value of "0" is used (no hold time).

Output Ports

- **(Out)** "Out" is the Event output port for the Gate On and Gate Off Events.

13.22.3 Example: Gate Off Hold

Use the Logic NOT Module (↑13.11, Logic NOT) to turn Gate Off signals into "1" valued, Events, for example. This is shown in the Structure below. Further, you can use it to trigger a Hold Module (at the "Trig" input port). The Hold Module then sends an Event with the value "1" upon a Gate Off Event and another Event with the value "0" after the hold time has passed.

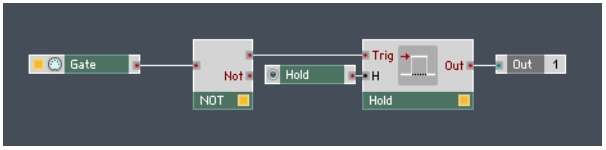


Fig. 13.53 Gate Off signals sent through the Logic NOT Module can be used to trigger Event processing, such as that of a Hold Module.

13.23 Event Table

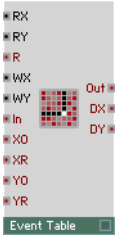


Fig. 13.54 Event Table Module

13.23.1 Overview

Holds a table of data values. Event values can be read from the table, event values can be stored in the table and the table's content can be displayed and edited graphically . The table can be 1-dimensional (a row of values) adressed by X, or 2-dimensional (a matrix of rows and columns, or a set of independent rows) adressed by X and Y.
The value at the output is taken from the table by reading at the position given by the inputs RX and RY. Values arriving at the module's W input are stored in individual cells of the table according to the write position given by the inputs WX and WY.
X is the horizontal position from left to right and Y is the vertical position from top to bottom. The count always starts at 0 for the first element.
The module's panel display can show all the data or a limited region of it. Many options in the properties allow customizing the behaviour.
For full details on properties, menus and keyboard shortcuts, please see the section Table Modules on page 165.

13.23.2 Overview

The Event Table Module holds a table of data values and offers read and write access to these values. The values can be read out as an Event signal and a Monophonic signal can be loaded from a file or written into the table via Events at the "In" input port. Supported file types are the native table file (*.ntf), text file (*.txt), and the *.aif and *.wav audio files. Stored values can be displayed and edited graphically with draw and data manipulation functions on the Module's Panel representation.
The internal table can be 1-dimensional (a row of values) where each cell is addressed by the "RX" (read X) and "WX" (write X) input port for read and write operations, respectively. It can also be 2-dimensional (a matrix of rows and columns, or a set of independent rows). In the 2-dimensional case each cell is addressed by the "RX" (read X) and "RY" (read Y) input ports for read operations and by the "WX" (write X) and "WY" (write Y) input ports for write operations.
The value at the "Out" output port is taken from the table by a read operation at the position given by the "RX" and "RY" input ports. With a moving "read pointer" to the table cells you can record arbitrary Event signals.
Incoming values at the "In" input port are written to individual cells according to the position determined by the "WX" and "WY" input ports at the time of the Event's arrival.

In the Panel representation's graphic display, "X" is the horizontal table cell position, counted from left to right and "Y" is the vertical table cell position, counted from top to bottom. The table cells are always counted starting with "0" for the first table cell. The Event Table Module's Panel representation can show all the data or a limited region of it. You can control the range of displayed table cells in the horizontal dimension by specifying the "X" coordinate of the first displayed table cell at the "XO" (X origin) input port and the number of displayed table cells in the "X" dimension at the "XR" (X range) input port. If you have chosen a graph style that shows 2-dimensional tables, you can analogously specify the coordinate of the first table cell in the "Y" dimension at the "YO" (Y origin) input port and the number of displayed table cells in the "Y" dimension at the "YR" input port. Many options in the Properties pages allow customizing the behavior.

For full details on the Properties pages of the Event Table Module and a general tutorial on the Table Modules by example of recording and playing back an Audio signal with the Audio Table Module, please refer to chapter 11 in the Application Reference.

Application

The most common application for the Event Table Module is building all types of custom sequencers. Connect the read pointer to a MIDI Clock signal (created with the help of a Song Position Module (↑3.10, Song Position Out) and a Clock Module (↑2.14, Clock), for example). Another, more advanced application is using several Event Table Modules to access a shared buffer from different points in the Structure. To do this, configure the table by setting the table size, units, etc. Then save the table file. Now when you copy and paste another instance of the Event Table Module elsewhere into the Structure, both Modules will use the same internal table. This is reflected by the Clients display in the Function page (shown in the figure below). Making changes to one table will be reflected by the state of the other Module. Note that even if you delete the table file from your disk, the Modules will still share the same internal state.

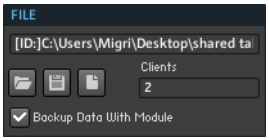


Fig. 13.55 The Clients display in the Function page shows how many other Table Modules (of the same kind) are using the internal table.

13.23.3 Ports

Input Ports

- **(RX)** "RX" (read X) is the Audio input port for the "X" position of the table cell from which the data is read. A read operation takes place when an Event arrives at the "R" (read) input port.
- **(RY)** "RY" (read Y) is the Audio input port for the "Y" position of the table cell from which the data is read. This is used in 2-dimensional tables for addressing the row number if more than one row exists. A read operation takes place when an Event arrives at the "R" (read) input port.
- **(R)** "R" (read) is the Event input port for triggering a read operation from the table cell specified by the "RX" and "RY" input ports. Every Event at this input port triggers an Event at the "Out" output port, carrying the value in the addressed table cell.
- **(WX)** "WX" (write X) is the Audio input port for the "X" position of the table cell to which the value of the Event at the "In" input port is written. Write operations are executed at the moment when the Event arrives at the "In" input port.
- **(WY)** "WY" (write Y) is the Audio input port for the "Y" position of the table cell to which the value at the "In" input port is written. Write operations are executed when a positive value lies at the "W" (write) input port.
- **(In)** "In" is the input port for the Events whose values are to be written into the table cells specified addressed by the "WX" and "WY" input ports.
- **(XO)** "XO" (X origin) is the Event input port for the horizontal offset of the displayed data region. "XO" controls which table cell appears in the left edge of the Panel representation's graphic display. The values at this input port should be specified in the units set with the X Units drop-down menu in the Function page.
- **(XR)** "XR" (X range) is the Event input port for the horizontal range of the displayed data region. "XR" controls how many units of data fit in the display, i.e. it lets you graphically zoom into the data. The values at this input port should be specified in the units set with the X Units drop-down menu in the Function page.
- **(YO)** "YO" (Y origin) is the Event input port for the vertical offset of the displayed data region. "YO" controls which table cell appears in the bottom edge of the Panel representation's graphic display. The values at this input port should be specified in the units set with the Y Units drop-down menu in the Function page.

- **(YR)** "YR" (Y range) is the Event input port for the vertical range of the displayed data region in a 2-dimensional graphic display. "YR" controls how many units of data fit in the display, i.e. it lets you graphically zoom into the data. The values at this input port should be specified in the units set with the Y Units drop-down menu in the Function page.

Output Ports

- **(Out)** "Out" is the output port for the Event carrying the value in the table cell at the position controlled by the "RX" and "RY" input ports. Events at the "Out" output port can only be triggered by Events at the "R" (read) input port.
- **(DX)** "DX" (dimension X) is the Event output port for the horizontal size of the table in units set with the X Units drop-down menu in the Function page. Use this output port to forward the information about the table dimension to the Structures for the read and write pointers such that they agree with the number of table cells.
- **(DY)** "DY" (dimension Y) is the Event output port for the vertical size of the vertical table in the units set with the Y Units drop-down menu in the Function page. Use this output port to forward the information about the table dimension to the Structures for the read and write pointers such that they agree with the number of table cells.

14 Auxiliary

If you can't find it anywhere else, it's probably here. First you'll find tape decks for recording and playing back audio files. Then there are Modules for managing Polyphonic Voices, for converting Audio signals to Control signals, and for reporting the status of various REAKTOR processes such as tuning and tempo.

14.1 Tapedeck 1 Channel

Fig. 14.1 Tapedeck 1 Channel Module

14.1.1 Overview

The Tapedeck 1 Channel Module comprises a mono tape deck for recording and playback of Audio signals. Audio files can be read from and written to either the internal memory (Internal Memory Mode) or the hard disk (Hard Disk Mode), depending on the settings in the Function page.

When Hard Disk Mode is enabled, the files are written into the "Audio Files" folder you specify in the REAKTOR Preferences. Please refer to subsection 14.4.2 in the Application Reference to learn how to specify the "Audio Folder". REAKTOR does sample rate conversion on the fly and writes the file in the sample rate which is currently used by your audio system.

In Internal Memory Mode audio data is kept only in your computer's random access memory (RAM) and you have to set the maximum recording length in the Function page. This puts an upper limit on the amount of RAM that is consumed by the Tapedeck 1 Channel Module. How big this value can be depends on the amount of available RAM storage in the computer. At a sample rate of 44.1 kHz you need 86 kB of RAM for each second of buffer length; for one minute you need 5 MB.

If the buffer memory for the Tapedeck 1 Channel Module has been set to too large a value, virtual memory swaps performed by your computer's operating system can cause significant hard disk activity during recording. This can prevent REAKTOR from processing audio smoothly, leading to pops, crackles, and in worst cases to audio drop-outs.

When importing an audio file into the Tapedeck 1 Channel Module, the length of the Module's memory buffer is adjusted to match the loaded data. If you later want to make a recording which is longer than this, you will first need to go to the Function page and set a bigger value for the maximum recording length. When loading an audio file, REAKTOR will automatically convert its playback to the current REAKTOR Sample Rate.

Be aware that REAKTOR converts all audio files stored in Tapedeck Modules which are working in Internal Memory Mode to the new Sample Rate whenever REAKTOR switches to a new Sample Rate. So you should avoid changing the Sample Rate if your ensemble contains memory based audio samples. If the audio file is stored on the hard disk, you can use the Reload button after changing the Sample Rate to import the file again.

A recording can be exported using the Save button in the Function page. If the file has already been exported, you will be asked if you want to overwrite the file, e.g. if you have made a new recording in the Tapedeck 1 Channel Module. With the Save as... button you can store the file under a new name.

The Tapedeck 1 Channel Module's View page provides the same options as that of a Sampler Module. Please refer to subsection 10.3.2 in the Application Reference for a detailed overview of the Tapedeck 1 Channel Module's View page.

Application

Use the Tapedeck 1 Channel Module to quickly achieve audio file recording and playback functionality that you can control from the Instrument Panel and synchronize to other processes. You could, for example, create a sampler Instrument that sends the output of a Sampler Module through many effects which then end up being recorded to the hard-disk by a Tapedeck Module. The point would be to record the Sampler's processed output, save it and then to load it again into the Sampler Module. Now you playback the processed Sample and add another iteration of effects (or mix it up with the original signal to create interesting characteristics stemming from phase effects). The Tapedeck Module is essential in such a case since you can easily synchronize the playback of the Sampler Module to the recording of the Tapedeck Module. For an example, please refer to the Tapedeck 2

Channel Module's (↑14.2, Tapedeck 2 Channel) entry. The Tapedeck 1 Channel Module (↑14.1, Tapedeck 1 Channel) is meant for mono signals. Use the Tapedeck 2 Channel Module for stereo signals.

14.1.2 Ports

Input Ports

- **(Rec)** "Rec" (record) is the Event input port for switching recording on and off, with a Gate signal, for example. Recording is started upon the arrival of a positive valued Event. Recording is ended when an Event with a negative or zero value arrives or when the maximum recording time is reached.
- **(Play)** "Play" is the Event input port for switching playback on and off, with a Gate signal, for example. Playback is started upon the arrival of a positive valued Event. Playback is ended when an Event with a negative or zero value arrives or when the maximum recording time is reached.
- **(Lp)** "Lp" (loop playback) is the Event input port for switching loop playback on and off. When this input port receives a positive valued Event, loop playback is turned on. When during playback the end of the recording or file is reached, playback starts from the beginning again. The result is an infinite loop while playback is switched on. Send a zero or negative valued Event to turn looped playback off.
- **(In)** "In" is the Monophonic Audio input port for the signal to be recorded. The signal should be within the range [-1 … 1]. To record a Polyphonic signal, a Voice Combiner Module (↑14.3, Audio Voice Combiner) has to be inserted.
- **(Pse)** "Pse" (pause) is the Event input port for pausing the recording or playback. An Event with a value greater than zero pauses the tape deck, an Event with zero or negative values continues the playback.
- **(Pos)** "Pos" (position) is the Event input port for setting the playback position in milliseconds. The typical range of values at this input port is [0 … 20000]. When disconnected, the default value for this input port is "0" (start of the recording).
- **(Spd)** "Spd" (speed) is the Event input port for controlling the playback speed of the tape deck. Values at this input port must be greater than "0". A value of "1" corresponds to normal playback speed. The typical range of values at this inptu port is [0.8 … 1.2]. When this input port is disconnected, the default value that is used, is "1" (normal playback speed).

Output Ports

- **(Out)** "Out" is the Audio output port for the playback signal or the signal being recorded.
- **(Rec)** "Rec" (recording) is the Event output port for the recording state of the tape deck. An Event with the value Rec = 1 is sent when the tape deck starts recording, an Event with the value Rec = 0 is sent when recording is stopped.
- **(Play)** "Play" (playing) is the Event output port for the playback state of the tape deck. An Event with the value Play = 1 is sent when the tape deck starts playing, an Event with the value Play = 0 is sent when playback is stopped.
- **(Wrp)** "Wrp" (wrap) is the Event output port for counting the times that playback has looped. An Event with value "1" is sent each time the tape wraps around to the beginning when in looped playback.
- **(Pse)** "Pse" (pause) is the Event output port for the pause state of the tape deck. An Event with the value Pse = 1 is sent when recording or playback is paused, an Event with the value "0" is sent when recording or playback is continued.
- **(Pos)** "Pos" (position) is the Audio output port for the current playback/recording position in milliseconds [0 ... <length of sample in ms>].
- **(Len)** "Len" (length) is the Event output port for the length of the recording in milliseconds.

14.1.3 Properties: Function Page

Keeping the Tapedeck 1 Channel Module Always Active

The Tapedeck 1 Channel Module only records and plays back signals if it is part of an active signal flow.

► To keep the Tapedeck 1 Channel Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Hard Disk Mode

In Hard Disk Mode, audio files are written to and read from hard disk directly.

► To activate Hard Disk Mode, engage the Stream audio from/to hard disk checkbox (shown in the figure below).

► To load an audio file for playback, press the Select File... button (shown in the screenshot below).
► To create a new file named "untitled", press the New File button (shown in the screenshot below). If a file with the name already exists in your "Audio Folder" a number will be appended to the new name.
► To create a new file with a custom name, enter the desired fie name into the Name edit field (shown in the screenshot below). The file will automatically be created if no file with the same name exists in the "Audio "Folder".

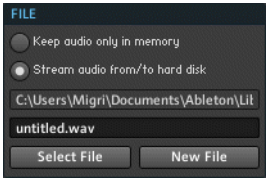


Fig. 14.2 The File area of the Function page lets you choose the operation Mode. Additionally you can load audio files for playback and create new audio files on the hard disk to which a signal may be recorded.

Internal Memory Mode

In Internal Memory Mode the Tapedeck Module records and plays back signals from your computer's random access memory (RAM).
► To activate Internal Memory Mode, engage the Keep audio only in memory checkbox (shown in the figure below).
► To load an audio file for playback, press the Select File... button (shown in the screenshot below). The length of the Module's memory buffer (see next point) is adjusted to match the loaded data.

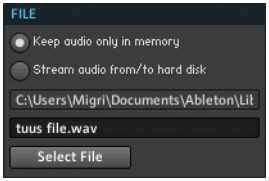


Fig. 14.3 The File area of the Function page allows you to activate the Internal Memory Mode. Additionally you can use the Select File button to load audio files into the Module's internal memory.

When in Internal Memory Mode, you need to set the size of the Module's internal memory buffer.

► To set the size of the Module's memory buffer, enter the corresponding maximum recording length into the Max Size (sex) edit field (shown in the screenshot below).

Changing the maximum recording length will erase any audio data stored in the memory.

► If you wish to reload an audio file that has been loaded with the Select File button (when it has been changed using a sample editor, for example), press the Reload button. The reload button is shown in the screenshot below.
► To save the internal state of the memory into a file with the name as set in the Name edit field (shown in the screenshot above), press the Save button (shown in the screenshot below.
► To save the internal state of the memory into a new file, press the Save as... button, shown in the screenshot below.

Fig. 14.4 The Recording area in the Function page becomes active when the Tapedeck Module is in Internal Memory Mode. You can set the memory buffer size with the Max Size (sec) edit field. Above this edit field appear three buttons in the following order: the Save button, the Save as... button, and the Reload button.

14.2 Tapedeck 2 Channel

Fig. 14.5 Tapedeck 2 Channel Module

14.2.1 Overview

The Tapedeck 2 Channel Module comprises a stereo tape deck for recording and playback of stereo Audio signals. Audio files can be read from and written to either the internal memory (Internal Memory Mode) or the hard disk (Hard Disk Mode), depending on the settings in the Function page.

When Hard Disk Mode is enabled, the files are written into the "Audio Files" folder you specify in the REAKTOR Preferences. Please refer to subsection 14.4.2 in the Application Reference to learn how to specify the "Audio Folder". REAKTOR does sample rate conversion on the fly and writes the file in the sample rate which is currently used by your audio system.

In Internal Memory Mode audio data is kept only in your computer's random access memory (RAM) and you have to set the maximum recording length in the Function page. This puts an upper limit on the amount of RAM that is consumed by the Tapedeck 2 Channel Module. How big this value can be depends on the amount of available RAM storage in the computer. At a sample rate of 44.1 kHz you need 86 kB of RAM for each second of buffer length; for one minute you need 5 MB.

If the buffer memory for the Tapedeck 2 Channel Module has been set to too large a value, virtual memory swaps performed by your computer's operating system can cause significant hard disk activity during recording. This can prevent REAKTOR from processing audio smoothly, leading to pops, crackles, and in worst cases to audio drop-outs.

When importing an audio file into the Tapedeck 2 Channel Module, the length of the Module's memory buffer is adjusted to match the loaded data. If you later want to make a recording which is longer than this, you will first need to go to the Function page and set a bigger value for the maximum recording length. When loading an audio file, REAKTOR will automatically convert its playback to the current REAKTOR Sample Rate.

Be aware that REAKTOR converts all audio files stored in Tapedeck Modules which are working in Internal Memory Mode to the new Sample Rate whenever REAKTOR switches to a new Sample Rate. So you should avoid changing the Sample Rate if your ensemble contains memory based audio samples. If the audio file is stored on the hard disk, you can use the Reload button after changing the Sample Rate to import the file again.

A recording can be exported using the Save button in the Function page. If the file has already been exported, you will be asked if you want to overwrite the file, e.g. if you have made a new recording in the Tapedeck 2 Channel Module. With the Save as... button you can store the file under a new name.

The Tapedeck 2 Channel Module's View page provides the same options as that of a Sampler Module. Please refer to subsection 10.3.2 in the Application Reference for a detailed overview of the Tapedeck 2 Channel Module's View page.

Application

Use the Tapedeck 2 Channel Module to quickly achieve audio file recording and playback functionality that you can control from the Instrument Panel and synchronize to other processes. You could, for example, create a sampler Instrument that sends the output of a Sampler Module through many effects which then end up being recorded by a Tapedeck Module. The point would be to record the Sampler's processed output, save it and then to load it again into the Sampler Module. Now you playback the processed Sample and add another iteration of effects (or mix it up with the original signal to create interesting characteristics stemming from phase effects). The Tapedeck Module is essential in such a case since you can easily synchronize the playback of the Sampler Module to the recording of the Tapedeck Module. The Tapedeck 2 Channel Module is meant for stereo signals. Use the Tapedeck 1 Channel Module (↑14.1, Tapedeck 1 Channel) for mono signals.

14.2.2 Ports

Input Ports

- **(Rec)** "Rec" (record) is the Event input port for switching recording on and off, with a Gate signal, for example. Recording is started upon the arrival of a positive valued Event. Recording is ended when an Event with a negative or zero value arrives or when the maximum recording time is reached.
- **(Play)** "Play" is the Event input port for switching playback on and off, with a Gate signal, for example. Playback is started upon the arrival of a positive valued Event. Playback is ended when an Event with a negative or zero value arrives or when the maximum recording time is reached.

- **(Lp)** "Lp" (loop playback) is the Event input port for switching loop playback on and off. When this input port receives a positive valued Event, loop playback is turned on. When during playback the end of the recording or file is reached, playback starts from the beginning again. The result is an infinite loop while playback is switched on. Send a zero or negative valued Event to turn looped playback off.
- **(In L)** "In L" (input left) is the Monophonic Audio input port for the left channel of the stereo signal to be recorded. The signal should be within the range [-1 … 1]. To record a Polyphonic signal, a Voice Combiner Module (↑14.3, Audio Voice Combiner) has to be inserted.
- **(In R)** "In R" (input right) is the Monophonic Audio input port for the right channel of the stereo signal to be recorded. The signal should be within the range [-1 … 1]. To record a Polyphonic signal, a Voice Combiner Module (↑14.3, Audio Voice Combiner) has to be inserted.
- **(Pse)** "Pse" (pause) is the Event input port for pausing the recording or playback. An Event with a value greater than zero pauses the tape deck, an Event with zero or negative values continues the playback.
- **(Pos)** "Pos" (position) is the Event input port for setting the playback position in milliseconds. The typical range of values at this input port is [0 … 20000]. When disconnected, the default value for this input port is "0" (start of the recording).
- **(Spd)** "Spd" (speed) is the Event input port for controlling the playback speed of the tape deck. Values at this input port must be greater than "0". A value of "1" corresponds to normal playback speed. The typical range of values at this inptu port is [0.8 … 1.2]. When this input port is disconnected, the default value that is used, is "1" (normal playback speed).

Output Ports

- **(L)** "L" (left) is the Audio output port for the left stereo channel of the playback signal or the signal being recorded.
- **(R)** "Out" (right) is the Audio output port for right stereo channel of the playback signal or the signal being recorded.
- **(Rec)** "Rec" (recording) is the Event output port for the recording state of the tape deck. An Event with the value Rec = 1 is sent when the tape deck starts recording, an Event with the value Rec = 0 is sent when recording is stopped.

- **(Play)** "Play" (playing) is the Event output port for the playback state of the tape deck. An Event with the value Play = 1 is sent when the tape deck starts playing, an Event with the value Play = 0 is sent when playback is stopped.
- **(Wrp)** "Wrp" (wrap) is the Event output port for counting the times that playback has looped. An Event with value "1" is sent each time the tape wraps around to the beginning when in looped playback.
- **(Pse)** "Pse" (pause) is the Event output port for the pause state of the tape deck. An Event with the value Pse = 1 is sent when recording or playback is paused, an Event with the value "0" is sent when recording or playback is continued.
- **(Pos)** "Pos" (position) is the Audio output port for the current playback/recording position in milliseconds [0 ... <length of sample in ms>].
- **(Len)** "Len" (length) is the Event output port for the length of the recording in milliseconds.

14.2.3 Properties: Function Page

Keeping the Tapedeck 2 Channel Module Always Active

The Tapedeck 2 Channel Module only records and plays back signals if it is part of an active signal flow.

► To keep the Tapedeck 2 Channel Module always active, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Hard Disk Mode

In Hard Disk Mode, audio files are written to and read from hard disk directly.

► To activate Hard Disk Mode, engage the Stream audio from/to hard disk checkbox (shown in the figure below).

► To load an audio file for playback, press the Select File... button (shown in the screenshot below).

► To create a new file named "untitled", press the New File button (shown in the screenshot below). If a file with the name already exists in your "Audio Folder" a number will be appended to the new name.

► To create a new file with a custom name, enter the desired fie name into the Name edit field (shown in the screenshot below). The file will automatically be created if no file with the same name exists in the "Audio "Folder".

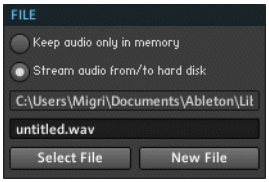


Fig. 14.6 The File area of the Function page lets you choose the operation Mode. Additionally you can load audio files for playback and create new audio files on the hard disk to which a signal may be recorded.

Internal Memory Mode

In Internal Memory Mode the Tapedeck Module records and plays back signals from your computer's random access memory (RAM).

► To activate Internal Memory Mode, engage the Keep audio only in memory checkbox (shown in the figure below).

► To load an audio file for playback, press the Select File... button (shown in the screenshot below). The length of the Module's memory buffer (see next point) is adjusted to match the loaded data.

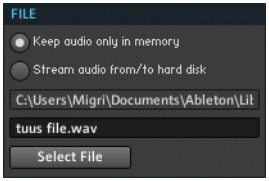


Fig. 14.7 The File area of the Function page allows you to activate the Internal Memory Mode. Additionally you can use the Select File button to load audio files into the Module's internal memory.

When in Internal Memory Mode, you need to set the size of the Module's internal memory buffer.

► To set the size of the Module's memory buffer, enter the corresponding maximum recording length into the Max Size (sex) edit field (shown in the screenshot below).

Changing the maximum recording length will erase any audio data stored in the memory.

► If you wish to reload an audio file that has been loaded with the Select File button (when it has been changed using a sample editor, for example), press the Reload button. The reload button is shown in the screenshot below.
► To save the internal state of the memory into a file with the name as set in the Name edit field (shown in the screenshot above), press the Save button (shown in the screenshot below.
► To save the internal state of the memory into a new file, press the Save as... button, shown in the screenshot below.

Fig. 14.8 The Recording area in the Function page becomes active when the Tapedeck Module is in Internal Memory Mode. You can set the memory buffer size with the Max Size (sec) edit field. Above this edit field appear three buttons in the following order: the Save button, the Save as... button, and the Reload button.

14.2.4 Example: Grain Resynth Resampler

The Structure in the figure below shows how to achieve synchronization between playback of a Resynth Module (↑7.4, Grain Resynth) and recording of a Tapedeck Module. Controls have been created for the most relevant parameters of the Resynth Module by right-clicking the corresponding input port and choosing the *Create Control* menu entry. The "Rec" Button Module is set to Toggle Mode and sends a Gate On and Gate Off signal to the "G", "Rec", and "Play" input ports. This way recording and sample playback are synchronized. Additionally, since a Gate On signal also arrives at the "Play" input port, what is being recorded by the Tapedeck can be heard at its "L" and "R" output ports. Note that a Merge Module (↑13.16, Merge) has been used at the "Play" input port such that the recorded signal can be played back independently from the playback of the Resynth Module.

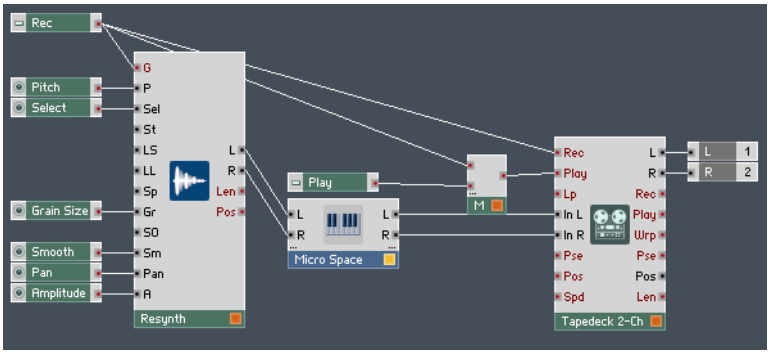


Fig. 14.9 The Structure for resampling the output of a Resynth Module with a Tapedeck 2 Channel Module.

14.3 Audio Voice Combiner

Fig. 14.10 Audio Voice Combiner Module

14.3.1 Overview

The Audio Voice Combiner Module converts a Polyphonic Audio signal to a Monophonic Audio signal by summing up all Voices.

Application

Use the Audio Voice Combiner Module to connect Polyphonic signals to Monophonic input ports, if the sum of all Voices still serves the purpose. Instruments can only output Monophonic signals, for example. This means that you need to insert an Audio Voice Combiner Module before an Instruments outputs if you wish to send them Polyphonic Audio signals.

14.3.2 Ports

Input Ports

- **(In)** "In" is the Polyphonic Audio input port for the signal whose Voices are to be combined to make up the outgoing Monophonic signal.

Output Ports

- **(Out)** "Out" is the Monophonic Audio output port for the Voice combined signal.

14.3.3 Example: New Instrument Outputs

Instruments can only output Monophonic signals, even if the internal Instrument Structure is Polyphonic. For this reason, when you create a new Instrument, the Audio Voice Combiner Modules are already placed in front of the outputs so that outgoing Polyphonic signals are automatically converted to Monophonic signals. This is shown in the figure below.

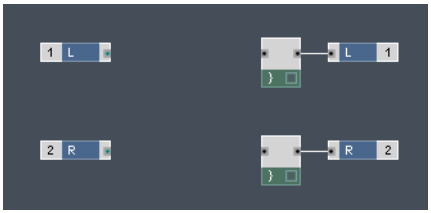


Fig. 14.11 The Structure of a new Instrument.

14.4 Event V.C. All

Fig. 14.12 Event Voice Combiner All Module

14.4.1 Overview

The Event Voice Combiner All Module converts a Polyphonic Event signal to a Monophonic Event signal by summing up all Voices.

Application

Use the Event Voice Combiner All Module to connect Polyphonic signals to Monophonic input ports where you wish to use the total sum of all Voices. Instruments can only output Monophonic signals, for example. This means that if you internally process Polyphonic Event signals and wish your Instrument to output the resulting Event signal, you need to insert an Event Voice Combiner All Module before the Output Modules.

14.4.2 Ports

Input Ports

- **(In)** "In" is the Polyphonic Event input port for the signal whose Voices are to be combined to make up the outgoing Monophonic signal.

Output Ports

- **(Out)** "Out" is the Monophonic Event output port for the Voice combined signal.

14.4.3 Example: Instrument Event Outputs

Instruments can only output Monophonic signals, even if the internal Instrument Structure is Polyphonic. For this reason, when you create an Instrument that outputs Event signals (such as a MIDI effect that creates an arpeggiating pitch signal), you need to place an Event Voice Combiner All Module in front of the output, as shown in the figure below. Note that this MIDI effect can then only control the pitch of a synthesizer one note at a time.

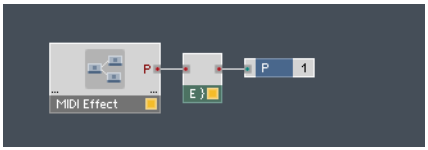


Fig. 14.13 The Event Voice Combiner All Module turns Polyphonic Event signals into Monophonic Event signals before outputting them from the Instrument.

14.5 Event V.C. Max

Fig. 14.14 Event Voice Combiner Max Module

14.5.1 Overview

The Event Voice Combiner Max Module converts a Polyphonic Event signal to a Monophonic Event signal by forwarding the Voice with the greatest value. Connect the Gate Module's (↑2.3, Gate) Polyphonic output signal to the "G" (gate) input port to tell the Event Voice Combiner Max Module which Voices are active.

Application

Use this Module to implement the legato play feature in a synthesizer. As long as at least one is pressed on the keyboard, any new played notes will not retrigger the synthesizer envelopes. If want velocity dependent parameters in the synthesizer to be set according to the note that has been played with the highest velocity, you route the output signal from the Gate Module (↑2.3, Gate) to both the "In" input port (where the velocity values to be selected from arrive) and also to the "G" input port. Note that you need to use an Order Module (↑13.12, Order) to define the Event Order such that the Events arrive first at the "In" input port and then at the "G" input port.

14.5.2 Ports

Input Ports

- **(In)** "In" is the Polyphonic Event input port for the signal whose maximum value is to be output.
- **(G)** "G" (gate) is the Polyphonic Event input port for the Gate signal (from the Gate Module, see also ↑2.3, Gate) of the Polyphonic Instrument.

Output Ports

- **(Out)** "Out" is the Monophonic Event output port for the greatest value carried by the incoming Voices at the "In" input port.

14.5.3 Example: Select Highest Velocity

Use the Event Voice Combiner Max Module to select the highest velocity value from a Polyphonic Gate signal. This is shown in the Structure below where the source of the Gate signal is the Gate Module (↑2.3, Gate). Note that the Order Module (↑13.12, Order) has been used to first forward the velocity information to the "In" input port and only then to the "G" (gate) input port which triggers the selection process for the active Voices.

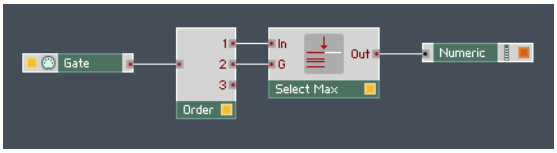


Fig. 14.15 The Event Voice Combiner Max Module can be used to select the highest velocity carried by the Polyphonic Voices from the Gate Module.

14.6 Event V.C. Min

Fig. 14.16 Event Voice Combiner Min Module

14.6.1 Overview

The Event Voice Combiner Min Module converts a Polyphonic Event signal to a Monophonic Event signal by forwarding the Voice with the smallest value. Connect the Gate Module's (↑2.3, Gate) Polyphonic output signal to the "G" (gate) input port to tell the Event Voice Combiner Min Module which Voices are active.

Application

Use this Module to implement the legato play feature in a synthesizer. As long as at least one is pressed on the keyboard, any new played notes will not retrigger the synthesizer envelopes. If want velocity dependent parameters in the synthesizer to be set according to the note that has been played with the lowest velocity, you route the output signal from the Gate Module (↑2.3, Gate) to both the "In" input port (where the velocity values to be selected from arrive) and also to the "G" input port. Note that you need to use an Order Module (↑13.12, Order) to define the Event Order such that the Events arrive first at the "In" input port and then at the "G" input port.

14.6.2 Ports

Input Ports

- **(In)** "In" is the Polyphonic Event input port for the signal whose minimum value is to be output.

- **(G)** "G" (gate) is the Polyphonic Event input port for the Gate signal (from the Gate Module, see also ↑2.3, Gate) of the Polyphonic Instrument.

Output Ports

- **(Out)** "Out" is the Monophonic Event output port for the smallest value carried by the incoming Voices at the "In" input port.

14.6.3 Example: Select Lowest Velocity

Use the Event Voice Combiner Min Module to select the lowest velocity value from a Polyphonic Gate signal. This is shown in the Structure below where the source of the Gate signal is the Gate Module (↑2.3, Gate). Note that the Order Module (↑13.12, Order) has been used to first forward the velocity information to the "In" input port and only then to the "G" (gate) input port which triggers the selection process for the active Voices.

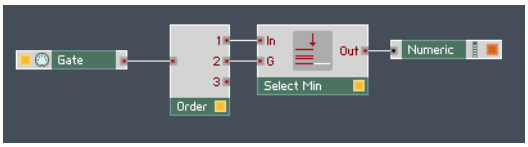


Fig. 14.17 The Event Voice Combiner Min Module can be used to select the lowest velocity carried by the Polyphonic Voices from the Gate Module.

14.7 A to E

Fig. 14.18 A to E Module

14.7.1 Overview

The A to E Module converts Audio signal into Event signals. The audio signal is sampled using the Control Rate specified in the Settings menu and the values are sent out as a stream of Events. Not only does it down-sample the signal from the Sample Rate to Control Rate, but also enables the signal to be processed by Event Processing Modules and even trigger other processes.

Application

Use the A to E Module when you wish to use Audio signals as control signals that do not need to be processed at the Audio Rate. Processing at the Control Rate saves CPU. For example, you can convert the output of an Envelope Module to an Event signal that can be used for further processing or modulating parameters at the Control Rate.

14.7.2 Ports

Input Ports

- **(In)** "In" is the Audio input port for the signal to be converted into a stream of Events.

Output Ports

- **(Out)** "Out" is the Event output port for the stream of Events resulting from the converted Audio input signal.

14.7.3 Example: Modulation Envelope

A good application for the A to E Module arises when you wish to control a parameter that has an Event input port (such as the cutoff pitch, shown in the figure below) with an envelope, which is inherently an Audio signal. In such a case just run the output of the Envelope Module (Audio signal) through the A to E Module to get an Event signal which you can then feed to the Event input port.

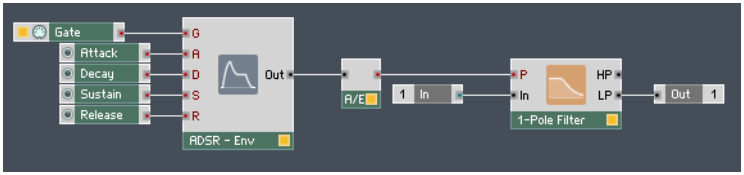


Fig. 14.19 Use the A to E Module to convert an Audio signal to an Event signal.

14.8 A to E (Trig)

Fig. 14.20 A to E Trig Module

14.8.1 Overview

The A to E Trig Module converts an Audio signal to Events, but only when a trigger signal arrives at the "Trig" (trigger) input port. When the trigger input signal goes from negative value to positive values (positive zero crossing) the Audio signal at the "In" input port is sampled and output as an Event.

Application

Use the A to E Trig Module when you wish to use Audio signals as Event signals where Events are sent at arbitrary intervals. This is the advantage of the A to E Trig Module that you can choose when and at which rate the conversion takes places. Since Events can trigger a lot of CPU-intensive signal processing, it sometimes efficient to control the rate of the converted Event stream.

14.8.2 Ports

Input Ports

- **(Trig)** "Trig" (trigger) is the Audio input port for triggering the Audio to Event conversion. The Audio signal is sampled when the input signal goes from a negative value to a positive value (positive zero crossing). Route the output of the Ramp Module (↑6.36, Ramp Oscillator) to this input to be able to exactly control the frequency of the conversion. Subtract a small value from the ramp signal to create a positive zero crossing in its waveform.
- **(A)** "A" (Audio) is the Audio input port for the signal to be sampled and converted to Events.

Output

- **(E)** "E" (Event) is the Event output port for the Events resulting from the sampled Audio input signal.

14.8.3 Example 1: Event Sample and Hold

Use the A to E Trig Module if you wish to sample an Audio signal upon an incoming trigger signal and send this value as an Event. The Structure below illustrates one example how this might be realized. The Differentiator Module (↑10.22, Differentiator) has a positive zero crossing at its output port when the slope of the incoming signal goes from falling to

rising. Such a positive zero crossing is interpreted as a trigger signal at the A to E Trig Module's "Trig" input port and upon such an event the signal at the "A" input port is sampled. Here the same signal is used for the trigger source (ultimately at the Differentiator Module's (↑10.22, Differentiator) input port) as well as for the sample and hold source (at the "A" input port).

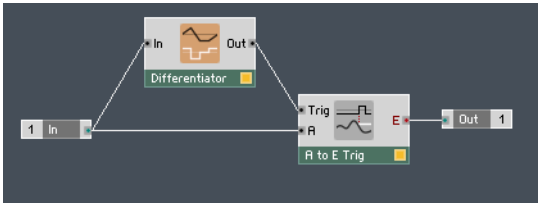


Fig. 14.21 A simple sample and hold mechanism that samples an Audio signal and outputs an Event signal.

14.8.4 Example 2: Simple Scope

This example shows how to display a periodic Monophonic Audio signal on a scope. The incoming signal should have periodic (regular) zero crossings in order for the A to E Trig Module to create a trigger signal from it. Of course, you could use other mechanisms for triggering the scope trace. The Structure below shows the essential components for the scope. The A to E Trig Module creates a trigger signal from a positive zero crossing of the incoming signal. Upon sending a trigger signal to the Scope Module (↑1.15, Scope), an interval of the signal incoming at the Scope Module's "In" input port is buffered that corresponds to the time interval set in the Module's Function page with the Buffer edit field. A knob with the range [0 … 1] and label "X" sends the scaling of the time axis of the scope to the "TS" input port. The scaling for the "Y" (signal level) axis is set in the logarithmic scale with the knob labeled "Y" (range [-32 … 32]), converted to linear scale with the Exp Lvl-to-A Module (↑4.15, Exp (Lvl-to-A) Module), and then sent to the "YS" input port.

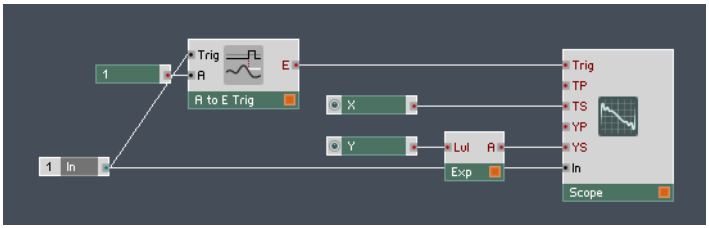


Fig. 14.22 The Structure for a simple scope that displays a Monophonic periodic Audio signal with regular zero crossings.

14.9 A to E (Perm)

Fig. 14.23 A to E Perm Module

14.9.1 Overview

The A to E Perm Module permanently samples and converts an Audio signal into Events at regular intervals and with a sampling frequency set at the "F" (frequency) input port. The Audio signal is sampled with the given frequency and output as a stream of Events. When running this module at a high frequency (above 1000 Hz) the CPU-Load caused by Event processing can rise significantly. Typically F = 200 Hz is sufficient.

Application

Use the A to E Perm Module when you wish to use Audio signals as control signals with an arbitrary sampling rate. Events can trigger a lot of CPU-intensive signal processing. The A to E Perm Module allows you to control the rate of the Events at the output, thus giving you the option of reducing CPU load when necessary.

14.9.2 Ports

Input Ports

- **(F)** "F" (frequency) is the Audio input port for controlling the frequency at which the Audio signal is sampled. Thus it also controls the rate of Events from the "E" (Event) output port. Values at this input port should be greater than zero and specified in units Hz (Events per second).
- **(A)** "A" (Audio) is the input port for the Audio signal to be sampled and converted to Events.

Output Ports

- **(E)** "E" (Event) is the Event output port for the Events resulting from the sampled Audio input signal.

14.9.3 Example: Modulation Envelope

A good application for the A to E Perm Module arises when you wish to control a parameter that has an Event input port (such as the cutoff pitch, shown in the figure below) with an envelope, which is inherently an Audio signal. The advantage of the A to E Perm Module over the A to E Module (↑14.7, A to E) is that you can choose at what rate the Audio signal is sampled. You can either increase or decrease the frequency as you wish (even from the Instrument Panel), independent of the Control Rate. In such a case just run the output of the Envelope Module (Audio signal) through the A to E Perm Module to get an Event signal which you can then feed to the Event input port. The value at the "F" (frequency) input port determines the rate of Events at the A to E Perm Module's output port.

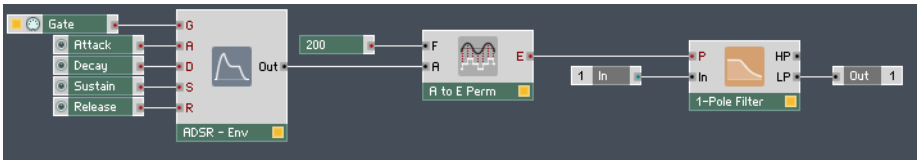


Fig. 14.24 Use the A to E Perm Module to convert an Audio signal to an Event signal with an arbitrary Event rate.

14.10 A to Gate

Fig. 14.25 A to Gate Module

14.10.1 Overview

The A to Gate Module samples and converts an Audio signal to an Event Gate signal. When the trigger signal at the "In" input port goes from negative to positive values (positive zero crossing), the Gate signal is switched on with an amplitude of the current value of the "A" (amplitude) input port. When the trigger input returns to negative values (negative zero crossing) the Gate signal is switched off.

Application

Use the A to Gate Module when you wish to create an Event Gate signal by sampling an Audio signal. The difference between the A to Gate Module and the A to E Trig Module (↑14.8, A to E (Trig)) is that although both sample the Audio input signal upon a trigger, the output of the A to Gate Module always returns to zero before the next time that the input is sampled. Use this Module to create a Gate signal that has its velocities derived from the amplitude of an incoming Audio signal, for example.

14.10.2 Ports

Input Ports

- **(In)** "In" is the Audio input port for triggering the Gate signal. The Audio signal is sampled when the input signal goes from a negative value to a positive value (positive zero crossing). Upon a negative zero crossing an Event with the value zero is sent from the output. Route the output of the Ramp Module (↑6.36, Ramp Oscillator) to this input to be able to exactly control the frequency of the conversion. Subtract a small value from the ramp signal to create a positive zero crossing in its waveform.
- **(A)** "A" (amplitude) is the Audio input port for controlling the amplitude of the Gate Events at the output port.

Output Ports

- **(G)** "G" is the Event output port for the Event signal.

14.10.3 Example: Single Trigger Gate

For the legato playing style there is only the Gate On signal for the first MIDI Note played. As long as at least one key is pressed, all subsequent Gate signals are suppressed for the synthesizer. The Structure below implements this feature by first combining the Gate signals from all Voices with the Voice Combiner and then using this signal to trigger the A to Gate Module. Since only for the first key pressed the signal goes from zero to a nonzero value, the signal at the "A" input port is sampled only upon such an Event. Note that for the Structure below the velocity information is discarded and the outgoing Event gate signal only switches between zero and "1".

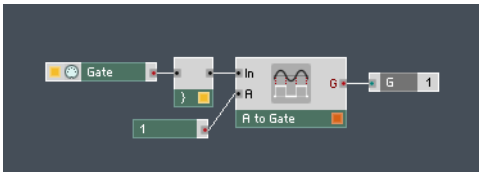


Fig. 14.26 The Structure for a custom single trigger Gate signal used in the legato playing style.

14.11 To Voice

Fig. 14.27 To Voice Module

14.11.1 Overview

The To Voice Module takes Monophonic input signals and transmits them to one Polyphonic Voice of the output signal. The Voice number is set at the "V" (Voice) input port. Signals are discarded when the value at the "V" input port is not a valid Voice number.

Application

The To Voice Module allows you to compose the makeup of Polyphonic signals, Voice by Voice. To set each Voice to a certain value, just use a To Voice Module for each Voice and subsequently use the Add Module (↑4.2, Add, +) to sum up the outputs of all the To Voice Modules. The output signal from the Add Module is then the Polyphonic signal that has the desired makeup. For an additional example, please refer to the Voice Info Module's (↑14.18, Voice Info) entry which shows how Polyphony can be used for parallel signal processing. This is also useful for making Polyphonic sequencers, for example.

14.11.2 Ports

Input Ports

- **(V)** "V" (Voice) is the Monophonic Audio input port for defining the number of the Polyphonic Voice to which the value at the lower input is to be applied. The range of values at this input port should be in the range [1 ... <number of Polyphonic Voices set in the parent Instrument>].

- **(In)** "In" is the Monophonic hybrid input port for signals to be sent to one of the Polyphonic Voices.

Output Ports

- **(Out)** "Out" is the Polyphonic hybrid output port for the Polyphonic signal which has the value at the "In" input port carried by the Voice set by the "V" (Voice) input port. All other Voices carry the value zero.

14.11.3 Example 1: Note to Chord

The To Voice Modules give you direct access to the values carried by each Voice. A handy application is to generate chords from incoming Monophonic MIDI Notes. The Structure that implements this feature is shown in the figure below for an Instrument that has its number of Polyphonic Voices set to "3". The chord that is played when a key is pressed is the minor chord: the incoming MIDI Note is assigned to the first Voice, the minor third to the second Voice, and the perfect fifth to the third Voice. As can be seen in the Structure below, Add Modules (↑4.2, Add, +) and the consequent To Voice Modules are used to assign each Voice the corresponding transposed MIDI Note. The output of each To Voice Module carries a Polyphonic signal where each Voice, except the one specified by the "V" input port, is zero. The outputs of the To Voice Modules are added together to result in a Polyphonic signal where each Voice has a defined value.
To get the base note of the chord, the "Nr" output port of the Channel Message Module (↑2.17, Channel Message) is used. Although this output can send values other than MIDI Notes, triggering the value to be used only when a Gate On Event from the Single Trig Gate Module (↑2.4, Single Trig Gate) arrives, ensures that only MIDI Note values are used. The Separator Module (↑13.14, Separator), on the other hand, ensures that only Gate On Events trigger the note number to be forwarded to the chord generation algorithm.

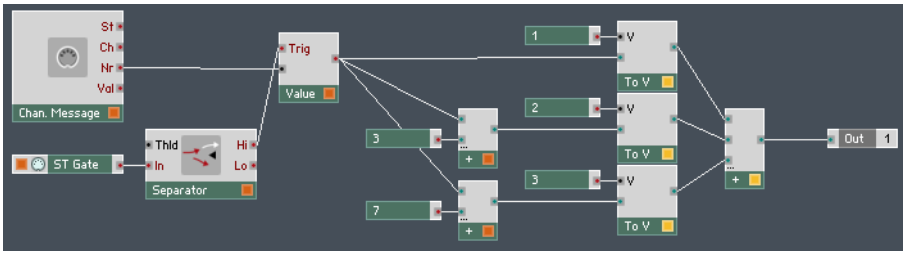


Fig. 14.28 The Structure to create a chord from an incoming Monophonic MIDI Note.

14.11.4 Example 2: Displaying Pitch Values

Since the Poly Display Module (↑1.17, Poly Display) gives you direct access to the values carried by Polyphonic Voices, it is ideal for tasks like displaying the pitch values carried by the Voices, for example. This example shows how to make a simple display that shows the pitch values carried by four Voices.

First, let's decide that we want to have a bar graph of the voice values, as can be seen in the last figure at the end of this example. For this, connect the constant value "-1" to the Poly Display Module's (↑1.17, Poly Display) "Obj" input port. We want each bar to have unit width, start at Y1 = 0 and reach up to the value of the corresponding MIDI Note. Since Y1 = 0 for all Voices, just connect the Constant (↑4.1, Constant) "0" to the "Y1" input port. Also, the Note Pitch Module (↑2.1, Note Pitch) forwards the right MIDI Note values already carried by each Voice, so directly connect the Note Pitch Module to the "Y2" input port.

Next, let's send the "X1" and "X2" values. For the four Voices in the Instrument below, the pairs (X1, X2) should be the following (remember we wanted a bar width of "1"): (0, 0 + 1), (1, 1 + 1), (2, 2 + 1), and (3, 3 + 1). In essence, we just need one signal which has the values {0, 1, 2, 3} for its four Voices. This would be the signal for the "X1" input port. Then you just add the Constant (↑4.1, Constant) "1" and forward the result to the "X2" input port. The Structure that achieves this is shown in the figure below.

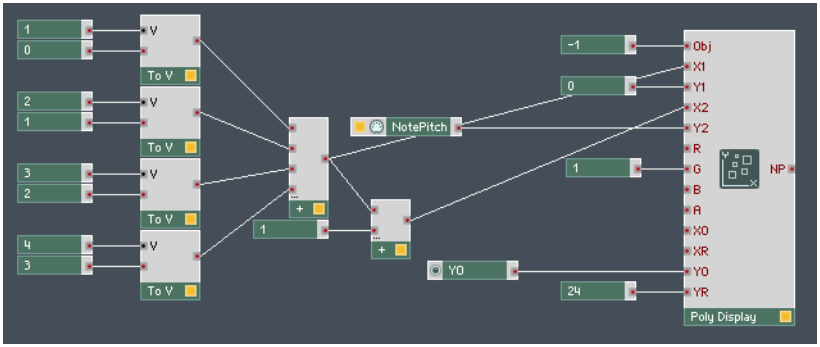


Fig. 14.29 The Structure for a simple Poly Display Structure that displays played MIDI Notes.

To set the values for each Voice separately, use To Voice Modules (although in this case you could achieve the same "X1" values using the Voice Info Module (↑14.18, Voice Info)). Place four To Voice Modules into the Structure, one for each Voice. Feed the values "1" to

"4" to the "V" (Voice) input ports to designate the Voice which the To Voice Module addresses. The lower input port of each To Voice Module is meant for the value to which the Voice is set. Set the values as shown in the Structure above for each Voice: "0", "1", "2", and "3". Each To Voice Module outputs a Polyphonic signal where each Voice is zero except the Voice which it addresses. Use the Add Module (↑4.2, Add, +) to sum up the Polyphonic signals from the To Voice Modules, Voice by Voice. This yields the desired signal for the "X1" input port. To get the signal for the "X2" input port, just add "1" and forward that to said port.
Next, to set all bars to have bright green coloring, forward the constant "1" to the "G" input port and make sure that the Ignore RGB checkbox in the Poly Display Module's (↑1.17, Poly Display) Function page is engaged. One more thing: you need to make sure that the range of displayed values matches the incoming values. The X Origin and X Range values (set in the Poly Display Module's Function page) are "0" and "4", respectively (since we are using four Voices). For the Y Origin and Y Range values, it depends on which keys you are pressing and wish to be displayed. Let's choose the range of displayed MIDI Note values to be two octaves. This corresponds to a MIDI Note interval of "24", so connect a Constant (↑4.1, Constant) carrying this value to the "YR" input port. You might want some flexibility for the starting point of the displayed range of notes, so just create a Knob Module with the range [0 ... 127] and Stepsize "1" at the "YO" input port. The resulting Instrument panel of this example is depicted in the figure below.

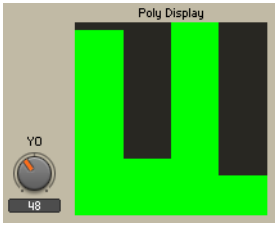


Fig. 14.30 The Poly Display displays played MIDI Notes starting with MIDI Note "48" and ending with MIDI Note "72".

14.12 From Voice

Fig. 14.31 From Voice Module

14.12.1 Overview

The From Voice Module takes the signal from one Voice of the Polyphonic input signal and forwards it as a Monophonic signal to the output port. Signals from all other Voices are discarded. The Voice number of the Voice whose signal is forwarded, is set at the "V" (Voice) input port. When the value at the "V" input port is not a valid Voice number, all incoming signals are blocked.

14.12.2 Ports

Input Ports

- **(V)** "V" (Voice) is the Monophonic Audio input port for defining the number of the Polyphonic Voice at the lower input that is to be forwarded to the output. The range of values at this input port should be in the range [1 ... <number of Polyphonic Voices set in the parent Instrument>].
- **(In)** "In" is the hybrid input port for the Polyphonic signal from which a Voice is to be selected and sent to the output.

Output Ports

- **(Out)** "Out" is the hybrid output port for the Monophonic signal carried by the selected Polyphonic Voice at the "In" input port.

14.12.3 Example: Voice Spread

For unison spread effects, the Randomize Module (↑13.3, Randomize) gives a random value offset for each incoming Voice. This example, shown in the figure below, demonstrates this effect by making the offset for each Voice visible from the Instrument Panel using a From Voice Module and a Numeric Display Module (↑1.8, Meter). Use a Button Module (↑1.2, Button) for the incoming signal. In its Function page, set it to Trigger Mode and both the "Min" and "Max" values to "0", using the corresponding edit fields. This way every time you press the button on the Instrument Panel, an Event with the value "0" (at each Voice) is sent to the "In" input port of the Randomize Module. Now create a control for the "Rng" (range) input port by right-clicking the input port and choosing the *Create Control* menu entry.

In the example, the number of Polyphonic Voices of the parent Instrument has been set to "3" or higher. To extract the value carried by a Polyphonic Voice, use the From Voice Module. Insert three From Voice Modules and connect the lower input of each to the Randomize Module's output. Connect Constants with the values "1", "2", "3", each to one From Voice Module's "V" (Voice) input port. As shown in the Structure below, the first From Voice Module outputs the value (including random offset) carried by the first Voice, the second From Voice Module outputs the value carried by the second Voice, and so on. Insert three Numeric Display Modules (↑1.8, Meter) into the Structure to display these values on the Instrument Panel. Remember to engage the Always Active checkbox for at least one of the Numeric Display Modules to activate the Structure. The second figure below shows the Instrument Panel where the button has sent a value "0" and the "Rng" value is "0.52". Sending another zero valued Event to the "In" input port creates a new random offset for each Voice.

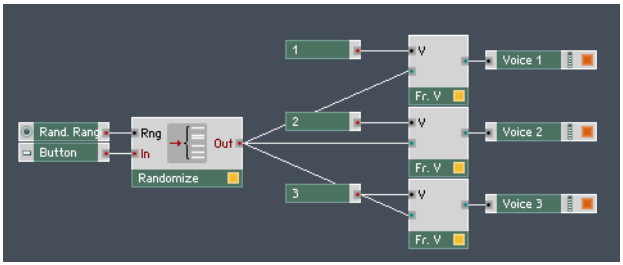


Fig. 14.32 The Structure to test the Voice spreading function of the Randomize Module.

Fig. 14.33 With the From Voice Modules you can see how the Randomize Module spreads out the values for the different Voices.

14.13 Voice Shift

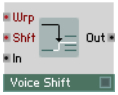


Fig. 14.34 Voice Shift Module

14.13.1 Overview

The Voice Shift Module can rearrange the signals carried by Polyphonic Voices incoming at the "In" input port, so that the signal at each Voice is remapped to an arbitrary Voice Number. For example, you could use the Voice Shift Module to remap the signals carried by Voices 1, 2, 3, 4 to the outgoing Voices 2, 3, 4, 1, or to 3, 4, 2, 1, and so on. If multiple incoming Voices are shifted to the same outgoing Voice, they are summed. For example, if incoming Voices 1 and 2 are both shifted to the outgoing Voice 3, Voice 3 will consist of: Voice 1 signal + Voice 2 signal.

Application

When working with sequencers that make use of Polyphony to process different sequencer tracks in parallel (triggering different samples, for example), you can use the Voice Shift Module to quickly reroute the destination of such a sequencer track by assigning it to another Voice. In such a situation you could even sum two or more sequencer tracks together to create a new sequence.

14.13.2 Ports

Input Ports

- **(Wrp)** "Wrp" (warp) is the Monophonic Event input port for turning wrapping of the shifted Voices on and off. When Wrp ≤ 0, wrapping is turned off and Voices can be shifted to invalid Voice numbers (i.e. less than 1 or greater than the number of Polyphonic Voices of the parent Instrument) and are therefore discarded. When Wrp > 0, wrapping is turned on and Voices shifted to invalid Voice numbers are wrapped around to valid Voice numbers. For example, +1 shifting in a 3-voice Instrument will cause Voice 3 to be remapped to Voice 1, and so on. The default value at this input port is "0", which corresponds to wrapping turned off.
- **(Shft)** "Shft" (shift) is the Polyphonic Event input port for the Voice shift parameter. Positive "Shft" values shift voices up (e.g. Shft = 1 would shift the incoming Voice 1 to outgoing Voice 2), and negative "Shft" values shift Voices down (Shft = -2 would shift the incoming Voice 3 to outgoing Voice 1). Since "Shft" is a Polyphonic input port, each Voice can be shifted by its own amount. The default for the "Shft" value is "1".
- **(In)** "In" is the input port for the Polyphonic signal to be Voice-shifted.

Output Ports

- **(Out)** "Out" is the output port for the Polyphonic Voice-shifted signal.

14.13.3 Example: Voice Shift of Voice Info Values

In the example shown by the Structure in the figure below, the incoming values for the Voice Shift Module are "1", "2", "3", and "4" for the first, second, third, and fourth Voice, respectively. Since the value "2" is at the "Shft" (shift) input port, these values are shifted up by two Voices. Warping has been engaged by supplying the value "1" to the "Wrp" (warp) input port. The result, as can be seen in the figure, comprises the values "3", "4", "1", and "2" for the Voices "1", "2", "3", and "4", respectively.

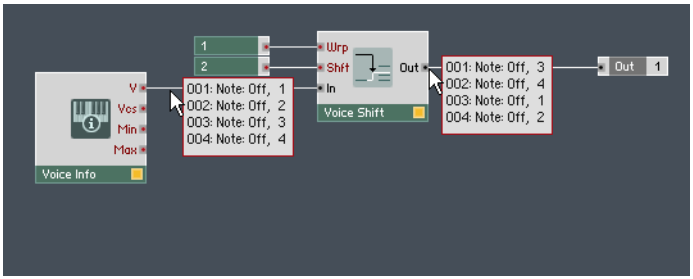


Fig. 14.35 An example of the effect of the Voice Shift Module using the "V" output of the Voice Info Module .

14.14 Audio Smoother

Fig. 14.36 Audio Smoother Module

14.14.1 Overview

The Audio Smoother Module smoothes Monophonic Event signals and converts them to Audio signals. Jumps in the value of the input Event signal are smoothed to ramps. The transition time (in milliseconds) can be adjusted in the Module's Function page. Exactly after the transition time has elapsed, the output signal reaches the same value as the input, assuming that no further jumps occurred at the input. The larger the transition time, the stronger the smoothing effect.

Application

Typically, a smoother is connected after a fader or button to get smooth transitions. Use the Audio Smoother Module to convert Event signals into Audio signals.

14.14.2 Ports

Input Ports

- **(In)** "In" is the Monophonic Event input port for the signal to be smoothed.

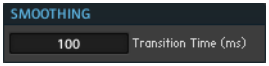
Output Ports

- **(Out)** "Out" is the Monophonic Audio output port for the smoothed signal.

14.14.3 Properties: Function Page

Setting the Transition Time

► To set the transition time of the outgoing smoothed signal from one input value to the next, use the Transition Time (ms) edit field (shown below).



14.14.4 Example: Comb Filter

Use the Audio Smoother Module to convert Event signals to Audio signals as is the case when feeding the delay time to the "Dly" (delay time) Audio input port of a Single Delay Module (↑11.1, Single Delay). This is the case in the comb filter Structure, shown below. A comb filter has a frequency response that looks similar to a comb. This means that certain frequencies at regular intervals are amplified, whereas other frequencies are attenuated. The amplification peaks are multiples of the "base frequency" of the comb filter, determined by the delay time set at the Single Delay Module's (↑11.1, Single Delay) "Dly" (delay time) input port.

The Structure of a comb filter is simple. As can be seen in the figure below, the incoming "dry" signal is combined with the feedback signal using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) and fed into the Single Delay Module (↑11.1, Single Delay). With the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+) you multiply the feedback signal with a value in the range [0 … 1] (received from the "Feedback" knob) and so determine the amount of feedback signal that is sent to the Single Delay Module. Since feedback Struc-

tures can cause the signal to "explode", a Saturator Module (↑12.1, Saturator) has been placed between the Single Delay Module's output and the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+). This way the feedback signal never exceeds the bounds [-2 … 2], which is reached for input values outside the range [-4 … 4].
The "base frequency" is calculated from a pitch value which is received from the knob labeled "pitch" (range [1 … 128]) and converted to the linear frequency scale using the Exp P-to-F Module (↑4.16, Exp (P-to-F)). The output of the Exp P-to-F Module delivers the pitch in Hertz (oscillations per second) but the we need the corresponding delay time in milliseconds. Therefore we use the Divide Module to divide 1000 by the pitch value in Hertz, giving us an Event signal that gives the delay time in milliseconds. Subsequently we convert this Event signal to an Audio Signal using the Audio Smoother Module.
Last, the "wet" signal, derived from the Single Delay Module's (↑11.1, Single Delay) output, is added to the "dry" incoming signal. This is done again using the Mult/Add Module (↑4.6, Mult/Add (a * b + c), X+), where the amount of "wet" signal mixed to the "dry" signal can be controlled with the "Mix" knob (range [0 … 1]).

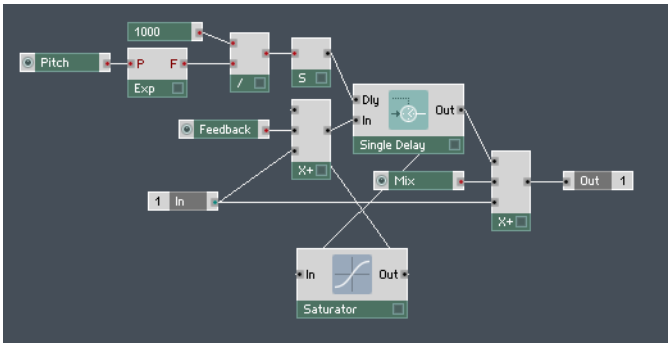


Fig. 14.37 The Structure of a comb filter.

14.15 Event Smoother

Fig. 14.38 Event Smoother Module

14.15.1 Overview

The Event Smoother Module smoothes Monophonic Audio signals and converts them to Event signals. Jumps in the value of the input Audio signal are smoothed to ramps. The transition time (in milliseconds) can be adjusted in the Module's Function page. Exactly after the transition time has elapsed, the output signal reaches the same value as the input, assuming that no further jumps occurred at the input. The larger the transition time, the stronger the smoothing effect. During the transition ramp, Events are output with the Control Rate. The higher the rate, the higher the resolution of the smoothing transition.

Application

Typically, a smoother is connected after a control signal source such as a knob or fader. In the case of the Event smoother, you might want to smooth envelope signals and have them delivered as Events. For more possibilities regarding Audio to Event conversion, please refer to the entries for the A to E (↑14.7, A to E), A to E Trig (↑14.8, A to E (Trig)), A to E Perm (↑14.9, A to E (Perm)), and A to E Gate Modules.

14.15.2 Ports

Input Ports

- **(In)** "In" is the Monophonic Audio input port for the signal to be smoothed.

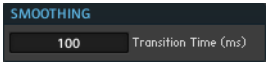
Output Ports

- **(Out)** "Out" is the Monophonic Event output port for the smoothed signal.

14.15.3 Properties: Function Page

Setting the Transition Time

► To set the transition time of the outgoing smoothed signal from one input value to the next, use the Transition Time (ms) edit field (shown below).



14.15.4 Example: Smoothing an Envelope Signal

In addition to the A to E Modules (↑14.7, A to E), you can also convert an Audio signal to an Event signal using the Event smoother. The figure below shows how to convert the output of an Envelope Module, which is an Audio signal, to an Event signal which you can then use to control parameters via Event input ports (such as the cutoff pitch parameter of a Filter Module, for example). Remember that you can set the transition time for the smoothed ramps in the Function page of the Event smoother Module.

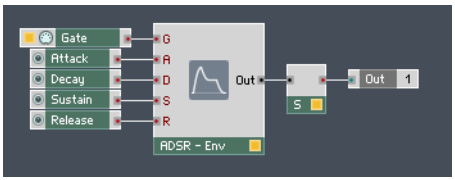


Fig. 14.39 Use the Event smoother Module to convert the Audio signal from the ADSR Module to a smoothed Event signal running at the Control Rate.

14.16 Master Tune/Level

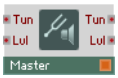


Fig. 14.40 Master Tune/Level Module

14.16.1 Overview

The Master Tune/Level Module allows you to control the master output signal level and tuning of your Ensemble. It also outputs the corresponding tuning and level information at its output ports. Changes in the master level are reflected by the position of the Audio Level Meter's fader in the Main Toolbar, shown in the first figure below, and vice versa. Changes in the master tuning parameter are added to the current tuning value of all Instruments contained in the Ensemble such that relative tuning of the Instruments remains intact. The Tune edit field in the Function page of these Instruments reflects the changes in the master tuning parameter (shown in the second figure below). However, the value at the "Tun" (tune) output port only reflects the global master tuning parameter (and not the resulting tuning of the parent Instrument).

Fig. 14.41 The fader below the Audio Level Meter reflects the master output level settings of your Ensemble. Changes made with the fader are also reflected by the Master Tune/Level Module.

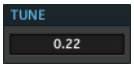


Fig. 14.42 The Tune edit field that lies in the Function page of all Instruments contained in the Ensemble reflects changes in the master tuning parameter.

Application

When creating a new Ensemble, the "Master" Instrument is automatically inserted. This Instrument contains the Master Tune/Level Module with faders connected to its input ports. This way you can easily access the overall tuning and master output signal level from the Instrument Panel.

14.16.2 Ports

Input Ports

- **(Tun)** "Tun" (tune) is the Event input port for controlling the master tuning parameter. Values at this input are given in the logarithmic pitch scale where 1 unit corresponds to 1 semitone. At Tun = 0.0 the MIDI Note A3 is tuned to 440 Hz. The typical range of values at this input port is [-1 ... 1] for fine tuning within a two semitone range. When disconnected, this input port uses the default value "0".
- **(Lvl)** "Lvl" (level) is the Event input port for controlling the master output level of your Ensemble. The values at this input port are given in the logarithmic decibel scale where 1 unit corresponds to 1 dB. For Lvl = 0.0 you have unity gain at 0.0 dB. The typical range of values at this input port is [-60 ... 0]. When disconnected, this input port uses the default value "0".

Output Ports

- **(Tun)** "Tun" (tune) is the Event output port for the master tuning parameter, given in the logarithmic pitch scale.
- **(Lvl)** "Lvl" (level) is the Event output port for the master output level of your Ensemble, given in the logarithmic decibel scale

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

14.16.3 Properties: Function Page

Keeping the Master Tune/Level Module Always Active

If the Master Tune/Level Module is not connected to an active signal flow, then the incoming values at its input ports will not have any effect on the tuning and the master level. Connecting it directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Master Tune/Level Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

14.16.4 Example: Master Instrument

When you create a new Ensemble, it will have an Instrument labeled "Master" in it. The Structure of this Instrument is shown in the figure below. The two faders labeled "Tune" and "Level" let you control the master tuning and output volume of the Ensemble. Since the output ports of the Master Tune/Level Module are connected to IC Send Modules (↑15.5, IC Send), these values are available via the Internal Connection Protocol Ensemble-wide.

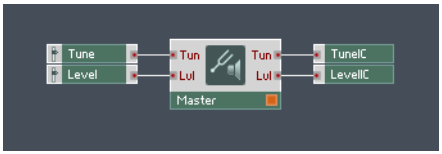


Fig. 14.43 The Structure of the "Master" Instrument.

14.17 Tempo Info

Fig. 14.44 Tempo Info Module

14.17.1 Overview

The Tempo Info Module sends from its output port the song tempo in units beats per second. The song tempo information is either received from REAKTORs internal MIDI Clock or from an external device (such as your host sequencer when running REAKTOR in plug-in mode). In stand-alone mode REAKTOR allows you to manually set the BPM (tempo in beats per minute) of its internal MIDI Clock with the BPM edit field, shown in the figure below. When using the internal MIDI Clock, these changes are reflected by the Tempo Info Module's output.

Fig. 14.45 Use the BPM edit field to change the BPM of the MIDI Clock.

Application

The Tempo Info Module is useful for displaying the song tempo, especially when this information if received from an external device. It may also be necessary to use this information in your Instrument's Structure. This could be the case when using running a sequencer Instrument, for example. To get the BPM value from the Tempo Info Module, simply multiply the output by "60".

14.17.2 Ports

- **(Out)** "Out" is the Event output port for the current song tempo in beats per second (Hz).

14.17.3 Example: Tempo in BPM

By convention, the tempo in music is given in units beats per minute. The Tempo Info Module delivers the tempo (as received from the internal MIDI Clock or the parent Instrument's MIDI In) in beats per second. To get the tempo in beats per minute, multiply it by the value "60", as shown in the figure below. A Numeric Display Module (↑1.8, Meter) has been used to display the BPM value on the Instrument Panel.

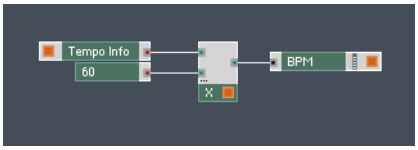


Fig. 14.46 Multiply the output of the Tempo Info Module by "60" to get the tempo in units BPM.

14.18 Voice Info

Fig. 14.47 Voice Info Module

14.18.1 Overview

The Voice Info Module enables you to use the parent Instrument's Voice information in the Structure. The Polyphonic "V" (Voice number) output port delivers a signal where the value of each Voice is its Voice number. The other three output ports deliver (from top to bottom) the parent Instrument's Voice settings: number of Polyphonic Voices, Min Voices, and Max Voices.

Application

The Voice Info Module is essential when building a custom framework for your Polyphonic synthesizer that allows for different Voice allocation schemes. Although you could assign values to the different Voices of your Instrument "manually" with To Voice Modules (↑14.11, To Voice), this would be complicated to achieve in a case where the number of Voices may change or for a large number of Voices. In such a case you could use the "V" (Voice number) output where each Voice already carries a different value, namely its Voice number. Dividing the "V" (Voice number) output port by the "Vcs" (Voices) output, for example, gives you a Polyphonic signal where the values of all Voices are evenly distributed between "0" to "1".

Please refer to section 9.3 in the Application Reference for more information on Polyphony.

14.18.2 Ports

Output Ports

- **(V)** "V" (Voice number) is the Polyphonic output port for the Voice number of each Polyphonic Voice. The first Voice carries the Voice number "1" and naturally the last Voice carries the number of Polyphonic Voices of the parent Instrument as its value.
- **(Vcs)** "Vcs" (Voices) is the Monophonic Event output port for the current number of Polyphonic Voices of the parent instrument. This value is set with the Voices edit field in the parent Instrument's Function page.
- **(Min)** "Min" (Min Voices) is the Monophonic Event output port for the minimum number of Unison Voices of the parent Instrument. It is the minimum number of Voices assigned to one key for Unison playing.
- **(Max)** "Max" (Max Voices) is the Monophonic Event output port for the maximum number of Unison Voices of the parent Instrument. It is the maximum number of Voices assigned to one key for Unison playing.

14.18.3 Example: Stereo Spread

Polyphonic signals in REAKTOR consist of a number of parallel Voices, as mentioned in subsection 9.3.1 in the Application Reference. These signals can be used to specify different values for all the different Voices. For example, you could position each Voice differently in the stereo (or even 5.1 surround sound) panorama field. This example shows how to build a Structure that does just that: a stereo spread structure that takes a Polyphonic Audio or Event signal and spreads it out in the stereo panorama.

We will use the Distributor Module (↑5.4, Distributor) to control the panning of the Voices between the left and right stereo channel. Let's assume you have a destination for the left and right stereo channels, such as output ports, labeled "L" and "R". Place a Distributor Module into the Structure and route the “0” output of the Distributor Module to the “L” output port. Hold down Ctrl if you are using Windows (Cmd for Mac OS X) while dragging a wire from the “R” output to the three dots on the bottom right of the Distributor Module. This will create an additional output port for the right stereo channel. Connect the source of the signal which you wish to "stereo spread" to the "In" input port of the Distributor Module. What you have now should be comparable to the screenshot below.

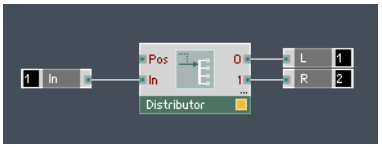


Fig. 14.48 The Distributor Module distributes the signal from the "In" input port between the "L" and "R" output ports.

The Structure now distributes the signal at the "In" input port between the "L" and "R" output ports, depending on the value that is sent to the "Pos" input port of the Distributor Module (↑5.4, Distributor). If Pos = 0, the signal is exclusively routed to "L" and if Pos = 1, the signal is exclusively routed to "R". For intermediary values between "0" and "1" for "Pos" (for example, Pos = 0.24) the distribution depends on the interpolation curve set with the Curve radio button selector in the Function page of the Distributor Module. For the Voices to be spread out more towards the edge of the panorama, click on the Sine radio button in the Function page of the Distributor Module, shown in the figure below.

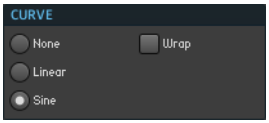


Fig. 14.49 The Curve radio button selector in the Function page of the Distributor Module's Function page lets you set the interpolation curve between the different output ports.

Now you need to assign each Voice a different pan setting so that the Voices are spread out in the stereo panorama. For you this means that the value arriving at the "Pos" input port of the Distributor Module (↑5.4, Distributor) needs to be different for each Voice. Furthermore, we want the mechanism that assigns each Voice its "Pos" value to be independent of the number of Voices that the parent Instrument is using. For this purpose, insert the Voice Info Module and a Divide Module (↑4.8, Divide, /) into the Structure.
Connect the “V” output port of the Voice Info Module (↑14.18, Voice Info) to the upper input port of the Divide Module and connect the "Vcs" output port of the Voice Info Module to the lower input port of the Divide Module. Next connect the output port of the Divide Module to the "Pos" input port of the Distributor Module (↑5.4, Distributor). Your Structure should look similar to the screenshot below.

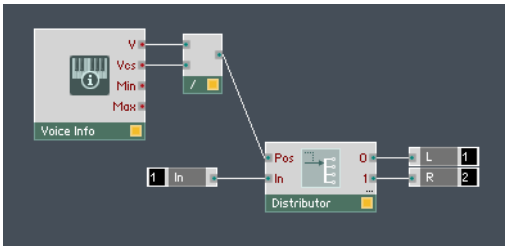


Fig. 14.50 The Voices in the Polyphonic signal sent from the "V" output port of the Voice Info Module carry their Voice number as their values. Dividing those values by the number of Voices yields a Polyphonic signal where each Voice carries a value in the range [0...1].

The value sent from the "V" output port of the Voice Info Module is different for each Voice. Each Voice carries the value of its Voice number. Furthermore, the "Vcs" output port of the Voice Info Module sends the number of Voices of the Instrument as its value (this is value is carried by all Voices). Let us look at this information for each Voice separately. Say you have an Instrument with the Polyphony, or number of Voices, set to "4". This means that at "Vcs" output port of the Voice Info Module all Voices carry the value "4", shown in the figure below.

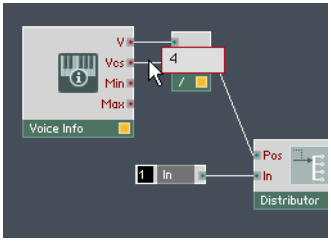


Fig. 14.51 The signal from the "Vcs" output port is Monophonic and carries the number of Polyphonic Voices, in this case, "4."

Now look at the "V" output port of the Voice Info Module. The first Voice carries the value "1", the second Voice carries the value "2", the third Voice carries the value "3", and the fourth Voice carries the value "4". This is shown in the screenshot below.

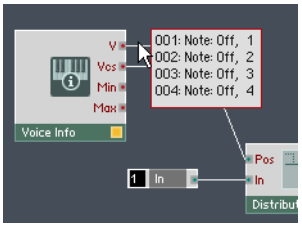


Fig. 14.52 The signal from the "V" output port is Polyphonic. Each Voice carries its Voice number as its value.

Next look at the output of the Divide Module (↑4.8, Divide, /). The first Voice carries the value "1 / 4" or "0.25", the second Voice carries "2 / 4" or "0.5", the third Voice carries "3 / 4" or "0.75", and the fourth Voice carries "4 / 4" or "1". Since each Voice receives a different value between "0" and "1" at the "Pos" input port of the Distributor Module (↑5.4, Distributor), each Voice is panned to a different coordinate in the stereo panorama. Clicking on the Linear radio button in the Function page of the Distributor Module you can move the panoramic distribution more to the center.

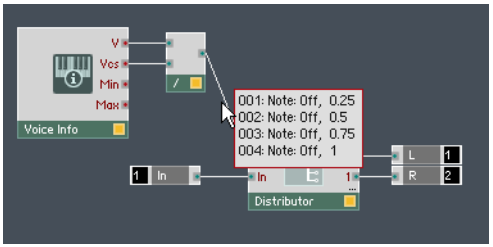


Fig. 14.53 The Structure for a Polyphonic stereo spread signal.

The final result should look like the Structure shown above. You can see that the Polyphonic signal going from the output of the Divide Module (↑4.8, Divide, /) to the "Pos" input port has each Voice carrying different values between "0" and "1". This means that each Voice receives a different pan setting and is distributed differently between the "L" and "R" output ports.

14.19 Tuning Info

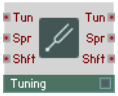


Fig. 14.54 Tuning Module

14.19.1 Overview

The Tuning Module allows you to adjust the parent Instrument's tuning as well as its Unison Spread and Note Shift parameters. Additionally, it delivers information on these parameters at its output ports.

When a note is played, the Instrument's MIDI pitch is defined first according to the settings in the MIDI In area of the Instrument's Connections page, in particular the Note Shift value. Then the pitch is transposed according to the Tune edit field in the Instrument's Function page. Finally, when playing in Unison mode, the pitches of the individual Voices are shifted according to the Spread edit field in the Instrument's Function page.

Please refer to subsection ý13.2.1 in the Application Reference for more information on the Note Shift edit field, as well as configuring an Instrument's MIDI In settings.

Application

If you want the Unison feature in your synthesizer Instrument, you don't necessarily have to create the Voice handling framework for this from ground up. Instead, you could use the Instrument's built-in Unison feature and control the Unison Spread parameter directly from the Instrument Panel, using the "Spr" (spread) input port of the Tuning Module.

Please refer to subsection 9.3.2 in the Application Reference for more information on Instrument tuning and Unison Playing.

14.19.2 Ports

Input Ports

- **(Tun)** "Tun" (tune) is the Event input port for adjusting the tuning the parent Instrument. Values at this input are given in the logarithmic pitch scale where 1 unit corresponds to 1 semitone. At Tun = 0.0 the MIDI Note A3 is tuned to 440 Hz. The typical range of values at this input port is [-1 ... 1] for fine tuning within a two semitone range. When disconnected, this input port uses the default value "0".
- **(Spr)** "Spr" (spread) is the Event input port for controlling the Unison Spread parameter of the parent Instrument. The "Spr" (spread) value determines the amount of detuning between Voices that are playing in Unison. It is only effective when the number of Unison Voices currently playing is greater than one. Values at this input are given in the logarithmic pitch scale where 1 unit corresponds to 1 semitone. The typical range of values at this input port is [-0.1 ... 0.1] which corresponds to slight detuning to create a fuller sound. When disconnected, this input port uses the default value "0".
- **(Shft)** "Shft" (shift) is the Event input port for controlling the amount by which MIDI Notes incoming at the Instrument's MIDI In are shifted. Values at this input are given in the logarithmic pitch scale where 1 unit corresponds to 1 semitone. When disconnected, this input port uses the default value "0".

Output Ports

- **(Tun)** "Tun"(tune) is the Event output port for the parent Instrument's tuning parameter, given in the logarithmic pitch scale.
- **(Spr)** "Spr" (spread) is the Event output port for the parent Instrument's Unison Spread parameter, given in the logarithmic pitch scale.
- **(Shft)** "Shft" (shift) is the Event output port for the parent Instrument's Note Shift parameter (in the Connections page), given in the logarithmic pitch scale.

Please refer to section 9.4 in the Application Reference for more information on the different scales in REAKTOR.

14.19.3 Properties: Function Page

Keeping the Tuning Module Always Active

If the Tuning Module is not connected to an active signal flow, then the incoming values at its input ports will not have any effect on the Instrument's pitch parameters. Connecting it directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Tuning Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

14.19.4 Example: Instrument Tuning

To control the tuning, Unison Spread, and shifting of incoming MIDI Notes of an Instrument from its Panel, place the Tuning Module into the Instrument's Structure, as shown in the figure below. Right-click the input ports and choose the Create Control menu entry to efficiently create knobs with the right resolution and range for the corresponding parameter. Note that the Unison Spread parameter has no effect if the maximum number of Unison Voices set in the parent Instrument's Function page is "1". In the example Structure below, IC Send Modules have been used to make it possible to forward the values of the three parameters to other Instruments via the Internal Connection protocol.

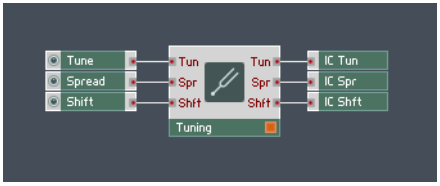


Fig. 14.55 Use the Tuning Module to control the parameters relevant to an Instrument's tuning, including Unison Spread.

14.20 System Info

Fig. 14.56 System Info Module

14.20.1 Overview

The System Info Module is an information source about your system. It delivers from its output ports (from top to bottom) the Sample Rate (in samples / sec), the Control Rate (in Hz), the display update rate, and the current CPU load (in %).

Application

The System Info Module is an important source of system parameters for exact DSP calculations. The "CR" (Control Rate) output port sends Events at the Control Rate and each Event carries the Control Rate as its value. You can use this to create a timer that runs at the Control Rate.

Please refer to section 3.5 in the Application to learn how to change the Sample Rate and Control Rate.

14.20.2 Ports

Output Ports

- **(SR)** "SR" (Sample Rate) is the Event output port for the current Sample Rate in samples per second. An Event is sent during Initialization and when the Sample Rate is changed.
- **(CR)** "CR" (Control Rate) is the Event output port for the current Control Rate in Hz. Events carrying the Control Rate as their value are sent from this output port. The Events are sent at the Control Rate.
- **(DClk)** "DClk" (Display Clock) is the Event output port for the current display rate in frames per second. An Event carrying the current display update rate as its value is sent just before each display update.

- **(CPU)** "CPU" is the Event output port for the current CPU load in percentages.

14.20.3 Example: Stopwatch

In this example (shown in the Structure below) you will build a stopwatch that runs at Control Rate. The output of this Structure is a constant stream of Events at the Control Rate, where each Event carries the total time that has passed since the last stopwatch reset Event.

First, you need a constant stream of Events at the Control Rate. The "CR" (Control Rate) output port of the System Info Module (↑14.20, System Info) is the perfect solution, since each Event additionally carries the Control Rate as its value. After converting the value carried by each Event to the time interval between them in milliseconds, you will use the Accumulator Module to sum up these time increments, resulting in real clock that runs at Control Rate.

First, you need to convert the Control Rate value (in Hz) carried by the Events stemming from the "CR" output port to the corresponding time interval (in milliseconds). As can be seen in the Structure below, you just need to divide 1000 by the Event signal from the "CR" output port). However, this calculation needs to be done only once after the Control Rate has been set. Therefore you can conserve CPU resources by making the 1/ CR calculation only once for every new Control Rate by using the Step Filter Module (↑13.17, Step Filter). Connect the Step Filter Module between the System Info and Divide Modules (↑4.8, Divide, /) (shown in the Structure below) and leave the "Tol" (tolerance) input port disconnected (zero). This way you make sure that a new "CR" Event is sent to the Divide Module only if the Control Rate has been changed. This is a handy trick for conserving CPU resources.

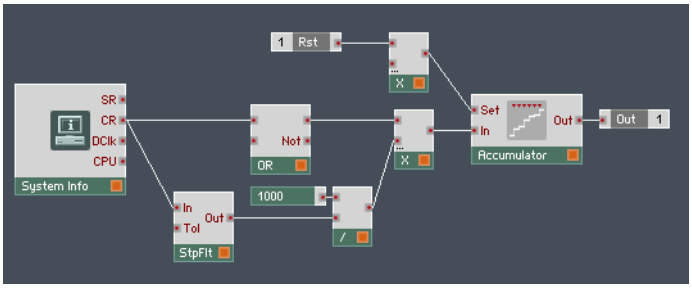


Fig. 14.57 The Structure for a simple Control Rate stopwatch.

Now that you have calculated the time increment in milliseconds between two Events at Control Rate, you need to make sure that Events carrying this value are sent at the Control Rate to the Accumulator Module (↑13.1, Accumulator). You could use a Value Module (↑13.15, Value) to do this, or alternatively, just connect the "CR" output port to an input port of a Logic OR Module (↑13.9, Logic OR). Since the "CR" value is always positive, the Logic OR Module will sent an Event with the value "1" (true) from its output port. Multiply the output of the Divide Module (↑4.8, Divide, /) with the "1" sent from the Logic OR Module and feed it into the "In" input port of the Accumulator Module. The Divide Module multiplies 1 * T at the Control Rate, where "T" is the time interval in milliseconds that you calculated above. Now the Accumulator Module adds up the time interval values at a constant rate and the resulting output is a stopwatch that counts time at the Control Rate, in unit milliseconds. Additionally, add an input for the reset signal for the Accumulator. Again, you could use the Value Module, or just connect the "reset" input port to a Multiply Module (↑4.5, Multiply, X) with one input port disconnected. Any incoming Event at the "reset" input will then cause the Multiply Module to send the value "0" to the "Set" input port of the Accumulator Module (↑13.1, Accumulator) and in turning resetting it to "0", as would be the case for a simple stopwatch.

14.21 Note Range Info

Fig. 14.58 Note Range Module

14.21.1 Overview

The Note Range Module allows you to adjust the parent Instrument's Upper Limit and Lower Limit parameters (as they appear in its Connection page) for the incoming MIDI Notes. Only MIDI Notes inside the corresponding range are assigned to Voices of the Instrument. Additionally, the Note Range Module delivers the values of these parameters at its output ports.

Please refer to subsection ý13.2.1 in the Application Reference for more information on the Upper Limit and Lower Limit edit fields, as well as configuring an Instrument's MIDI In settings.

Application

Use the Note Range Module to set up an Ensemble that features split-keyboard playing, directly from the Instrument Panel or using information from your Ensemble Structure. You can, for example, create an Ensemble that consists of two synthesizer Instruments. When playing MIDI Notes C-2 to C3 (lower half of your keyboard) the first synthesizer is played and when playing MIDI Notes C3 to G8 (upper half of your keyboard), the second synthesizer is played. As mentioned above, you can set the Lower Limit and Upper Limit values for each Instrument either by placing a Note Range in its Structure and sending the desired MIDI Note numbers to the corresponding input ports. You can also use the Upper Limit and Lower Limit edit fields Instrument's Connections page.

14.21.2 Ports

Input Ports

- **(Upr)** "Upr" (upper limit) is the Event input port for setting the upper limit for MIDI Notes received at the parent Instrument's MIDI In.
- **(Lwr)** "Lwr" (lower limit) is the Event input port for setting the lower limit for MIDI Notes received at the parent Instrument's MIDI In.

Output Ports

- **(Upr)** "Upr" (upper limit) is the Event output port for the current value of the Upper Limit parameter.
- **(Lwr)** "Lwr" (lower limit) is the Event output port for the current value of the Lower Limit parameter.

14.21.3 Properties: Function Page

Keeping the Note Range Module Always Active

If the Note Range Module is not connected to an active signal flow, then the incoming values at its input ports will not have any effect on the Instrument's MIDI Note limit parameters. Connecting it directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Note Range Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

14.21.4 Example: Split Keyboard

This example shows how to use the Note Range Module to set up an Instrument within an Ensemble such that it features split-keyboard playing. The Structure shown below is that of an Instrument in an Ensemble with two or more Instruments. The Instrument shown has incoming MIDI Notes assigned to it starting from the MIDI Note value "Lower" to a value that is one or more octaves above. The number of octaves can be set using the "Oct" knob. Since the point of split-keyboard playing is that separate areas of the keyboard are assigned to different Instruments, then the "Upper" value (as determined by "Lower" and "Oct") is forwarded to the other Instrument which then uses it as its "Lower" value.

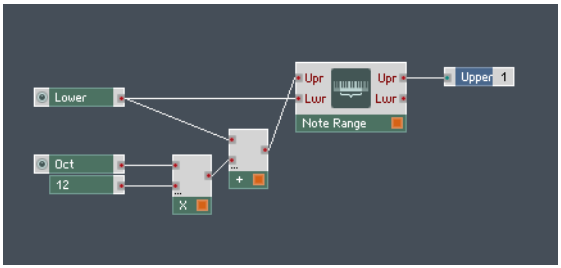


Fig. 14.59 The Structure to set up an Instrument for split-keyboard playing.

14.22 MIDI Channel Info

Fig. 14.60 MIDI Channel Info Module

14.22.1 Overview

The MIDI Channel Info Module's output ports send the numbers corresponding to the MIDI Channels that have been selected for the parent Instrument's MIDI In and MIDI Out. Additionally, you can use the input ports to change the MIDI Channels of the parent Instrument's MIDI In and MIDI Out.

Application

The MIDI Channel Info Module is useful when you want to build a MIDI sequencer that has several Instruments on different MIDI Channels hooked up to its MIDI Out. You could then use the MIDI Channel Info Module to choose the Instrument to which the sequencer sends its MIDI signals. Similarly when you have an Instrument that receives MIDI Information on several MIDI Channels you can use the MIDI Channel Module to choose which MIDI Channel the Instrument should "listen" to.

14.22.2 Ports

Input Ports

- **(ICh)** "ICh" (incoming MIDI Channel) is the Event input port for selecting the parent Instrument's MIDI Channel for incoming MIDI signals.
- **(OCh)** "OCh" (outgoing MIDI Channel) is the Event input port for selecting the parent Instrument's MIDI Channel at which it sends MIDI signals.

Output Ports

- **(ICh)** "ICh" (incoming MIDI Channel) is the Event output port that sends the parent Instrument's MIDI Channel number for incoming MIDI signals.
- **(OCh)** "OCh" (outgoing MIDI Channel) is the Event output port that sends the parent Instrument's MIDI Channel number at which it sends MIDI signals.

14.22.3 Properties: Function Page

Keeping the MIDI Channel Info Module Always Active

If the MIDI Channel Info Module is not connected to an active signal flow, then the incoming values at its input ports will not have any effect on the Instrument's MIDI Channels. Connecting it directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate the MIDI Channel Info Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

14.22.4 Example: Set MIDI Out Channel

You can not only use REAKTOR for sound generation and processing, but also to generate MIDI messages, such as for algorithmic sequencing. In such a case, if you have setup more than one MIDI device to receive the MIDI messages from REAKTOR, you might want to assign different MIDI Channels to these devices. This way you can use the MIDI Channel Module, as shown in the Structure below, to select directly from the Instrument Panel to which MIDI Channel the outgoing MIDI messages are sent and in turn which MIDI device ends up receiving these messages.

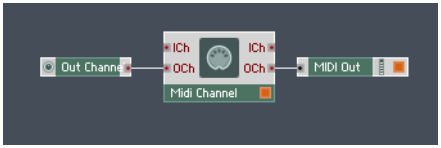


Fig. 14.61 Setting the outgoing MIDI Channel directly from the Instrument Panel.

14.23 Snapshot

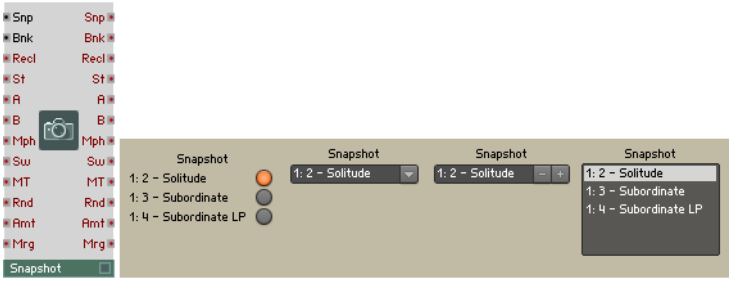


Fig. 14.62 Snapshot Module

14.23.1 Overview

The Snapshot Module gives you access to Snapshot store, recall, morph, randomize, and merge operations directly from the Structure. Its Panel representation allows you to switch between Snapshots exactly like the switching between list entries in a List Module (↑1.3, List).

Please refer to chapter 4 in the Application Reference for a detailed description of the various Snapshot operations available in REAKTOR.

The Snapshot Module has four different Panel representation styles: Buttons, Menu, Spin, and Text Panel. These styles are shown in that order in the figure above. Below you will learn how to switch between these styles and how to change the area of the Panel representation. For more detailed information on editing the Snapshot Module's Panel appearance using the View page, please refer to subsection 8.2.3 in the Application Reference. Also, custom control styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.
When a Snapshot operation is done either using the Snapshot tab, the Snapshot Module's input ports, or its Panel representation, an Event is sent from the corresponding output port. For example, selecting a Snapshot entry from the Panel representation causes Events with the Snapshot number, Bank number, and the value "1" to be sent from the "Snp" (Snapshot), "Bnk" (Bank), and "Recl" (recall) output ports, respectively.

Application

The Snapshot Module enables you to implement custom Snapshot behavior in your Ensemble. For example, you can build a sequenced synthesizer Ensemble that consists of three Instruments: a sequencer, a synthesizer, and an Instrument for managing Snapshots. Use the Snapshot manager Instrument to link different sequences to different synthesizer sounds, stored as sequencer and synthesizer Snapshots, respectively. The sequencer and synthesizer Instruments both should have a Snapshot Module which is controlled by the Snapshot manager Instrument using wires or IC Send (↑15.5, IC Send) and IC Receive Modules (↑15.6, IC Receive).

14.23.2 Ports

Input Ports

- **(Snp)** "Snp" (Snapshot) is the Audio input port for selecting the Snapshot to be recalled or stored. The range of values at this input port should be [1 … 128].
- **(Bnk)** "Bnk" (Bank) is the Audio input port for selecting the Snapshot Bank from which the Snapshot is to be recalled or stored. The range of values at this input port is [1 … 16].
- **(Recl)** "Recl" (recall) is the Event input port for the Snapshot recall operation. A positive Event at this input port recalls the Snapshot specified by the "Snp" (Snapshot) and "Bnk" (Bank) input ports.
- **(St)** "St" (store) is the input port for the Snapshot store operation. A positive Event at this input port stores the current Snapshot to the position specified by the "Snp" (Snapshot) and "Bnk" (Bank) input ports.
- **(A)** "A" is the Event input port for selecting the first of the two Snapshots (Snapshot A) between which you want to morph. A positive Event at this input port takes the Snapshot from the position specified by the "Snp" (Snapshot) and "Bnk" (Bank) input ports and loads it as Snapshot A.
- **(B)** "B" is the Event input port for selecting the second of the two Snapshots (Snapshot B) between which you want to morph. A positive Event at this input port takes the Snapshot from the position specified by the "Snp" (Snapshot) and "Bnk" (Bank) input ports and loads it as Snapshot B.

- **(Mph)** "Mph" (morph) is the Event input port for controlling the morph position between the Snapshot A and Snapshot B. Mph ≤ 0 corresponds to Snapshot A, Mph ≥ 1 corresponds to Snapshot B, and 0 < Mph < 1 results "morphed" settings between Snapshot A and Snapshot B. The actual morph position follows the value set at this input port in a smoothed transition and reaches it after the morph time set at the "MT" input port.
- **(Sw)** "Sw"(switch) is the Event input port for choosing if switch and buttons settings are to be taken either from Snapshot A or from Snapshot B. Events with negative and zero values at this input port set switches and buttons to their position in Snapshot A. Positive Events set switches and buttons to their position in Snapshot B.
- **(MT)** "MT"(morph time) is the Event input port for the morph time in milliseconds. Typical values at this input port are in the range [0 … 60 000].
- **(Rnd)** "Rnd" (randomize) is the Event input port for randomizing the current Snapshot. A positive Event at this input port triggers the Snapshot randomize operation where the randomization amount is set with the "Amt" input port.
- **(Amt)** "Amt" (random amount) is the Event input port for the randomization amount. The values at this input port should be in the range [0 … 1] which corresponds to [0 % … 100 %] of a control's range.
- **(Mrg)** "Mrg" (merge) is the Event input port for randomly merging Snapshot A and Snapshot B (set with the "A" and "B" input ports, respectively). A positive Event at this input port triggers the random merge operation where the randomization amount applied during merging is set with the "Amt" input port.

Output Ports

- **(Snp)** "Snp" (Snapshot) is the Event output port for the position of the current Snapshot in the Snapshot Bank. An Event is sent every time a Snapshot is recalled or stored.
- **(Bnk)** "Bnk" (Bank) is the Event output port for the number of the Snapshot Bank which holds the current Snapshot. An Event is sent every time a Snapshot is recalled or stored.
- **(Recl)** "Recl" (recall) is the output port that sends an Event with the value "1" each time a Snapshot is recalled.
- **(St)** "St" (store) is the output port that sends an Event with value "1" each time a Snapshot is stored.
- **(A)** "A" is the Event output for the position of Snapshot A in its Snapshot Bank. An Event is sent each time a Snapshot is recalled or selected for position A.

- **(B)** "B" is the Event output port for the position of Snapshot B in its Snapshot Bank. An Event is sent each time a Snapshot is recalled or selected for position B.
- **(Mph)** "Mph" (morph) is the Event output port for the current morph position. Mph = 0 corresponds to Snapshot A, Mph = 1 corresponds to Snapshot B, and 0 < Mph < 1 corresponds to a morphing position between Snapshot A and Snapshot B.
- **(Sw)** "Sw" (switch) is the Event output port that informs if switch and button positions from Snapshot A or from Snapshot B are used. If Sw = 0 switches and buttons are set to their position in Snapshot A. If Sw = 1 switches and buttons are set to their position in Snapshot B. An Event is sent each time the state of the "Sw" value is set and when a new Snapshot is set for the morph slot whose switch and button states are currently used (Snapshot A or Snapshot B, depending on "Sw").
- **(MT)** "MT" (morph time) is the Event output port for the morph time, in milliseconds. An Event is sent each time the morph time is changed.
- **(Rnd)** "Rnd" (randomize) is the output port that sends an Event with the value "1" each time a Snapshot randomize operation is triggered.
- **(Amt)** "Amt" (random amount) is the Event output port for the randomization amount. The range of values at this output port is [0 ... 1] which corresponds to [0 % ... 100 %] of a control's range. An Event is sent each time the random amount is changed.
- **(Mrg)** "Mrg" (merge) is the output port that sends an Event with the value "1" each time the random merge operation is triggered.

14.23.3 Properties: View Page

Choosing the Snapshot Module's Panel Representation

You can choose between four styles for the Snapshot Module's Panel representation. This is done by choosing the desired menu entry from the Style drop-down menu in the Module's View page, as shown in the figure below. The following styles are available:

- **Button:** Each Snapshot entry of the currently selected Snapshot Bank has a button. All buttons will be arrayed vertically in the Instrument Panel and the number of displayed buttons is taken as the number of Snapshot entries of the largest loaded Snapshot Bank. The currently activated Snapshot's button will be displayed in the Indicator color of the Instrument (please refer to section 8.3 in the Application Reference for more information on changing the different colors of the Instrument). The size of the buttons can be adjusted in the Module's View page.

- **Menu:** The Snapshots of the currently selected Snapshot Bank are listed in a drop-down menu on the Instrument Panel.
- **Text Panel**: Each Snapshot of the currently selected Snapshot Bank has an entry in a list on the Instrument Panel. The list displays several entries at once. If you have created more entries than are able to fit into the text panel display specified by the Width and Height edit field in the View page, you will get scrollbars in the panel.
- **Spin:** Each Snapshot of the currently selected Snapshot Bank has an entry in a list on the Instrument Panel. The list only displays the currently active Snapshot. You can switch through the list of Snapshots using a + and a - button to the right of the list entry in the Panel representation.

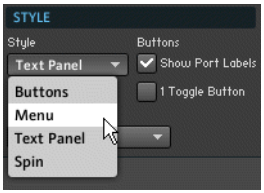
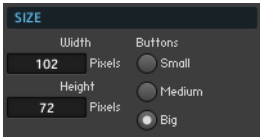


Fig. 14.63 Use the Style drop-down menu to choose the Panel representation of the Snapshot Module.

Changing the Size of the Snapshot Module's Panel Representation

Depending on the size of your selected Snapshot Bank, you might need a small or large area for the Snapshot Module's Panel representation.

► To change the width and height of the Snapshot Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



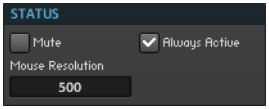
For more detailed information on editing the Snapshot Module's Panel appearance using the View page, please refer to subsection 8.2.3 in the Application Reference. Also, custom control styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

14.23.4 Properties: Function Page

Setting the Mouse Sensitivity of the Spin Panel Representation

If you have chosen the Spin Panel representation in the View page, you can switch between input Snapshots on the Instrument Panel, by clicking and dragging the Spin menu. For this, you might want to change the sensitivity of the Spin menu in respect to your mouse movements.

► To set the Spin Panel representation to have low mouse sensitivity, set the Mouse Resolution edit field in the Function page to a high value, perhaps "500". For high sensitivity, you might set the Mouse Resolution edit field to "10". The Mouse Resolution edit field is shown in the figure below.



Keeping the Snapshot Module Always Active

If the Snapshot Module is not connected to an active signal flow, then the incoming values at its input ports and actions on its Panel representation will not have any effect on the Instrument's Snapshots. Connecting it directly to Knob Modules, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate the Snapshot Module, go to its Function page and engage the Always Active checkbox, shown in the figure above.

14.23.5 Example: Snapshot Control from Panel

This example shows how to make Snapshot recalling, storing, and morphing accessible from the Instrument Panel. The fastest way is to insert a Snapshot Module and right-click the ports shown in the figure below and choose the *Create Control* menu entry. The Instrument Panel corresponding to this Structure is shown in the second figure below.

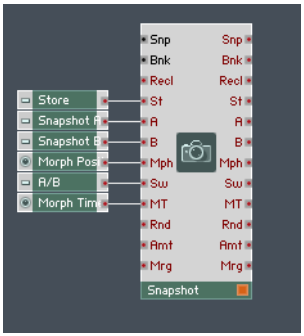


Fig. 14.64 The Snapshot Module with some basic controls.

Fig. 14.65 The Panel appearance of a simple Snapshot feature.

14.24 Set Random

Fig. 14.66 Set Random Module

14.24.1 Overview

The value at the Set Random Module's input port is used as the random seed for the pseudo-random number generator used in all Randomize, Slow Random (↑9.2, Slow Random) and Geiger Modules (↑6.40, Geiger Oscillator). Only Modules in the same Instrument are affected. Each unique value starts a different pseudo-random sequence.

Application

Pseudorandom number sequences are constructed using an algorithm that uses an initial value, called random seed, to create the subsequent sequence. When a single seed is used, the sequence will be exactly the same every time it is restarted. In some cases this predictability defeats the purpose of your "random" Module. The Set Random Module lets you set a new seed whenever you need a new pseudorandom sequence.

14.24.2 Ports

- **(In)** "In" is the Event input port to set the random seed to be used by all "random" Modules in the same Instrument.

14.24.3 Properties: Function Page

Keeping the Set Random Module Always Active

If the Set Random Module is not connected to an active signal flow, then the values incoming at its input port will not be applied to the random seed. Connecting it directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate Set Random Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

14.24.4 Example: Random Pitch Deviation

In old analog synthesizers variations in the internal state of voltage controlled oscillators resulted in a random deviation from the intended oscillator pitch. This effect can be simulated in REAKTOR using the Slow Random Module (↑9.2, Slow Random). As shown in the Structure below, just add the output of a Slow Random Module to the intended pitch value (here from a Note Pitch Module, also see ↑2.1, Note Pitch). The extent of the random deviation can be controlled by a knob at the "A" (amplitude) input port of the Slow Random Module. The frequency has been set quite arbitrarily to 3 Hz, as indicated by the value "3" at the "F" (frequency) input port. The resulting pitch is then fed to the "P" (pitch) input port of a parabolic oscillator, for example. The Set Random Module lets you choose the seed for the pseudorandom sequence created by the Slow Random Module.

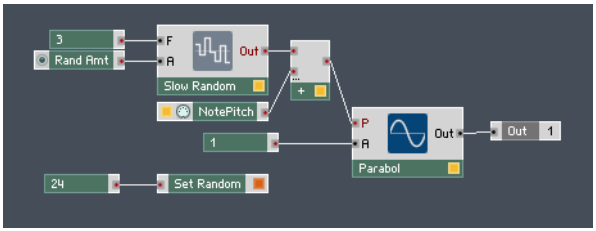


Fig. 14.67 A simple Structure to emulate random pitch deviations, as they appear in old analog synthesizers.

14.25 Unison Spread

Fig. 14.68 Unison Spread Module

14.25.1 Overview

The Unison Spread Module provides a Polyphonic signal for which Voices playing in Unison (assigned to the same MIDI Note) all carry a different constant value. These values are grouped around "0" and are multiples of the value at the input port. When a Voice is the only one playing a MIDI Note, its value at the Unison Spread Module's output is zero.

Application

In Unison mode, the different Voices that are assigned to play the same MIDI Note need to have their parameters spread out a little to make them slightly different so that their sum results in a sound that is fatter, and not just louder. For note pitch values this happens automatically and is controlled by the Unison Spread parameter in the Instrument's Function page. Other parameters can be spread out as well by adding an offset generated by the Unison Spread Module.

14.25.2 Ports

Input Ports

- **(In)** "In" is the Event input port for the spread amount. Voices playing in Unison will carry multiples of this value at the Unison Spread Module's output port.

Output Ports

- **(Out)** "Out" is the Event output port for the Polyphonic spread signal. Each Voice which is playing in Unison carries a different offset value which is a multiple of the value at the input port. Events at the input port as well as played notes trigger Events to be sent from this output port.

14.25.3 Example: Cutoff Spread

This example shows the use of the Unison Spread Module for spreading out a filter's cutoff pitch for different Polyphonic Voices. The "P Cutoff" knob sends the Monophonic value "69". To spread the cutoff parameter around this value in increments of "0.1", send the value "0.1" to the Unison Spread Module's input port and add the Polyphonic output to the signal from the "P Cutoff" knob (which happens to be at "69"). The resulting values for the four Voices in the example Instrument are shown in the figure below.

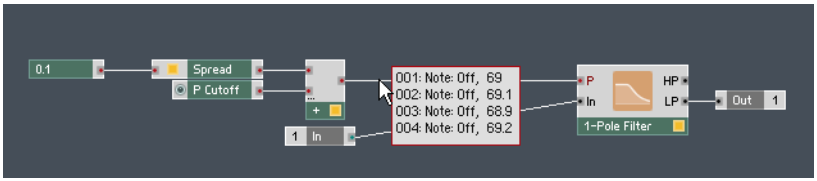


Fig. 14.69 You can use the Unison Spread Module for spreading out all sorts of parameters such as the cutoff pitch of a Filter Module.

14.26 Snap Value

Fig. 14.70 Snap Value Module

14.26.1 Overview

The Snap Value Module stores the value at its input port with a Snapshot (during a Snapshot store operation) and outputs that value as an Event when the Snapshot is recalled. The output value also depends on the morph position and can be set using the random merge operation. The randomize operation does not affect the Snap Value Module. The

Snap Value Module's Clipping feature offers clipping of incoming values with a variable range. Forwarding of incoming Events to the output can be turned on and off in the Function page.

Application

Use the Snap Value Module to make positions of custom controls storable in a Snapshot. Often the Mouse Area Module (↑1.18, Mouse Area) (running in Incremental Mouse Mode) is used in conjunction with a graphic display such as the Multi Display (↑1.16, Multi Display) or Multi Picture Module (↑1.11, Multi Picture). You would then connect the "BX" and "BY" input ports of the Mouse Area Module to Snap Value Modules and route the outgoing values of your custom control to the input ports of the Snap Value Modules. Sometimes the Snap Value can come in handy when faced with problems during Initialization. Sometimes inserting the Snap Value Module in the right place can delay an Initialization Event in a favorable way. Please refer to subsection 9.2.5 in the Application Reference for more information on Initialization.

14.26.2 Ports

Input Ports

- **(In)** "In" is the hybrid input port for the value to be stored in a Snapshot. The input signal is sampled at the moment when the Snapshot is stored. If connected to an Event source, input Events can be passed to the output when the Events Thru checkbox in the Function page is engaged.

Output Ports

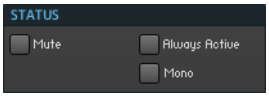
- **(Out)** "Out" sends an Event carrying the value stored with a Snapshot when it is recalled. Secondly, when the Events Thru checkbox in the Function page is engaged, incoming Events are also sent from this output port. Third, when the At Store Time checkbox has been engaged, an Event carrying the stored value is output right upon a Snapshot store operation. Last, Events are sent from this output when Snapshot morphing or random merge operations take place. When the Clipping checkbox in the Function page is engaged, the range of output values at this port is [Min … Max] (except values beyond this range that have been stored without clipping). "Min" and "Max" are set with the corresponding edit fields in the Function page.

14.26.3 Properties: Function Page

Keeping the Snap Value Module Always Active

If the Snap Value Module is not connected to an active signal flow, values at its input port will not be stored with the Snapshot. Connecting it directly to a Knob Module, for example, does not suffice. Also, it will not send any stored values upon a Snapshot recall operation. In such a case you can activate the Module with the Always Active checkbox.

► To activate Snap Value Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Setting the Range of Incoming Values

You might only want to work with values in a certain range defined by [Min … Max]. The Snap Value Module offers the Clipping feature where all incoming values greater than "Max" are replaced by "Max" and all incoming values less than "Min" are replaced by "Min". With the Clipping feature active and when Snapshots values are stored, the clipped values are used.

► To activate the Clipping feature, go to the Snap Value Module's Function page and engage the Clipping checkbox, shown in the figure below.

► To set the clipping range [Min … Max], go to the Function page and enter the desired values in the Max and Min edit fields, respectively. This is shown in the figure below.

If you have stored values which are outside of the range [Min … Max] prior to engaging the Clipping checkbox or defining that range, the stored values will remain intact. They will be output by the Snap Value Module upon a Snapshot recall operation unclipeed, even if they lie outside of the clipping range.

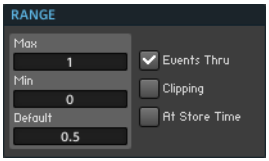


Fig. 14.71 Use the Min and Max edit fields to set the clipping range. Engage the Clipping checkbox to activate clipping. To have incoming Events be forwarded to the output, engage the Events Thru checkbox.

Events Thru

The Snap Value Module only passes incoming Events to its output when the Events Thru checkbox (shown in the figure above) is engaged. Regardless of this setting, Audio signals do not trigger Events at the Module's output (other than Events triggered by Snapshot recall and store operations).

When both the Events Thru and Clipping checkboxes are engaged, incoming Events with values outside of the Clipping range will be clipped and sent to the output as such. For the settings in the figure above, for example, an incoming Event with the value "1.5" will be clipped at "1" and sent from the output carrying the value "1.

Sending Events at Store Time

Upon a Snapshot store operation the Snap Value Module can output an Event with the stored value. This can be useful when sampling and storing a fast Audio signal with the Snap Value Module or when the stored value needs to be processed further for some reason. When the Clipping checkbox also is engaged, the values are stored after clipping and therefore output as the "At Store Time" Events.

► To cause the Snap Value Module to output an Event carrying the stored value right after a Snapshot store operation, engage the At Store Time checkbox, shown in the screenshot below.

Setting the Default Value

Like all controls with Snapshot capability, the Snap Value Module has a Default Value which it will output when the *Recall default values* menu entry in the Edit Bank drop-down menu is clicked.

► To set the Snap Value Module's Default Value, go to its Function page and enter the desired value into the Default edit field, shown in the figure above.

14.26.4 Example: Value Edit Field

This example shows how to build a custom value edit field that displays numbers with one decimal point precision. The edit field also enables you to change the values by clicking and dragging the mouse over the edit field and to reset the value to "0" by double clicking on the field. Start by creating a Macro with an output port labeled "Out". Make sure the Macro is set to Mononophonic Mode before you start building this example Structure. First we will start with building the value display. We want to display decimal numbers from -9.9 to 9.9, in 0.1 step increments. For this purpose you will need three Multi Picture Modules (↑1.11, Multi Picture): one for the sign (+ and -), one for the integers (0 to 9), and one for the first decimal place after zero (.0 to .9).

After inserting the three Multi Picture Modules into the Structure, designate one for the sign and go to its Function page. There, in the Select Picture menu you will find the *-> Open from file...* menu entry, click on it. This opens the Load Image File dialog box. Go to the *Tutorial Ensembles* folder and from there enter the folder labeled *Module Reference* where you will find the file "plus_minus.tga". Select the file and click the Open button. In the following Picture Properties dialog window, make sure that the Alpha Channel checkbox is activated. Next, make sure that the Animation Width and Animation Height edit fields both carry the value "32". You should now see a "+" and a "-" with transparent backgrounds next to each other in the Picture Preview display. If so, click the OK button to continue.

Fig. 14.72 The Picture Properties dialog window.

Now follow this same procedure for loading the digit animations to the other the Multi Picture Modules. Designate one of the Modules for integer digits and load the "numbers.tga" file from the same location as "plus_minus.tga" into the Multi Picture Module whilst making sure that the same settings for the Alpha Channel checkbox and Animation Height and Animation Width edit fields apply. For the third Multi Picture Module, repeat the aforementioned procedure, but this time loading the file "decimals.tga" into the Module. This last Module will be dedicated to display the first position after the decimal point. Go to the Instrument Panel and arrange the signs and digits in a way that makes sense when reading from left to right: first the sign, then the integer, and then the decimal fraction.

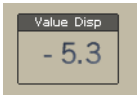


Fig. 14.73 The display part of the value edit field.

Now we need a way to split up the information that the value display should display into three parts: the sign (if the value is positive or negative), the integer part, and the decimal fraction. The first split will happen between the sign and the numbers. For that, insert a Rectify/Sign Module into the structure. The incoming value to the value display goes into the "In" input port of the Rectify/Sign Module. The "Sign" output port of that Module outputs a "1" if the incoming value is positive and "-1" if the incoming value is negative. We need to convert these values to specify the picture index of the animation loaded into the Multi Picuture Module. For that purpose, add an Add Module (↑4.2, Add, +) into the structure and use it to add "1" to the signal from the "Sign" output port of the Rectify/Sign Modue. The output of the Add Module will then be "0" if the value at the input of the Rectify/Sign Module is negative and "2" if the value is positive. After connecting the output port of the Add Module to the "Sel" input port of the Multi Picture Module, the "0" will address the first picture in the animation, the minus sign, and the value "2" will address the last picture of the animation, the plus sign. To test out the functionality of the value display, create a Knob Module at the "In" input of the Rectify/Sign Module by right-clicking on the "In" input port and clicking on the *Create Control* menu item. Set the range of the knob to [-9.9 … 9.9] by going to the Module's Function page and setting the Max Value edit field to "9.9" and the Min Value edit field to "-9.9".
On to the numbers: insert two Modulo Modules (↑4.9, Modulo) into the structure. Connect the "|x|" output port of the Rectify/Sign Module the "A" input port of one Modulo Module. Since we want to split the number into integers and fractions, it is only natural to create a Constant with the value "1" at the "B" input port of the first Modulo Module. Now the "Div" output port of the Modulo Module (↑4.9, Modulo) carries the value that is the largest multiple of "1" and is at the same time smaller than the value received from the "|x|" output port of the Rectify/Sign Module. For example, if |x| = 5.2, then Div = 5 or if |x| = 3.8, then Div = 3. Connect the "Div" output port to the "Sel" input port of the Multi Picture Module that is meant to display the integer part of the incoming signal value. Since we want to display signals within the range -9.9 to 9.9, then the "Div" value has a range [0 … 9] which exactly matches the indices of the pictures of the animation file "numbers.tga" loaded into that Multi Picture Module. Go to the Instrument Panel and move the knob that you

previously created and watch the sign and the integer numbers change in the Panel display of the Multi Picture Modules. Next, hook up the decimal fraction part of the incoming signal, namely the value sent to the "Mod" output of the Modulo Module (↑4.9, Modulo), to the "A" input port of the second Modulo Module. Next, create a Constant with the value "0.1" at the "B" input port of that Modulo Module. Wire the "Div" output port that same Modulo Module to the "Sel" input of the last remaining Multi Picture Module. With the "B" input port of the first Modulo Module (↑4.9, Modulo) receiving a constant value of "0.1", the "Mod" output port of the Modulo Module outputs the remainder of the incoming value, divided by "0.1". To demonstrate this with the numbers used in the previous example, if |x| = 5.2, then for the first Modulo Module, Mod = 0.2 and sending that value to the "A" input of the second Modulo Module, we get 0.2 / 0.1 = 2 at the "Div" output of the second Modulo Module. In the same way, if |x| = 3.8, then for the first Modulo Module, Mod = 0.8 and when that value is sent to the "A" input port of the second Modulo Module, the "Div" output of that Module is 0.8 / 0.1 = 8. Note that since the range of the "Mod" output of the first Modulo Module is [0 ... 0.9] and in the second Modulo Module we divide by "0.1", the range of the "Div" output of the second Modulo Module is [0 ... 9]. In this case "Mod" has a range [0 ... 9] which exactly matches the indices of the pictures of the animation file "decimals.tga" loaded into that Multi Picture Module. Go to the Instrument Panel and move the previously created knob to test out the functionality of the value display.

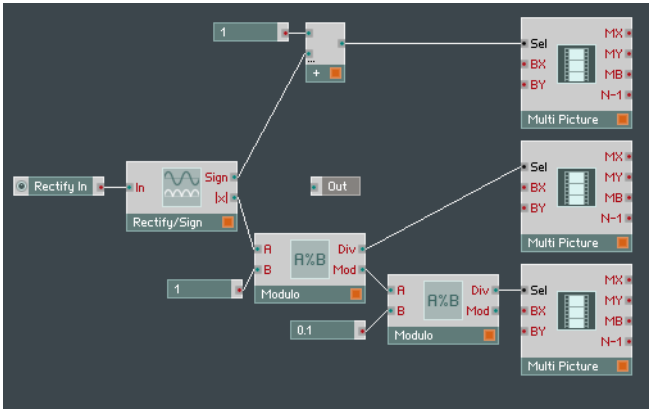


Fig. 14.74 The Structure for sorting out the sign and decimal parts of the value.

Now that you have a functioning value display, you probably want a way to manipulate the values directly through the value display itself, not with an extra knob cluttering your Panel layout. This can conveniently be achieved with the Mouse Area Module. Place it into the structure in front of the Rectify/Sign Module, leaving a bit of space for a Merge Module that you will insert later. By default, the Mouse Area Module is transparent in the Instrument Panel. To make testing the functionality easier you should make the mouse area visible. First, the Mouse Area Module needs to have a graphic element in the Instrument Panel. In the Appearance page of the Module, click on the Bar Outline Style selector to create a filled rectangle as the graphical element of the mouse area in the Instrument Panel. Now you have a 100 % transparent rectangle marking the mouse area in the Instrument Panel; for ease of manipulation you should to make the appearance more opaque. To do this, enter the value "80" into the Inactive Transparency edit field and the value "70" into the Active Transparency edit field of the Module's Appearance page. To add one more touch, click on the Left Button selector to make the active state of the mouse area appear during a left-click on the mouse area in the Instrument Panel. Now you should be able to see the mouse area in the Instrument Panel and see it become darker when you click on it. Unlock the Panel and move the mouse area over your value display. Set the dimensions of the mouse area to cover all digits of the value display using the Size X and Size Y edit fields in the Appearance page of the Mouse Area Module. After finding good settings for the dimensions, ("30" for Size X and "60" for Size Y should be good) lock the Panel again.

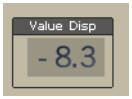


Fig. 14.75 The value edit field with the mouse area.

The Mouse Area Module can supply us with the values that we want to display with the structure that you built above. When you click and drag the mouse over the mouse area in the Instrument Panel, the "X" and "Y" output ports output events carrying the "x" and "y" coordinates, respectively, as specified in the Function page of the Module. For our purpose, we are only interested in clicking and dragging the mouse vertically to change the values in the value display since that seems the most intuitive setup for most Instrument layouts. For this reason we will leave the "X" output (and hence the settings in the Range X area of the Function page) alone and focus on the "y" coordinate. Connect the "Y" output port of the Mouse Area Module to the "In" input port of the Rectify/Sign Module. In order to be able to control the output values of the "X" and "Y" output ports of the Mouse Area

Module over the mouse area in the Instrument Panel much like a fader, that is, that clicking and dragging changes the output value, engage the Incremental Mouse checkbox in the Function page of the Mouse Area Module.
The range of values that we want to get out of the Mouse Area Module is [-9.9 ... 9.9]. To specify these parameters, go to the Function page of the Mouse Area Module and enter "-9.9" into the Min Y Value edit field and "9.9" into the Max Y Value edit field in the Range Y area. Next, we want to determine the precision with which one can manipulate the output values. The relevant parameters to this are edited with the Num Steps X and Mouse Reso X edit fields. If Num Steps X and Mouse Reso X are too large, you would have to drag the mouse all the way to the top of the screen to see any change in the output value. On the other hand, if these parameters are too small, dragging the mouse over the mouse area just a small distance already effects a great change in the output value. In order to have the right value increments at the output, the Num Steps X value should be an integer multiple of the Mouse Reso X value. Try playing around with the values and seeing the effect it has on the value display, but in the end settling with a value of "1000" in the Num Steps X edit field and "500" in the Mouse Reso X edit field should be fine.
You now have a structure that displays values and lets you directly manipulate the values by clicking and dragging the mouse over the display. We would like to add one additional feature to this setup, namely, the possibility to double-click the mouse area to reset the value to a certain number. The Mouse Area Module has an output port labeled "Db" that sends out an event with the value "1" every time the mouse area on the Panel is double-clicked. You now have two sources of events to determine the number that your value edit field is supposed to display. To combine them, place a Merge Module into the structure between the Mouse Area Module and the Rectify/Sign Module. Replace the wire to the "In" input port of the Rectify/Sign Module with a wire from the "Out" output port of the Merge Module. Next, connect the "Y" output port of the Mouse Area Module to the input port of the Merge Module and finally, while holding Ctrl, drag a wire from the "Db" output of the Mouse Area Module to the three dots on the left side of the Merge Module. The latter action creates a second port for the Merge Module. When you click and drag the mouse over the mouse area in the Instrument Panel, a value is sent to the Merge Module from the "Y" output port of the Mouse Area Module. This value is then forwarded in the form of an event (carrying the same value as was received from the "Y" output port) to the value display part of your structure which starts with the Rectify/Sign Module. If you now double click the mouse area in the Instrument Panel, the value "1" is sent to the Merge Module.

Now the output port of the Merge Module sends an event with the value "1". This should be reflected in the Instrument Panel where the value display should display "+1.0". Each new event at any of the input ports of the Merge Module triggers an event at the output port with the same value as the last incoming event. During initialization, the value from the lowest input port of the Merge Module is sent last to its output port so in the current structure, if you press the Run/Stop Audio button twice (turning the audio off and then on again) to reinitialize the structure, your value display will show the initialization value coming from the "Db" output port of the Mouse Area Module, namely "1". Depending on the circumstances, this might not be what you want, so changing the port order of the wires connected to the Merge Module can be a way to get the desired reinitiation behavior. In this case switching the port orders around will yield the initialization value "0" from the "Y" port of the Mouse Area Module to be sent to the Merge Module's output last.
Regardless of the port order of the wires at the Merge Module's input, you might not want the value edit field to be reset to "+1.0" when you double click the display. Having a reset-behavior of "0" might be much more natural. This can be changed very easily with the help of only one additional Module. Insert a Value Module into the structure and connect the "Db" output port of the Mouse Area Module to the "Trig" input port. Any event at the "Trig" input port will cause an event with the value lying at the "In" input port of the Value Module to be sent from its output port. In this case we leave the "In" input port disconnected which means that an event from the "Db" output port of the Mouse Area Module causes an event with the value "0" to be sent from the output of the Value Module. This is exactly what you want: you replaced an event carrying the value "1" with an even carrying the value "0"! Now replace the wire at the input port of the Merge Module that is connected to the "Db" output port of the Mouse Area Module with a wire to the output port of the Value Module. Now go to the Instrument Panel, make sure the value display is nonzero and double click it. Voila! The value display should now reset to the value "0".
Your value edit field is almost complete. You are probably asking, “Why almost and not fully complete??” The reason lies yet again in the initialization behavior. If you reinitialize the structure or load the Instrument anew, regardless of the value that the display was set to, it will always show "0". If you want to use this structure in an Instrument, it is desirable to have the Instrument's state upon saving fully recalled. Knob, fader, and value display settings count among the things you usually do not want to change when you reload an Instrument. This is where the Snap Value Module comes into play. Insert one into the structure and connect its input port to the output port of the Merge Module. The output port of the Snap Value Module should now be connected to the input port of the Rectify/

Sign Module and the "Out" Out Port that you added to your Macro's structure in the very beginning. The Snap Value Module takes the last value that was received at its input port and stores it in memory. When the structure is reinitialized or the Instrument is reloaded, this value will be sent to the output port of the Snap Value Module (and on to the structure downstream) instead of the initialization values from the Mouse Area Module. To ensure that the Snap Value Module properly deals with the values that it is supposed to store, go to its Function page and set the Max Value edit field to "9.9" and the Min Value edit field to "-9.9". This range is the one we have been using in this example. Also, to retain the functionality that we have achieved until now, that is, that the values from the Mouse Area Module are sent to the rest of the structure (and now, to the Out Port), make sure that the Event Thru checkbox in the Function page of the Snap Value Module is engaged. If the Event Thru checkbox would be disengaged, the values arriving at the Snap Value Module's input port would not be forwarded to the output port except during initialization. Your complete structure should now look like the following picture.

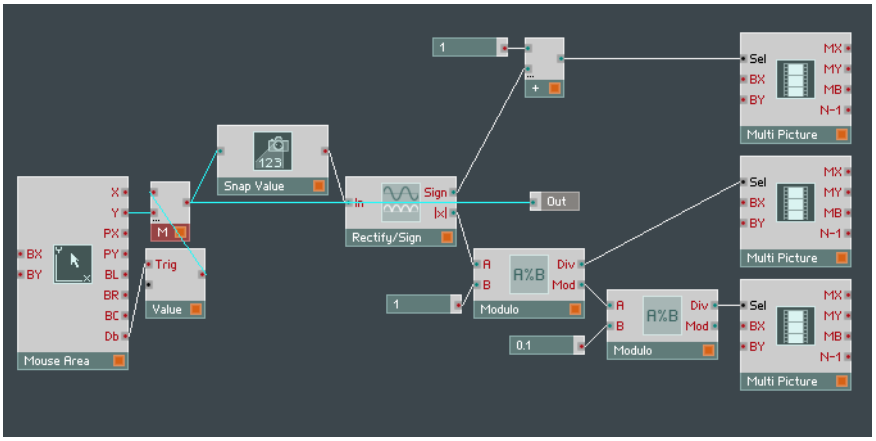


Fig. 14.76 The final Structure of the value edit field.

14.27 Snap Value Array

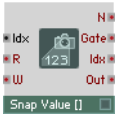


Fig. 14.77 Snap Value Array Module

14.27.1 Overview

The Snap Value Array is the "big brother" of the Snap Value Module. It stores and recalls arrays of floating-point (fractional) numbers to and from the edit buffer and the current Snapshot. Additionally it holds the very handy feature of Self-Iteration.

A single Snap Value Array Module can hold 1 - 40 arrays. All arrays contain the same number of elements which can be arbitrarily large, limited only by the availability of system memory. Both the number of arrays and the number of elements per array are defined in the Function page. For each array there is a corresponding input port which receives the values to be written into the array and an output port which sends values in the array. The write and read operations are applied to the array element specified at the "Idx" (index) input port. Per default there is initially only one array. It is labeled "W", as can be seen by the lowest input port shown in the figure above (the "R" (read) input port does not belong to an array).

To write a value to the first element of the array labeled "W", you first have to specify that you are writing to the first array element by sending the value "1" to the "Idx" (index) input port. After the "Idx" (index) value has been specified, sending an Event to the "W" input port will write the Event's value to the first element of the "W" array. To write to the array's second element, you first need to send the value "2" to the "Idx" input port before sending the next Event to the array's input port. The range of "Idx" (index) values is [1 ... N] where "N" is the number of elements per array.

To trigger a read operation of the element specified by the "Idx" of each array, first send the index value to the "Idx" input port. Then send an Event of any value to the "R" (read) input port. For each array, an Event carrying the value in the specified array element is sent from its output port and in the order of array output ports, from top to bottom.

When the Self-Iteration checkbox in the Function page is engaged, all array elements are output in a serial fashion (i.e. the first element, then the second element, then the third etc.) when any of the following operations occur: Initialization, Module activation, Snapshot recall, randomize, random merge, and morph. The Self-Iteration feature enables you to update a complete set of data with Snapshot operations.
The Snap Array Module's Clipping feature allows you to set a different clipping range for each array and choose if incoming values at the array's input port which are outside of the clipping range should be clipped. Also forwarding of incoming Events at array input ports to the corresponding array output ports can be turned on and off in the Function page.
Memory for the Snap Value Array Module is allocated dynamically. That is, creating additional Snapshots causes additional memory to be allocated and deleting Snapshots results in memory being freed. This keeps memory requirements as low as possible.
When the Snap Isolate checkbox in the Function page is engaged, only the most recent value written to each array element is stored (and recalled during Ensemble Initialization). No data is stored or recalled in response to Snapshot operations. All Snap Value Array input ports accept Monophonic signals only.

Application

A typical application of the Snap Value Array is storing sequencer data in Snapshots. For this purpose, it is often used in conjunction with Event Table or Multi Display Modules. The Self-Iteration feature makes it possible to almost instantaneously update parts of the Structure, much like the Iteration Module (↑13.13, Iteration). Another common application is supplying Array Modules in REAKTOR Core with data upon Initialization, since these Modules do not retain their internal state after an Ensemble is closed. Whenever using the Snap Value Array Module, make sure that the index of the addressed array element is specified at the "Idx" input port before carrying out any read or write operations.

14.27.2 Ports

Input Ports

- **(Idx)** "Idx" (index) is the Audio input port for specifying the index of the array element to address (for both read and write operations). Arrays are 1-based; i.e. the index of the first element is 1 (not 0). Fractional values are rounded to the nearest integer. Where

"Idx" values are outside the range [1 … N] (where "N" signifies the number of elements per array), the behaviour depends on the Index Behaviour radio button selector in the Function page.

- **(R)** "R" (read) is the Event input port for triggering read operations. When an Event (of any value) arrives at the "R" (read) input port, the array element specified by the "Idx" input port is read for each array and its value is output from the output port corresponding to the array to which the element belongs. In other words, each Event at the "R" (read) input port triggers a single Event at each array's output port. These Events are sent in the order of the array output ports from top to bottom. Thus, multiple arrays are addressed in parallel by the same "Idx" value and read trigger Event. It is essential to set the desired "Idx" value before Events arrive at the "R" (read) input.
- **(W1, W2 …)** "W1, W2 …" (write 1, write 2…) are array input ports for writing values into the elements of the corresponding array. When an Event arrives at an array input port (such as "W", the default array), its value is written to the corresponding array's element specified by the value at the "Idx" input port, overwriting any data that was previously there. The Snap Value Array Module provides a separate "write" input port for each array it contains. Each input port can be renamed as appropriate. When the Events Thru checkbox in the Function page is engaged, Events arriving at the array input ports are passed to the corresponding array output ports.

Output Ports

- **(N)** "N" is the Event output port for "N", the number of elements per array.
- **(Gate)** "Gate" sends an Event with value "1" before array output ports start sending Events and then sends an Event of value "0" after all array output ports have finished sending their Events.
- **(Idx)** "Idx" (index) is the Event output port that reports the index number of the array elements currently being read or written to. Events arriving at the "R" and array input ports as well as Snapshot operations such as recall and morph trigger an Event at the "Idx" output port. For read and write operations, the "Idx" output port sends its Event before any array output ports start sending their Events (but after Gate On Event at the "Gate" output port).

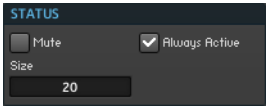
- **(Out1, Out2 ...)** "Out1, Out2 ..." (array out 1, array out 2 ...) are the Event output ports which send the value stored array element addressed by "Idx" of the corresponding array. Events are transmitted from these array output ports in response to Events at the "R" input port and Snapshot operations (recall, morph etc.). Upon a read operation the Events are sent from these array output ports in the order from top to bottom.

14.27.3 Properties: Function Page

Keeping the Snap Value Array Module Always Active

If the Snap Value Array Module is not connected to an active signal flow, values at its input ports cannot be stored and arrays cannot be read. Connecting it directly to a couple of Knob Modules, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate Snap Value Array Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Setting the Number of Elements per Array

All arrays have the same number of elements which can be arbitrarily large, limited only by your computer's memory.

► To set the number of elements per array, go to the Function page and enter the desired value into the Size edit field, shown in the figure above.

Note that reducing the number of array elements will cause the values stored by Snapshots in the deleted array elements to be lost forever.

Events Thru

The Snap Value Array Module only passes Events incoming at its array input ports to the corresponding output ports when the Events Thru checkbox (shown in the figure below) is engaged. When the Clipping checkboxes have been engaged for one or more arrays, incoming Events at these arrays with values outside of the clipping range will be clipped and sent to the array output as such.

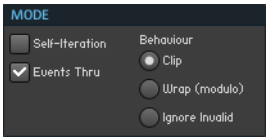


Fig. 14.78 Snap Value Array Module modes

► To have incoming Events be forwarded to the output, engage the Events Thru checkbox. Engage the Self-Iteration checkbox to serially output the whole array data upon the operations mentioned in the text below. Use the Behaviour radio button selector to determine how the Module deals with "Idx" (index) values that exceed the allowed range.

Self-Iteration

The Self-Iteration feature causes the values of all array elements to be output one after another when any of the following operations occur: Initialization, Module activation, Snapshot recall, randomize, random merge, and morph. The values are output before the next Audio sample is processed, therefore almost instantanously.

► To activate the Self-Iteration feature, engage the Self-Iteration checkbox in the Function page, shown in the figure above.

Setting the Index Behaviour

To choose the way to treat "Idx" (index) values at the "Idx" input port that exceed the number of array elements ("N" value), press the corresponding radio button from the Behaviour selector (shown in the figure above). The three choices are as follows:

- **Clip:** "Idx" (index) values that exceed the allowed range address the array element with the highest or lowest index, depending on if the higher or lower bound has been exceeded.
- **Wrap (modulo):** "Idx" values that exceed the array size are "wrapped" around the "N" (number of elements) value. This means that the value "N" is subtracted from the "excessive" index value: Idx' = Idx - N.
- **Ignore Invalid:** "Idx" (index) values that exceed the number of array elements are ignored.

Adding and Deleting Arrays

The Snapshot area in the Snap Value Array's Function page gives you an overview of all arrays in the Module and their settings. A screenshot of a list of arrays is shown below.

► To append an array at the bottom of the list, press the Append button.
► To insert an array after the array corresponding to the selected list entry, press the Insert button.
► To delete the selected array, press the Delete button.

Note that deleting an array will cause the contained values stored by Snapshots to be lost forever.

► To rename an array, double-click its entry in the Label column, enter the desired name, and press Enter.

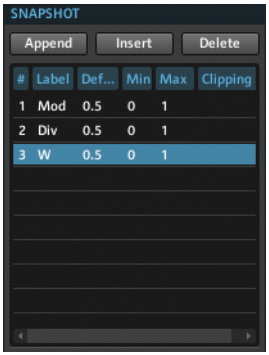


Fig. 14.79 The Snapshot area of the Snap Value Array's Function page allows you to add and remove arrays. Additionally for each array you can toggle the Clipping feature on and off, define the clipping range, and the array's Default Value.

Setting the Range of Incoming Values

You might only want to work with values in a certain range defined by [Min … Max]. The Snap Value Array Module offers the Clipping feature separately for each array where all incoming values greater than "Max" are replaced by "Max" and all incoming values less than "Min" are replaced by "Min". With the Clipping feature activate and when Snapshots values are stored, the clipped values are used.
► To activate the Clipping feature for an array, engage the Clipping checkbox by clicking on the array's entry in the Clipping column, shown in the figure below.
► To set an array's clipping range [Min … Max], its entries in the Min and Max columns and enter the desired "Min" and "Max" values, respectively. These entries are shown in the figure below.

If you have stored values in an array which are outside of the range [Min ... Max] prior to engaging its Clipping checkbox or defining that range, the stored values will remain intact. They will be output by the Snap Value Array Module unclipped even if they lie outside of the clipping range.

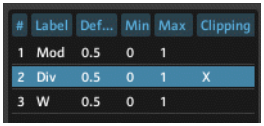


Fig. 14.80 Use an array's entries in the Min and Max columns to set its clipping range. Engage the Clipping checkbox in the row corresponding to the desired array to activate its Clipping feature. Use the Default column to set the Default Value for all arrays.

Setting the Default Value

Like all controls with Snapshot capability, the Snap Value Array Module's arrays each have a Default Value which it will be output when the *Recall default values* menu entry in the Edit Bank drop-down menu is clicked.

► To set an array's Default Value, double-click its entry in the Default column (shown in the figure above), enter the desired value, and press Enter.

14.27.4 Example: Saving 2 Values

This example shows how to save two values in a Snap Value Array Module with one array of two entries. The Structure that achieves this functionality is shown in the figure below. Note that for each incoming signal (at an input port) a combination of an Order Module (↑13.12, Order) and a Value Module (↑13.15, Value) is used to first send the index of the array's entry and only then the actual value to be saved. Since both values are saved in one array Merge Module's (↑13.16, Merge) have been used to merge the Events at the Snap Value Array's input ports. The values in the array are saved with Snapshots. In order for the values to be forwarded to the corresponding output ports upon a Snapshots recall operation or during Structure Initialization, engage the Self-Iteration checkbox in the Snap Value Array's Function page. Additionally, to forward the incoming Events to the output ports, engage the Events Thru checkbox.

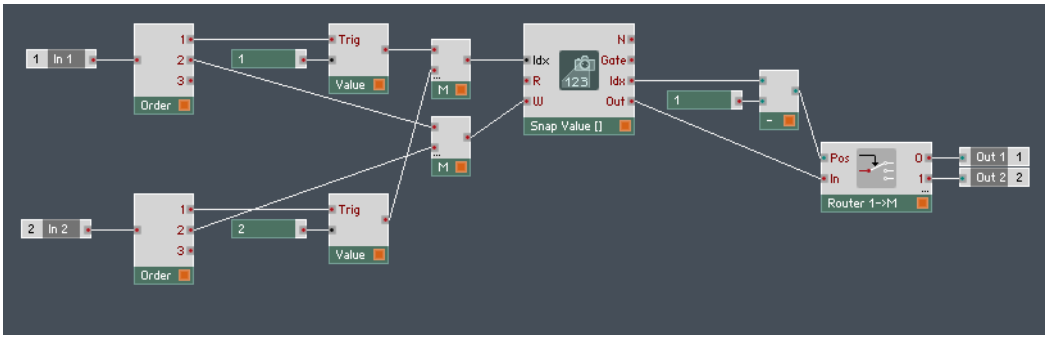


Fig. 14.81 The Structure for saving two values in a Snap Value Array Module.

15 Terminal

Terminal Modules are used to route Audio and Control signals in and out of REAKTOR's Instrument and Macro sub-Structures. The most common type of Terminal Modules are the In and Out Modules which serve as input and output ports for your Instruments and Macros. There are also Send and Receive as well as IC Send and IC Receive Modules to create "wireless" connections within and across Instruments, respectively. Additionally, you will find here the OSC Send and OSC Modules which make it possible to create custom OSC connections to and from your Instrument. You can create In and Out Modules automatically by using the dynamic ports feature. This is done by dragging a wire to the three dots on either lower corner of the Instrument or Macro Object (depending if you want input or output ports) while holding down the Ctrl key in Windows (Cmd key in Mac OS X).

15.1 In Module

15.1.1 Overview

The In Module is the most common input terminal for Audio and Event signals. To each In Module inside an Instrument or Macro corresponds an input port on the Instrument or Macro Object.

Application

Place an In Module inside an Instrument or Macro Object to create an input port for it. Through this input port you can now route signals from outside the Object into its Structure.

In Modules that are placed in the top-level Instrument's Structure create connections to the Audio Bridge between REAKTOR and your soundcard. The input and output connections of the Audio Bridge can be configured in the Audio and MIDI Settings dialog window (see chapter 2 in the Getting Started Manual). In such a case the Info page of the Module shows which audio input channel of your soundcard it is connected to.

15.1.2 Ports

(In) "In" is the hybrid input port for the signal to be sent from the outside into the Instrument or Macro.

15.1.3 Properties: Function Page

Changing the Port Order

The number on the In Module (shown in the Module picture above) denotes the position at which the input port appears on the Instrument or Macro when counted from the topmost input port. This number can be changed using the Port Index edit field in the Module's Function page (shown below).

Please refer to subsection 7.3.4 in the Application Reference for more information on changing the port order including an example for working with input and output ports.

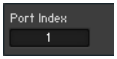


Fig. 15.1 Use the Port Index edit field in the Module's Function page to set the position at which the input port appears at the outside of the Instrument or Macro when counted starting from the topmost input port.

15.2 Out Module

Fig. 15.2 Out Module

15.2.1 Overview

The Out Module is the most common output terminal for Audio and Event signals. To each Out Module inside an Instrument or Macro corresponds an output port on the Instrument or Macro Object.

Application

Place an Out Module inside an Instrument or Macro Object to create an output port for it. Through this output port you can now route signals from the Object's internal Structure to the outside. Note that Instruments can only output Monophonic signals. In Instrument's send outgoing Polyphonic signals through a Voice Combiner Module (↑14.3, Audio Voice Combiner) before connecting them to the Out Module.

Out Modules that are placed in the top-level Instrument's Structure create connections to the Audio Bridge between REAKTOR and your soundcard. The input and output connections of the Audio Bridge can be configured in the Audio and MIDI Settings dialog window (see chapter 2 in the Getting Started Manual). In such a case the Info page of the Module shows which audio output channel of your soundcard it is connected to.

15.2.2 Ports

- **(Out)** "Out" is the hybrid output port for the signal to be sent from the Instrument's or Macro's internal Structure to the outside.

15.2.3 Properties: Function Page

The number on the Out Module (shown above in the Module picture) denotes the position at which the output port appears on the Instrument or Macro when counted from the topmost output port. This number can be changed using the Port Index edit field in the Module's Function page (shown below).

Please refer to subsection 7.3.4 in the Application Reference for more information on changing the port order including an example for working with input and output ports.

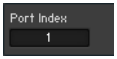


Fig. 15.3 Use the Port Index edit field in the Module's Function page to set the position at which the output port appears at the outside of the Instrument or Macro when counted starting from the topmost output port.

15.3 Send

Fig. 15.4 Send Module

15.3.1 Overview

The Send Module is an outgoing "wireless" connection for Audio and Event signals within an Instrument. Signals sent to the hybrid input port can be retrieved at the output port of any Receive Module within the same Instrument. Each inserted Send Module creates an entry in a list of available signal sources in the Function page of all Receive Modules (↑15.4, Receive) in the Instrument. The label of the Send Module is used as the name of

the corresponding entry in the signal sources list. Choosing the corresponding entry from the signal sources list of a Receive Module automatically creates a "wireless" connection between the Send Module and that Receive Module.

Application

Use the Send and Receive Modules to create "wireless" connections within an Instrument. This might be desirable to keep Structures tidy by eliminating wires or just to save you the hassle of wiring signals over great distances and many hierarchy levels. Also, you could create a modular synthesizer Instrument for which the connections between the individual modules can be controlled from the Instrument Panel.

For an example regarding the Send Module, please refer to the entry on the Receive Module (↑15.4, Receive).

15.3.2 Ports

- **(In)** "In" is the hybrid input port for the signal to be sent to a Receive Module.

15.3.3 Properties: Function Page

Keeping the Send Module Always Active

If the Send Module is not connected to an active signal flow, values at its input port are not sent to connected Receive Modules. Connecting it directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate Send Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

15.4 Receive

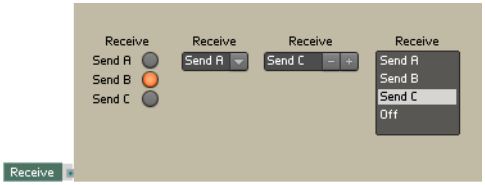


Fig. 15.5 Receive Module

15.4.1 Overview

The Receive Module is an incoming "wireless" connection for Audio and Event signals within an Instrument. Use the signal sources list in the Module's Panel representation or the Function page to select the Send Module whose input signal should be sent from the Receive Module's hybrid output port. Each Send Module creates an entry in the signal sources list seen by Receive Modules within the same Instrument. Choosing a Send Module's entry from the signal sources list automatically creates a "wireless" connection between that Send Module and the Receive Module.

Application

Use the Send and Receive Modules to create "wireless" connections within an Instrument. This might be desirable to keep Structures tidy by eliminating wires or just to save you the hassle of wiring signals over great distances and many hierarchy levels. Also, you could create a modular synthesizer Instrument for which the connections between the individual modules can be controlled from the Instrument Panel.

15.4.2 Ports

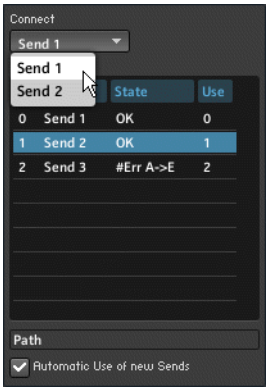
- **(Out)** "Out" is the hybrid output port for the signal received from a Send Module within the same Instrument.

15.4.3 Properties: Function Page

Selecting a Send Module

For each Send Module inserted in the same Instrument a list entry is created (shown in the figure below). The list entry name will be created according to the Label of the corresponding Send module.

► To choose a Send Module from which to receive the incoming signal, choose the corresponding menu entry from the Connect drop-down menu, as shown in the screenshot below. The "wireless" connection is created right after choosing the signal source. Note that only compatible connections show up in the Connect drop-down menu.



The Signal Sources List

The signal sources list contains the following columns:

- **#**: This column indicates the position of the Send Module in the list. The Default Value refers to this number.
- **Label**: This column displays the name of the Send Module. If you rename a Send module the Label in the list will be updated.
- **State**: This column indicates if a connection to this Send Module is possible. Only establish a connection if this field displays OK. Note that the "State" corresponding to "Send 3" shows "#Err A->E", indicating that a connection to the Send 3 Module is incompatible since "Send 3" sends an Audio signal, but the output port of the Receive Module at hand has already been connected to an Event input port.

- **Use**: This column allows you to set the position of the list entry in the Receive Module's Panel representation. "0" corresponds to the first entry, "1" to the second, and so on. To remove an entry from the Panel representation, double-click he corresponding "Use" value and press the Del or Backspace key on your computer keyboard.

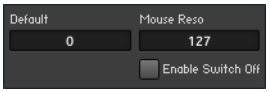
Automatic Adding of New Send Entries to Panel's List

► To automatically add entries for newly created Send Modules to a Receive Module's Panel representation, engage the Automatic Use of new Sends checkbox, shown in the figure above.

Setting the Mouse Sensitivity of the Spin Panel Representation

If you have chosen the Spin Panel representation in the View page, you can switch between signal sources by clicking and dragging the Spin menu. For this, you might want to change the sensitivity of the Spin menu in respect to your mouse movements.

► To set the Spin Panel representation to have low mouse sensitivity, set the Mouse Reso edit field in the Function page to a high value, perhaps “500”. For high sensitivity, you might set the Mouse Reso edit field to “10”. The Mouse Reso edit field is shown in the figure below.



Setting the Default Signal Source

The Default Value is used whenever an Initialization of the control happens. In this case the entry in the signal sources list that has the same value for "#" as the Default Value will be chosen as the signal source.

► To set the Default Value, enter the integer corresponding to the desired default signal source's "#" value into the Default edit field (shown in the figure above).

Adding the Switch Off List Entry

You can also create a menu entry in the Panel representation's signal sources list that corresponds to receiving no signal from any of the Send Modules.

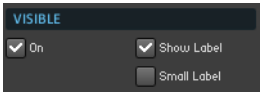
► To insert a menu entry to the Module's Panel representation that corresponds to switching the receive functionality off, engage the Enable Switch Off checkbox, shown in the figure above. A menu entry labeled "Off" will appear in the Receive Module's Panel representation.

15.4.4 Properties: View Page

Activating the Receive Module's Panel Representation

The Receive Module allows you to choose the signal source from the Instrument Panel. For this you need to make the Panel representation visible.

► To activate the Receive Module's Panel representation, go to the View page and engage the On checkbox, shown in the figure below.



Choosing the Receive Module's Panel Representation

You can choose between four styles for the Receive Module's Panel representation. This is done by choosing the desired menu entry from the Style drop-down menu in the Module's View page, as shown in the figure below. The following styles are available:

- **Button:** Each signal source entry creates a button. All buttons will be arrayed vertically in the Instrument Panel. The currently selected signal source's button will be displayed in the Indicator color of the Instrument (please refer to section 8.3 in the Application Reference for more information on changing the different colors of the Instrument). The size of the buttons can be adjusted in the Module's View page.
- **Menu:** The signal sources are listed in a drop-down menu on the Instrument Panel.
- **Text Panel**: Each signal source has an entry in a list on the Instrument Panel. The list displays several entries at once. If you have created more entries than are able to fit into the text panel display specified by the Width and Height edit field in the View page, you will get scrollbars in the panel.
- **Spin:** Each signal source has an entry in a list on the Instrument Panel. The list only displays the currently selected signal source. You can switch through the list of signal sources using a + and a - button to the right of the list entry in the Panel representation.

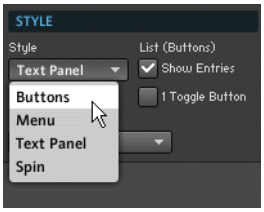
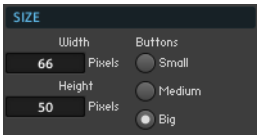


Fig. 15.6 Use the Style drop-down menu to choose the Panel representation of the Receive Module.

Changing the Size of the Snapshot Module's Panel Representation

Depending on the size of your signal sources list, you might need a small or large area for the Receive Module's Panel representation.

► To change the width and height of the Receive Module's Panel representation, go to the Module's View page and enter the desired values into the Width and Height edit fields, as shown in the figure below.



For more detailed information on editing the Receive Module's Panel appearance using the View page, please refer to subsection 8.2.3 in the Application Reference. Also, custom control styles can be created via skins. Please refer to section 8.5 in the Application Reference for more information on how to do this.

15.4.5 Example: Send and Receive within an Instrument

This example briefly outlines how to create a connection between Send (↑15.3, Send) and Receive Modules (↑15.4, Receive). First, let's say you have three signal sources coming from a Macro's output ports (shown in the Structure below). To be able to use these sources with Receive Modules elsewhere in the Instrument, place a Send Module for each signal into the Structure. Rename the Send Modules as you like.

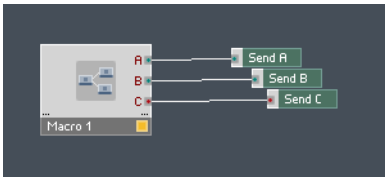


Fig. 15.7 Fetch signals for using Send Modules.

Navigate to the place where you wish to use one or more of these signals connected to Send Modules and insert a Receive Module (↑15.4, Receive), as shown in the figure below.

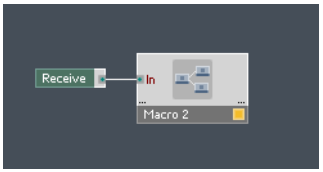


Fig. 15.8 Signals connected to Send Modules elsewhere in the Structure can be retrieved using the Receive Module.

For the example shown, three sources are available for the Receive Module. These sources can be chosen in the Receive Module's Function page, depicted in the figure below. Note that the signal types should be compatible between the chosen Send Module (↑15.3, Send) and the Receive Module.

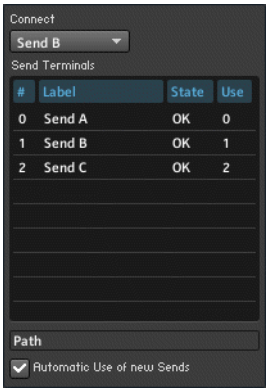


Fig. 15.9 Use the Receive Module's Function page to choose which Send Module's signal it should output.

15.5 IC Send

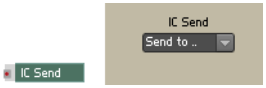


Fig. 15.10 IC Send Module

15.5.1 Overview

The IC Send Module transmits Monophonic Event signals to any Module in the Ensemble capable of receiving IC (Internal Connection) signals. Such Modules include IC Receive Modules, but also various Panel elements such as knobs and switches. The IC Send Module has a Panel display allowing connections to be configured from the Instrument panel. All Modules in the Ensemble capable of receiving IC signals will appear in this Panel representation, except those with the No Entry in IC Send Menu checkbox in their Function page engaged. Connections can also be established in the Connections page of the Modules to be connected.

Application

Internal Connections work globally. This means that the IC Send Module can be used to make wireless connections between different Instruments within the Ensemble. Connections between Send (↑15.3, Send) and Receive Modules (↑15.4, Receive) ↑15.6, IC Receive(described above) are limited to the bounds of the parent Instrument. The advantage of the IC protocol is that connections between parts of different Instruments can be made and between a number of Modules, not only IC Send and IC Receive Modules (↑15.6, IC Receive) (as is the case for Send and Receive Modules). However, the IC protocol only supports Event streams whereas Send and Receive Modules can transmit Audio signals as well. Use the IC protocol to transmit control messages between instruments ranging from macro control parameters for several knobs and faders to Monophonic modulation sequences.

► To choose a destination to which to route the Event signal at the IC Send Module's input port, click on its Panel representation and select the desired destination from the drop-down menu (shown below).

Fig. 15.11 The incoming signal at this IC Send Module has been routed to two faders in the Master Instrument one of which controls the Master Level the other the Master Tune parameters (as indicated by the fader Labels).

The menu is sorted into submenus, one for each Instrument in the Ensemble. Selected menu entries have a dot next to them. As can be seen in the screenshot above, you can select several menu entries. Clicking on a selected menu entry again disconnects the IC Send Module from the corresponding destination. The screenshot below shows how the two Internal Connections created above are reflected in the IC Send Module's Connections page.

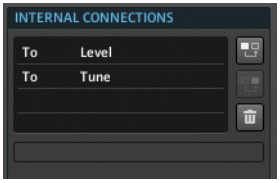


Fig. 15.12 The Internal Connections area of the IC Send Module shown in the previous figure indicates that two Internal Connections have been made to faders labeled "Level" and "Tune", respectively. These are both outgoing connections.

Please refer to section 14.2 in the Application Reference for more information on the IC Send Module's Function page and in particular for an example on how to create Internal Connections from the Connections page of a compatible Module.

15.5.2 Ports

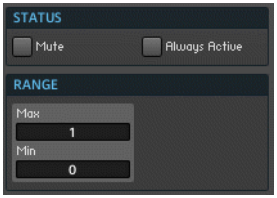
- **(In)** "In" is the Event input port for the Event signal to be sent to an IC compatible Module anywhere in the Ensemble.

15.5.3 Properties: Function Page

Keeping the IC Send Module Always Active

If the IC Send Module is not connected to an active signal flow, values at its input port are not sent to the chosen destination. Connecting its input port directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate IC Send Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.



Please refer to subsection 14.2.2 in the Application Reference for an explanation on how ranges are treated in the IC protocol and how the Min and Max edit fields of the IC Send Module affect IC transmission.

15.5.4 Properties: View Page

Visibility of the Panel Representation

The IC Send Module's Panel representation consists of a drop-down menu and a picture.

► To make the whole Panel representation disappear, disengage the On checkbox, shown in the figure below.

► To make the Label above the drop-down menu disappear, disengage the Show Label checkbox, shown in the screenshot below.

► To make the drop-down menu disappear, independent of the Label, disengage the Show Picture checkbox, also shown in the figure below.

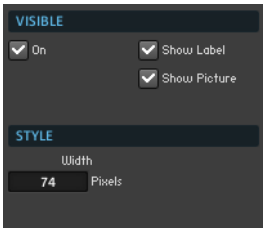


Fig. 15.13 The Visible area of the IC Send Module's lets you choose the visibility of the Label and drop-down menu separately. The Width edit field lets you choose the width of the drop-down menu in pixels.

Choosing the Size of the Drop-Down Menu

Depending on the number of characters that you wish to be visible in the IC Send Module's drop-down menu, you might want to increase or decrease the width of the Panel representation.

► To set the width of the IC Send Module's Panel representation, enter the desired value in pixels into the Width edit field, shown in the figure above.

15.6 IC Receive

Fig. 15.14 IC Receive Module

15.6.1 Overview

The IC Receive Module is used to transmit Monophonic Event signals from any Module in the Ensemble capable of sending IC (Internal Connection) signals. Such Modules include IC Send Modules (↑15.5, IC Send), but also various Panel elements such as knobs and switches. The IC Receive Module is typically used together with IC Send Modules. In such a case the connection is usually made using the IC Send Module's Panel representation (shown in its entry in the Module Reference). However, connections to other IC compatible Modules can also be established in the Connections page of the Modules to be connected.

Application

Internal Connections work globally. This means that the IC Receive Module can be used to make wireless connections between different Instruments within the Ensemble. Connections between Send Modules (↑15.3, Send) and Receive Modules (↑15.4, Receive)are lim-

ited to the bounds of the parent Instrument. The advantage of the IC protocol is that connections between parts of different Instruments can be made and between a number of Modules, not only IC Send and IC Receive Modules (as is the case for Send and Receive Modules). However, the IC protocol only supports Event streams whereas Send and Receive Modules can transmit Audio signals as well. Use the IC protocol to transmit control messages between instruments ranging from macro control parameters for several knobs and faders to Monophonic modulation sequences.

Please refer to section 14.2 in the Application Reference for an example treating how to create Internal Connections from the Connections pages of Modules to be connected as well as how ranges set in the IC Module's Function page are processed.

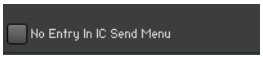
15.6.2 Ports

- **(Out)** "Out" is the Event output port for the IC signal incoming from an IC Send Module (↑15.5, IC Send) or another source configured in the IC Receive Module's Connections page.

15.6.3 Properties: Connections Page

All knobs, faders, and switches automatically appear in the IC Send Module's (↑15.5, IC Send) menu in the Instrument Panel. Luckily you can keep unwanted entries from appearing in the IC Send Menu.

► Engage the No Entry in IC Send Menu checkbox in a Module's Connections page (shown below) to keep its menu entry from appearing in the IC Send menu.



The No Entry In IC Send Menu checkbox is found in the Connections page of all IC compatible Modules except the IC Send Module.

15.7 OSC Send

Fig. 15.15 OSC Send Module

15.7.1 Overview

The OSC Send Module is used to transmit incoming Event signals as OSC messages. These OSC messages can either be sent to other Modules in your Ensemble (Panel elements, for example) or to one or more external OSC devices, including other computers. In the Connections page you can use the OSC Target and OSC Source drop-down menus (shown in the figure below) to select Modules or external devices to which to send and from which to receive OSC signals. Established OSC connections show up in the same display in the Connections page as Internal Connections. . This is shown in the figure below, which also serves as an example of how to connect an OSC Send Module placed in the Structure. Use the trashcan icon to delete existing connections. The OSC send and receive settings in the Connections pages of other OSC compatible REAKTOR Modules are the same as for the OSC Send and OSC Receive Modules.

Fig. 15.16 Established OSC Connections appear in the Internal Connections area of the Connections page.

In order for the desired destinations and sources to appear in the OSC Target and OSC Source drop-down menus, you need to configure REAKTOR's OSC Settings. Please refer to section 13.4 in the Application Reference to learn how to do this and how to set up an external OSC device with REAKTOR.

The OSC Send Module supports dynamic input ports. When all existing input ports are connected, new input ports can be added by wiring to the three dots in the input port area while holding down the Ctrl key in Windows (Cmd key in Mac OS X). The maximum number of input ports is 10.

When the OSC Send Module receives signals at more than one input port, it sends OSC messages with multiple arguments. For example, if you wire the "MX" and "MY" output ports of an XY Module (↑1.14, XY) to two input ports of the OSC Send Module, each outgoing OSC message from the OSC Send Module will carry both the "MX" and "MY" value. When recollecting such multiple arguments with an OSC Receive Module, you must create

the same number of output ports as is the multiplicity of the incoming OSC message: one for each argument. The maximum number of arguments for the OSC protocol is 10. So naturally, it is the maximum number of input and output ports for the OSC Send and OSC Receive Modules as well.

Application

Use the OSC protocol to set up communication between REAKTOR and other computers and OSC devices. You can even have two separate instances of REAKTOR running on different computers in different parts of the world and control one from the other. OSC is also a convenient communication protocol for multi-touch devices and other advanced controllers.

15.7.2 Properties: Function Page

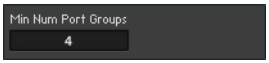
Keeping the OSC Send Module Always Active

If the OSC Send Module is not connected to an active signal flow, values at its input port are not sent to the chosen destination. Connecting its input port directly to a Knob Module, for example, does not suffice. In such a case you can activate the Module with the Always Active checkbox.

► To activate OSC Send Module, go to its Function page and engage the Always Active checkbox, shown in the figure below.

Changing the Minimum Number of Input Ports

► To change the minimum number of input ports, type the desired number into the Min Num Port Groups edit field (shown in the figure below). The maximum number of input ports for the OSC Send Module is 10.



15.8 OSC Receive

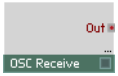


Fig. 15.17 OSC Receive Module

15.8.1 Overview

The OSC Receive Module is used to transmit incoming OSC signals to the Structure Events. Incoming OSC messages can either stem from other Modules in your Ensemble (Panel elements, for example) or from one or more external OSC devices, including other computers. In the Connections page you can use the OSC Source and OSC Target drop-down menus (shown in the figure below) to select Modules or external devices from which to receive and to which to send OSC signals. Established OSC connections show up in the same display in the Connections page as Internal Connections. This is shown in the figure below, which also serves as an example of how to connect an OSC Receive Module placed in the Structure. Use the trashcan icon to delete existing connections. The OSC send and receive settings in the Connections pages of other OSC compatible REAKTOR Modules are the same as for the OSC Send and OSC Receive Modules.

Fig. 15.18 Established OSC Connections appear in the Internal Connections area of the Connections page.

In order for the desired destinations and sources to appear in the OSC Target and OSC Source drop-down menus, you need to configure REAKTOR's OSC Settings. Please refer to section 13.4 in the Application Reference to learn how to do this and how to set up an external OSC device with REAKTOR.

The OSC Receive Module supports dynamic output ports. When all existing output ports are connected, new output ports can be added by wiring to the three dots in the output port area while holding down the Ctrl key in Windows (Cmd key in Mac OS X). The maximum number of output ports is 10.
The OSC Receive Module can receive OSC messages with multiple arguments. In this case you must create the same number of output ports as is the multiplicity of the incoming OSC message: one for each argument. The maximum number of arguments for the OSC protocol is 10. So naturally, it is the maximum number of input and output ports for the OSC Send and OSC Receive Modules as well.

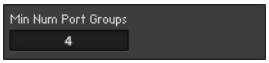
Application

Use the OSC protocol to set up communication between REAKTOR and other computers and OSC devices. You can even have two separate instances of REAKTOR running on different computers in different parts of the world and control one from the other. OSC is also a convenient communication protocol for multi-touch devices and other advanced controllers.

15.8.2 Properties: Function Page

Changing the Minimum Number of Output Ports

► To change the minimum number of output ports, type the desired number into the Min Num Port Groups edit field (shown in the figure below). The maximum number of output ports for the OSC Receive Module is 10.



Index

Numerics

A

B

S

T

U

V

X