TASSMAN

USER MANUAL

A A S

Applied Acoustics Systems

Information in this manual is subject to change without notice and does not represent a commitment on the part of Applied Acoustics Systems DVM Inc. The software described in this manual is furnished under a license agreement. The software may be used only in accordance of the terms of this license agreement. It is against the law to copy this software on any medium except as specifically allowed in the license agreement. No part of this manual may be copied, photocopied, reproduced, translated, distributed or converted to any electronic or machine-readable form in whole or in part without prior written approval of Applied Acoustics Systems DVM Inc.

Tassman is a Trademark of Applied Acoustics Systems DVM Inc. Windows 98, 2000, NT, ME, XP and DirectX are either trademarks or registered trademarks of Microsoft Corporation. Macintosh, Mac OS, QuickTime and Audio Units are registered trademarks of Apple Corporation. VST Instruments, ASIO, Cubase and Cubasis are trademarks of Steinberg Soft Und Hardware GmbH. Sonar is a trademark of Twelve Tone Systems Inc. ProTools and DirectConnect are registered trademarks of Digidesign. Digital Performer and MAS are registered trademarks of Mark of the Unicorn. Logic is a trademark of Emagic Soft-un Hardware GmbH. Adobe and Acrobat are trademarks of Adobe Systems incorporated. Unauthorized copying, renting or lending of the software is strictly prohibited.

Visit Applied Acoustics Systems DVM Inc. on the World Wide Web at

www.applied-acoustics.com

Contents

1 Introduction

The *Tassman* is a modular software synthesizer based on physical modeling. The modular architecture of the software reproduces the very powerful features of vintage analog synthesizers letting you construct instruments "à la carte" by patching modules together. The module library includes many analog-type objects but also modules simulating acoustic objects and instruments. The *Tassman* makes no distinction between different object types which means that you can reproduce your favorite vintage analog synthesizer, recreate or invent acoustic instruments but also combine analog and acoustic modules and create very innovative hybrid instruments.

The *Tassman* generates sound by simulating the different modules through physical modeling. This technology uses the laws of physics to reproduce the behavior of an object. In other words the Tassman solves, in real-time, mathematical equations describing how an object functions. The Tassman uses no sampling or wavetable, it just calculates the sound as you play in accordance to the controls it receives. For example, if you choose to hit a plate with a mallet, the *Tassman* simulates (1) the impact of the mallet at a particular point, (2) the resulting displacement of the plate due to wave motion (taking into account the geometry and physical parameters of the plate related to its material), and (3) sound radiation at a particular listening point.

Physical modeling is a very general and powerful approach since the result is obtained by reproducing how an object generates sound rather than trying to reproduce the sound signal itself using, for example, wavetables, additive synthesis or samples. This implies that a module can generate very different sounds depending on the driving signals it receives. For example, different sounds will be produced by a plate of a given geometry and material, depending on the strength of the mallet impact and its impact point. It will behave differently again if you hit the plate when it is at rest or when it is already vibrating. Physical modeling takes all these parameters into account naturally since it reproduces the behavior of the real object. This results in very natural and realistic sounds and reproduces the control musicians have on real acoustic instruments.

The *Tassman* software is comprised of three tightly integrated views, in a single window. Instruments are created in the Builder by patching together modules imported from the Browser. Modules are just "building block" having inputs and outputs which you connect together using wires. The fully modular architecture of the *Tassman* lets you expand the Browser library by letting you define groups of modules as sub-patches for later inclusion in your constructions. Instruments created with the *Tassman* can be exported as short text files, which means that you can very easily exchange them with other users.

Once an instrument has been constructed, the Player is opened, displaying the instrument's controls. The Player interprets the files generated in the Builder and automatically generates the playing interface and the computational code corresponding to a particular instrument. The panels of the different modules were inspired by vintage hardware making them very easy to use. All the controls appearing on the screen can be moved with the mouse and keyboard but can also be linked to external MIDI controllers. As with sub-patches, and modules in the builder view, presets for each instrument can be easily "drag and dropped" from the browser, which means you don't have to worry about searching through an endless stream of "load" dialog boxes to find the components you need.

Before discussing the *Tassman* in more detail, we would like to take the opportunity to thank you for choosing an Applied Acoustics Systems product. We hope that you have as much fun playing with the *Tassman* as we had developing it!

1.1 System requirements

The following computer configuration is necessary to run the *Tassman*:

Mac OSX :

- Mac OSX 10.2 (Jaguar) or later.
- G3 Processor (G4 strongly recommended)
- 256 Mb RAM
- 800 x 600 or higher screen resolution
- MIDI Keyboard (recommended)
- Ethernet Port
- Quicktime 4.0

PC :

- Win98SE/2000/XP
- PIII 500 or better processor
- 128 Mb RAM
- 800 x 600 or higher screen resolution
- Direct X or ASIO supported sound card
- MIDI Keyboard (recommended)

This computer configuration will enable you to play fairly elaborate instruments. Keep in mind that the computational power required by the *Tassman* depends on the complexity of the patch you are playing and the number of voices of polyphony. Although it is not absolutely necessary it is strongly recommended to play the *Tassman* with a MIDI keyboard or controller.

1.2 Installation

Mac OSX

Insert the *Tassman* program disc into your CD ROM drive. Open the CD icon once it appears on your desktop. Click on the *Tassman* Install icon and follow the instructions of the installer.

For download installations simply click on Install icon and follow the instructions of the installer.

PC

Insert the *Tassman* program disc into your CD ROM drive. Click on the CD ROM icon on your desktop. Click on Install icon and follow the instructions of the installer.

For download installations simply click on Install icon and follow the instructions of the installer.

1.3 Registration

Upon launching the *Tassman* for the first time the registration page will appear. The *Tassman* uses a proprietary challenge/response copy protection system which requires you to provide some information before using the program. You will require:

- A valid email address
- Your product serial number (on the back of your CD or in your confirmation email for downloads)
- The challenge key supplied by the registration page

A *challenge key* is a long string of capital letters and numbers that is generated uniquely for each machine during the registration process. In other words, for each machine you install this program on you will receive a different challenge key. If you are a PC user, please note that a new challenge key will be generated should you change your computer's hard drive or upgrade your operating system. Once you have registered you will not have to do so again unless:

- You reformat or upgrade your hard drive (on PC)
- You change or upgrade your operating system (on PC)
- You uninstall the program (Mac and PC)

If the machine you have installed on is connected to the internet:

Enter your email address and serial number, and click next. The registration system will now automatically launch your internet browser and connect to the AAS unlock page. The information you have provided, along with your challenge key will be filled into the online form. Verify that this information is correct and click next. The next page asks you to provide information about yourself including your mailing address and phone number. While this part of the registration process is optional, we recommend this information be provided to ensure our support team is able to contact you to resolve any future support issues, and notify you of product updates promptly. This information is kept completely confidential.

You will then receive your response code on-screen. Copy and paste this information into the registration window, and the process is complete. Your response code will also be sent to you by email for your records.

If you wish to register your software from another computer:

Enter you email address and serial number into the registration window and click next. You will then be provided your challenge key. Take note of this information and proceed to your internet connected computer. Launch your browser and go to the unlock page of the AAS website at:

http://www.applied-acoustics.com/unlock.htm

Enter your email address, serial number, and challenge key, and click next. The next page asks you to provide information about yourself including your mailing address and phone number. While this part of the registration process is optional, we recommend this information be provided to ensure our support team is able to contact you to resolve any future issues, and notify you of product updates promptly. This information is kept completely confidential.

You will then receive your response code on-screen. Copy this information into the registration window, and the process is complete. Your response code will also be sent to you by email for your records.

If you wish to register by fax or over the phone:

Should you not have access to the internet, AAS support representatives are available to assist you in the registration process Monday to Friday, 9am to 6pm EST. You may contact us by phone at:

- North America Toll-free number: 1-888-441-8277
- Outside North America: 1-514-871-8100
- Fax Number: 1-514-845-1875
- Email: support@applied-acoustics.com

1.4 Getting started

There's no mistaking it. Getting a new piece of gear is an occasion for excitement. At a time when music technology is getting faster and more powerful by the day, however, the task of going through the process of figuring out how your new toy actually works can be a little daunting to say the least. Enter the realm of software and virtual instruments and you could be looking at learning the "ins and outs" of the equivalent of an entire studio full of gear, in a single piece of software! As is so often the case, the most important thing is to know yourself and the pace at which you feel comfortable working. The *Tassman* has been developed with the needs of a very wide spectrum of individuals in mind. Whether you're new to the world of soft synths and computer music production, or are a seasoned industry professional, the *Tassman* is sure to be a source of creative inspiration, and offer up hours of virtual knob twiddling fun.

When you launch the *Tassman* for the first time you'll be greeted with a familiar "file-browser-like" interface, not unlike those your operating system generates to display the contents of your hard disk, or your email program uses to organize your mail and address book. The left side of the screen contains the *Tassman*'s browser, a “tree view” organization of all the relevant components the *Tassman* uses, including:

- **Imports -** destination folder for imported instruments, presets and sub-patches.
- **Instruments -** ready to play, pre-configured groups of modules of all shapes and sizes including physical modeled acoustic instruments, electro-mechanical emulations, analog, FM synths, and more.
- **Modules -** The basic building blocks of the *Tassman* including Oscillators, Filters, etc.
- **Sub-patches -** smaller groups of pre-patched modules that are saved for use in various instruments such as your favorite effects, filter banks and EQs, etc.

This approach makes it possible for you to explore the *Tassman* at your own pace. From scanning through the included instruments and presets, to constructing your own synths from the ground up, the *Tassman* grows right along with you.

Audio and MIDI Configuration

Before you start exploring the included instruments and presets, take a moment to set up your system configuration, under the Edit drop down menu, selecting Preferences:

General Settings - This menu allows you to select whether exported folders from the browser contain the contents of any sub-folders located in their branch of the browser tree, i.e. when you export a synth, its presets and sub-patches will be included in the export with it if this option is selected. For more details on import and export functions please refer to the browser section of this manual. The General page also includes a slider which allows you to choose between smoother graphical response or better audio performance. Save preferences and the default sound file directory are also set from this menu. Finally, this menu lets you resize the window of the different plug-in versions of *Tassman* by adjusting the height and width of the window. Note that in order for these adjustments to take effect, you need to unload and then reload the plug-in.

Audio Settings - This menu allows you to select from the installed audio ports on your computer, by driver type. If you have ASIO drivers available, these should be selected for optimum performance. Multi-channel interfaces will have their outputs listed as stereo pairs.

MIDI Settings - This menu lists all of the available MIDI ports installed on your system. Select the port or ports you wish to use and click OK. *Tassman* can receive up to 16 simultaneous channels of MIDI data.

Latency Settings on PC - Audio Control Panel on Mac - This panel allows you to select the bit depth (16, 24, or 32 bit audio) sample rate (22.05, 44.1, 48, or 96 kHz) and buffer size, which affects how quickly the *Tassman* responds to the control information it receives. The smaller the buffer size, the shorter the latency, and vice versa. Why would you ever want to introduce more latency you might ask? In some situations, such as large, self generating ambient synths and other stand-alone applications, you may wish to 'trade' the resources *Tassman* normal utilizes to maintain low latency response for more raw processing power. Note that the content of the dialog depends on the driver selected in the Audio Settings menu.

ASIO Driver Configuration - Some sound cards provide their own ASIO control panel, in which case the above information will differ from card to card. Some sound-cards also require that you close all programs before making changes to the buffer size, sampling rate, or bit depth. If you discover this is the case with your sound card, please refer to the manufacturer's documentation for details on configuring it for optimum performance. Most sound card manufacturers also update their drivers regularly. It's is strongly recommended that you visit your sound card manufacturer's website regularly to ensure you are using the most up to date drivers and support software.

96k hz Sample Rate Support - It will literally take twice as much CPU power to process audio at a sampling rate of 96 kHz as it would to process the same data at 48 kHz, simply because you have twice the processing to do. As a result of this, 96 kHz support is only recommended for PIV, and G4 processor based systems running at 800Mhz or more.

Exploring the Factory Instruments and Presets

The *Tassman* comes with 50 factory instruments and around 1000 presets right out of the box, which amounts to a huge range of sounds before you've even turned a single knob. As you'd expect, the best way of coming to grips with the possibilities each synth offers is simply to go through them one at a time. Open the Instruments folder by clicking on the "disclosure" symbol to the left. This will expand the browser to reveal the folder's contents. Select the type of synth that interests you (acoustic, analog, etc) and double click on the first synth in the list. The Builder, as the name implies, shows all of an instrument's included modules and internal connections, while the Player displays its editing and performance controls. You can switch back and forth between these views from the **View** menu. Clicking on the "disclosure" symbol to the left of any instrument reveals its presets. You can switch between presets by double clicking on the preset of your choice. If you fall upon an instrument you're having trouble understanding, or would simply like to have more information about how it works, choose **Get Instrument Info** from the **Edit** menu. For detailed information on the functionalities of the Builder, Browser, and Player, please refer to the dedicated chapters on each later in manual.

As was mentioned earlier, the *Tassman* has been designed to meet the needs of a wide range of users. Similarly, the included synths and presets have been created to cover an equally wide range of tastes. Once you've had a chance to explore the included synths in some detail, you may find that some of them produce sounds you feel you will use very rarely in your work, or simply aren't quite your style. The Browser makes it easy to organize your synths and presets in whatever manor you choose. Click in the browser, and choose **New Folder** from the **File** menu. Name this folder "Archive". You can now place all those "specialty" synths in the archive, freeing up space in your instruments folder and making it quicker and easier to find the sounds you need while you work.

Building your Own Instruments

One of the *Tassman*'s greatest strengths is its modularity. As you explore the various factory instruments and presets, ideas for your own creations are sure to come up. The tutorials section of the manual provides an excellent basis for getting your ideas off the ground, and coming to grips with the basic functionalities of the *Tassman*'s Builder. Regardless of your knowledge of synthesis and previous experience with modular environments, we recommend that you at least scan the tutorials to learn about the basic conventions of the builder. By answering a few key questions before beginning your first constructions, you will be able to spend less time pondering the near infinite possibilities the *Tassman* offers, and more time making music.

Does one of the included instruments contain some of the elements of the idea you have in mind?

You can easily duplicate groups of modules used in one synth in another using the **Copy** and **Paste** menu commands. If there is a group of modules you find you're using quite frequently in your constructions, copy and paste it to a new instrument, add the necessary inlets and outlets from the **In/Out** folder of the browser, and save it as a new sub-patch.

Have you checked the sub-patch folder for elements you want to include in your synth?

The *Tassman* comes with a large collection of sub-patches ranging from various oscillator and filter configurations, common output setups, and effects chains. You don't have to reinvent the wheel every time you sit down to build a new instrument. Save time, use a sub-patch. Is this synth going to be used as a plug-in?

If you plan to use your synth as plug-in, a well thought out, efficient design will provide better performance when it's running along side several other audio and MIDI tracks. Even if you're using a top of the line system with the fastest processor on the market, efficient patch design means your instruments will run more smoothly in plug-in mode, and take less of your system's resources. Does your patch really need 16 reverbs !?

For detailed instructions on building your first synths, please refer to the Tutorials section of the manual. A detailed description of each module's functionalities is also provided in chapter 6.

1.5 Using the *Tassman* as a Plug-in

The *Tassman* integrates seamlessly into the industry's most popular multi-track recording and sequencing applications as an instrument plug-in, making it easy to use the factory synths, as well as your own instruments and presets in your arrangements. Below you'll find step by step instructions on how to open the *Tassman* as plug-in in several of these applications. Instructions have been provided for the most recent versions of each product. For further details on using virtual instruments, please refer to the documentation of your host sequencer.

Opening *Tassman* in Cakewalk's Sonar as a DXi plug-in

- Start Sonar,
- Select **DXi Synth** > **Tassman** from the Insert menu,
- The **Insert DXi Synth Options** dialog will appear, click OK,
- The *Tassman* window will appear.

Opening *Tassman* in FL Studio as a DXi or VST plug-in

- Start Fl Studio,
- Select **More ...** from the Channel menu,
- Click the **refresh** button and select **Fast Scan (Recommended)**,
- Double-click on *Tassman* either from the **VST plug-ins** or **DXi plug-ins** categories,
- The *Tassman* window will appear.

Opening Tassman in MOTU's Digital Performer as an Audio Units plug-in

- Start Digital Performer,
- Create or open a project,
- Select **Add Track Instrument Track** > **Tassman** from the **Project** menu,
- The *Tassman* window will appear,
- Click the output of a MIDI channel, select *Tassman* and the MIDI Channel; make sure that the MIDI channels are matching.

Opening Tassman in Digidesign's ProTools using RTAS

- Start Protools,
- Create or open a project,

- Create a Stereo Audio Track,
- In the mix window, on the newly created track, click on the first insert button and choose **multi-channel plugin** > *Tassman*,
- Create a MIDI Track,
- In the mix window, on the newly created track, assign the MIDI output to *Tassman*; make sure that the MIDI channels are matching.

Opening Tassman in Emagic's Logic Audio a VST plug-in

- Start Logic,
- Create or open a project,
- In the main window of Logic, select an **audioinst** track by double-clicking on it. This will bring up the mixer,
- Choose the track you want to appear on, hold-click on the inserts button and select **Stereo** > **VST** > *Tassman*.

Opening Tassman in Emagic's Logic Audio an Audio Units plug-in

- Start Logic,
- Create or open a project,
- In the main window of Logic, select an **audioinst** track by double-clicking on it. This will bring up the mixer,
- Choose the track you want to appear on, hold-click on the I/O button and select **Stereo** > **Audio Units** > **Applied Acoustics Systems** > **Tassman**.

Opening Tassman in Steinberg's Cubase SX as a VST plug-in

- Select **VST Instruments** from the **Devices** menu,
- Select *Tassman* from the drop down menu,
- Create a MIDI Track,
- Assign the output of the MIDI track to *Tassman*.

1.6 Getting help

Applied Acoustics Systems technical support representatives are on hand from Monday to Friday, 9am to 6pm EST. Whether you've got a question regarding a new synth you are building, or need a hand getting the Tassman up and running as a plug-in in your favorite sequencer, we're here to help. Contact us by phone, fax, or email at:

- North America Toll Free:1 888 441 8277
- Worldwide: 1 514 871 8100
- Fax: 1 514 845 1875
- Email: support@applied-acoustics.com

Our online support pages contain downloads of the most recent product updates, and answers to frequently asked questions on all AAS products. The support pages are located at:

www.applied-acoustics.com/faq.htm

The AAS community site also contains the Tassman user forum, an excellent place to meet new users and get answers to your questions, as well as the synth download page, tutorials, and MP3s create with AAS products. The community site is located at:

www.applied-acoustics.com/community.htm

1.7 About this book

This User Manual begins with a tutorial to help you learn quickly how to create and play instruments with the Tassman. Four examples are included in the tutorial and every patch presented in the tutorial has been pre-constructed for you. You can find the corresponding files in the **Tutorial** folder of the Tassman browser. The Tassman comes with a certain number of pre-patched instruments and presets. We strongly recommend that you have a look at these instruments as a complement to the tutorial.

Chapters 5, 3 and 4 of this manual describe the Tassman Browser, Builder and Player respectively. Chapter 6, a reference guide, contains a description of every module included in the Tassman. Builder and Player menus and shortcuts are listed in Chapter 7.

2 Tutorials

The following tutorials will teach you the basics of constructing and playing synthesizers with the *Tassman*. We recommend that you build the different synthesizers from scratch as you go along the different examples. If you have any problem, you can find the patches described in the different tutorials in the **Tutorials** folder under the Instruments folder in the *Tassman* Browser. The *Tassman* comes with many pre-constructed instruments and presets. We strongly recommend that you have a look at the patches of these instruments for more elaborate examples.

2.1 Tutorial 1. A Simple Analog Synth

In this first example we will build a simple analog type synthesizer. You will learn to:

- Select modules.
- Connect modules.
- Switch to the Player view.
- Use modulation signals.
- Delete modules and wires.
- Monitor the output of an instrument.
- Save an instrument.

Open the *Tassman* by clicking on its icon or from the **Start** menu. The Builder contains three different parts. The main section of the Builder is the construction area on which you will make your patch. The Browser at the left, contains all the folders needed in *Tassman*. You will find the **Imports**, **Instrument**, **Modules** and **Sub-patches** folders.

The different modules that you can assemble appear on the left in the Browser and are listed under the different headings of **Effects**, **Envelopes**, **Filters**, **Generators**, etc . . . Just above the construction area is a help area that will give you basic information on the currently selected module.

Step 1: VCO and Audio Out

Description

We will construct what is probably the simplest synthesizer one can build. The basis of our first synth is a **VCO** (Voltage Controlled Oscillator). This module is a wave generator and constitutes the sound source in our example. To hear the output of the **VCO** we will connect it to an **Audio Out** module, which represents the output of your sound card. This module takes the digital signal produced by the **VCO** and sends it to the computer sound card so that it can be heard. It is, in fact, necessary to have one **Audio Out** in any instrument you make if you want to hear it.

Construction

- In the **Generators** section, click-hold on the **VCO** and then drag it in the construction area. A **VCO** module then appears in the construction area. You can select it by dragging the icon and placing it anywhere you want in the construction area. Note that the module has three inputs and one output. You can have some information on the use of these inputs and outputs by positioning the mouse over them on the icon. You can also find basic information on the currently selected module in the help area located above the construction area.
- Select an **Audio Out** module in the **Outputs** folder of the **In/Out** section and drag it in the construction area. Note that this module has one input and no output.
- Now we need to connect the two modules together so that the **VCO** output signal can be sent to the **Audio Out** module. Click on the **VCO** output and move the mouse toward the **Audio Out** module and you will see a wire appear in the construction area. Now click on the **Audio Out** input and the two modules will be connected as shown in the following figure.

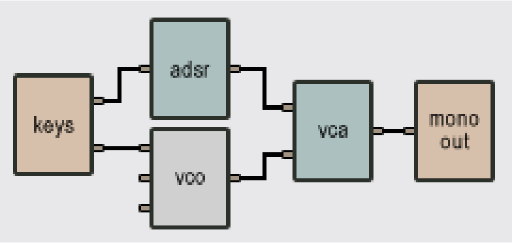


Figure 1: Tutorial 1, step 1

Playing

To play and hear the instrument you need to display the *Tassman* Player view. In the **View** menu choose the **Show Player** command which will switch to the *Tassman* Player view. You can now see the playing interface of the two modules you have connected in the Builder. You can turn off your new synth by clicking on the switch of the **Audio Out** module.

You can now play with the **VCO**. To change the frequency of the sound generated by the **VCO**, move the *coarse* or *fine* knob. There are many different ways to move a knob. First select it by clicking on it and, keeping button down, move the mouse upwards or downwards. Once the knob has been selected, you can also move it by using the arrow keys. To position the cursor in a particular spot, **Shift-click** (**Option-click** on Mac) on the circumference of the knob where you want it to point. The frequency can also be varied by one or more octaves by using the range selector.

The color of the sound produced by the **VCO** depends on the waveform you choose. To change the waveform, click on the *wavetype* selector on the right of the front panel and drag the cursor up and down. You have a choice between four waveforms: sine, pulse, sawtooth, and noise. The sine wave consists of a single fundamental harmonic and is a very soft sound. The pulse wave is made by combining a fundamental and the whole harmonics series; this is very rich in tone and is good for woodwind sounds. The sawtooth wave contains all the harmonics, but its higher frequencies are softer than in the pulse wave; it is good for brass-like sounds and strings. The noise wave, as its name indicates, consists of white noise; it is good for unpitched percussion. Note that when a pulse wave has been chosen, its shape can be changed with the *PWM* (Pulse Width) knob.

Before going on to the next step, go back to the Builder by choosing **Show Builder** from the **View** menu or use the **Ctrl-T/Apple-T** shortcut.

Step 2: Adding a LFO

Description

To add some life to our synthesizer, we will use the output of a **LFO** (Low Frequency Oscillator) to modulate the input of the **VCO**. A **LFO** is an oscillator which generates sine, triangle, square and random waves with a frequency varying between 0.1 Hz and 35 Hz. The frequency of a **LFO** signal is so low that it cannot be heard; a **LFO**, therefore, is not used to produce sound but rather to generate control signals which modulate other signals. In our patch, we will use this signal to vary the frequency of the **VCO** (i.e., to produce a vibrato effect).

Construction

- In the **Generators** section of the Browser, choose the **LFO** module and connect its output to the first input of the **VCO**. This input is a pitch modulation signal. This means that the pitch variation will follow the shape of the input signal.

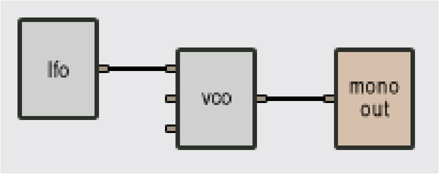


Figure 2: Tutorial 1, step 2

Playing

Now switch back to the *Tassman* Player. To hear the effect of the **LFO**, turn the *mod1* knob of the **VCO** to the right and you will start to hear the frequency varying. The *mod1* knob is simply a gain knob that adjusts the effect of the input signal by multiplying it by a gain. When the knob is fully turned to the left, the gain is zero, which means that the input has no effect. As you turn the knob to the right, the amplitude of the modulation signal affecting the **VCO** increases so that you hear a deeper vibrato. The frequency variations of the vibrato are relative to the settings of the *coarse* and *fine* knobs on the **VCO** panel. The speed of the vibrato can be adjusted with the *frequency* knob on the **LFO** panel, the oscillation of the red LEDs on the panel giving you an indication of the speed of the vibrato. As with the **VCO**, the shape of the output waveform of a **LFO** can be varied with the *wavetype* selector. Try the different waveforms and you will hear how the frequency variations of the **VCO** follow the modulation signal.

Note

The *mod2* knob is a gain knob for the second pitch modulation input of the **VCO**. This input works exactly like the first one, which means that you can use two signals to modulate the pitch variations of the **VCO**. The resulting modulation signal is the sum of the two inputs. The third input signal modulates the pulse width of the pulse wave relative to the setting of the *PW* knob. Try connecting the output of a **LFO** to this input and then vary the frequency of the **LFO** to hear the change in the waveform.

Step 3: Adding a Filter

Description

Filters are important elements of synthesizers. They are used to color the sound by altering its harmonic content. Now, instead of sending the output of the **VCO** directly to the **Audio Out**, we will first filter it with a low-pass filter. This type of filter, as the name suggests, filters out high frequency components from a sound so that only frequencies below a so-called *cutoff* frequency are able to escape.

Construction

- In the **Filters** section, select a **Vlowpass2** filter and drag it into the construction area.
- You must now disconnect the **VCO** from the **Audio Out**. To do so, click on the corresponding wire to select it and then delete it by pressing on the **Del** or **BkSp** key (**delete** key on Mac) of your computer keyboard.
- Place the filter between the **VCO** and the **Audio Out** and then pull one wire between the **VCO** output and the first **Vlowpass2** input and another one from the **Vlowpass2** output to the **Audio Out** input.

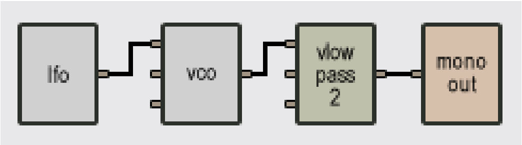


Figure 3: Tutorial 1, step 3

Playing

Vlowpass2 stands for “variable second-order low-pass filter”. This means that the cutoff frequency of the filter can be controlled with an external modulation signal. It can, however, also be adjusted with the *cutoff frq* knob on the filter panel. Launch the Player and move the *cutoff* knob to the right and you will hear the sound become brighter as the cutoff frequency increases (to hear the effect you can use any waveform from the **VCO** except the sine wave, which only has one frequency component). When this knob is turned completely to the left there is no sound (since the cutoff frequency becomes so low that all the components are filtered out from the sound). When the knob is fully turned to the right, the full spectrum of sound is heard.

The *resonance* knob is used to enhance the frequency components near the cutoff frequency. If you turn this knob to the right and then slowly tweak the *frequency* knob, you will hear the different frequency components of the sound come out as the cutoff frequency is equal to these frequencies. Try changing waveforms and playing with the *frequency* knob to hear the differences in harmonic content between all waveforms. If, when working with a sine wave, you adjust the cutoff frequency close to the signal frequency and then increase the filter resonance, the filter will act more like an amplifier than a filter. Interesting effects are achieved as the resonance rises and the filter starts to self-oscillate and the sound to saturate. Filters can be effectively used with the noise waveform because of all the harmonics they contain (try it with a lot of resonance).

Note

Modules can be deleted in the construction area the same way as the wires can be. Click on a module to select it and then use the **Del** or **BkSp** (**delete** key on Mac) key on your computer keyboard.

Step 4: Modulating the filter

Description

As was the case with the **VCO**, the **Vlowpass2** filter has a modulation input that can be used to modulate the cutoff frequency of the filter.

Construction

- Pull a wire from the output of the **LFO** that we already have in the construction area to the second input of the **Vlowpass2** filter.

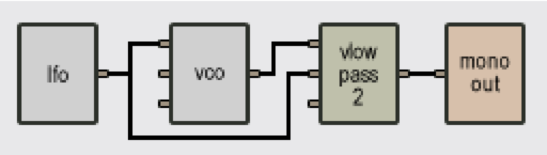


Figure 4: Tutorial 1, step 4

Playing

The amplitude of the modulation signal is controlled with the *mod1* gain knob on the filter panel. As you turn this knob to the right you will start hearing the effects of the cutoff frequency variations. This cutoff frequency increases with the amplitude of the modulation signal. The filter also has a second modulation input which modulates the cutoff frequency. The resulting modulation signal is the sum of the two signals modulated by their respective gain values.

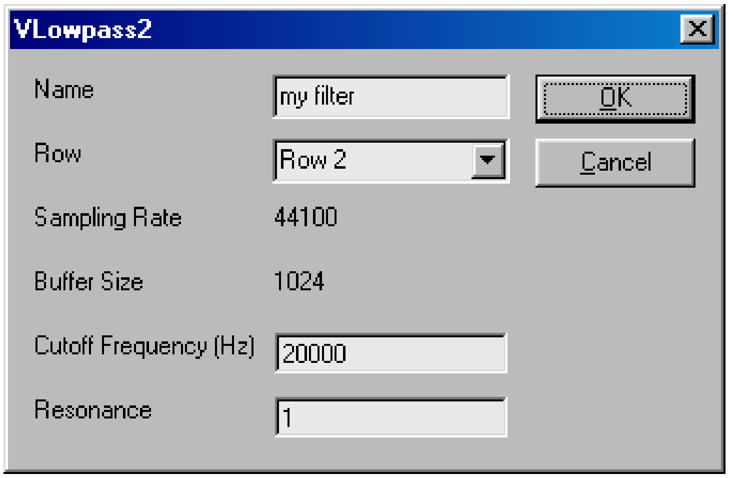
Step 5: Adjusting default module parameter values

Description

You have probably noticed that when you switch to the Player, the different modules appear with their panel controls adjusted to specific values. In this example we will change the default cutoff frequency, the name of the **Vlowpass2** filter, and the location of its control panel on the Player. Keep in mind that you can apply the same procedure to all the modules appearing in the construction area.

Construction

- In the construction area, double click on the **Vlowpass2** module. A dialog box appears with certain fields that you can edit.
- Set the value of the cutoff frequency to 20000.
- Change the name of the module to “my filter”.
- Change the display row to Row2 in the display row combo box.
- Click OK to exit the edition window.



Name: my filter
Row: Row 2
Sampling Rate: 44100
Buffer Size: 1024
Cutoff Frequency (Hz): 20000
Resonance: 1

Figure 5: Tutorial 1, step 5

Playing

When you launch the Player, the filter module appears on the second row. Note that the *cutoff frq* knob on the **Vlowpass2** front panel is completely turned to the right, which means that the filter is fully open. Note also that the name appearing at the top of the front panel has been changed to "my filter". Naming modules can be very helpful when using several modules of the same type in a patch.

Step 6: Monitoring the output signal

Description

We will now add two last modules to our patch, a **Volume** and a **Level** meter. These modules do not produce sound, but are very useful for monitoring the output from a synth and they appear in practically every instrument made with the *Tassman*.

Construction

- Select the **Level** module from the **Output** folder in the **In/Out** section of the Browser and the **Volume** module from the **Envelopes** section and then place them in the construction area.
- Select and delete the wire going from the output of the **Vlowpass2** filter to the **Audio Out**.
- First pull a wire from the output of the **Vlowpass2** filter to the input of the **Volume** module and then pull two wires from the output of the **Volume** module to the inputs of the **Level** and **Audio Out** modules.

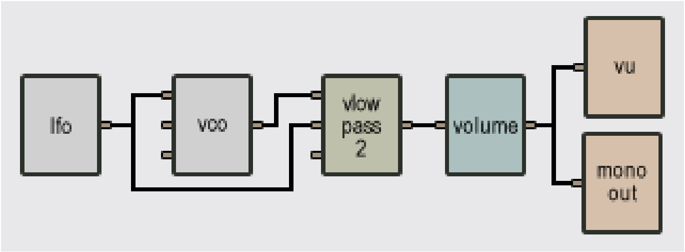


Figure 6: Tutorial 1, step 6

Playing

When sound is produced by the synth, you will see the needle of the level meter move with the amplitude of the signal. The red section of the meter indicates the saturation zone. The **Volume** slider is used to change the amplitude of the output signal from the synthesizer.

Step 7: Saving your synth

We will conclude this example by saving the instrument you have just made.

- To save a patch, use the **Save Instrument** or **Save Instrument As** command from the **File** menu. A dialog box will appear with “untitled instrument” highlighted, write the name you want your instrument to have and click OK. A new instrument with the name you wrote will appear under the Instrument section of the Browser.

You can open an instrument by double-clicking on its name in the Browser. If you want to work again on your instrument in the Builder, hit **Ctrl-T/Apple-T**.

2.2 Tutorial 2 Playing a Synth with a Keyboard

The synthesizer we have constructed in the preceding tutorial gives some interesting results but is not very convenient for playing melodies. In this second tutorial we will replace the **LFO** which controlled the **VCO** with a keyboard so that the pitch of the **VCO** changes according to the note being played on the keyboard. You will learn to:

- Use a MIDI keyboard in your instrument.
- Create envelopes.
- Link MIDI controllers to the Player interface controls.
- Create polyphonic instruments.

Step 1: Connecting a keyboard

Description

The **Keyboard** module reads and interprets control signals coming from a MIDI keyboard or host sequencer. MIDI stands for Musical Instrument Digital Interface and is a communication protocol used by most electronic musical instruments, computers and sound cards. Using MIDI, the keyboard sends messages such as the notes played, the status of the pitch or modulation wheel. The keyboard communicates with your computer through a MIDI cable connecting the MIDI output of the keyboard to the MIDI connector of the computer sound card. Some keyboards also use USB to communicate with the sound card.

It is assumed in this tutorial that you have a MIDI keyboard which can be connected to your computer. This is, of course, the best way to take full advantage of the *Tassman*. This is not a limit, however, since there are lots of things to do with the *Tassman* even without a keyboard. In the next tutorial we will replace the **Keyboard** module by a **Sequencer** module. But read this one first, even if you do not have a keyboard.

Construction

- We will first open the instrument we constructed in the last tutorial. Double-click on it in the Browser, this will open the instrument in the Player view, hit **Ctrl-T/Apple-T** to switch to the Builder view. If you did not save the preceding synth open the “Tutorial1 Step6” in the **Tutorial** folder of the instrument section.
- Select the wire linking the **LFO** to the **VCO** and delete it.
- In the module library section, select the **Keyboard** module from the **MIDI** folder in the **In/Out** section. You will notice that there are four different types of keyboard: a **Keyboard**, a **Vkeyboard**, a **Polykey** and a **Polyvkey** first select the **Keyboard** module and place it in the construction area.
- Pull a wire between the second **Keyboard** output (pitch signal) and the first input of the **VCO**.

Playing

Use your keyboard to play melodies. The **Keyboard** module we have selected behaves like a classic monophonic keyboard, which means that it can only play one note at a time. The note signal output corresponds to the highest note played when one or more keys are pressed and to the last key played when no key is pressed. In order to change the pitch of the **VCO**, remember that the *mod1* gain knob of the first modulation input of the **VCO** must be turned to the right. In its center position (green LED on), the gain is equal to 1 and the pitch variations will follow the notes you play on the keyboard. If you do not want to play an equal tempered scale, tweak the *mod1*

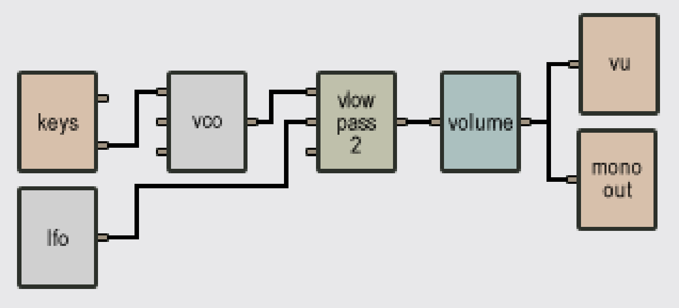


Figure 7: Tutorial 2, step 1

knob to the left for microtonal variations or to the right for larger variations. You can also use the pitch bend wheel of your keyboard to change the pitch. Note that the sound is uninterrupted even after you have released a key on the keyboard. This is because a monophonic keyboard holds the last note played. To remedy this, we will use the gate signal of the **Keyboard** module to stop the note at the right time.

Step 2: Add a VCA

Description

A **VCA** is a Voltage Controlled Amplifier. More simply said, it is a module that multiplies two signals. In order to obtain sound only when a key is pressed, we will connect the gate signal from the **Keyboard** to the first input of the **VCA**. The gate signal is simply a signal that indicates whether a note is pressed or not; its value is 1 when a key is pressed and 0 when the key is released. In this way, if we connect a second signal to the **VCA**, the output of the **VCA** will produce no sound when no key is pressed and will be equal to the second input signal when a key is pressed.

Construction

- Select a **VCA** module in the **Envelopes** section of the module library section in the Browser.
- Pull a wire between the first **Keyboard** output, the gate signal, and the first input of the **VCA**.
- Select the wire between the **Vlowpass2** output and the **Volume** input and delete it.
- Now connect the output of the **VCO** to the second input of the **VCA** and the output of the **VCA** to the first input of the **Volume**.

Playing

Make sure the *mod1* knob of the **VCO** is in its center position. You should now be able to play melodies on your keyboard with the sound going on or off as keys are pressed and released.

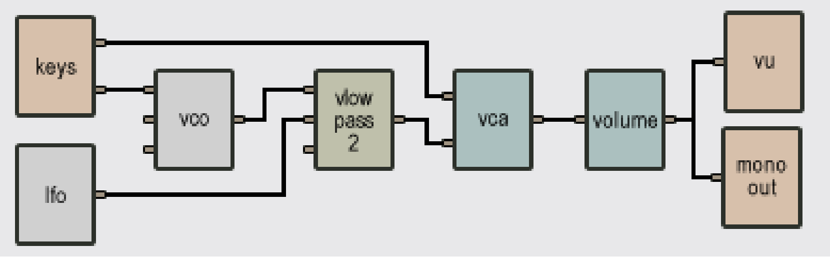


Figure 8: Tutorial 2, step 2

Step 3: Add an ADSR

Description

Now that we are able to trigger the sound with the keyboard, we would like to be able to shape the sound with different types of envelopes. To achieve this we will use an **ADSR** module (Attack, Decay, Sustain, Release). This module shapes the amplitude of a note according to the settings you chose. The Attack is the time it takes for the envelope of a sound to go from zero to its maximum value. The time it takes for the sound to go from this peak value to the sustain level is referred to as the Decay. As to the Sustain level, it is held as long as the note is held on the keyboard. And, finally, the Release time is the time the sound takes to vanish once the note has been released.

Construction

- Choose an **ADSR** module from the **Envelopes** section and place it in the construction area.
- Delete the wire that connects output 1 of the **Keyboard** (gate signal) to input 1 of the **VCA**.
- Pull a wire from output 1 (gate signal) of the **Keyboard** and connect it to the input of the **ADSR**.
- Pull a wire from the output of the **ADSR** and connect it to input 1 of the **VCA** (input 2 remains connected to the **Vlowpass2**).
- The signal to the **VCA** will now be shaped by the **ADSR** which is itself triggered by the gate signal of the **Keyboard**.

Playing

The **ADSR** can shape the amplitude in many different ways. This is one of the most important components in the characterization of a sound. For example, a piano sound has a completely different envelope than does a violin or a trumpet. They are, of course, very different in timbre too, but for now let's concentrate on the envelope. A piano has a very sharp attack, a long decay, no

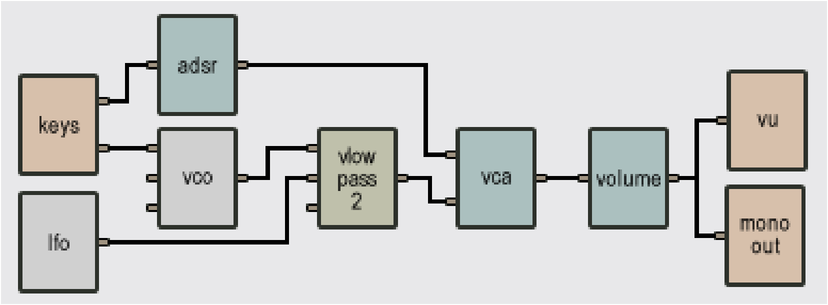


Figure 9: Tutorial 2, step 3

sustain and no release. Try to set the **ADSR** to those settings. Because there is no sustain in this example, the sound starts to decay shortly after you press keys.

Now let’s try to produce a violin-like envelope. This envelope is very different from that of the preceding example, since on a string instrument a note is held as long as the string is bowed. Since a violin’s sound doesn’t appear immediately, select a long Attack. Now choose a short Decay, a high Sustain level (since the note is held as long as the bow plays the string), and a Release that is not too long. Now the note will play until you release the key.

Step 4: Filter Modulation

Description

There are no new modules in this step, but we’ll explore other possibilities of modulation with the modules we already have. First, we’ll have the **Keyboard** modulating the **Vlowpass2**, which will cause the **Vlowpass2** to change the cut-off frequency according to the notes played on the **Keyboard**. This enables one to change the harmonic content of a sound with the pitch of the note played, which is a behavior found in many acoustic instruments (piano for example). Second, the **ADSR** will also modulate the **Vlowpass2** so that the cut-off will move according to the envelope. Again, this is a very natural acoustic and musical phenomenon.

Construction

- First select and delete the wire linking the **LFO** and the **Vlowpass2** modules.
- Pull a wire between output 1 of the **LFO** and input 2 of the **VCO** module.
- Pull a wire from output 2 (pitch signal) of the **Keyboard** and connect it to input 3 of the **Vlowpass2** filter. Note that you can pull as many wires as you want out of an output, but you can only connect one wire to an input.
- Pull a wire from output 1 of the **ADSR** and connect it to input 2 of the **Vlowpass2**.
- Save this instrument so that we can use it in Tutorial 3.

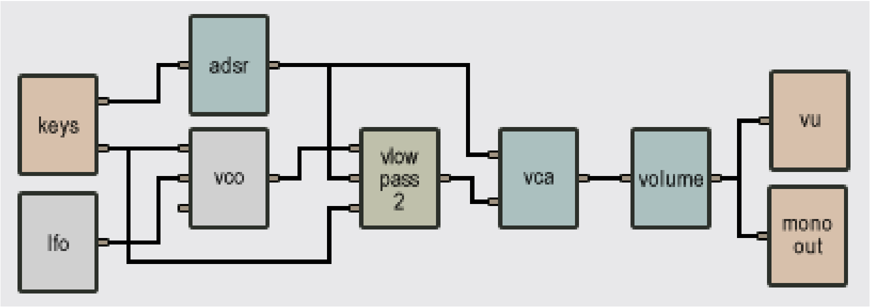


Figure 10: Tutorial 2, step 4

Playing

Try changing the parameters of the **ADSR** and hear the changes in the filter response. You can control the amount of modulation of both the **ADSR** and the **Keyboard** with the two knobs, *mod1* and *mod2*, on the filter module. Set a very slow attack and a long release on the **ADSR** to hear the filter open and close at the same speed as the **ADSR** (you have to turn the *mod1* knob to the right to hear the effect). You can hear the keyboard modulation effect by turning the *mod2* knob all the way to the right and playing low notes followed by high notes. The higher ones will have a richer harmonic content than the lower ones. You will hear the effect better if the resonance knob is turned halfway up.

Step 5: Add a MIDI controller

Description

Knobs on the Player interface can be tweaked with the mouse, although this is not the most natural way to play a synth. This method has the further limitation of allowing you to tweak only one knob at a time. *Tassman*, however, allows you to link all the controllers on the front panel of the Player to any hardware MIDI controller (such as a modulation wheel, sustain pedal, breath controller or a knob box). In this example we will control the *mod2* knob of the **VCO** (controlling the modulation signal from the **LFO**) with the modulation wheel of the keyboard.

Construction

- In the Player, right-click (control-click on Mac) on the *mod2* knob of the **VCO**, a contextual menu appears. Choose **Learn MIDI link**.
- Move the modulation wheel on your keyboard controller. This will link it to the *mod2* knob.
- To edit the MIDI link, right-click/Ctrl-click again on the *mod2* knob and select “Edit MIDI link”. This opens the Edit window for the MIDI links.

- Click on **Edit** to modify the MIDI link. You can also click on **New** to create a new MIDI link.
- The MIDI controller number specified in the **MIDI Ctrl** textbox is set by default to a value of 1. This is the MIDI controller number corresponding to the modulation wheel; you do not have to change this. In case you want to assign a new controller to the knob, specify the number here.
- You can also assign a different MIDI channel to the controller in the MIDI channel textbox. By default this value will be set to channel 1.
- There are two other parameters one can adjust: the **Minimum Value** and **Maximum Value** of the controller, which are used to limit the range of MIDI controllers. The **Minimum Value** field determines the position on the *Tassman* controller to which corresponds the minimum value sent by the MIDI controller; the **Maximum value** determines the position to which corresponds the maximum value sent by the MIDI controller. A value of 0 corresponds to the *Tassman* Player controller minimum position (left position for a knob) and a value of 1 to the *Tassman* controller maximum position (right position for a knob). Set the **Maximum value** parameter to 0.75.
- Click OK and the link appears in the list of controllers linked to this **VCO** module.
- Click OK again to confirm the change and to leave the edition window.

Playing

While you are playing, move the modulation wheel of the keyboard to control the amount of modulation of the **LFO** on the **VCO**. As you move the wheel, observe that the *mod2* knob also moves on the screen. Remember that any knob of the interface on the *Tassman* Player can be linked to any controller. You can, furthermore, link as many interface controls as you want and you can move them simultaneously.

Step 6: Creating a polyphonic instrument

Description

So far we have been using a monophonic keyboard, which means that we can play melodies but not chords. The **Polykey** or **Polyvkey** modules are used to create polyphonic instruments. Creating a polyphonic patch involves three steps. First you have to connect the polyphonic keyboard module to the other modules of the synth. You then need to connect the output of your polyphonic patch to a **Polymixer** module. Finally, you have to set the number of voices. The number of voices is set at construction time and the number of voices you will be able to run on your computer without loosing real time depends on both the power of your computer and the complexity of the instrument you are playing.

Construction

- Select and delete the **Keyboard** module in the construction area.
- Select the **Polykey** module in the MIDI folder of the **In/Out** section and place it in the construction area.
- Pull a wire between the first output of the **Polykey** module, the gate signal, and the input of the **ADSR**.
- Pull a wire between the second output of the **Polykey** module, the pitch signal, and the first input of the **VCO** and another one between the same keyboard output and the third input of the **Vlowpass2** filter.
- Select and delete the wire connecting the output of the **VCA** filter to the input of the **Volume** module.
- Select the **Polymixer** module in the **Mixer** section and place it in the construction area between the **VCA** and the **Volume** module.
- Pull a wire between the **VCA** output and the **Polymixer** input.
- Pull a wire between the output of the **Polymixer** module and the **Volume** input.
- Double-click on the **Polykey** module and set the number of voices to 4 in the **Number of Voices** textfield.

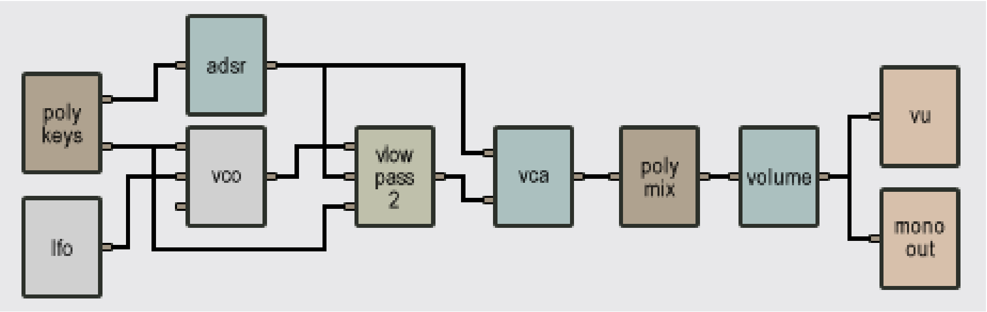


Figure 11: Tutorial 2, step 6

Playing

You can now start playing with the synth again, it now has four voices of polyphony.

Note

When you create a polyphonic patch, the *Tassman* duplicates the modules appearing between the **Polykey** and **Polymixer** as many times as there are voices of polyphony. Although only one polyphony line will be mapped on the player (since the voices all share the same controls) other modules have indeed been created. This means that the computing load can quickly become very heavy when using polyphony. If you find you are loosing real time, try reducing the number of voices. You can also try to move some elements out of the polyphonic section. For example, instead of using a filter at the end of a polyphonic section, you can connect the output of the **Polymixer** to the input of a filter.

Step 7: Playing with presets

Description

With a given synthesizer you can achieve very different sounds by choosing different settings. But once you have obtained a sound you like, you will find it very convenient to save those current settings and reuse them when you play your synthesizer again later. To save presets for the whole instrument, select the **Save Preset** or **Save Preset As** commands from the **File** menu. This will display a dialog where you can write the name for your preset. After that, the preset will be listed under its instrument in the Browser. You can recall it at any time simply by double-clicking on it or by dragging it into the Player. Note that you can also save presets for individual modules by clicking on the down arrow located in the lower left corner of each module. The presets you save for a given module can be loaded by all other modules of the same type.

Playing

We now conclude this tutorial by playing some presets that we have made for you. To try them, load the following presets from the tutorial folder in the Browser.

patch2_6_1

Our first preset gives a classic analog sound. The **ADSR** modulates the filter on which the resonance is quite high. You can hear the sweep of the harmonics as the **ADSR** modulates the cutoff frequency in accordance with its parameters (a long attack and a long release). The amount of modulation is set with the *mod2* knob on the **VCO**; here it is set at its maximum. Note that the **ADSR** also modulates the **VCA**. The vibrato in the sound is created by the modulation of the **LFO** on the *mod2* knob of the **VCO**. This *mod2* knob is set to a very low value so that the vibrato is not excessive. Try changing the setting of the *mod2* knob either manually or with the modulation wheel on the keyboard. Change the waveform on the **VCO** to noise and you will hear some sounds that resemble the wind.

patch2_6_2

Here, we have a marimba-like sound. This is achieved mainly with the **ADSR**. Note that there is no sustain in this patch, so the sound starts fading just after the notes have been pressed. You can make the decay time longer by turning the *decay* knob to the right. The **VCO** is set to a sine waveform, which is good for pitched percussion sounds. Set the **VCO** to the noise waveform and you will hear a kind of hi-hat sound. With the **VCO** set to the noise waveform, you will not hear any change in the pitch while playing the keyboard. This is because noise has no pitch.

patch2_6_3

First, note that the *mod1* knob on the **VCO** is turned so that the keyboard has no effect on the pitch change. The pitch change is caused by the keyboard modulation on the filter **VCO** (*mod2* knob). When the resonance on the filter is very high, the cutoff frequency is boosted and the cutoff itself is modulated by the keyboard (*mod2* knob) so that it follows the pitch of the keyboard.

patch2_6_4

In this sound, you can hear the fifth of any note you play. This is achieved by adjusting the cutoff on the **Vlowpass2** on the third harmonic of the fundamental (interval of a twelfth) and enhancing it by using a high resonance. The cutoff frequency of the filter is adjusted with the **Keyboard** pitch output so that the effect is transposed throughout the whole range of the keyboard. With the noise wave on the **VCO**, you can still hear the isolated harmonic and play melodies.

2.3 Tutorial 3 Using a Sequencer

In the last tutorial we played the synthesizer we constructed with a keyboard. It is possible, however, to use a sequencer instead. A **Sequencer** is a module that records note patterns and then plays them automatically. In this tutorial, we will use the same synth again as this will give us the opportunity to create a sub-patch and to include it in the *Tassman* module library. There are combinations of modules that you will be using often or even some instruments that you will want to use as a basis for constructing other instruments. To save you the trouble of connecting everything again, the *Tassman* allows you to save patches and include them in the module library. They will appear in the Browser, in the **Sub-Patches** section, like any other elementary module (such as a **LFO** or a **VCO**) and you can use them in any other assembly you make. Note that you can move them in another folder if you want. There is really no limit to the number of modules you can add! This feature enables you to take full advantage of the modular architecture of the *Tassman*. You can expand your module library as you use the *Tassman* or exchange modules easily with other users. Do not forget to have a look regularly at our web site for new sub-patches to download.

In this tutorial you will learn to:

- Create a sub-patch.
- Use a sub-patch from the Browser.
- Use a sequencer.

Step 1: Creating a sub-patch

Description

We will use the synthesizer we constructed in the first tutorial and define it as a sub-patch to use in our new patch. This operation involves four steps. First, the modules that will constitute the sub-patch must be selected. Next, the number of inputs and outputs of this sub-patch must be determined. Finally the sub-patch must be named and saved. In this example we will create a new module with two inputs and one output. Note that a sub-patch can have between 0 and 12 inputs and between 0 and 12 outputs, but must always have at least one input or one output. If it does not, it cannot be connected to any other module.

Construction

- Open the patch you saved in the first tutorial. If you did not save this synth, open the Tutorial2Step4 from the Tutorial folder in the Browser.
- Select and delete the **Polykey** module.
- Select and delete the **Polymixer** module.
- Select and delete the **Volume** module.
- Select and delete the **Level** module.
- Select and delete the **Audio Out** module.
- In the **Inlet** folder of the **In/Out** section of the module library area, select the **Inlet2** and place it in the construction area.
- In the **Outlet** folder of the **In/Out** section of the module library area, select the **Outlet1** modules and place it in the construction area.
- Pull a wire between the first output of the **Inlet2** module and the input of the **ADSR** module.
- Pull a wire between the second output of the **Inlet2** module and the first input of the **VCO** module and the third input of the **Vlowpass2** module.
- Pull a wire between the output of the **Vlowpass2** module and the input of the **Outlet1** module.
- Select the **Save as** command in the **File** menu. Note that the file will be saved in the **SubPatches** folder in the Browser.

Note

In this example we have saved the whole synthesizer as a sub-patch. Some elements of this new module could be sub-patches themselves.

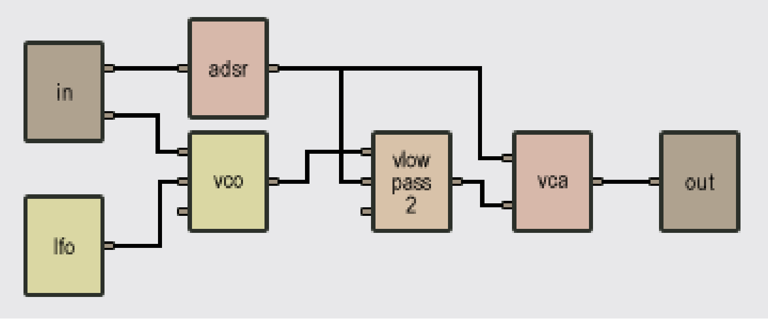


Figure 12: Tutorial 3, step 1

Step 2: Using a sub-patch

Description

The sub-patch you have just created appears in the Sub-Patch section of the Browser. It can be used just like any other module, this will enable you to connect it to other modules in the construction area. In this example, we will connect this sub-patch to a **Multi Sequencer** module.

Construction

- Choose the **New** command in the **File** menu to start another instrument.
- Drag the sub-patch you just created in the construction area.
- Select a **Multi Sequencer** module in the **Sequencers** folder and drag it in the construction area.
- Pull a wire between the fourth output of the **Multi Sequencer** module (a gate signal) and the first input of the **Sub-patch** module.
- Pull a wire between the fifth output of the **Sequencer** module (a pitch signal) and the second input of the **Sub-patch** module.
- Choose a **Level** and **Audio Out** module from the **In/Out** section and a **Volume** module from the **Envelopes** section.
- Pull a wire between the output of the **Sub Patch** module and the input of the **Volume** and another one between the output of the **Volume** and the inputs of the **Level** and the **Audio Out**.
- Save your patch for step 4.

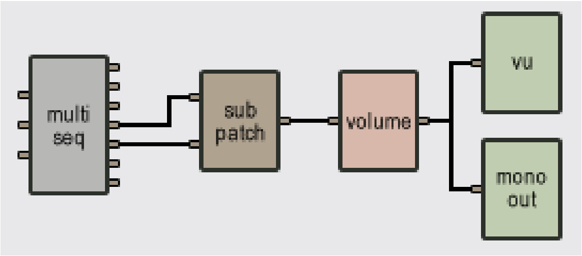


Figure 13: Tutorial 3, step 2

Playing

You have now connected the sequencer to your synth and are ready to play. In the next step you will learn to use the **Sequencer**. You can view the internal connection of the new library module by right-clicking (PC) or Ctrl+double-click (Mac) on the module in the construction area.

Step 3: Using the Sequencer

Description

In this step we will play with pre-recorded sequences.

- From the **Tutorials** folder in the Browser, select **Tutorial3** and double-click on the **Sequence1** preset. This will launch the patch in the Player.

Playing

Load presets for the **Sequencer** module by clicking on the upwards pointing arrow in the lower left corner of the **Sequencer** module, this will take you the **Multi Sequencer** module in the Browser. You can now select sequences on the keypad of the **Multi Sequencer** module. You can change them as you want. The **Sequencer** is always in loop mode (unless the *once* button is pressed), so the sequence keeps repeating itself. When the end of the current loop is reached, the new sequence will start playing. The speed or tempo of the sequences can be varied with the frequency display (the upper green window at left) on the panel. Change the settings of the controls of the different modules and experiment with the different sequences!

Step 4: Programming the Sequencer

Description

The **Multi Sequencer** module is a very complete 16-step sequencer. You must manually enter each note to be played and patterns can be up to 16 note (not every step has to play a note, some may remain silent). The 16 steps are displayed in one row, each step representing a sixteenth-note. A pattern can be of any length between 1 and 16 steps. To set the loop, click on the loop button below the step you want the loop point to be. You have four banks of sequences (A, B, C, and D), each containing eight more sequences (1 to 8) for a total of 32.You will now learn to enter your own sequences. Let's start with a simple sequence!

Programming

- Open the patch you saved in step 2 in the Player.
- Click on the gate button of the first step. The gate buttons are the green rounded below each step. When clicked, the gate button will highlight in green.
- Click-Hold on the small green display at the top of the first step (you should see C3 in the display). Drag the cursor up, the notes should start to change in the display. Select F3, this will be the note played by the first step.
- You should now hear the **Sequencer** play an F at the first step of the sequence. Note that the **Sequencer** plays as soon as the synth is open, so you do not have to turn it on. If you want to stop it, press on the *stop* button on the front panel.
- Repeat the same operation (clicking on the gate button and selecting an F3) with steps 5, 9, 13. You will now hear four Fs playing on quarter notes.
- In the pitch display of step 5, select G3.
- In the pitch display of step 9, select G#3.
- In the pitch display of step 13, select C4.
- You are now hearing a very simple melody played by the **Multi Sequencer**. Add notes to the other steps and experiment with different settings.
- Play with all the modules parameters while the sequence is playing. You can hear the changes as you make them.
- Each step can be shifted forward in time with the *shift* knob. This can create unusual rhythmic patterns.
- You can also change the playing mode of the sequence with the *mode* display. Five modes are available: forward, backwards, pendulum, random 1 (this will play the steps randomly but will repeat the random pattern, to change the random pattern click on the *reset* knob at the left of the mode display) and random 2 (this mode will play the steps randomly without ever repeating a pattern).
- The *swing* knob will introduce shuffle in the pattern.
- Use the *frequency* display to set the tempo to 72.
- To create another sequence, click 2 on the pad numbers and start again!

2.4 Tutorial 4 Playing with Acoustic Objects

We will now build an instrument with modules simulating acoustic objects such as plates, strings and mallets. The acoustic objects included in the *Tassman* library react just like their real physical counterparts. The *Tassman*, however, allows you to do things that would be impossible to achieve in real life and that, for sure, will stimulate your creativity! For example, modules can have a time-varying behavior, a mallet can get stiffer or softer, a plate can change material or geometry as you play it.

Although the acoustic modules represent objects that are completely different from those simulated by the more traditional electronic modules, the *Tassman* makes no distinction between the different type of objects. You connect together the acoustic modules exactly as we have done so far with the electronic modules. Furthermore, you can combine electronic and acoustic objects without any restrictions. Of course, you might need to experiment a little while inventing new instruments, but that is part of the fun!

In this tutorial you will learn to:

- Use acoustic modules
- Play acoustic modules
- Create sympathetic instruments

Step 1: A mallet and an Audio Out

Description

Acoustic objects such as plates, strings, beams, and membranes produce sound as a result of an excitation. This driving signal can be very short, such as the impact of a hammer on a plate, or continuous, such as when a string is bowed. In this example we will consider the excitation of a plate by a mallet.

Construction

- In the **Generators** section of the browser, select a **Noise Mallet** module and place it in the construction area.
- In the **Output** folder from the **In/Out** section, select an **Audio Out** module and connect the output of the mallet to the **Audio Out** input.
- Launch the *Tassman* Player.

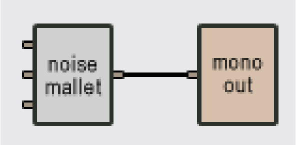


Figure 14: Tutorial 4, step 1

Playing

The **Noise Mallet** module can be triggered by the output signal from another module, but it can also be triggered manually by clicking on the *trig* button from its front panel (which is how we will use this object for the moment). The stiffness of the mallet is adjusted with the *stiffness* knob on the module front panel, while the amplitude of the impact is determined by the *strength* knob. To obtain a soft mallet, made out of cotton for example, turn the *stiffness* knob to the left; for a harder one, which could be made out of wood, turn this knob to the right. To hear a sound, set the stiffness and the strength in the middle and click on the trigger knob. What you hear is the impact signal from the mallet and a certain amount of superimposed noise. You only hear a faint sound because the mallet is not hitting anything for now. Try different settings on the stiffness knob to hear the mallet change. For now, the two little knobs at the bottom of the module are not connected to anything, so they have no effect on the sound.

Step 2: Add a plate

Description

We will now add a **Plate** module and listen to the sound it produces when hit by a **Noise Mallet**.

Construction

- In the **Resonators** section of the module library, select a **Plate** module and place it in the construction area.
- Select and delete the wire connecting the **Noise Mallet** to the **Audio Out**.
- Pull a wire between the output of the **Noise Mallet** and the second input of the **Plate**.
- Select a **Constant** module from the **Generators** section of the library area. By default the **Constant** module outputs a value of 1 (you can check this by double-clicking on the module). We will keep this value.
- Pull a wire between the output of the **Constant** module and the first input of the **Plate** module. This input determines if external dampers are lowered on the **Plate** or not. A value of 0 lowers the dampers and a value of 1 keeps them above the **Plate**.
- Select a **Level** module from the output folder in the **In/Out** section of the Browser.

- Select a **Volume** module from the **Envelopes** section in the Browser.
- Pull one wire between the output of the **Plate** module and the input of the **Volume** module.
- Pull a wire between the output of the **Volume** module and the input of the **Audio Out** and the **Level** module.
- Launch the *Tassman* Player with the Ctrl-T/Apple-T shortcut.

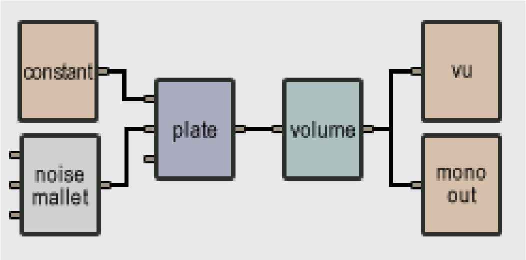


Figure 15: Tutorial 4, step 2

Playing

Now click on the *trig* button of the **Noise Mallet** and you will hear the sound of the **Plate** being hit by the mallet. You can change the sound produced by the plate by changing the settings of the different knobs appearing on the control panel. These settings control parameters that are directly related to the material of the plate.

The damping of a plate affects the decay time of the sound produced by the object. This parameter is adjusted using the *decay* knob on the front panel. When the knob is turned right, the damping is light and the decay time long. Turning the knob to the left increases the damping and reduces the decay time. Damping is characteristic of the material of an object. In wood, for example, damping is high and the decay time is short; in steel, on the other hand, a lower damping results in a longer decay time. But damping also varies for a given material depending on how the object is used or connected to other objects. As an example, the response of a gong is longer than that of the top plate of a steel guitar.

In a mechanical structure, the damping (or decay time) also varies for the different frequency components of the oscillating motion. The variation of damping with frequency is yet another characteristic of the material of a structure and is adjusted with the *damp/freq* knob on the module front panel. In the left position, the decay time of low frequencies is shorter than that of high frequencies; in the right position it is greater. As a rule of thumb, steel and glass are found in the left position, nylon in the center position, and wood in the right position.

Try different combinations and tweak the knobs while you play so that you can hear the changes gradually. Don’t forget to try different mallet parameters, since this too can change drastically the resulting sound.

Step 3: Add a keyboard

Description

Triggering the mallet manually is rather limiting, so we will now use a keyboard to control the mallet and “play” the Plate.

Construction

- Select a **Vkeyboard** module from the MIDI folder in the **In/Out** section of the Browser.
- Pull a wire from the first output of the **Vkeyboard** module (gate signal) and connect it to the first input of the **Noise Mallet** module.
- Pull a wire from the second output of the **Vkeyboard** module (pitch signal) and connect it to the third input of the **Plate** module.
- Pull a wire between the third output of the **Vkeyboard** (velocity signal) and the second and third inputs of the **Noise Mallet**.
- Switch to the *Tassman* Player.

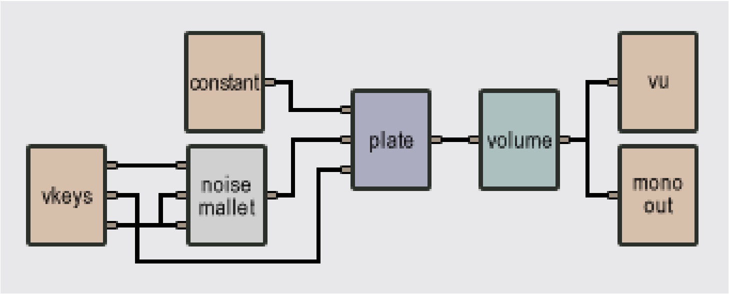


Figure 16: Tutorial 4, step 3

Playing

You can now trigger the mallet from the keyboard. Since the **Vkeyboard** sends a note signal to the **Plate** module, the pitch of the sound it produces can also vary. The pitch variations are controlled by the modulation signal entering the third input of the module; the higher the amplitude of the signal, the higher the pitch. In other words, the size of the object is varied in order to obtain the requested pitch. The *mod* knob is a gain knob affecting the amplitude of the pitch modulation signal. In its center position (green LED on), the pitch variation will follow an equal temperament scale.

In this patch, the velocity signal from the **Vkeyboard** is connected to the *strength* modulation input of the **Noise Mallet** and to the *stiffness* modulation input. Try pressing the keyboard keys at different velocities and note how the sound of the plate changes as the excitation signal varies. In this configuration, the higher the velocity the higher the strength of the impact and the stiffer the mallet.

Step 4: Add a second plate

Description

In many acoustic instruments, certain elements are used as passive resonators to amplify certain components of the sound. They are called sympathetic resonators. We will now add a second **Plate** which will be connected to the output of the first one. We will keep the geometry of this second plate fixed so that it will have a different behavior depending on the note played on the keyboard.

Construction

- Select and delete the wire linking the **Plate** and the **Volume** module.
- Select a second **Plate** from the Browser
- Select a **Mix2** module from the **Mixer** section.
- Select a second **Volume** module.
- Pull a wire between the output of the first **Plate** and the second input of the second **Plate** and the input of the new **Volume** module.
- Pull a wire between the second **Plate** module and the first input of the **Mix2** module.
- Pull a wire between the new **Volume** and the second input of the **Mix2** module.
- Pull a wire between the **Constant** module and the first input of the second **Plate**.
- Connect together the **Mix2** module and the remaining **Volume** module.

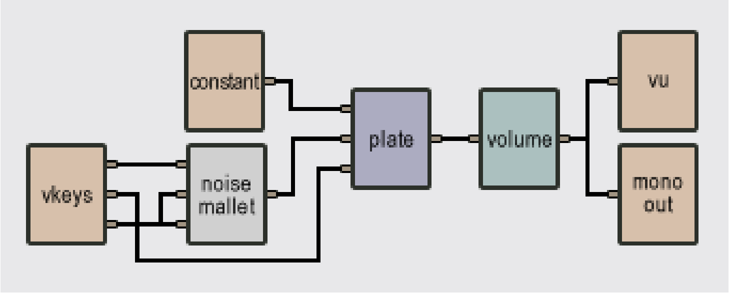


Figure 17: Tutorial 4, step 4

Playing

Change the parameters on both plates and experiment with different settings on each. Because the sympathetic **Plate** responds differently to different notes being played (having a fixed geometry, it resonates at specific frequencies), many interesting and unexpected sounds are possible. Try different mallet types and mix the output from the two plates with the **Volume** module connected on the first plate.

Step 5: Playing with presets

We now conclude this tutorial with some presets that we have made for you. To try them, load the following presets from the **tutorials/tutorial4/Step4** folder of your *Tassman* browser.

patch4_4_1

In this preset, the first **Plate** has a very short *decay* time, so you only hear it on the initial thump of the sound. The second **Plate**, on the contrary, has a long decay time so you hear it resonating for a longer time. Note the difference in the settings of the *damp/frq* knobs on the two plates.

patch4_4_2

In this example, the *damp/frq* knobs of the two **Plates** are set in their center position. The first **Plate** has a short decay, while the second has a longer one. Note the setting of the *stiffness* on the **Noise Mallet**, which results in a noisy attack.

patch4_4_3

In this sound, the two **Plates** have long decay times. The *mod2* knob on the **Noise Mallet** is a gain knob for the strength modulation signal. In this patch, the velocity output of the **Vkeyboard** module is linked to this input, which means that you can hit the plate with different levels of force. Try playing softer and harder to hear how the sound change.

3 The *Tassman* Builder

The *Tassman* Builder is used to create instruments. Constructing instruments is very easy and straightforward. One first drags modules from the Modules section of the Browser and then connects them together in order to create a patch. Modules are units that either produce or transform sound in a particular manner. They each have a certain number of inputs and outputs which are used to transmit signals from one module to another and which can be interconnected by using wires . The Player view is then displayed to play instruments that have been made with the Builder.

The different modules of the library are just like building blocks that can be connected to other blocks any way you desire. The only limit is your imagination! Of course creating instruments implies a certain amount of experimentation, but that is part of the fun! You can start with simple instruments, trying them right away with the Player, and then come back to the Builder to modify the patch. Very rapidly you will be able to construct the instruments you have always dreamed of.

And that's not all! The architecture of the *Tassman* is entirely modular, which makes it a powerful evolutionary creation tool. As you create patches, you can save them as sub-patches and then reuse them in another patch just like any other elementary module. Very rapidly your library will expand and contain many different types of instruments and synthesizers.

When you save an instrument, the Builder creates an entry in the Browser under the Instruments folder describing the patch you have just made. This information contained in this file is used by the *Tassman* to generate the instrument front panel and the computation code necessary to simulate the instrument. These instrument files are very light and can be exchanged with other users (using the Export/Import command) via e-mail. Check the Applied Acoustics Systems website often for new instruments to download.

3.1 The Builder area

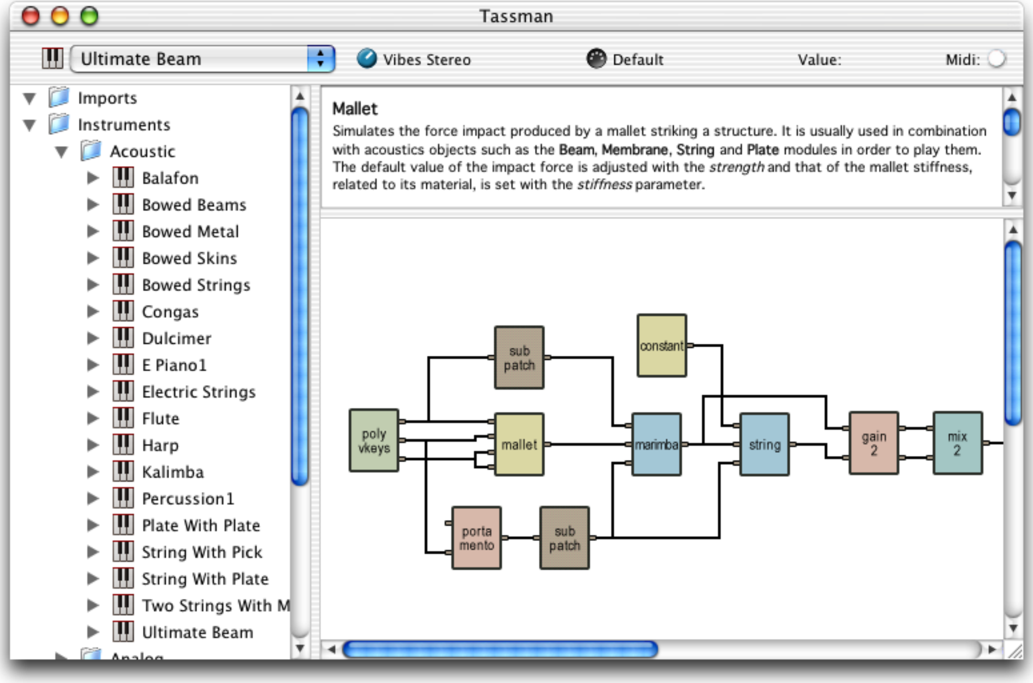
The Builder is divided into two different areas: the construction area and the help area. The Browser contains the module library.

The library area

The module library, included in the Browser, contains the different modules of the library. The modules are divided into eleven categories: **Effects**, **Envelopes**, **Filters**, **Generators**, **In/Out**, **Logic**, **Mixers**, **Resonators**, **Routing**, **Sequencers** and **Sub-patches**. The different sections are selected by clicking on the corresponding folder in the Browser, to open a section, click on the sign at the left of the folder, this will display the list of modules contained in the folder.

The construction area

The construction area is where you build your patch. After dragging modules from the module section of the Browser, you connect them with wires using the mouse.



Mallet

Simulates the force impact produced by a mallet striking a structure. It is usually used in combination with acoustics objects such as the **Beam**, **Membrane**, **String** and **Plate** modules in order to play them. The default value of the impact force is adjusted with the *strength* and that of the mallet stiffness, related to its material, is set with the *stiffness* parameter.

Figure 18: The Builder area.

The help area

The help area is located above the construction area. This is where information about the module currently selected is displayed. The information found here is limited to what is needed to create patches in the Builder. For more information on the functioning of a module or the controls appearing on its front panel, please consult the reference section of this manual or its online version, which can be opened from the *Tassman* **Help** menu.

3.2 Creating an instrument

Choosing modules

The first step in creating an instrument consists in adding the modules it will be made from.

- To add a module, drag it from the Browser to the construction area, it will appear at the specified location.
- If you want to include many identical modules in your instrument, you can copy it using the Copy/Paste commands from the **Edit** menu or use the **Duplicate** command (Ctrl-D/Apple-D).

- An **Audio Out** module must always be included in your patch. You can save an unfinished patch without an **Audio Out**, but you will not be able to play the instrument.

Connecting modules

Modules have a certain number of inputs (on the left of the module) and outputs (on the right of the module) which are used to exchange signals between modules.

- To view a description of an input or output, position the arrow-cursor over it.

Modules are connected together using wires. Wires are drawn using horizontal and vertical segments.

- To pull a wire, click on the output of a module and move the jack-cursor to the input of another module and then click to make the connection. The Builder will automatically draw the wire.
- If you want the wire to follow a specific path, click as you pull the wire in order to insert breakpoints in it and then move the mouse either horizontally or vertically.
- To stop pulling a wire after clicking on an output, right-click (PC) or double-click (Mac) anywhere in the construction area. Hitting the **Escape** key on your keyboard has the same effect.

Editing wires

Once a wire has been drawn, it is possible to change its layout.

- To select a wire, click on it. To deselect a wire, click anywhere in the construction area.
- To delete the wire, select the wire and press the **Del** (PC, Mac) or **BkSp/delete** key of the computer keyboard.
- To move a segment, click on it, hold, and move the mouse.
- To introduce breakpoints in a wire, shift-click (PC) or Option-click (Mac) on the point where you want the new breakpoint and move the mouse horizontally or vertically depending on whether the cursor is positioned on a vertical or horizontal segment.

Editing modules

Once they have been placed on the construction area, modules can be moved in a number of ways.

- To select a module, click on it and it will be surrounded by a red square (PC) or the highlight color chosen by the user (Mac). To deselect a module, click in the construction area.

- To select more than one module at once, click on the construction area, keep the left button down and drag the mouse in order to surround the modules you want to select with the rectangle appearing on the area. You can also click on different modules while pressing on the **Shift** key to achieve the same results. To select all the modules at once, use the **Select All** command from the **Edit** menu or use the **Ctrl-A/Apple-A** shortcut.
- When many modules are selected, you can remove one module from the selection by pressing the **Shift** key and clicking on the module you want to remove.
- Once modules have been selected, you can move them on the construction area by dragging them.
- To delete modules, select them and press on the **Del** (Mac, PC) or **BkSp/delete** key or use the **delete/Clear** command from the **Edit** menu.
- Modules can be aligned horizontally by selecting them and using the **CenterVertically** command from the **Arrange** menu or its **F9** shortcut (PC only).
- Modules can be aligned vertically by selecting them and using the **CenterHorizontally** command from the **Arrange** menu or its **Shift-F9** shortcut (PC only).
- To make copies of modules, select them and use the **Copy** command from the **Edit** menu or its keyboard shortcut, **Ctrl-C/Apple-C**. Use the **Paste** command from the **Edit** menu, or its shortcut **Ctrl-V/Apple-V** keys to paste the last copied modules. You can also use the Duplicate command from the **Edit** menu or its shortcut **Ctrl-D/Apple-D**
- To place a module in a precise manner on the construction area, select it and use the arrow keys to move it on the construction area.

Adjusting module default values

Most of the module parameters can be adjusted on their front panel in the Player. It is possible, however, to adjust their default values during construction in the Builder.

- To edit a module in the construction area, double-click on the module and a dialog will appear. You can also select the module and use the command **Module Settings** in the **Edit** menu.
- The first editable field is the name of the module. By default the name given to a module is made out of the module type followed by an integer. You can choose any name you like for the module. The name you choose will appear on the module front panel in the *Tassman* Player.
- The second editable field is the row number. This number corresponds to the row on which the module panel will be mapped in the Player. There is a maximum of 16 rows in the Player and if you do not want to see the module on the Player, you can choose the *invisible* option to hide the modules. Note that this field is not editable for certain modules which do not have a front panel. Note that within a row, the modules will be placed depending on their position in the Builder (left-to right and up-down priority).

- Finally, depending on the module you are currently editing, there might be a certain number of parameters which can be set at construction. For more information on the effect of each parameter see the help area or the module description in the user manual.

Saving an instrument

To save an instrument use the **Save** (**Ctrl-S/Apple-S**) or **Save As** commands from the **File** menu of the Builder.

- The instrument will be saved in the Browser under the Instruments folder. If the patch contains an **Inlet** or **Outlet** module, it will be saved under the Sub-Patches folder.
- You can save an unfinished instrument, but remember that if you want to hear it, you need an **Audio Out** or **Stereo Audio Out** in your patch.

Playing an instrument

Once you have completed your instrument, just choose the **Show Player** command from the **View** Menu (or use the **Ctrl-T/Apple-T** shortcut) to display the corresponding Player view and have fun! You can always come back to the Builder from the Player by choosing choose the **Show Builder** command from the **View** Menu (or use the **Ctrl-T/Apple-T** shortcut).

3.3 Setting MIDI Links

Every control of every module that appears on the module front panel of the *Tassman* Player can be linked to an external MIDI controller.

- To link a control to a specific MIDI controller, choose the **Edit MIDI Links** command from the **Edit** menu. A Midi links dialog for the current patch will appear.
- To add a link, click on the **New** button which will open the **Edit MIDI Link** window. From the **Name** menu, choose the front panel control you want to link.
- In the **Controller** and **Channel** fields, indicate the number of the external MIDI controller you want to use and its MIDI channel.
- To limit the range of a MIDI controller edit the **Minimum Value** and **Maximum Value** fields. The Minimum Value field determines the position on the *Tassman* control which corresponds to the minimum value sent by the MIDI controller, while the **Maximum Value** determines the position which corresponds to the maximum value sent by the MIDI controller. A value of 0 corresponds to the *Tassman* Player controller minimum position (left position for a knob) and a value of 1 to the *Tassman* controller maximum position (right position for a knob). Note that the range of knob can be inverted by setting the value of **Maximum Value** to a smaller value than that of **Minimum Value**.

- Click on the **OK** button of the **Edit MIDI Link** window and the MIDI link you have just edited will appear in the **MIDI Links** window. If you wish to activate this MIDI link, click on the **OK** button of the **MIDI Links** window.
- If you want to change a MIDI link, select it by clicking on it in the **MIDI links** window and press the **Edit** button or simply double-click on it.
- To delete a MIDI link, select it by left-clicking on it in the **MIDI links** and press the **Delete** button.

3.4 Making Polyphonic Instruments

A polyphonic instrument is created by including modules between a polyphonic keyboard module (**Polykey** or **Polyvkey** module) and a **Polymixer** module.

- Link the input modules of your patch to the output of a **Polykey** or **Polyvkey** module.
- Link the output of your patch to the input of a **Polymixer** module.
- Link the output of the **Polymixer** module to other modules or to an **Audio Out** module.
- Double-click on the **Polykey** module and set the number of voices you want.
- You can use more than one **Polymixer** in an instrument.

You can choose any number of voices for your instrument but keep in mind that the computational load of a patch increases with the number of voices you choose. Basically, adding a voice is roughly equivalent to adding another copy of the polyphonic modules in your patch. The number of voices you will be able to run depends on the complexity of the patch you are currently using and the power of your computer.

3.5 Using Sub-Patches

indexsub-patch

A very powerful feature of the modular or “building-block” architecture of the *Tassman* is that you can define patches as new modules of the *Tassman* library. This means that you can reuse patches you have already made in new patches. Using sub-patches is very useful if you often use the same combination of modules in many patches. It will also save you a lot of time when you want to include a complex patch into another.

Making a Sub-Patch

An instrument is saved by the Builder in an entry under the Instruments folder. When you double click on the instrument icon, the Player will be launched with the control panel corresponding to the instrument you have just chosen. A sub-patch, on the other hand, is saved under the Sub-Patches folder. When you double-click on it, the Builder will be displayed. Sub-patches can be included in other patches saving you the trouble of redoing the patch again. The only difference between a an instrument and a sub-patch is that a sub-patch, like any other elementary module from the library, has inputs, or outputs, or both, so that it can be connected to another patch. To create a new sub-patch:

- Choose an **Inlet** or **Outlet** module or both in the **In/Out** folder of the Browser and place them in the construction area. A sub-patch module may have between 0 and 12 inputs and between 0 and 12 outputs, but it must always have at least one input or one output.
- Determine the inputs of your patch which you will be using to connect this patch to another one. These inputs will appear on the sub-patch icon as the inputs of the **Sub-patch** module. Connect the inputs you have chosen in your patch to the outputs of an **Inlet** module (**Inlet 1-12** depending on the number of inputs of the new module).
- Determine the outputs of your patch which you will be using to connect this patch to another one. These outputs will appear on the sub-patch icon as the outputs of the **Sub-patch** module. Connect the outputs you have chosen in your patch to the inputs of an **Outlet** module (**Outlet 1-12** depending on the number of outputs of the new module).
- Use the **Save** (or the **Ctrl-S/Apple-S** shortcut) or **Save As** command from the **File** menu. Since you have at least one **Inlet** or **Outlet** in your patch, the Builder will automatically save this patch in the **Sub-Patches** folder in the Browser.

It might be useful to document a sub-patch so that you will have a reminder of its purpose or functioning when you use it in another patch. To have text appear in the help area of the *Tassman* Builder when you select a sub-patch.

- In the browser, **right-click/control-click** on the sub-patch icon and select **SubPatch Info** from the menu. Fill the required fields and they will appear in the help area when you click on the module in the Builder.
- You can also change the names of the inputs or outputs which appear when you position the mouse on the inputs or outputs of the **Inlet**, **Outlet** or **Sub-Patch** module in the inlet or outlet text field. To do so double-click on the **Inlet** or **Outlet** of the sub-patch.

Including a Sub-Patch

Once you have defined a certain number of patches as new modules, you can reuse them in other patches.

- In the Browser, open the **Sub-Patches** folder and drag and drop the sub-patches you want in the construction area just like any other module.
- To view the patch inside a sub-patch module, select it and choose the **Open Sub-patch** command from the **File** menu of the Builder. You can also right-click (PC) or Ctrl-click (Mac) on the module and choose **Open sub-patch** from the contextual menu.

4 The *Tassman* Player

4.1 Introduction

The Player is the view used by the *Tassman* to play instruments. It appears on the screen as an instrument front panel with knobs, buttons, sliders and switches which you can tweak to play the instrument.

The Player can viewed in the following manners:

- Double click on the *Tassman* icon on your desktop. The default performance is then launched.
- To launch the Player from the Builder view, choose the **Show Player** command from the View menu or use the keyboard shortcut **Ctrl-T/Apple-T**.

Note that it is possible to switch back and forth between the Player and the Builder when you want to modify your instruments.

- To view the patch corresponding to the current synthesizer in the Builder, click on the **Show Builder** button in the toolbar, choose the **Show Builder** command from the **View** menu or use the keyboard shortcut **Ctrl-T/Apple-T**.
- Once the current patch is modified in the Builder, return to the Player by clicking on the **Show Player** button in the toolbar, choosing the **Show Player** command from the **View** menu or using the keyboard shortcut **Ctrl-T/Apple-T**.

4.2 The *Tassman* Player

The Player area is displayed as a rack into which you mount rows of modules. The front panels which appear in the rack correspond to the modules you have used to make your patch in the Builder. The name appearing at the top of each module is the one you chose when constructing the instrument in the Builder. There is no a priori limit to the number of modules you can use in an instrument, this will be determined by the power of your machine.

- There are sixteen rows accessible in the Player view. You can navigate through them horizontally and vertically with the two scroll bars at the right and bottom of the Player.
- The row on which a module is displayed is set in the Builder. To place a module on a specific row, double-click on the module in the Builder. In the edit window, choose the **Display** row (1-16). Within a row, modules are placed according to their position in the Builder view with a left-to-right and up-down priority.

More modules can be placed in a given row than actually appear in the *Tassman* Player area. The number of modules you will be able to place on one row will be around 100-150 (depending on which modules you include).

- To move the rows horizontally when they are wider than the Player area, use the bottom scroll bar.
- To move the Player view vertically to access bottom rows, use the right scroll bar.

It is also possible to save screen space if you have used sub-patches to construct your instrument.

- To open and close modules encapsulated in a sub-patch, click on the arrow appearing on the upper left corner of the sub-patch.

4.3 Tweaking knobs

Each of the different knobs, buttons, and switches appearing on the module front panel can be tweaked with the mouse. There are different ways to control them depending on the effect you want to achieve.

- For coarse adjustment of a knob, click on it and, keeping the left-button down, move the mouse upwards or downwards to move the knob to the right or to the left.
- For fine adjustment of a knob, click on the knob to select it and move it counter-clockwise by using the left or down arrow and clockwise with the right or up arrow. You can also move the switches by selecting them and using the arrows. The **Page Up** and **Page Down** keys give the same result, but the knobs then move a little faster.
- To move a knob or switch directly to a given position, place the mouse at this position and Shift-click (PC) or Option-click (Mac). For the knob or switch to reach this position slowly, do the same, but use the middle button of the mouse (PC only).
- Knobs with a green LED above are moved directly to their center position by clicking on the LED.
- To adjust switches, click on them and, keeping the button down, move the mouse upwards or downwards. You can also select them by clicking on them and using the arrows just like the knobs.
- To change the position of buttons and switches click on them.

Remember that the keyboard shortcuts only affect the most recently selected controller. The value of the controller currently selected is displayed on the toolbar at the top of the *Tassman* window. The number displayed on the counter is a value corresponding to the setting of the controller currently selected. For knobs, the reading is a value between 0 (left) and 127 (right); for switches it is a value that depends on the adjustment, for buttons it is 0 or 1 depending whether it is off or on.

4.4 Audio Device Settings

To select the audio device used by the *Tassman*:

- Go to the **Edit** menu, choose **Preferences** and then **Audio Settings**. On Mac OSX, the same command is under the *Tassman* menu. A list of the audio devices installed on your computer will appear in the **Audio Configuration** window.
- Click on the Audio device you wish to use and click on the **OK** button.

4.5 MIDI Settings

Selecting a MIDI device

To select the MIDI device used by the *Tassman*:

- Go to the **Edit** menu, choose **Preferences** and then **MIDI Settings**. On Mac OSX, the same command is under the *Tassman* menu. A list of the MIDI devices installed on your computer will appear in the **MIDI Configuration** window.
- Select the MIDI devices you want to use and click on the **OK** button.

Setting MIDI links

Every control that you see in the Player can be manipulated by an external MIDI controller . In most cases, this is, of course, much more convenient than using the mouse, especially if you want to move many controllers at once. For example, you can map the motion of a knob from a knob box to that of the modulation wheel from a keyboard or link a switch to a sustain pedal. As you use the specified MIDI controllers, you will see the controls move on the Player area just as if you had used the mouse. MIDI links are set in the Player but can be edited in the Player or the Builder.

To assign a MIDI link to a controller:

- In the Player, right-click (Control-click on Mac) on a control (knob, button, or slider), a contextual menu appears. Select **Learn MIDILink**.
- Move a knob or slider on your MIDI controller (this can be a keyboard, a knob box or any Device that sends MIDI). This will link the control of the *Tassman* to the MIDI controller you just move.
- To edit the MIDI link, right-click/Control-click again on the control and choose **Edit MIDILinks**. This opens the Edit window for the MIDI links.
- Click on the MIDI link you wish to modify and then on the **Edit** or **Delete** button to modify or delete the MIDI link. You can also click on **New** to create a new MIDI link.

- The MIDI controller number specified in the **Controller** textbox is set by default to a value of 1. This is the MIDI controller number corresponding to the modulation wheel. If you want to assign a new controller to the knob, specify the number here.
- You can also assign a different MIDI channel to the controller in the **Channel** textbox. By default this value will be set to channel 1.
- There are two other parameters one can adjust: the **Minimum Value** and **Maximum Value** of the controller, which are used to limit the range of MIDI controllers. The **Minimum Value** field determines the position on the *Tassman* controller to which corresponds the minimum value sent by the MIDI controller; the **Maximum Value** determines the position to which corresponds the maximum value sent by the MIDI controller. A value of 0 corresponds to the *Tassman* Player controller minimum position (left position for a knob) and a value of 1 to the *Tassman* controller maximum position (right position for a knob). Note that the range of knob can be inverted by setting the value of **Maximum Value** to a smaller value than that of **Minimum Value**. This can be useful, for example, if you want to control the cutoff and the resonance of a filter with the same knob but you want the resonance to increase as the cutoff decreases.
- Click OK and the link appears in the list of controllers linked to the control.
- Click OK again to confirm the change and to leave the MIDI links window.
- Note that the **Minimum Value** and **Maximum Value** of a MIDI link can also be set by right/control clicking on the corresponding control and selecting the **Set MidiLink Minimum Value** or **Set MidiLink Maximum Value** command. The value corresponding to the control position will then be saved as the minimum or maximum value of the MIDI link.
- To remove a MIDI link, right-click/Control-click again on the control and choose **Forget MIDILinks**.

Creating a MIDI map

MIDI links for a given instruments can be saved into a MIDI map by using the **Saved MIDI Links As** from the **File** menu. Different MIDI maps corresponding to different MIDI controllers can be saved for the same instruments. A MIDI map can be loaded by double clicking on the MIDI connector icon under an instrument in the browser. Furthermore a MIDI map can be loaded by automatically when an instrument is launched.

- To assign a default MIDI map to an instrument, **right-click/Ctrl-click** on the MIDI map icon and choose the **MIDI Link Info** command, in the **Edit Information Window**, click on **Mark As Default**.

Creating the MIDI program change map

MIDI program changes can be used to switch between performances while playing (more on performances in a moment). To associate a program change to a performance:

- Choose the **EditMidiChanges** from the **Edit** menu,
- The list of performances appears in the left of the **Program Changes** window while the program change numbers (from 1 to 128) appear on the right.
- To associate a performance to a given program change, click on the performance icon and drag-and-drop it on the corresponding number.
- To unassign a program change, right-click/Ctrl-click on the performance name on the right of the **Program Changes** window and click on **unassign**.

4.6 Latency Settings

The latency is the time delay between the moment you send a control signal to your computer (for example when you hit a key on your MIDI keyboard) and the moment when you hear the effect. Roughly, the total latency is due to three factors: the time taken by the sound card driver to send MIDI signals to the *Tassman*, the time taken by the *Tassman* to compute the requested number of sound samples and finally the time taken by the sound card driver to send back the sound samples to the card and play them.

Within the Player you can control the amount of latency introduced by the *Tassman*.

- Choose the **Latency/Audio Control Pannel** command from the **Configuration** menu.
- Adjust the buffer size.
- The total latency is equal to the number of buffers multiplied by the number of samples per buffer divided by the sampling rate.

You can also choose the sampling rate and the audio format (16, 24, 32 bits) in the Latency window. The panel and settings may look different depending on which sound card you are using.

It is of course desirable to have as little latency as possible. The *Tassman* will however require a certain latency to be able calculate sound samples in a continuous manner. This time depends on the power of your computer and the size and nature of the patch you are playing.

Note that the content of the dialog depends on the driver selected in the **Audio Settings** menu.

4.7 Instruments and Presets

Instruments are created in the Builder and saved in the Browser under the instruments folder. A given instrument can be loaded into the player in the following way:

- Double-click on an instrument in the Browser. Note that the instrument will load in the current view selected (Player or Builder) in the *Tassman*.

It is possible to obtain very diverse sounds with a given instrument depending on the settings of the different controls. When you obtain a sound that you like, it is possible to save the configuration of the different controls as **Preset** for the instrument so that you can rapidly reproduce the same sound. This is one of the advantages of software over hardware: you can find again different configurations without having to tweak all the knobs again.

The modular architecture of the *Tassman* allows you to save and load presets for all hierarchy level of your instruments. It is possible to save and load control settings for a given module, for a sub-patch or for the whole instrument. Presets are saved in the corresponding folder under the instrument, sub-patch or module name. As an example, if you create a preset for a **Plate** module, it will be saved in the **Module** section in the **Resonators** folder under **Plate**.

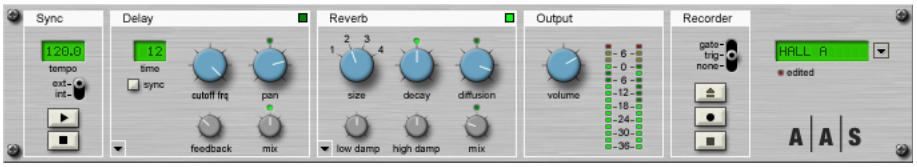
- To save settings for a module, click on the downward pointing arrow on the lower left corner of the module.
- To load settings for a module, click on the upward pointing arrow on the lower left corner of the module, this will highlight the module in the Browser. Then, drag and drop a preset from the Browser on the module. You can also drag and drop a preset directly from the Browser without clicking on the arrow on the module.

When you load presets for a module, all the presets saved under the module will appear in the browser. This means that settings saved for one module can be loaded by another module of the same type. This is very helpful when you have several identical modules in an instrument and you want to set them all exactly the same way (filters, for example).

Presets for sub-patches are saved and loaded (just as for modules) by clicking on the upward or downward pointing arrows appearing in the lower left corner of the sub-patch. To save settings for the entire instrument, use the **Save Preset** or **Save Preset As** commands from the **File** menu.

Note that a default preset can be assigned to a given instruments. In this way, the controls of the instrument front panel will be adjusted according to this default preset when the instrument is launched. To assign a default preset right-click/control-click on the preset you want to use as default, select **Preset Info** from the menu and check the **Mark as Default** box in the **Edit** window.

4.8 Output Effect Stage



tempo 120.0

time 12

HALL A

The output effect stage is always displayed in the top row of the *Tassman*. This effect stage is added to each *Tassman* synth and allows one to add effects to the sound, record on the fly, export loops as wave or aiff files for further processing and control the tempo and sync sources (internal or sync to host) of sequencer or effect modules.

The Sync module

This module is used to control the tempo of the **Sequencer**, **Sync LFO** and **Sync Delay** modules when they are connected to the **Master Sync Input** module. The *ext/int* switch is used to determine if the sync signal comes from an external source or from the internal clock of the module. When the *Tassman* is used as a plug-in in a host sequencer and the *ext* source is chosen, the clock signal will be that sent by the host sequencer while in standalone mode the clock will be the one received on the MIDI channel selected in the Player toolbar. When the *int* source is chosen, the clock is adjusted in the green *tempo* display in beats per minute. To change the tempo, click-hold on the display and drag up or down or use the up and down arrows of the computer keyboard after clicking on the display. The *play* and *stop* buttons are used to start or stop **Sequencer** modules in your patch that have been connected to a **Master Sync Input** module while the *reset* button is used to send a reset signal.

The Delay module

This module is a standard ping pong delay and is based on two delay lines. The *time* display sets the length of the lines. When the *sync* button is pressed, the sync signal from the **Sync** module of the output stage is used to determine the length of the delay line which is adjusted to fit the number of steps appearing in the display, four steps representing a quarter note. The *feedback* knob is used to adjust the amount of signal re-injected from the output of a line into the other one while the *cutoff* knob controls the cutoff frequency of the low-pass filter applied to the signal in each line. The *pan* knob is used to adjust the panning of the echoes between the left and right position. Finally the *mix* knob controls the relative amount of “dry” and “wet” signal in the output signal. For more details on the algorithm implemented in this module, please refer to the **Sync Ping Pong Delay** module description in Section 6.86.

The Reverb module

This Reverberator module is the same as that included in the module library and described in more details in Section 6.70. The *size* button is used to choose the size of the room from small (1) to large hall (4). The *decay* knob controls the reverberation time of the room (note that the range of this knob depends on the setting of the size of the room). The *diffusion* knob is used to adjust the time density of the echoes which is related to the geometrical complexity of the room from simple (left) to very complex (right). The *low damp* and *high damp* knobs control the relative decay time of the room in the low and high frequencies respectively, a characteristic associated with the absorption of the wall of a room. Turning these knobs to the right decreases the decay time of the low/high frequencies. Finally the *mix* knob is used to set the relative amount of “dry” and “wet” signal which is related to the proximity of the sound source.

The Output module

This is where the adjustments of the overall level is made. The best dynamic range is obtained when the level meters are around 0 dB for loud sounds.

Master Recorder

This section is used to record the output of the *Tassman* to a wave or aiff file. The *eject* button, is used to choose the name and location of the destination file and it should always be used before starting a recording. The *record* and *stop* buttons are used to start or stop the recording. By using the **Master Recorder Trig** in a *Tassman* patch and setting the selector on the *gate* or *trig* position, you can also automate the start and stop of the recording in order to create loops precisely. For more detailed information on cutting loops, please refer to Section 6.41.

4.9 Performances

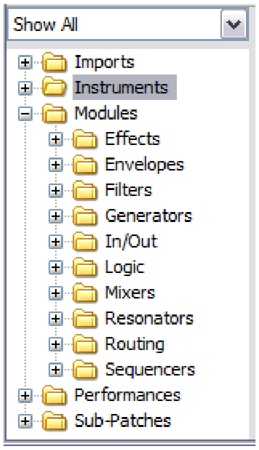
A Performance consists in a specific Synth/Preset combination, a given setting of the different effects of the output stage and a MIDI map. Performances enable the grouping of presets from different synths in the same folder and therefore to rapidly and efficiently switch between different sounds.

- To save a performance, choose the **Save Performance** or **Save Performance As** command from the **File** menu.
- The current synth/preset combination, the settings of the output stage and the current MIDI maps will be saved under the **Performances** folder of the browser.
- To load a performance, simply double-click on the corresponding performance icon in the browser, use the **Load Performance** command from the **File** menu or use program changes as explained in Section 4.5.

Note that you can select a default performance that will be loaded when you first start the Tassman.

- To set a performance as default, right-click/Control-click on the performance icon in the browser and select the **Set Performance As Default** command.

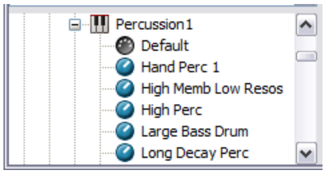
5 The Browser



The *Tassman*'s Browser is similar to those found in most email programs. Using a hierarchical tree structure, all the objects and files used in the building and playing of synths are available using a visually intuitive, drag and drop approach. These different elements have been organized under five root folders.

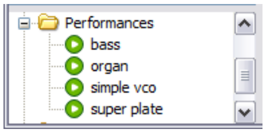
- Imports
- Instruments
- Modules
- Sub-Patches
- Performances

5.1 The Instruments folder



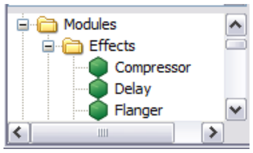
To explore the different synths and presets in *Tassman* click on the "+" icon to the left of the Instruments folder, in other words, *expand* this branch of the browser tree, in order to reveal the various instrument categories based on the different synthesis techniques used and instrument types. Increasingly specific categories can be found within by expanding each folder. Opening an individual instrument folder reveals the instrument file, and by expanding the individual instrument you'll find its presets. To play an instrument simply double click on the instrument icon (piano keys) or any of the preset icon (blue knob) in order to launch the *Tassman* Player.

5.2 The Performances folder



Performances consist in a synth/preset combination associated with a specific setting of the output effect stage and allow user to scroll rapidly between instruments and sounds (more on performances in Chapter 4. Playing the different performances is certainly the best way to explore the wide sonic possibilities of the *Tassman*. To load a performance, simply double click on the green icon.

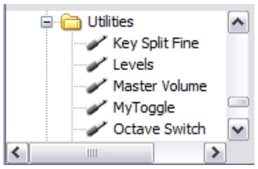
5.3 The Modules folder



The modules (green cell icon) are the elementary building blocks used to construct synths in the Builder (more on modules in Chapter 3 and 6. Expanding the modules folder reveals the following module categories:

- **Effects -** delay, stereo chorus, compressor, etc.
- **Envelopes** - ADSR, portamento, VCA, etc.
- **Filters** - low-pass, band-pass, high-pass, etc.
- **Generators** - VCO, VCS, mallet, etc.
- **In/Out -** Audio outs, MIDI ins, sub-patch inlets and outlets, etc.
- **Logic** - AND, OR, XOR, etc.
- **Mixers -** basic 2 to 5 signal mixers.
- **Resonators -** string, bowed string, membrane, etc.
- **Routing** - selectors (chicken heads!), and switches.
- **Sequencers -** Gate, CV and pitch sequencers.

5.4 The Sub-Patches folder



Finally the sub-patches folder organizes the various sub-patch (mini-jack icon) into similar categories as found in the modules folder. This is an excellent place to start when building a synth. Many basic module configurations have been saved as sub-patches.

5.5 The Import folder

The **Import** and **Export** commands, found in the **File** drop down menu, allow one to easily exchange synths with other *Tassman* users, or decrease the number of synths in your Browser by archiving older or rarely used instruments elsewhere, on CD-R, or a second hard disk for example.

While one can export sub-patches or presets individually, exporting an instrument extracts all of the associated files including sub-patches and presets, and places them in a single *.txf file, in the specified export folder on the computer. An instrument containing several sub-patches and 20 to 30 presets is equivalent in size to short text file, making it easy to send constructions to other users via email.

Importing instruments from the archive or from other users is just as easy. Simply click on the **Import** command from the **File** drop down menu, and select the file to import. A new folder will then appear under the “Imports” directory in the browser, containing all of the files contained within the imported package. These can then be dragged and dropped to a new instrument folder, or remain in the Imports directory. How things are ultimately organized, we leave entirely up to you!

5.6 Customizing the browser

The Browser structure can be customized in various ways. New folders can be created from the **File** drop down menu using the **Create New Folder** command. One can also move files from one place in the Browser to another using the **Copy** and **Paste** commands from the **Edit** drop down menu, or by simply dragging a file from one folder and dropping it into the folder of your choice. While this open ended format makes it very easy to organize your instruments and presets, there are some restrictions on what can go where:

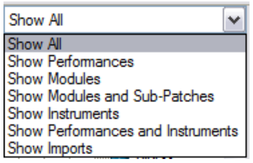
- **Modules** (green box icon) - May only appear within the modules folder. They cannot be moved.
- **Instruments** (piano keys icon) - May appear in the Instruments directory, in separate folders within the instruments directory, or in import folders (more on imports in a moment!).
- **Sub-patches** - May appear in the sub-patches directory, in separate folders with the sub-patch directory, or within an instrument file.
- **Presets** - May appear within an instrument, sub-patch, or module file.

While this all may seem a little convoluted on paper, the *Tassman*'s browser performs in very much the same way as various other programs you use everyday. The most important thing to consider when organizing your synths, sub-patches, and presets within the Browser is how you feel comfortable working. If you find yourself struggling to find the synths you're looking for, it might well be time to give the Browser a spring cleaning. As was mentioned in the 'Getting Started' guide, creating an archive folder for synths you find you're rarely using can help to eliminate unnecessary clutter. This becomes of particular importance if you're exchanging synths regularly with other *Tassman* users.

Lost?

If you find yourself struggling to find the modules or presets you're looking for, the browser's **Locate** function allows you to quickly jump to the instrument, module, or preset you currently have selected. Simply click on the module or sub-patch you wish to find in the Builder, or anywhere on the Player interface, hit **Ctrl-L/Apple-L**, and the browser will jump to the appropriate position.

5.7 Browser Filters

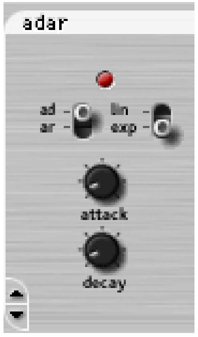


There are so many different entries in the browser that navigating can rapidly become confusing once a few folders have been expending. In order to simplify the browser view, you can apply different filters from the drop down menu at the top of the browser in order to view only certain categories of objects depending on what you are currently doing with the *Tassman*. The list of filters is as follows:

- Show All
- Show Performances
- Show Modules
- Show Modules and Sub-Patches
- Show Instruments
- Show Performances and Instruments
- Show Imports

6 Specifications for modules

6.1 ADAR



The **ADAR** is an envelope generator. It uses a gate signal for input and generates an output envelope signal. The **ADAR** module can generate two types of envelopes attack/decay or attack/release. The envelope type is set using the *ad/ar* selector. The behavior of the module is shown in Figure 1. In attack/decay mode the envelope signal rises from 0 Volt to 1 Volt when the gate is triggered and then immediately decreases form 1 Volt to 0. The time the output signal takes to go from 0 to 1 Volt is called the attack time, it is set with the *attack* knob and the time the signal takes to go from 1 to 0 Volt is the decay time and is adjusted using the *decay* knob. In attack/release mode the shape of the output signal is different since a sustain state is added. The output signal is held to 1 Volt until the gate signal falls to 0 and the release is then triggered. Finally, the *lin/exp* switch is used to determine the shape of the different segments of the envelope which can be either linear of exponential. Note that in Figure 19 the segments are exponential.

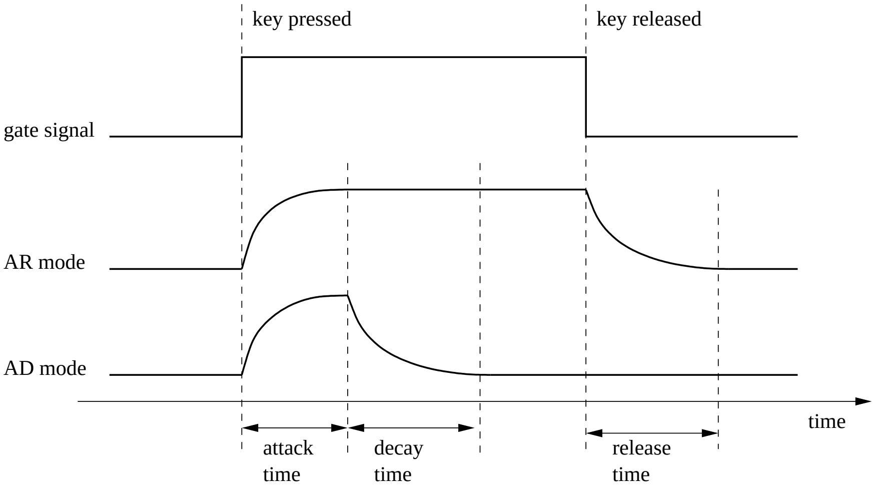


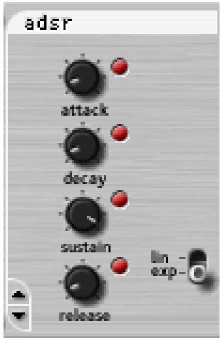
Figure 19: **ADAR** response curve

Typical Use

The **ADAR** is Typically used for generating amplitude envelopes through a **VCA**, or spectral envelopes by modulating the frequency of a filter module.

Note: See also the **ADSR**, **VADSR** and **VADAR** modules.

6.2 ADSR



The **ADSR** is an envelope generator. It uses a gate signal for input and generates an output envelope signal. An envelope is a time varying signal having a value between 0 and 1 Volt. It is divided into four, the Attack, Decay, Sustain and Release which can be adjusted as shown in Figure 2. The attack is triggered by an input signal exceeding a threshold value of 0.1 Volt. During this phase, the output signal goes from 0 to 1 Volt during the time set by the *Attack* knob. When the output reaches 1 Volt, the decay phase begins, and the output signal decreases from 1 Volt to the sustain level during the time set by the *Decay* knob. The sustain level, set by the *Sustain* knob, is then held until the input signal drops to less than 0.1 Volt. The output signal then decreases to 0 Volt during the time set by the *Release* knob. The red LED beside each knob indicates the current phase of the output envelope. Finally, the *lin/exp* switch is used to determine the shape of the different segments of the envelope which can be either linear of exponential. Note that in Figure 20 the segments are exponential.

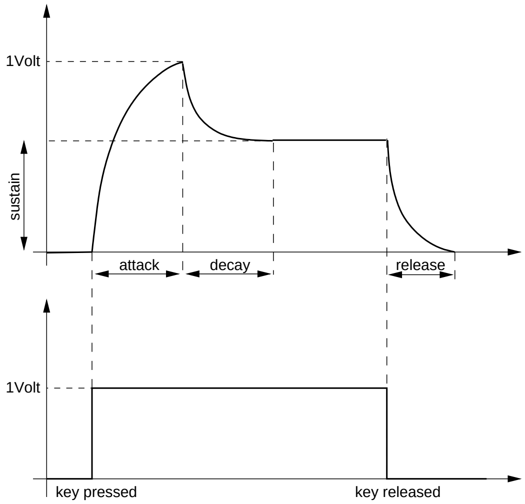


Figure 20: **ADSR** response curve

The default value of the following parameters is set during construction

- *Attack*: duration, in seconds, of the Attack phase.
- *Decay*: duration, in seconds, of the Decay phase.
- *Sustain*: level, in Volts, of the sustain phase.
- *Release*: duration, in seconds, of the release phase.

Typical Use

The **ADSR** is Typically used for generating amplitude envelopes through a **VCA**, or spectral envelopes by modulating the frequency of the filter modules. An **ADSR** can also be used to obtain an auto wah wah effect as shown in Figure 88 under **Vbandpass2**.

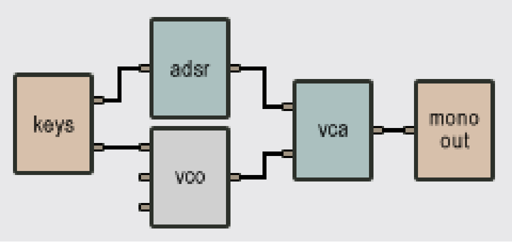


Figure 21: Amplitude envelope created with **ADSR**

Note: See also **ADAR**, **VADSR** and **VADAR** modules.

6.3 After Touch

The **After Touch** module is used to send the after touch control from a MIDI keyboard. It has one output, the after touch signal. It outputs a value ranging between 0 and 1 volt. This module has no front panel.

Typical Use

The After Touch module can be used to control modulation inputs on a **VCO** or a **VCF**.

6.4 And

The **And** module performs an AND logic operation. The one output of this module is either 1 (true) or 0 (false) depending on the values sent to the two inputs. To deliver 1 at the output, the two inputs must receive a value of 1 otherwise the output will deliver a value of 0. This module has no front panel. The following table shows the output value depending on the values in the two inputs.

Input signals are considered False (0) when smaller than 0.1 Volts and True (1) when greater than 0.1 Volts.

Input1	Input2	Output
1	1	1
1	0	0
0	1	0
0	0	0

Table 1: **And** module output as a function of its inputs.

6.5 Audio In

The **Audio In** module is used to process external audio in *Tassman*. The output of this module is a monophonic signal from a track or a bus of a host sequencer where the *Tassman* has been inserted as an effect. This signal can be then be processed on the fly by *Tassman* modules and then sent back to the track or the bus trough the use of an **Audio Out** or **Stereo Audio Out** module.

Note: See also **Stereo Audio In**.

6.6 Audio Out

The **Audio Out** module represents a digital to analog converter. This module has one input and no output. The input is converted to an analog signal by the sound card and is sent on the left and right channels of the sound card (the same signal on both channels). Before sending the input signal to the analog converters of the sound card, the **Audio Out** applies a saturation curve to the signal similar to that of an analog amplifier (Figure 4) in order to avoid undesirable digital saturation at high amplitudes. Inputs between -0.7V and +0.7V will be passed on linearly to the output, but higher voltages will be reduced to lie in the range of -1V to +1V. Subsequently, the voltage is transformed to a 16 bit integer range. If you do not want saturation to occur, you must ensure that the **Audio Out** input signal stays within the range where the curve is linear.

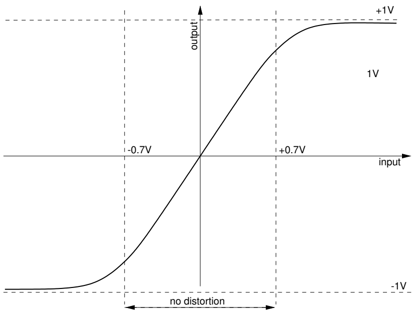


Figure 22: **Audio Out** module saturation curve

Typical Use

To ensure a good signal/noise ratio and avoid distortion due to excessive loudness, the **Audio Out** is often used in conjunction with a **Volume** and a **Level**.

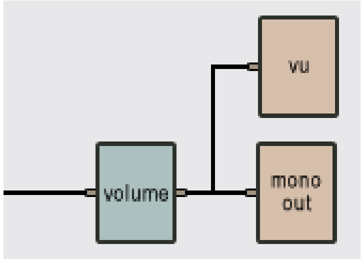
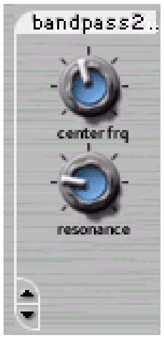


Figure 23: Use of an **Audio Out**

Note: There must be an **Audio Out** in your patch if you want to hear you instrument. See also **Stereo Audio Out**.

6.7 Bandpass2



The **Bandpass2** module is a second-order band-pass filter (-6dB/octave). Its one input is the signal to be filtered, and its one output is the filtered input signal.

Tuning the filter

The *center freq* knob tunes the center frequency of the filter to the desired level. The *resonance* button is used to adjust the resonance of the filter around the center frequency as shown in Figure 6. Note that as the resonance is increased, the amplitude of the filter response increases while the bandwidth of the filter decreases.

Typical Use

Bandpass2 filters can be used to make a parametric equalizer as in the patch of Figure 26. The response of the resulting filter is shown in Figure 25.

The default value of the following parameters is set during construction

- *Center Frequency*: middle frequency of the passing band.
- *Resonance*: resonance around the center frequency.

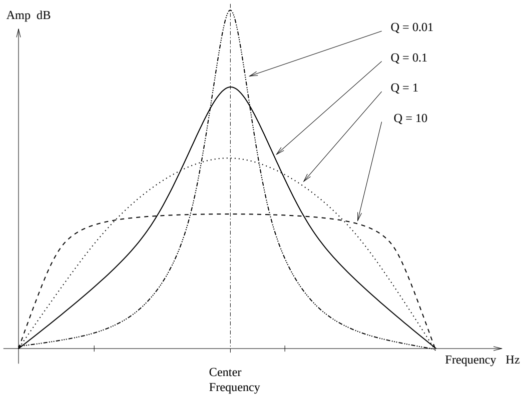


Figure 24: Frequency response of a **Bandpass2**.

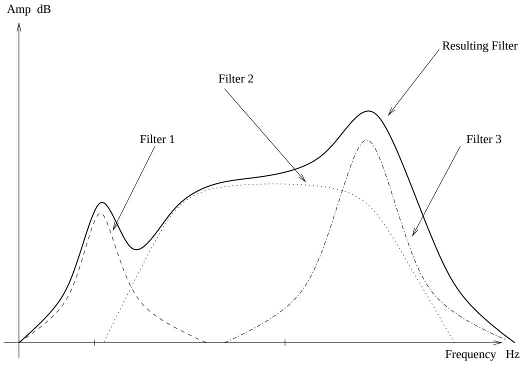


Figure 25: Response of the parametric equalizer shown in Figure 26.

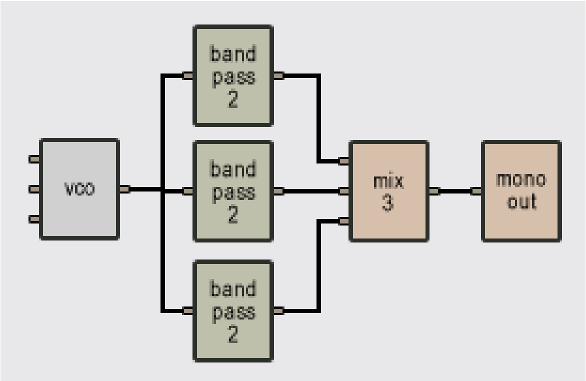


Figure 26: Parametric equalizer made with a **Bandpass2**.

Note: See also **Vbandpass2**.

6.8 Beam

The **Beam** module simulates sound produced by beams of different materials and sizes. This module first calculates the modal parameters corresponding to beam-shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Multimode** module to simulate sound production by the beam. The module has one output, the sound produced by the beam, and three inputs. The first input signal is a damping signal which, depending on its value, lowers or raises dampers on the structure. When the input signal is equal to 0, dampers are lowered on the beam which shortens the decay time of the sound produced by the structure; when the signal is greater than 0, dampers are raised. Note that this damping adds to the natural damping of the beam itself. If this input is not connected to any other module, the default value is set at 0, which implies that the beam motion will be damped. This input is, therefore, usually connected to a **Constant** module to obtain undamped motion or to a **Damper** module or the gate signal from a keyboard in order to vary the damping while playing. The second input signal is the force signal exciting the beam, the output from a **Mallet** module for example. The third input is a pitch modulation signal.

Typical Use. See Multimode module.

The default value of the following parameters is set during construction

- *Length*: the length, in meters, of the beam.
- *Frequency*: fundamental frequency, in Hertz, of the beam when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the length of the beam. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Beam** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the beam.
- *Number of modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point*: x-coordinate, in meters, of impact point from the extremity of the beam.
- *Listening point*: x-coordinate, in meters, of listening point from the extremity of the beam.

Note: For more details on this module and especially the front panel controls, see the **Multimode** module.

6.9 Bowed Beam

The **Bowed Beam** module simulates sound produced by bowed beams of different materials and sizes. This module first calculates the modal parameters corresponding to beam shaped objects depending on the value of the different parameters requested at construction time and, next, calls the **Bowed Multimode** module to simulate sound production by the beam. The module has one output, the sound produced by the beam, and three inputs. The first input signal is the bow velocity in the direction of the motion. The second input signal is a force signal which is considered to act perpendicularly to the motion of the beam. The third input is a pitch modulation signal.

Typical Use. See Bowed Multimode module.

The default value of the following parameters is set during construction

- *Length*: the length, in meters, of the beam.
- *Frequency*: fundamental frequency, in Hertz, of the beam when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the length of the beam. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Bowed Beam** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the beam. Number of Modes: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point*: x-coordinate, in meters, of bow from the extremity of the beam.
- *Listening point*: x-coordinate, in meters, of listening point from the extremity beam. To obtain proper functioning, the excitation and listening points should be the same.

Note: For more details on this module and especially the front panel controls, see the **Bowed Multimode** module.

6.10 Bowed Marimba

The **Bowed Marimba** module simulates sound produced by bowed marimba bars of different materials and sizes. This module reproduces the characteristic tuning of marimba bars overtones obtained with the deep arch-cut of the bars. This module, which constitutes a special case of the **Bowed Beam** module first calculates the modal parameters corresponding to beam-shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Bowed Multimode** module to simulate sound production by the bars. The module has one output, the sound produced by the beam, and three inputs. The first input signal is the bow velocity in the direction of the motion. The second input signal is a force signal which is considered to act perpendicularly to the motion of the beam. Third input is a pitch modulation signal.

Typical Use. See Bowed Multimode module.

The default value of the following parameters is set during construction

- *Length*: the length, in meters, of the beam.
- *Frequency*: fundamental frequency, in Hertz, of the beam when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the length of the beam. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Bowed Marimba** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the beam.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point*: x-coordinate, in meters, of bow from the extremity of the beam.
- *Listening point*: x-coordinate, in meters, of listening point from the extremity of the beam. To obtain proper functioning, the excitation and listening points should be the same.

Note: For more details on this module and especially the front panel controls, see the **Bowed Multimode** module.

6.11 Bowed Membrane

The **Bowed Membrane** module simulates sound produced by bowed rectangular membranes of different materials and sizes. This module first calculates the modal parameters corresponding to membrane-shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Bowed Multimode** module to simulate sound production by the membrane. The module has one output, the sound produced by the membrane, and three inputs. The first input signal is the bow velocity in the direction of the motion. The second input signal is a force signal which is considered to act perpendicularly to the motion of the membrane. The third input is a pitch modulation.

Typical Use. See Bowed Multimode module.

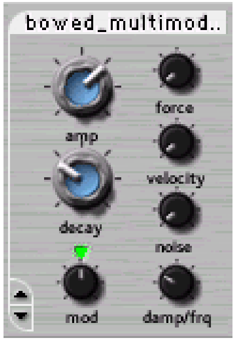
The default value of the following parameters is set during construction

- *length*: the length, in meters, of the membrane.
- *Width*: the width, in meters, of the membrane.
- *Frequency*: fundamental frequency, in Hertz, of the membrane when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the size of the membrane. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Bowed Membrane** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the membrane.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point-x*: x-coordinate, in meters, of bow from the lower left corner of the membrane.
- *Excitation point-y*: y-coordinate, in meters, of bow from the lower left corner of the membrane.
- *Listening point-x*: x-coordinate, in meters, of listening point from the lower left corner of the membrane. To obtain proper functioning, the excitation and listening points should be the same.
- *Listening point-y*: y-coordinate, in meters, of listening point from the lower left corner of the membrane. To obtain proper functioning, the excitation and listening points should be the same.

Note: For more details on this module and especially the front panel controls, see the **Bowed Multimode** module.

6.12 Bowed Multimode

The **Bowed Multimode** module is used by the *Tassman* to simulate mechanical objects such as strings, plates, beams and membranes that are excited as a result of the interaction with a bow. The output of this module is the acoustic signal that would be produced when these objects are bowed and given a certain geometry, material, listening point and damping. The functioning of this module is based on modal analysis. This technique is well-known in areas of physics and mechanics and is used to describe complex vibrational motion using modes. Modes are just elementary oscillation patterns that can be used to decompose a complex motion. By adding together modes having different frequencies, amplitudes and damping, one can reproduce the behavior of different type of structures. The accuracy of the resulting signal depends on the number of modes used in the simulation.



The **Bowed Multimode** module is not directly accessible to the user. Rather, other modules such as **Bowed String**, **Bowed Plate**, **Bowed Beam**, **Bowed Marimba** and **Bowed Membrane** use the **Bowed Multimode** module as their front. These other modules first calculate the different modal parameters corresponding to their respective structure type as requested at construction and, next, call the **Bowed Multimode** module in order to implement the parameters they require. Since these different object types are based on the same underlying simulation technique, they all have the same number of inputs and outputs and share the same controls (which appear on their front panel) for changing their physical properties.

Amplitude

The amplitude control is simply a gain which controls the amplitude of the output signal. It can be adjusted with the *amp* knob on the front panel.

Decay

The damping of an object affects the decay time of the sound produced by the object. This parameter is adjusted using the *decay* knob on the front panel. When the knob is turned left, the damping is strong and the decay time short; damping is light and decay time is long when the knob is turned right. The damping is characteristic of the material of the object. For example, damping in wood is strong and the decay time is short (knob turned to left) and in steel damping is weaker and, therefore, decay time is longer (knob turned to right). But damping also varies for a given material depending on how the object is used or connected to other objects. The oscillation of a string, for example, has a much shorter decay time when used on a violin than on a mandolin.

Playing frequency

The frequency of the sound produced by an object is dependent on its “useful” size. A large metal plate, for example, produces a sound with a lower pitch than does a smaller one. The pitch of the output of a **Bowed Multimode** object is determined by the signal entering the pitch input signal appearing on every such object. In other words, the size of the object is varied in order to obtain the requested pitch. The mod knob is a gain knob affecting the amplitude of the pitch input signal. When in the center position (green LED on), the gain equals 1 and the pitch variation is equal to 1 Volt/octave. This position is used to play an equal temperament scale when connecting the note output of a **Keyboard** to the pitch signal input of a **Bowed Multimode** object.

Force

The *force* knob is a gain knob acting on the force input of a Bowed Multimode object.

Velocity

The *velocity* knob is a gain knob acting on the velocity input of a Bowed Multimode object.

Noise

The *noise* knob is used to set the amount of irregularities in the bow structure.

Damping vs Frequency

In a mechanical structure, the damping, or decay time, varies for the different frequency components of the oscillating motion. The variation of the damping with frequency is characteristic of the material of a structure and is adjusted, in a **Multimode** object, with the *damp/frq knob* on the module front panel. In the left position, the decay time of low frequencies is shorter than that of high frequencies; in the right position the opposite holds. As a rule of thumb, steel and glass are found in the left position; nylon in the center position; and wood in the right position.

Typical Use

A good module to drive a **Bowed Multimode** module is an **ADSR**.

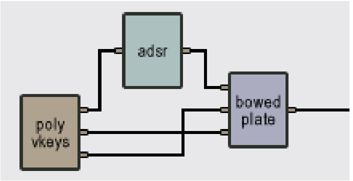


Figure 27: **Bowed Multimode** driven by an **ADSR**

Note: see also **Bowed Beam**, **Bowed Marimba**, **Bowed Membrane**, **Bowed Plate** and *Bowed String*.

6.13 Bowed Plate

The **Bowed Plate** module simulates sound produced by bowed rectangular plates of different materials and sizes. This module first calculates the modal parameters corresponding to plate shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Bowed Multimode** module to simulate sound production by the plate. The module has one output, the sound produced by the membrane, and three inputs. The first input signal is the bow velocity in the direction of the motion. The second input signal is a force signal which is considered to act perpendicularly to the motion of the beam. The third input is a pitch modulation signal.

Typical Use. See Bowed Multimode module.

The default value of the following parameters is set during construction

- *Length*: the length, in meters, of the plate.
- *Width*: the width, in meters, of the plate.
- *Frequency*: fundamental frequency, in Hertz, of the plate when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the size of the plate. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Bowed Plate** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the plate.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point-x*: x-coordinate, in meters, of bow from the lower left corner of the plate.
- *Excitation point-y*: y-coordinate, in meters, of bow from the lower left corner of the plate.
- *Listening point-x*: x-coordinate, in meters, of listening point from the lower left corner of the plate. To obtain a proper functioning, the excitation and listening points should be the same.
- *Listening point-y*: y-coordinate, in meters, of listening point from the lower left corner of the plate. To obtain a proper functioning, the excitation and listening points should be the same.

Note: For more details on this module and especially the front panel controls, see the **Bowed Multimode** module.

6.14 Bowed String

The **Bowed String** module simulates sound production by bowed strings of different materials and sizes. This module first calculates the modal parameters corresponding to string shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Bowed Multimode** module to simulate sound production by the string. The module has one output, the sound produced by the string, and three inputs. The first input signal is the bow velocity in the direction of the motion. The second input signal is a force signal which is considered to act perpendicularly to the motion of the string. The third input is a pitch modulation signal.

Typical Use. See Bowed Multimode module.

The default value of the following parameters is set during construction

- *Length*: the length, in meters, of the string.
- *Frequency*: fundamental frequency, in Hertz, of the string when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the length of the string. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Bowed String** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the string.
- *Inharmonicity*: detunes the partial, toward higher frequencies, with respect to the fundamental. This parameter varies between 0 and 1, where 0 represents a perfect string.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point*: x-coordinate, in meters, of bow interaction point from the extremity of the string.
- *Listening point*: x-coordinate, in meters, of listening point from the extremity of the string. The excitation and listening points should be the same in order to obtain a proper functioning.

Note: For more details on this module and especially the front panel controls, see the **Bowed Multimode** module.

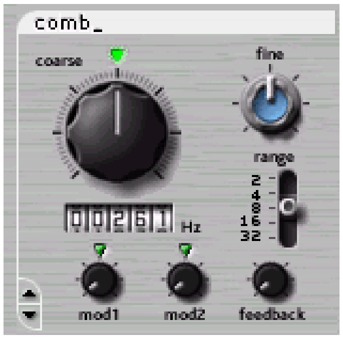
6.15 Breath Controller

This module is used to receive signal from a MIDI breath controller (MIDI controller number 2). It has no input. Its one output, the breath controller signal, lies between 0 and 1 depending on the blowing strength. This module has no front panel control.

The default value of the following parameters is set during construction

- *MIDI channel*: MIDI channel used by the breath controller.

6.16 Comb



The **Comb** filter enhances frequency components located at harmonic intervals. The frequency response of the filter is composed, as shown in Figure 29, of resonances around frequency components located at multiples of a fundamental frequency (hence its name). The effect of the filter is to color the sound with a change of apparent pitch. This module has one output, the filtered signal, and three inputs. The first input is the signal to be filtered. The second and third inputs are modulation signals used to vary the tuning (resonance frequency) of the filter.

The algorithm implemented in this module sends the input signal into a variable delay line. The output of this delay is then re-injected in this delay line with a gain factor.

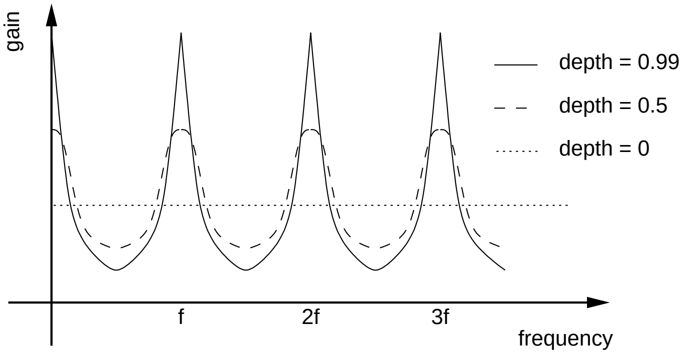


Figure 28: Frequency response of a **Comb** filter

Tuning

The *coarse* and *fine* knobs and the *range* switch are used to tune the resonance frequency of the filter. The variations in resonance frequency caused by changes in the modulation signals are relative to this level. The length of the filter delay line is calculated as the inverse of the resonance frequency (a period). When the two knobs are in their center position (green LEDs on for the *coarse* knob), the *range* switch is set to 8 and there is no modulation signal, the filter resonance frequency has a value of 261.6 Hz, which corresponds to the C3 key on a piano (middle C). The range switch transposes the resonance frequency one or two octaves up or down. The reading on the counter indicates the resonance frequency, in Hertz, of the filter. The length of the delay line can be modulated by using the modulation inputs of the module. The amount of variation of the resonance frequency obtained with the modulation inputs depends on the adjustment of the *mod1* and *mod2* gain knobs. The total modulation signal is the sum of the two inputs each multiplied by the gain corresponding to its respective *mod* knob. When the knobs are in the center position (green LEDs on), the gain equals 1 and the resonance frequency variation is 1 Volt/octave. This position is used to follow an equal temperament scale when connecting the output of a **Keyboard** module to a modulation input of a **Comb** module. The frequency variation with the modulation signal can be increased or decreased by turning the modulation knobs clockwise or anti-clockwise. Finally the *feedback* knob is used to fix the amount of "wet" signal re-injected into the delay line.

Typical Use

The **Comb** filter can be used to obtain a vibrato effect on a sound source. In the example of Figure 11, the output of the **LFO** modulates the resonance frequencies of the **Comb** filter. When the *feedback* knob is adjusted in the left position and the *mod1* knob is opened, the frequency of the filtered signal varies with the output from the **LFO**. When the *feedback* knob is in the right position, the filter may start to self-oscillate and be used as a sound source as in the patch of Figure . In this example, the output of the **Noise Mallet** is filtered and the resonance frequency of the filter is controlled with the pitch output from a keyboard. In order to obtain a tempered scale, do not forget to adjust the *mod* knob of the filter in the middle position.

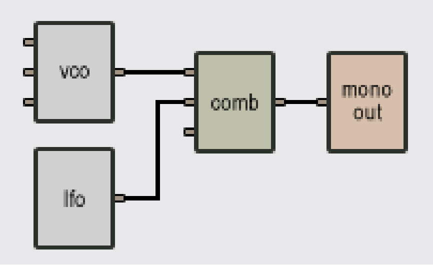


Figure 29: A vibrato effect obtained with a **Comb** filter

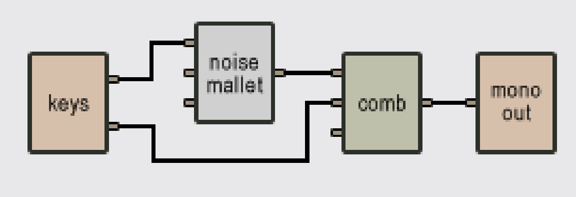
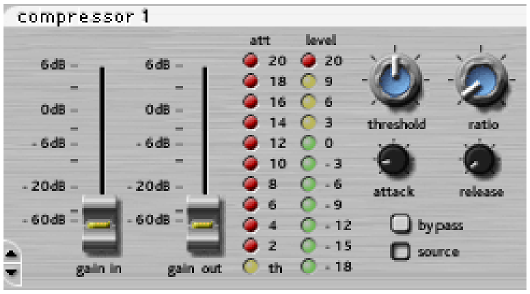


Figure 30: **Comb** and **Noise Mallet** as a sound source

6.17 Compressor



The **Compressor** module is used to automatically compress or expand the dynamics of a signal. This module has two inputs and one output. The first input is the signal to be compressed and the second input is a control signal which triggers the compression process when it rises above a given threshold. This control signal is usually the same as the input signal.

The *gain in* slider is used to adjust the level of both the input and control signal. The *gain out* slider is used to adjust the output signal of the unit. The *attenuation* level meter indicates the amount of reduction due to the gain reduction unit. It displays the difference between the level of the input signal multiplied by the input gain and the signal level after the compression. The *level* meter shows the level of the signal after the input gain or after the output gain depending on the position of the *source* button. The *threshold* and *ratio* knobs are used to control the behavior of the compression unit. The *threshold* button sets the level above which the gain reduction occurs while the *ratio* knob is used to adjust the ratio of compression to values ranging between 1:1 and 1:30. The *attack* and *decay* knobs set the amount of time it takes for the unit to respond to the variation of the control signal. The *attack* time is used when the control signal is above the threshold while the *decay* time is used when the control signal is below the threshold. Finally the *bypass* control is used to bring the compressor in or out of the audio chain. Note that this control has no effect on the output gain control which is always active.

6.18 Constant

The **Constant** module has 1 output, a constant value (DC) which is set during construction. This module has no input and no front panel control. To change the value of the constant in the *Tassman* Player, use the **Constant** module in combination with a Volume module. Its value can be positive or negative.

Typical Use

In the following example, the pitch of a **Plate** module is fixed to a constant value. The value of the constant is adjusted with the **Volume** slider.

The default value of the following parameters is set during construction

- *Output value*: value of the constant.

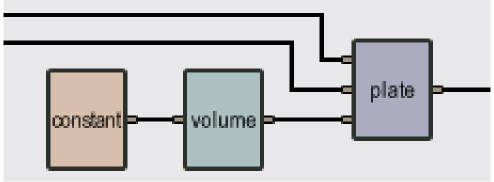
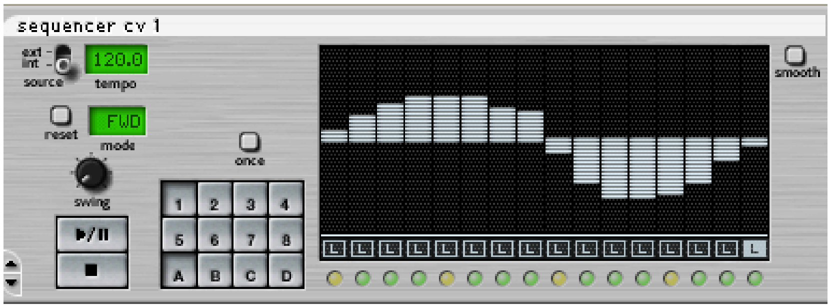


Figure 31: The pitch of a **Plate** module adjusted with a **Constant** module.

6.19 Control Voltage Sequencer



The **Control Voltage Sequencer** module enables you to record sequences of voltage. This module in itself does not produce sound but is used, usually instead of a **Keyboard** module, to control other modules such as **VCO**, **VCA** or filters. This module is a very complete 16-step sequencer, which means that it plays sequences or patterns of 16 notes in loop. Sequences can be set to have 1 to 16 steps. Because each sequence represents a bar containing four quarter notes, each step of the sequencer itself represents a sixteenth note. The module can memorize 32 different sequences between which you can switch while playing. This module has three inputs and four outputs. The first input is a sync signal which controls the tempo from an external source, the second is a start/stop input which will start the sequencer when it goes form 0 to 1 volt and stop it when it goes from 1 to 0 volt. The signal can come from another sequencer or a Keyboard. The third one is a reset input which will restart the sequence from beginning when it goes from 0 to 1 volt. The first three outputs are the same as the inputs (sync, start/stop, reset) and are used to control other sequencers. The fourth output is the control voltage signal.

This sequencer has 16 *vertical bar* sliders each controlling the output signal corresponding to a given step. The output value ranging between -1 and 1 Volt can be adjusted by click-holing and dragging or using the keyboard arrows once a bar is selected.

Creating Patterns

To create a pattern, you must first select its location. You can select it with a combination of letters (A, B, C, D) and numbers (1 to 8), on the front panel, giving you a total of 32 patterns.

The sequencer will loop each time a pattern ends. To make the sequencer stop at the end of a pattern the *once* button must be clicked.

The patterns can be played following 5 play modes using the *mode* control. Forward (FWD) plays the pattern incrementally. Backward (BWD) plays the pattern decrementally. Pendulum (PEND) plays the pattern forwards then backwards. Random 1 (RDN1) plays the pattern randomly, the same random sequence is repeated when looping. The *reset* button is used to generate a new random sequence. Random 2 (RDN2) plays the pattern randomly changing the random pattern when looping.

The *tempo* display will adjust the speed of the pattern. The *ext/int* switch will determine if it is the internal clock (int) that sets the tempo or an external source (ext) such as another sequencer or a **Sync Lfo**. The swing knob will introduce a swing feel to the rhythm of the pattern.

The *loop* buttons are used to set the length of the Pattern from 1 to 16 steps.

It is possible to draw across the bars by holding down the Shift key (Windows) or Option key (Mac OS). Also, by holding the Ctrl key (Windows) or Apple key (Mac OS) the value of every bar will be offseted. Holding the Shift key (Windows) or the Option key (Mac OS) will draw a line across the bars.

The *smooth* button enables the control voltage output to be gradually changed to the value of the next step.

Typical Use

In this example, the **CV Sequencer** is used to control the cutoff frequency of a **Vlowpass4** filter.

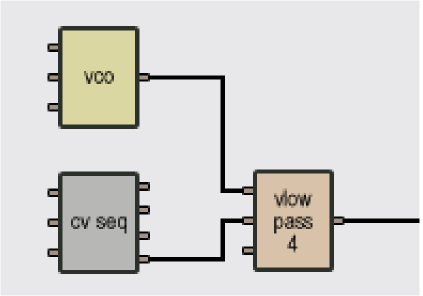
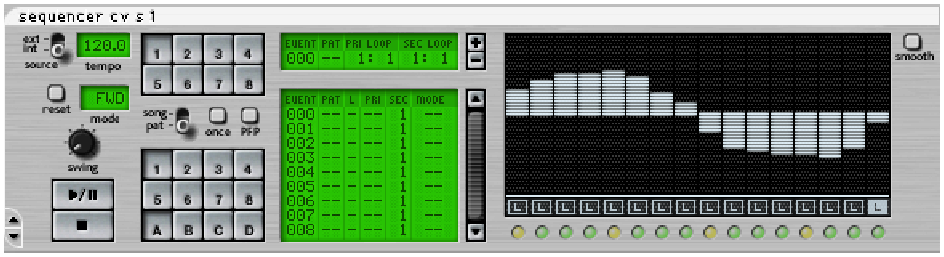


Figure 32: **CV Sequencer** controlling a **Vlowpass4** filter.

Note: see also **Multi-Sequencer**, **Control Voltage Sequencer with Songs**, **Single Gate Sequencer**, **Single Gate Sequencer with Songs**, **Dual Gate Sequencer** and **Dual Gate Sequencer with Songs**.

6.20 Control Voltage Sequencer with Songs

This module is the same as the **Control Voltage Sequencer** but with song mode added. For more information about the song mode, please refer to the **Multi Sequencer** module documentation.



Note: see also **Multi-Sequencer**, **Control Voltage Sequencer**, **Single Gate Sequencer**, **Single Gate Sequencer with Songs**, **Dual Gate Sequencer** and **Dual Gate Sequencer with Songs**.

6.21 Damper

This module is used to receive signal from a MIDI sustain pedal (MIDI controller number 64). It has no input and one output, the damper signal, which is equal to 1 when the sustain pedal is depressed and to 0 when it is released. This module has no front panel control.

Typical Use

The following patch is used to reproduce the behavior of a piano damper pedal with a MIDI sustain pedal. This combination of modules is often used with **Multimode** objects.

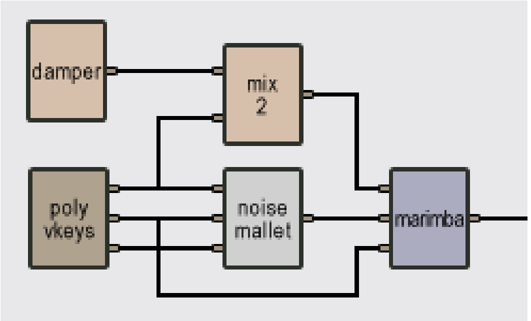
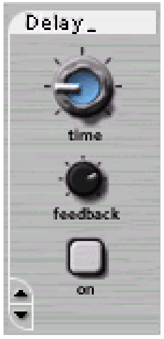


Figure 33: A **Damper** module used to play **Marimba**

The default value of the following parameter is set at construction

- *MIDI channel*: MIDI channel used by the sustain pedal.

6.22 Delay



The **Delay** module is a feedback loop with a variable delay in the feedback. There is one input and one output. The input signal is sent into the feedback loop. The output is the sum of the input signal and the returning signal from the feedback loop. The duration of the delay can be adjusted, with the *time* knob (between 10 ms and 1.5 s). The *feedback* knob sets the gain in the feedback loop (values between 0 and 1). If the *on/off* switch is in the off position, the module passes the input on with no effect. The following figure shows the effect of the **Delay** module on a pulse signal with a2/a1 = feedback gain.

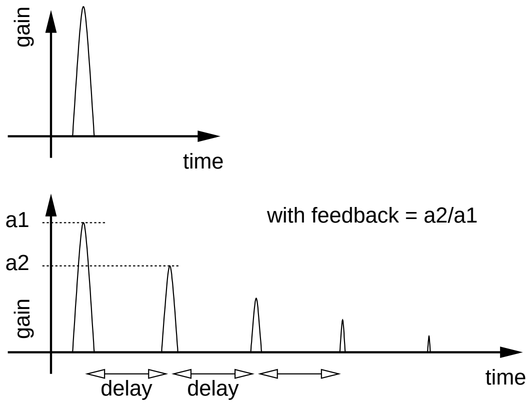


Figure 34: Effect of **Delay** Module.

Typical Use

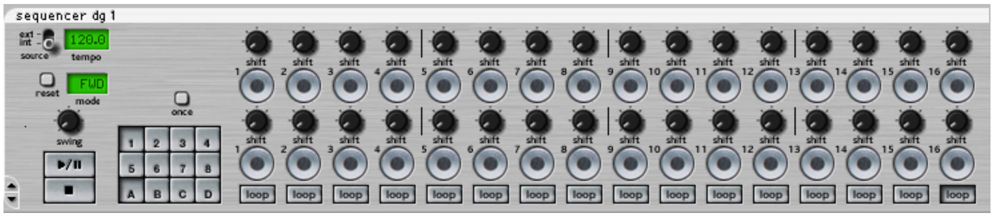
The **Delay** module is used to produce an echo effect when the delay is long (>100ms) or to color the sound when the delay time is short (< 100 ms).

The default value of the following parameters is set during construction

- *delay time*: time delay, in seconds, applied to the input signal (between 1 ms and 1.5 s).
- *feedback*: gain applied to the delayed signal (values between 0 and 1).

Note: See also **Sync Delay** and **Sync Ping Pong Delay**.

6.23 Dual Gate Sequencer



The **Dual Gate Sequencer** module enables you to record two sequences of gates at the same time. This module in itself does not produce sound but is used, usually instead of a **Keyboard** module, to trig other modules such as **Player** or drum sounds. This module is a very complete 16-step sequencer, which means that it plays sequences or patterns of 16 notes in loop. Sequences can be set to have 1 to 16 steps. Because each sequence represents a bar containing four quarter notes, each step of the sequencer itself represents a sixteenth note. The module can memorize 32 different sequences between which you can switch while playing.

This module has three inputs and five outputs. the first input is a sync signal which controls the tempo from an external source, the second is a start/stop input which will start the sequencer when it goes form 0 to 1 volt and stop it when it goes from 1 to 0 volt. The signal can come from another sequencer or a Keyboard. The third one is a reset input which will restart the sequence from beginning when it goes from 0 to 1 volt. The first three outputs are the same as the inputs (sync, start/stop, reset) and are used to control other sequencers. The fourth and fifth outputs are gate signals which can be used as control sources to trigger other modules.

This sequencer has 2 sets of 16 *gate* buttons. Each has its own gate output that will generate a square pulse of 1/8 of a quarter note for each active *gate* buttons. The two sets of 16 *shift* knobs delay the output of their respective gate. The *loop* buttons are used to set the length of the pattern from 1 to 16 steps.

Creating Patterns

To create a pattern, you must first select its location. You can select it with a combination of letters (A, B, C, D) and numbers (1 to 8), on the front panel, giving you a total of 32 patterns. The sequencer will loop each time a pattern ends. To make the sequencer stop at the end of a pattern, the *once* button must be clicked.

The patterns can be played following 5 play modes using the *mode* control. Forward (FWD) plays the pattern incrementally. Backward (BWD) plays the pattern decrementally. Pendulum (PEND) plays the pattern forward then backward. Random 1 (RDN1) plays the pattern randomly, the same random sequence is repeated when looping. The *reset* button is used to generate a new random sequence. Random 2 (RDN2) plays the pattern randomly changing the random pattern when looping.

The *tempo* display will adjust the speed of the pattern. The *ext/int* switch will determine if it is the internal clock (int) that sets the tempo or an external source (ext) such as another sequencer or a **Sync Lfo**.

The *swing* knob will introduce a swing feel to the rhythm of the pattern. The *gate* buttons control the gate output. The gate output will generate a square pulse of 1/8 of a quarter note for each active *gate* buttons. To hear a step, the gate button must be clicked (green light on). You have two sets of gate buttons, one for each pattern.

The *loop* buttons are used to set the length of the Patterns from 1 to 16 steps. Note that the loop point is set for both patterns at the same time. The *shift* knobs delays the output of their respective gate.

Typical Use

In this example, one **Dual Gate sequencer** is used to control two **Player** modules.

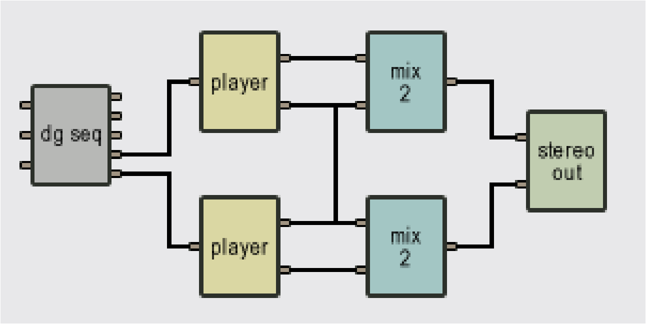
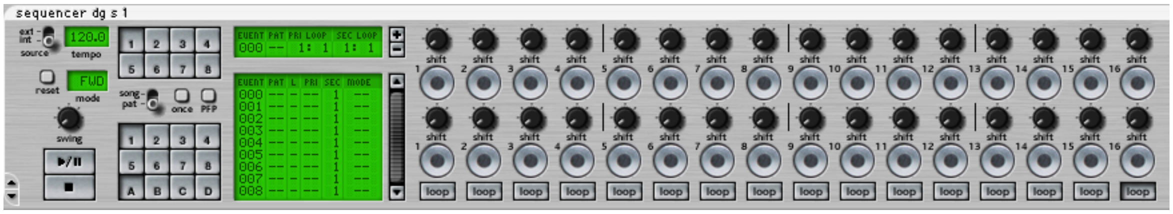


Figure 35: **Dual Gate Sequencer** controlling two **Player** modules.

Note: see also **Multi-Sequencer**, **Control Voltage Sequencer**, **Control Voltage Sequencer with Songs**, **Single Gate Sequencer**, **Single Gate Sequencer with Songs** and **Dual Gate Sequencer with Songs**.

6.24 Dual Gate Sequencer with Songs

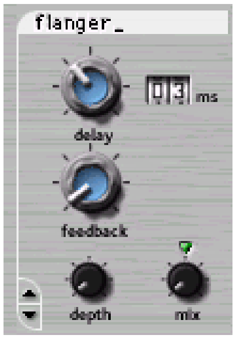


```
EVENT PAT PRI LOOP  SEC LOOP
000   --    1: 1    1: 1
```

```
EVENT PAT L PRI SEC MODE
000   --  - --   1  --
001   --  - --   1  --
002   --  - --   1  --
003   --  - --   1  --
004   --  - --   1  --
005   --  - --   1  --
006   --  - --   1  --
007   --  - --   1  --
008   --  - --   1  --
```

This module is the same as the **Dual Gate Sequencer** but with song mode added. To read more about song mode, please refer to the **Multi Sequencer** module documentation.

6.25 Flanger



The **Flanger** module implements the effect known as “flanging” which colors the sound with a false pitch effect caused by the addition of a signal of varying delay to the original signal. This module has two inputs and one output. The first input is the audio signal to be flanged and the second input is a modulation signal that varies the delay and affects the apparent pitch. The output is the flanged signal.

The algorithm implemented in this module is shown in Figure 36. The input signal is sent into a variable delay line. The output of this delay is then mixed with the “dry” signal and re-injected into the delay line with a feedback coefficient.

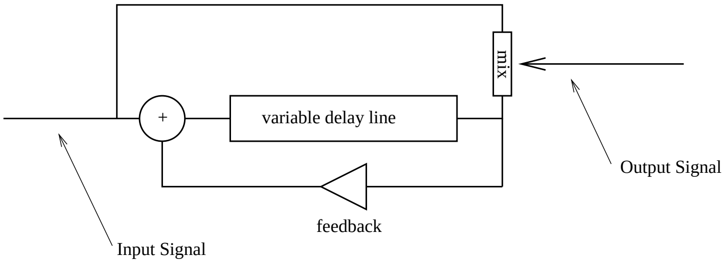


Figure 36: **Flanger** algorithm.

The effect of the **Flanger** module is to introduce rejection in the spectrum of the input signal at frequencies located at odd harmonic intervals of a fundamental frequency as shown in Figure 37. The location of the fundamental frequency $f0$ and the spacing between the valleys and peaks of the frequency response is determined by the length of the delay line ($f0$ = 1/(2delay)), the longer the delay, the lower is $f0$ and the smaller the spacing between the harmonics while decreasing the delay increases $f0$ and hence the distance between the harmonics.

The amount of effect is determined by the ratio of “wet” and “dry” signal mixed together as shown in Figure 38. As the amount of “wet” signal sent to the output is increased, the amount of rejection increases. Finally, the shape of the frequency response of the **Flanger** module is also influenced by the amount of “wet” signal re-injected into the feedback loop as shown in Figure 39. Increasing the feedback enhances frequency components least affected by the delay line and located at even harmonic intervals of the fundamental frequency. As the feedback is increased, these peaks become sharper resulting in an apparent change in the pitch of the signal.

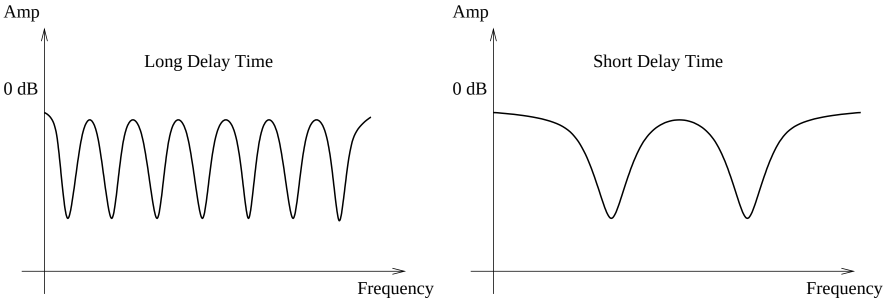


Figure 37: Frequency response of a **Flanger** module. Effect of the length of the delay line.

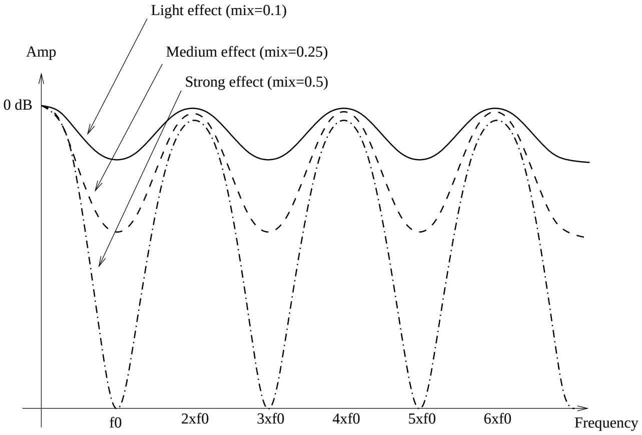


Figure 38: Effect of the mix between "wet" and "dry" signal on the frequency response of a **Flanger** module

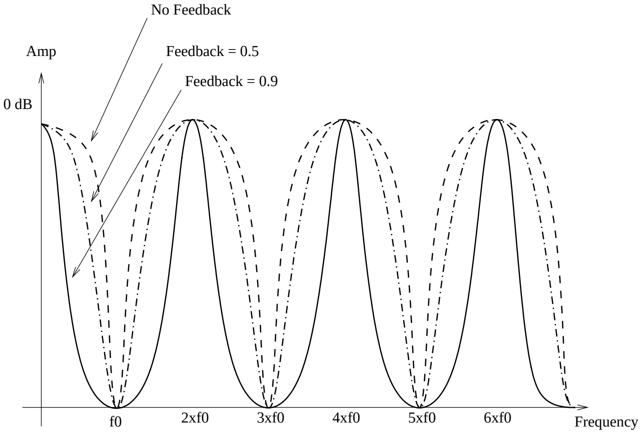


Figure 39: Effect of the amount of feedback on the frequency response of a **Flanger** module.

Tuning

The delay length is adjusted with the *delay* knob and is displayed, in milliseconds, in the counter next to the knob. The length of this delay can be modulated by using the second input of the module, the amount of modulation depending on the adjustment of the *depth* knob. In the left position, there is no modulation and the delay line remains fixed while in the right position, with a modulation signal varying between [-1,1] Volt, the delay line varies between 0 and twice the value set with the *delay* knob. The *feedback* knob is a gain knob used to fix the ratio of "wet" signal re-injected into the delay. Finally, the mix knob determines the amount of "dry" and "wet" signal sent to the output. When this knob is adjusted in the left position, only "dry" signal is sent to the output, in its center position (green LED On), there is an equal amount of "dry" and "wet" signal in the output and in the right position, only "wet" signal is sent to the output.

Typical Use

The output from a **LFO** module can be used to control the filtering of a signal (the output of a **VCO** for example) with a **Flanger** module. Different type of effects can be obtained with different settings of the **Flanger** module.

- A chorus effect is obtained by using a delay of roughly 60 ms, a feedback value of 0.4, a low frequency modulation signal (5 Hz) with *depth* adjusted to 0.3 and *mix* adjusted to a value of 0.4.
- A pure flanger effect is obtained with the following settings: a short delay length (5 ms), much feedback (0.75), a low frequency modulation signal (between 0 and 2 Hz) with the *depth* knob in the right position and a half and half mix between "wet" and "dry" signal (*mix* knob in center position).
- A Vibrato effect is obtained with 45 ms delay (*delay* knob in center position), no feedback (*feedback* knob in left position), a low frequency modulation signal (5Hz) with the *depth* knob adjusted to its center position and sending only the "wet" signal to the output (*mix* knob in right position).

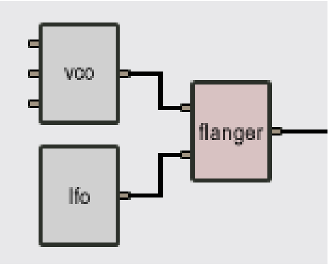
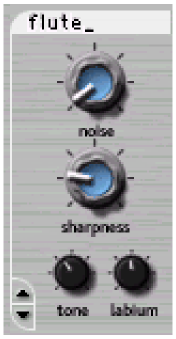


Figure 40: **Flanger** module with **LFO** module

The default value of the following parameters is set during construction

- *delay*: time delay, in seconds, applied to the input signal (values between [0, 92]ms).
- *feedback*: coefficient,[0, 1[, determining amount of “wet” signal re-injected into the delay line. If feedback = 0 there is no “wet” signal re-injected while if feedback = 0.99, maximum of “wet” signal re-injected.
- *depth*: gain coefficient, [0,1], multiplying the modulation signal. mix: amount of “dry” and “wet” signal sent to output. If mix = 0 there is only dry signal while if mix =1, there is only “wet” signal.

6.26 Flute



The **Flute** module simulates sound production by a flute having the geometry of a recorder. This flute is simulated using models of the air flow in the embouchure, wave propagation in the body of the instrument and note changes with different combinations of open and closed keyholes. Since this module simulates a real instrument, it also has the same range (C3 to G5). This module has 3 inputs and one output. The first input is a gate signal, generally that from a **Keyboard**. The second input is the driving pressure signal and is generally connected to the output from an **ADSR** module, the output from a **Breath Controller** module or the gate signal from a **Keyboard** module. Finally, the third input is a pitch signal generally connected to the pitch output from a **Keyboard**. The output signal is the sound produced by the instrument.

Four parameters can be adjusted while playing. The *noise* knob sets the amount of turbulence noise in the sound. The *tone* knob controls the jet behavior which affects the tone color of the recorder sound. The *labium* knob sets the position of the edge of labium of the recorder relative to the jet. In its center position, the jet blows exactly in front of the recorder labium. Finally the *sharpness* knob controls the sharpness of the edge of the labium (this parameter is adjusted by recorder makers since it affects the color of the tone produced by the instrument).

Typical Use

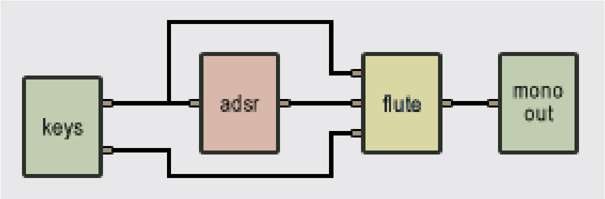
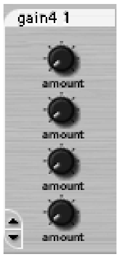


Figure 41: A **Flute** module controlled with a **Keyboard**.

In the following example, a Flute module is controlled with a **Keyboard** module. The **ADSR** is used to shape the driving pressure signal.

Note: For polyphonic flute-like sounds, use the **Organ** module.

6.27 Gain, Gain 2, Gain 3, Gain 4



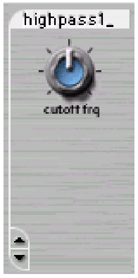
The **Gain**, **Gain 2**, **Gain 3** and **Gain 4** knob modules have respectively one to four inputs and one to four outputs. They are used to adjust the amplitude of a signal. The output signal is the input signal multiplied by a constant varying between 0 and 2 (+6dB).

Typical Use

The **Gain** modules are used whenever the level of a signal must be adjusted.

Note: See also **Slider** and **Volume**.

6.28 Highpass1



The **highpass1** module is a first order high-pass filter (-6dB/octave). Its one input is the signal to be filtered, its one output the filtered input signal.

Tuning the filter

The *cutoff frq* knob tunes the cutoff frequency of the filter to the desired level.

The default value of the following parameter is set at construction

- *Cutoff frequency*: value of the filter cutoff frequency.

6.29 Inlets (1-12)

These modules are used to define the inputs of a sub-patch so that it can be connected in another patch. These modules have no input but between 1 and 12 outputs which are connected to inputs in the sub-patch which you want to connect to outputs in another construction. These inputs will correspond to the inputs of the sub-patch icon which will appear in the construction window when later you include this sub-patch in another construction. These modules have no front panel.

Sub-patches may have between 0 and 12 inputs and 0 and 12 outputs but they must always have at least one input or output. As soon as an **Inlet** or **Outlet** module is included in a patch, the *Tassman* Builder will consider that you want to define the current patch as a sub-patch and will save it as so in the **Sub-Patches** folder of the Browser. You can then use it just like any other module.

Typical Use

A sub-patch is created with an **Inlet** or **Outlet** module or both. The inputs of the sub-patch are determined by connecting them to an Inlet module. In the example of Figure 24, a stereo-reverb sub-patch having two inputs and two outputs is created.

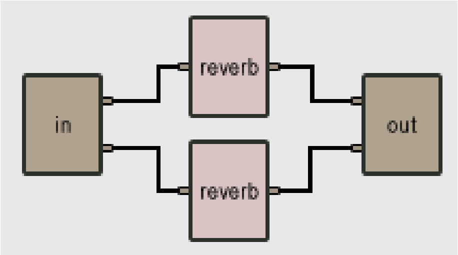


Figure 42: A reverb sub-patch.

Note: See also the **Outlet (1-12)** modules.

6.30 Inverter

The **Inverter** module has no front panel control. It has one input and one output. It is used to invert the phase of the input signal. In other words, the output is the input signal multiplied by -1.

Typical Use

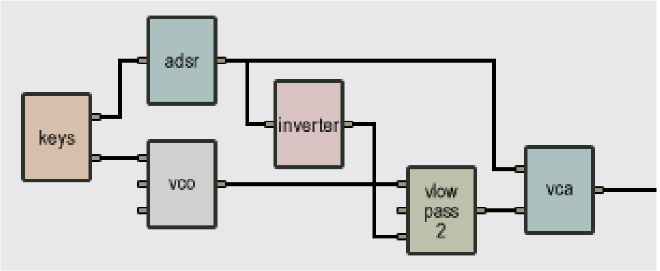


Figure 43: Modulation of the cutoff frequency of a filter using an **Inverter**.

This module is to invert the control voltage generated by an **ADSR** so that the cutoff frequency of a **VCF** module first goes down when triggering a new note as shown in Figure 43. The inverter can also be used to obtain a stereo tremolo effect (amplitude modulation) as illustrated in Figure 44.

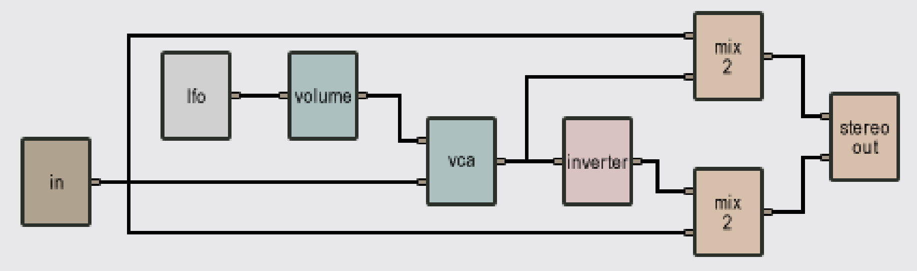
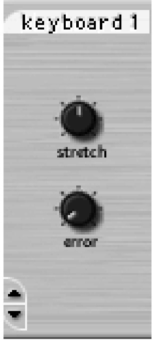


Figure 44: Stereo tremolo effect.

6.31 Keyboard



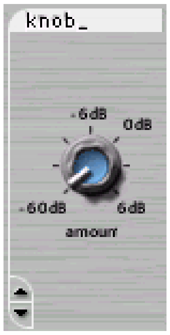
The **Keyboard** module simulates the outputs of a classic monophonic analog high-note priority keyboard. It has no input and two outputs. The first output is the gate signal. It is equal to 0 Volt when no key is played, and 1 Volt when one or more keys are played. The second output signal is the pitch signal. Its value corresponds to the highest key played when one or more keys are depressed and to the last key played when no key is depressed. The pitch signal varies by ±1 Volt per octave which implies a change of 1/12 Volt for a pitch variation of 1 semitone. The pitch signal is calculated with respect to the C3 key (middle C) which outputs a value of 0 Volt. This means that, for example, the C2 key signal is -1 Volt and that of the C4 key is +1 Volt. The *stretch* knob on the interface is used to simulate stretched tuning used on instruments such as pianos. Turned to the left, low notes will be tuned higher and high notes lower (inner stretch); turned to the right, low notes will be tuned lower and high notes will be higher (outer stretch). In the center position, the tuning will be equal. The *error* knob introduces some randomness in the pitch signal. Turned to the left, no error is outputted and the pitch signal is perfect; as the knob is turned to the right, errors will start to appear causing small fluctuations in pitch. The effect of this knob is to simulate pitch variations found in analog synths.

The default value of the following parameters is set at construction

- *pitch wheel range*: determines the range of pitch variations that can be obtained with the pitch wheel. The convention is 1 Volt/octave (maximum value is 2 Volts). A semitone is equal to a 0.08333 value.
- *MIDI channel*: MIDI channel used by the keyboard.

Note: see also the **Vkeyboard** and **Polykey** and **Polyvkey** modules.

6.32 Knob



The **Knob** module is used to adjust the amplitude of a signal. It acts in the same way as the **Slider** module. It has one input and one output. The output signal is the input signal multiplied by a constant varying between 0 and 2 (+6dB).

Typical Use

The **Knob** module is used whenever the level of a signal must be adjusted.

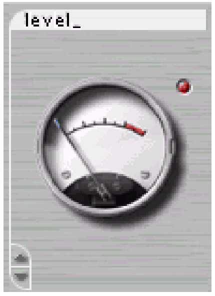
The default values of the following parameter is set at construction

- *gain*: default value of the volume gain (value between 0 and 2).

6.33 LESS

The **LESS** module performs the comparison between its two inputs. The one output of this module is either 1 (true) or 0 (false) depending on the values sent to the two inputs. The output is true (1) if the first input is lower than the second and false (0) otherwise.

6.34 Level

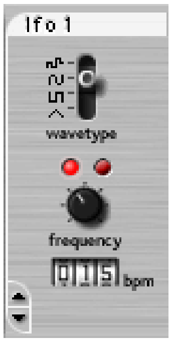


The **Level** module is a VU Meter used to show the RMS (root mean square) value of a signal (1 Volt full scale). This module has one input and no output. The red sector indicates the saturation zone of the signal which also triggers the red LED (see saturation curve of the **Audio Out** module).

Typical Use

A Level module is generally used to monitor the audio signal sent to the computer sound card through the **Audio Out** module as shown in Figure 5 under **Audio Out**.

6.35 LFO (Low Frequency Oscillator



The **LFO** module has no input and one output. The output is a periodic signal with frequency varying between 0.1 and 35 Hertz depending on the setting of the *frequency* knob. The oscillation of the two red LEDs on the front panel give an indication of the output frequency. The wave shape is set by the *wavetype* switch. Waveforms include triangle, square random and sine. The amplitude of the output signal is ±1 Volt.

Typical Use

The **LFO** is often used to generate tremolo (amplitude modulation) as shown in Figure 27 or vibrato (frequency modulation) as in Figure 28. Two **LFO** modules with a **Sample & Hold** can be used to generate a random signal as illustrated in Figure 53 under **Sample & Hold** module.

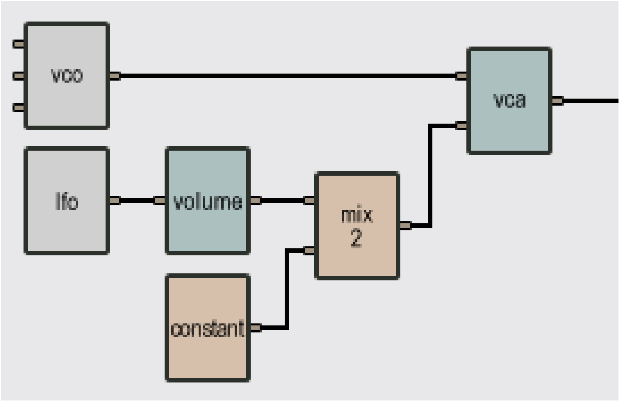


Figure 45: **LFO** used for tremolo.

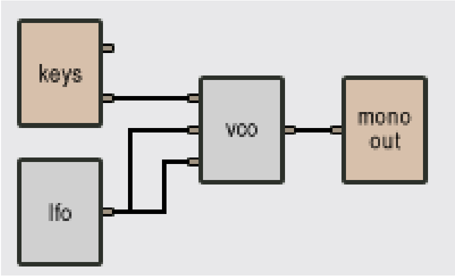
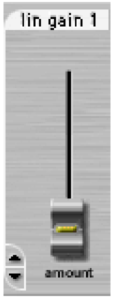


Figure 46: **LFO** used for Vibrato and Pulse Width modulation.

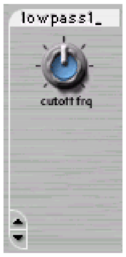
6.36 Lin Gain



This module is used to modify the amplitude of a signal. It has one input, the signal to be adjusted, and one output, the adjusted signal.

The amplitude of the signal is a controlled with the *amount* slider on the front panel. The output signal is the input signal multiplied by a gain having a value between the min and max range set in the dialog of the module in the Builder.

6.37 Lowpass1



The **Lowpass1** module is a first order low-pass filter (-6dB/octave). Its one input is the signal to be filtered, its one output is the filtered input signal.

Tuning the filter

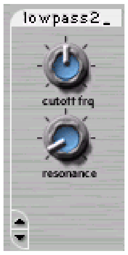
The *cutoff frq* knob tunes the cutoff frequency of the filter to the desired level.

The default value of the following parameter is set at construction

- *Cutoff frequency*: value of the filter cutoff frequency.

Note: see also **Lowpass2**, **Vlowpass2** and **Vlowpass4**.

6.38 Lowpass2



The **Lowpass2** module is a second order low-pass filter (-12dB/ octave). Its one input is the signal to be filtered, its one output is the filtered input signal.

The *cutoff frq* knob tunes the cutoff frequency of the filter to the desired level. The *resonance* button is used to adjust the resonance of the filter around the cutoff frequency as shown in Figure 47.

The default value of the following parameters is set at construction

- *Cutoff frequency*: value of the filter cutoff frequency.
- *Resonance*: resonance of the filter around its cutoff frequency.

Note: see also **Lowpass1**, **Vlowpass2** and **Vlowpass4**.

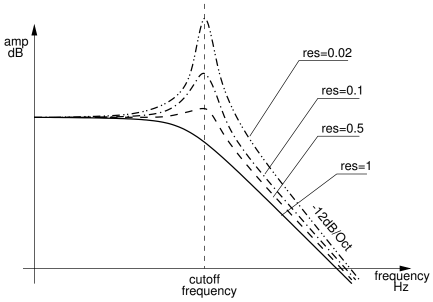
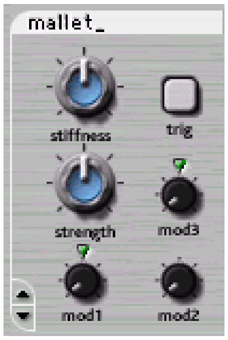


Figure 47: Frequency response of a **Lowpass2**.

6.39 Mallet



The **Mallet** module is used to simulate the force impact produced by a mallet striking a structure. It is usually used in combination with acoustic objects such as the **Beam**, **Membrane**, **Plate** and **String** modules in order to play them. The force of the impact is adjusted with the *strength* knob while the stiffness of the mallet (related to its material) is varied with the *stiffness* knob. Figure 30 shows the effect of the adjustment of the stiffness on the output signal. As the stiffness is increased the excitation signal becomes narrower. The effect of the *strength* parameter which determines the amplitude of the impact is also shown in the same figure.

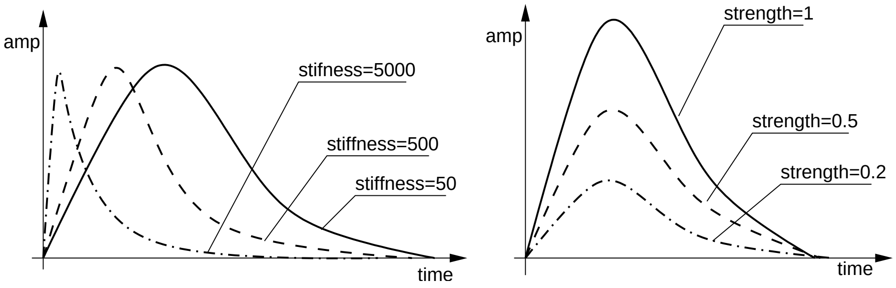


Figure 48: Effect of *stiffness* and *strength* knob on **Mallet** output.

This module has one output, the impact signal, and four inputs. The first input triggers the mallet every time a low-to-high transition is encountered in the input signal. This input is usually connected to the gate signal from a **Keyboard** module. Note that the mallet can also be triggered manually by using the *trig* button on the front panel. The second input signal modulates the stiffness of the mallet relative to the value selected with the *stiffness* knob. The amplitude of the modulation is adjusted with the *mod1* knob. The greater the amplitude, the greater the stiffness. This modulation input is used, for example, when a variation of the stiffness of the mallet with the note played is desired. When the knob is adjusted in its center position and when this input is connected to a pitch signal, the stiffness exactly follows the pitch variation so as to ensure that the spectral content (or color) of the sound produced by a structure is uniform when the pitch is varied. The third input also modulates the stiffness, but in the reverse manner as for the second input so that the stiffness of the mallet decreases when the input signal increases. This input is usually connected to the velocity output from a keyboard module which implies that the mallet will soften as the impact velocity increases. This is a behavior one observes, for example, on piano hammer heads due to the non-linearity of the felt. The amplitude of this input is adjusted with the *mod2* gain knob. The last input modulates the strength of the impact relative to the adjustment of the *strength* knob. This input is also generally connected to a velocity signal so as to increase the force of the impact with the velocity signal. The amplitude of this modulation signal is adjusted with the *mod3* gain knob.

Typical Use

A **Mallet** module is generally used to excite **Multimode** objects such as **Beams**, **Membranes**, **Strings** and **Plates**. See the example in Figure 51 under **Multimode**.

The default value of the following parameters is set at construction

- *Strength*: default value of the impact force (value between 0 and 2).
- *Stiffness*: default value of mallet stiffness (value between 1 and 20 000).

Note: see also **Noise Mallet**.

6.40 Marimba

The **Marimba** module simulates sound production by marimba bars of different material and sizes. This module reproduces the characteristic tuning of marimba bars overtones obtained with the deep arch-cut of the bars. This module, which constitutes a special case of the **Beam** module, first calculates the modal parameters corresponding to marimba bar shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Multimode** module to simulate the sound. The module has one output, the sound produced by the marimba bar, and three inputs. The first input signal is a damping signal which, depending on its value, lowers or raises dampers on the structure. When the input signal is equal to 0, dampers are lowered on the bar, thus shortening the decay time of the sound produced by the structure. When the signal is greater than 0, dampers are raised. Note that this damping adds to the natural damping of the marimba bar itself. If this input is not connected to any other module, the default value is set at 0, which implies that the marimba bar motion will be damped. This input is, therefore, usually connected to a **Constant** module to obtain undamped motion or to a **Damper** module or the gate signal from a keyboard in order to vary the damping while playing. The second input signal is the force signal exciting the marimba bar, while the third is a pitch modulation signal.

The default value of the following parameters is set at construction

- *Length*: the length, in meters, of the beam.
- *Frequency*: fundamental frequency, in Hertz, of the beam when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the length of the beam. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz, which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Marimba** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the beam.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point*: x-coordinate, in meters, of impact point from the extremity of the beam.
- *Listening point*: x-coordinate, in meters, of listening point from the extremity of the beam.

Note: For more details on this module and especially the front panel controls, see the **Multimode** module.

6.41 Master Recorder Trig

The **Master Recorder Trig** is used to trig the **Master Recorder** in the output stage of the Player. Its one input expects a gate signal such as the start/stop signal from a **Sequencer**. When the switch of the **Master Recorder** is in the *gate* position, recording will start whenever a low-to-high transition occurs in the gate signal; it will stop when a high-to-low transition is received. In the *trig* position, the recording will start when a low-to-high transition occurs in the gate signal and will continue until the *stop* button of the **Master Recorder** is pressed. When the switch is in the *none* position, the gate signal from the **Master Recorder Trig** is ignored and the **Master Recorder** is controlled by using the *record* and *stop* buttons.

Typical Use

In Figure 49, a **Master Recorder Trig** module is used in combination with a **Master Sync Input** and a **Sequencer** to cut a perfect loop. In this example, the **Sequencer** is triggered by the gate signal from the **Master Sync Input** when the *play* button of the **Master Sync** module is pressed. When it receives the triggering signal, the **Sequencer** sends a start signal to the **Master Recorder** module followed by a stop signal when the end of the sequence is reached. In order to respond to the signal sent by the **Sequencer**, remember to adjust the **Master Recorder** switch in the *trig* or *gate* position depending on if you want the recorder to react to the start signal only or both the start and stop signals. Since the **Sequencer** is using the clock from the **Master Sync Input** module, adjust its clock source switch to *ext*. Finally use the *pat* mode of the **Sequencer** and press on the *once* button in order to make the **Sequencer** stop at the end of the sequence. Note that the pitch output from the **Sequencer**, connected in this example to a **VCO** module, could be used to control any other modules you would like to record.

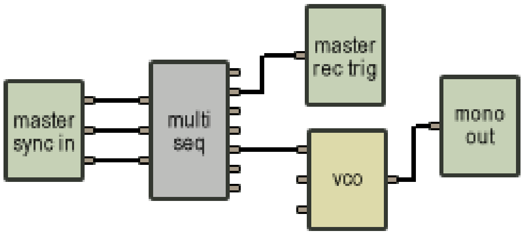


Figure 49: A **Master Recorder Trig** is used with a **Master Sync Input** and a **Sequencer** to cut a perfect loop.

Note: See the documentation of the **Recorder** of the output effect stage in Section 4.8.

6.42 Master Sync Input

The **Master Sync Input** module is used to route the synchronization signals generated by the **Sync** module of the output stage to other modules such as a **Sequencer**. The first output of the module is the sync signal itself. When the source switch of the **Sync** module of the output stages is set to *ext*, the clock signal will be that from a host sequencer when the *Tassman* is used as a plug-in or, in standalone mode, the clock signal received on the MIDI channel selected in the Player toolbar. When the *int* source is chosen, the clock is adjusted, in beats per minute, in the green *tempo* display of the **Sync** module. The second output is a gate signal equal to 1 volt when the *play* button of the **Sync** module is pressed and 0 volt when it is inactive. This signal can be used by any module that needs to be triggered such as a **Sequencer**, a **Player** or a **Master Recorder Trig** module. The third output of the module is a reset signal generated when the reset button of the **Sync** module is pressed and that can be used to re-sync modules that have a reset input.

Typical Use

In Figure 50, a **Master Sync Input** module is used to control a Sequencer. This module can also be used to cut perfect loops when used in conjunction with a Master Recorder Trig as shown in Figure 49.

Note: See the documentation of the **Sync** module of the output effect stage in Section 4.8.

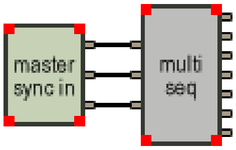


Figure 50: A **Master Sync Input** is used to synchronize a **Multisequencer** module.

6.43 Membrane

The **Membrane** module simulates sound production by rectangular membranes of different materials and sizes. This module first calculates the modal parameters corresponding to membrane shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Multimode** module to simulate sound production by this object. The module has one output, the sound produced by the membrane, and three inputs. The first input signal is a damping signal which, depending on its value, lowers or raises dampers on the structure. When the input signal is equal to 0, dampers are lowered on the membrane, thus shortening the decay time of the sound produced by the structure. When the signal is greater than 0, dampers are raised. Note that this damping is in addition to the natural damping of the membrane itself. If this input is not connected to any other module, the default value is set at 0, which implies that the membrane motion will be damped. This input is, therefore, usually connected to a **Constant** module to obtain undamped motion or to a **Damper** module or the gate signal from a keyboard in order to vary the damping while playing. The second input signal is the force signal exciting the membrane, while the third is a pitch modulation signal.

The default value of the following parameters is set at construction

- *Length*: the length, in meters, of the membrane.
- *Width*: the width, in meters, of the membrane.
- *Frequency*: fundamental frequency, in Hertz, of the membrane when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the size of the membrane. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz, which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Membrane** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the membrane.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point-x*: x-coordinate, in meters, of impact point from the lower left corner of the membrane.

- *Excitation point-y*: y-coordinate, in meters, of impact point from the lower left corner of the membrane.
- *Listening point-x*: x-coordinate, in meters, of listening point from the lower left corner of the membrane.
- *Listening point-y*: y-coordinate, in meters, of listening point from the lower left corner of the membrane.

Note: For more details on this module and especially the front panel controls, see the **Multimode** module.

6.44 Mix2, Mix3, Mix4 and Mix5

These modules are used to mix signals together with no relative gain coefficient. The number of inputs can vary between two and five, depending on the module chosen. The output is the sum of the inputs. This module has no front panel control.

Typical Use

A **Mix3** module is used for mixing the output of three filters in the example of Figure 26 under Bandpass2. In Figure 35 under **Damper**, a **Mix2** module is used to perform a logical OR function of two gate signals (output is on when either input is on) in order to reproduce the behavior of a piano damper pedal.

6.45 Modulation Wheel

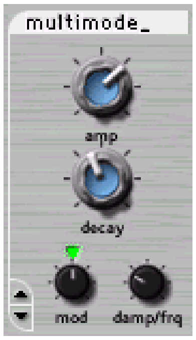
This module is used to receive signal from a MIDI modulation wheel (MIDI controller number 1). It has one output, the modulation wheel signal, which lies between 0 and 1, depending on the modulation wheel position. This module has no input and no front panel control.

The default value of the following parameter is set at construction

- *MIDI channel*: MIDI channel used by the modulation wheel.

6.46 Multimode

The **Multimode** module is used to simulate mechanical objects such as strings, plates, membranes, beams. The output of this module is the acoustic signal that would be produced by these objects given a certain geometry, material, type of excitation, listening point and damping. The functioning of this module is based on modal analysis. This technique is well-known in areas of physics and mechanics and is used to describe complex vibrational motion using modes (elementary oscillation patterns which can be used to decompose a complex motion). By adding together modes of different frequencies, amplitude and damping, one can reproduce the behavior of different type of structures. The accuracy of the resulting signal depends on the number of modes used in the simulation.



The **Multimode** module is not directly accessible to the user. Rather, other modules such, as **String**, **Plate**, **Beam**, **Marimba** and **Membrane**, use the **Multimode** as their front. These other modules first calculate the different modal parameters corresponding to their respective structure type as requested at construction time and, next, call the **Multimode** module in order to implement the parameters they require. Since these different object types are based on the same underlying simulation technique, they all have the same number of inputs and outputs and share the same controls (which appear on their front panel) for adjusting their physical properties.

Amplitude

The amplitude control is simply a gain which controls the amplitude of the output signal. It can be adjusted with the *amp* knob on the front panel.

Decay

The damping of an object affects the decay time of the sound produced by the object. This parameter is adjusted using the *decay* knob on the front panel. When the knob is turned left, the damping is high and the decay time short; damping is low and decay time is long when the knob is turned right. The damping is characteristic of the material of the object. Damping in wood, for example, is high and the decay time is short (knob turned to left) and in steel damping is lower and, therefore, decay time is longer (knob turned to right). But damping also varies for a given material depending on how the object is used or connected to other objects. The oscillation of a string, for example, has a much shorter decay time when used on a violin than on a mandolin.

Playing frequency

The frequency of the sound produced by an object is dependent on its “useful” size. A large metal plate, for example, produces a sound with a lower pitch than does a smaller one. The pitch of the output of a **Multimode** object is determined by the signal entering the pitch input appearing on every such object. In other words, the size of the object is varied in order to obtain the requested pitch. The *mod* knob is a gain knob affecting the amplitude of the pitch input signal. When in the center position (green LED on), the gain equals 1 and the pitch variation is equal to 1 Volt/octave. This position is used to play an equal temperament scale when connecting the pitch output of a **Keyboard** to the pitch signal input of a **Multimode** object.

Damping vs Frequency

In a mechanical structure, the damping, or decay time, varies for the different frequency components of the oscillating motion. The variation of the damping with frequency is another characteristic of the material of a structure and is adjusted, in a **Multimode** object, with the *damp/frq* knob on the module front panel. In the left position, the decay time of low frequencies is shorter than that of high frequencies; in the right position it is longer. As a rule of thumb, steel and glass are found in the left position, nylon in the center position, and wood in the right position.

Typical Use

These modules are often used with a **Mallet** or **Noise Mallet** module, as shown in Figure 51. The different **Multimode** modules can be cascaded to simulate coupling between structures as illustrated by the following example.

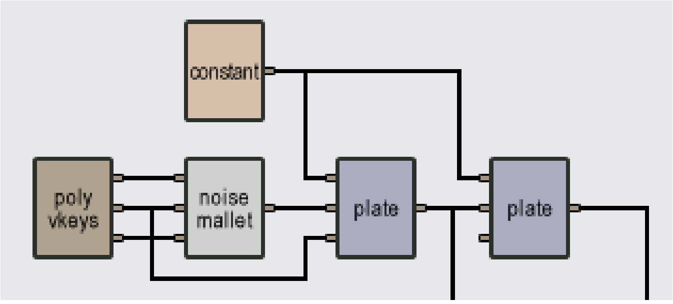
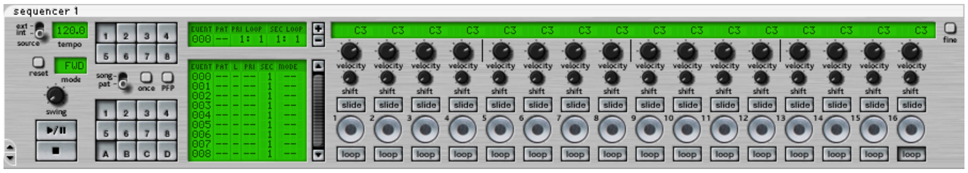


Figure 51: Cascading of Multimode objects.

Note: see also **Beam**, **Bowed Multimode**, **Marimba**, **Membrane**, **Plate** and **String**.

6.47 Multi-sequencer



EVENT	PAT	PRI LOOP	SEC LOOP
000	--	1: 1	1: 1

EVENT	PAT	L	PRI	SEC	MODE
000	--	-	--	1	--
001	--	-	--	1	--
002	--	-	--	1	--
003	--	-	--	1	--
004	--	-	--	1	--
005	--	-	--	1	--
006	--	-	--	1	--
007	--	-	--	1	--
008	--	-	--	1	--

The **Multi Sequencer** module enables you to record sequences of notes. This module in itself does not produce sound but is used, usually instead of a **Keyboard** module, to control instruments. This module is a very complete 16-step sequencer, which means that it plays sequences or patterns of 16 notes in loop. Sequences can be set to have 1 to 16 steps. Because each sequence represents a bar containing four quarter notes, each step of the sequencer itself represents a sixteenth note. The module can memorize 32 different sequences between which you can switch while playing. The sequences can also be chained in any order with the Song mode.

This module has three inputs and seven outputs. the first input is a sync signal which controls the tempo from an external source, the second is a start/stop input which will start the sequencer when it goes form 0 to 1 volt and stop it when it goes from 1 to 0 volt. The signal can come from another sequencer or a **Keyboard**. The third one is a reset input which will restart the sequence from beginning when it goes from 0 to 1 volt. The first three outputs are the same as the inputs (sync, start/stop, reset) and are used to control other sequencers. The fourth output is a gate signal which can be used to trigger events, the fifth is the pitch signal, the sixth is the velocity signal and the last is a slide signal used to trig a **Portamento** module to create a sliding effect between two notes.

Creating Patterns

To create a pattern, you must first select its location. You can select it with a combination of letters (A, B, C, D) and numbers (1 to 8), on the front panel, giving you a total of 32 patterns.

The patterns can be played following 5 play modes using the *mode* control. Forward (FWD) plays the pattern incrementally. Backward (BWD) plays the pattern decrementally. Pendulum (PEND) plays the pattern forward then backward. Random 1 (RDN1) plays the pattern randomly, the same random sequence is repeated when looping. The *reset* button is used to generate a new random sequence. Random 2 (RDN2) plays the pattern randomly changing the random pattern when looping. The sequencer will loop each time a pattern ends. To make the sequencer stop at the end of a pattern, the *once* button must be clicked.

The *tempo* display will adjust the speed of the pattern. The *ext/int* switch will determine if it is the internal clock (int) that sets the tempo or an external source (ext) such as another sequencer or a **Sync Lfo**. The swing knob will introduce a swing feel to the rhythm of the pattern.

The *pitch* display controls the pitch output associated with each step of the sequencer. The pitch signal varies by ± 1 Volt per octave which implies a change of 1/12 Volt for a pitch variation of 1 semitone. The value of the pitch can be changed by click-holding on the display and dragging. Arrows on the keyboard can also be used once a display has been selected. The pitch signal is calculated with respect to the C3 key (middle C) which outputs a value of 0 Volt. This implies that the C2 key signal is -1 Volt and the C4 key is +1 Volt. Holding the Ctrl key (Windows) or Apple key (Mac OS) will offset the value of all 16 pitch knobs. The *fine* buttons allows to adjust the value of the pitch output from -63 to 64 cents of the coarse pitch value.

The *velocity* knobs control the velocity output. The velocity output generates values from 0 to 1 Volt. The *shift* knobs delays the output signal of a gate. The *slide* buttons are used to change the duration of the output signal of a gate. When pressed, the *slide* button adjust the length of the corresponding gate signal to 1/4 of a quarter note instead of the usual 1/8 of a quarter note. The slide output signal will be equal to 1 Volt for duration of the gate signal. If the slide output is connected to a **Portamento** module, the *slide* knob will create a glide effect between two notes.

The numbered *gate* buttons control the gate output signal. The output will generate a square pulse of 1/8 of a quarter note with an amplitude of 1 Volt for each active *gate* buttons. To hear a step, the *gate* button must be clicked (green light on). The *loop* buttons are used to set the length of the Pattern from 1 to 16 steps.

Song Mode

The sequencer can play patterns individually or in a programmed order using a song. The song mode is activated by placing the mode selector to *song*. The sequencer can hold 8 songs. A song can hold up to 1000 events. The sequencer will loop each time a song ends. To make the sequencer stop at the end of a song, the *once* button must be clicked.

Five parameters are associated with each pattern on the square display: *pattern*, *primary loop point* (L), *primary loop times* (PRI), *secondary loop times* (SEC) and *play mode* (MODE). Each of the parameters on the window is adjusted by clicking-holding and dragging. The keyboard arrows can also be used once a parameter has been selected. The *pattern* parameter determines the pattern to be played for a given event, from A1 to D8.The primary loop is a group of events that will be played a number of times equal to the value of the *primary loop times* (PRI) parameter. In order to program a primary loop of events, the beginning and the end of the loop must be selected by the *primary loop point* parameter. This parameter has 3 values: none (-), start (S) and end (E). Selecting start sets the return point of the loop to a given event and activates the *primary loop times* parameter, which can be set from 1 to 99. Selecting end (E) in any subsequent event will make the sequencer return to the start event until the number of loops is reached. Note that placing a primary loop inside another one will not produce the desired effect. The *secondary loop times* parameter sets the number of times an event will be played before switching to the next. It can be set from 1 to 99. The *play mode* parameter change the play mode of the sequencer. The sequencer will stay in the selected play mode for the next events. In order to make song programming easier, use of the *play from position* (PFP) button is useful. It allows to start the song from a selected event. When clicked, the first event in the *edit* window will be remembered as the first and return event of the song.

Editing a song

The song *info* window at the top of the event window displays the event currently being played or selected with the wheel located at the right of the event window. Events can be removed from a song by clicking on the “-” button on the right of the song info window. Pressing on the “+” button inserts a new event after the event currently selected.

Note: see also **Control Voltage Sequencer**, **Control Voltage Sequencer with Songs**, **Single Gate Sequencer**, **Single Gate Sequencer with Songs**, **Dual Gate Sequencer** and **Dual Gate Sequencer with Songs**.

6.48 Nand

The **Nand** module performs the inverse of the logical AND operation. The one output of this module is either 1 (true) or 0 (false) depending on the values sent to the two inputs. This module has no front panel. The following diagram shows the output value depending on the values in the two inputs.

Input1	**Input2**	**Output**
1	1	0
1	0	1
0	1	1
0	0	1

Table 2: **Nand** module output as a function of its inputs.

Input signals are considered False (0) when smaller than 0.1 Volts and True (1) when greater than 0.1 Volts.

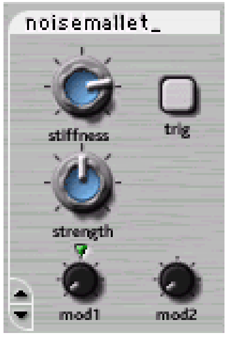
6.49 Noise

The **Noise** module outputs white noise. It has no input and one output, the noise signal. This module has no front panel.

Typical Use

The **Noise** module can be used as a sound source in analog synthesizers. It can be used to create percussion sounds or special effects.

6.50 Noise mallet



The **Noise Mallet** module can be used as an alternative to the **Mallet** module. This module outputs the same signal as the **Mallet** module, but, in addition to the impact noise, it also generates white noise. The force of the impact is adjusted with the *strength* knob, while the stiffness of the mallet, related to its material, is varied with the *stiffness* knob.

This module has one output, the impact signal, and three inputs. The first input triggers the mallet every time a low-to-high transition is encountered in the input signal. This input is usually connected to the gate signal from a **Keyboard** module. Note that the **Noise Mallet** can also be triggered manually by using the *trig* button on the front panel. The second input signal modulates the stiffness of the mallet relative to the value selected with the *stiffness* knob. The amplitude of the modulation is adjusted with the *mod1* knob. The greater the amplitude, the greater the stiffness. This modulation input is used, for example, when a variation of the stiffness of the mallet with the note played is desired. When the knob is adjusted in its center position and when this input is connected to a pitch signal, the stiffness exactly follows the pitch variation so as to ensure that the spectral content (or color) of the sound produced by a structure is uniform when the pitch is varied. The third input modulates the strength of the impact relative to the adjustment of the *strength* knob. This input is generally connected to a velocity signal which allows you to increase the force of the impact. The amplitude of this modulation signal is adjusted with the *mod2* knob.

The default value of the following parameters is set at construction

- *Strength*: value of the impact force (value between 0 and 2).
- *Stiffness*: value of mallet stiffness (value between 1 and 20 000).

Note: See also **Mallet**.

6.51 Nor

The **Nor** module performs the inverse of the OR logic operation. The one output of this module is either 1 (true) or 0 (false) depending on the values sent to the two inputs. This module has no front panel. The following diagram shows the output value depending on the values in the two inputs.

Input signals are considered False (0) when smaller than 0.1 Volts and True (1) when greater than 0.1 Volts.

Input1	Input2	Output
1	1	0
1	0	0
0	1	0
0	0	1

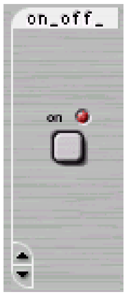
Table 3: **Nor** module output as a function of its inputs.

6.52 Not

The **Not** module performs the logical not operation. The one output of this module is either 1 (true) or 0 (false) depending on the values its one input receives. The module outputs false when the input is true and outputs true when the input is false.

Input signals are considered False (0) when smaller than 0.1 Volts and True (1) when greater than 0.1 Volts.

6.53 On/Off, On/Off2, On/Off3, On/Off4



The **On/Off**, **On/Off 2**, **On/Off 3** and **On/Off 4** switch modules have respectively one to four inputs and one to four outputs. Their behavior is very simple: when the buttons are in the *Off* position, the output is zero regardless of the input signals and when the buttons are pushed in the *On* position, the output signal is the exact copy of the input signals. The transition between the two states (*On/Off*) is smoothed in order to avoid clicks when turning the switches *On* or *Off*.

Typical Use

An **On/Off** switch can be used as a mute button or to create modulation matrices.

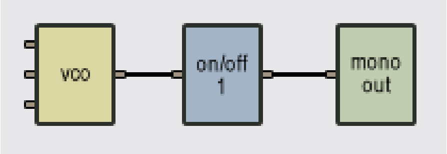


Figure 52: **On/Off** switch used to mute the output of a **VCO** module

Note: See also **Volume**, **Slider**, **Gain**, **Selector**.

6.54 Or

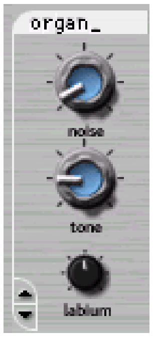
The **OR** module performs an OR logic operation. The one output of this module is either 1 (true) or 0 (false) depending on the values sent to the two inputs. To deliver 1 at the output, either one or the two inputs must receive a value of 1, otherwise the output will deliver a value of 0. This module has no front panel. The following diagram shows the output value depending on the values in the two inputs.

Input signals are considered False (0) when smaller than 0.1 Volts and True (1) when greater than 0.1 Volts.

Input1	**Input2**	**Output**
1	1	1
1	0	1
0	1	1
0	0	0

Table 4: **Or** module output as a function of its inputs.

6.55 Organ



The **Organ** module simulates a simple polyphonic street pipe organ. Every note played on the organ excites a pipe of different length, thereby changing the pitch. This module has three inputs and one output. The first input is a gate signal, generally that from a **Keyboard**. The second input is the driving pressure signal and is generally connected to the output from an **ADSR** module or the gate signal from a **Keyboard** module. Finally, the third input is a pitch signal generally connected to the pitch output from a **Keyboard**. The output signal is the sound produced by the instrument.

Three parameters can be adjusted while playing. The *noise* knob sets the amount of turbulence noise in the sound. The *tone* knob controls the jet behavior which affects the tone color of the organ sound. The *labium* knob sets the position of the edge of the labium of each organ pipe relative to the jet. In its center position, the jet blows exactly in front of the pipe labium

Typical Use

This module is generally played with a Polykey module as shown in Figure 53.

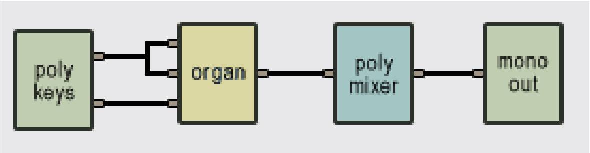


Figure 53: An **Organ** module controlled with a **Polykey**.

Note: This module is designed to be played with a **Polykey** or **Polyvkey** module. Although it works with one voice, you will hear clicks with note changes as the output switches abruptly from one pipe to the other. For monophonic flute-like sounds, use the **Flute** module.

6.56 Outlet (1-12)

These modules are used to define the outputs of a sub-patch so that it can be connected in another patch. These modules have no output but between 1 and 12 inputs which are connected to the outputs of the sub-patch which you want to connect to inputs in another construction. These inputs will correspond to the outputs of the sub-patch icon which will appear in the construction window when later you include this sub-patch in another construction. These modules have no front panel.

Sub-patches may have between 0 and 12 inputs and 0 and 12 outputs but they must always have at least one input or output. As soon as an **Inlet** or **Outlet** module is included in a patch, the

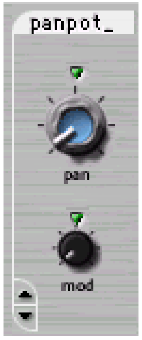
Tassman Builder will consider that you want to define the current patch as a sub-patch and will save it as so in the **Sub-Patches** folder of the Browser. You can then use it just like any other module.

Typical Use

A sub-patch is created with an **Inlet** or **Outlet** module or both. The outputs of the sub-patch are determined by connecting them to an **Outlet** module. See example of Figure 42 under **Inlet**.

Note: See also the **Inlets** (1-12) modules.

6.57 Panpot



The **Panpot** module (panoramic potentiometer) is used to position a sound source in stereo space by adjusting the relative amplitude of signals sent to the left and right channels of the sound card. This module has two inputs and two outputs. The first input is the signal to be panned, and the second is a modulation signal that can affect the relative amount of right and left distribution of the signal. The **Panpot** distributes the first input between the two outputs in such a way that the power remains constant, i.e. $(IN1)^2 = (OUT1)^2 + (OUT2)^2$. When there is 0 Volts on the second input or when the *mod* knob is set to zero, the *pan* knob alone controls the distribution of the input signal over the two outputs. When the *pan* knob is turned to the left, all the signal goes to output 1 (left); turned to the right, all the signal goes to output 2 (right); and in the center position (green LED on) the signal is equally distributed between the two channels.

The second input is used to control the **Panpot** through an external signal. The strength of this effect can be adjusted using the *mod* knob. A negative value of the input moves the source toward the left, while a positive input moves it to the right. The resulting source position depends on the combination of both the *pan* knob position and the input modulation signal.

Typical Use

The **Panpot** is often used for positioning a sound in stereo space such as shown in Figure 54. Or it can be used for producing special effects using the modulation input as illustrated in Figure 55.

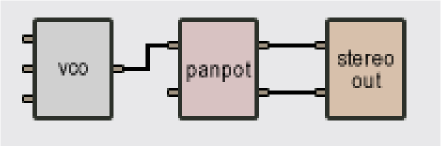


Figure 54: **Panpot** used for Stereo Output.

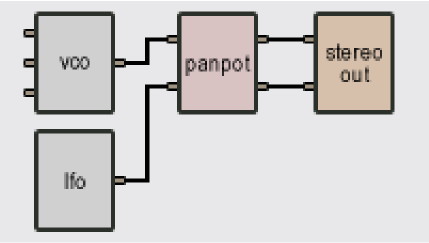
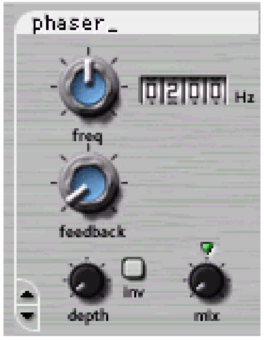


Figure 55: **Panpot** modulated by **LFO**.

The default value of the following parameters is set at construction

- *Angle*: default source position. A value of 0 positions the source on the left, 0.5 in the middle and 1 on the right.
- *Range*: determines the maximum possible amount of source excursion from its original position, varies between 0 and 0.5 (90 degrees).

6.58 Phaser



The **Phaser** module implements the effect known as “phasing” which colors a signal by removing frequency bands from its spectrum. The effect is obtained by changing the phase of the frequency components of a signal using an all-pass filter and adding this new signal to the original one. This module has two inputs and one output. The first input is the audio signal to be phased and the second input is a modulation signal that varies the phase variations introduced in the spectrum of the first input signal. The output is the phased signal.

The algorithm implemented in this module is shown in Figure 56. The input signal is sent into a variable fourth order all-pass filter. This “wet” signal is then mixed down with the original “dry” signal. A feedback line allows the resulting signal to be re-injected into the filter. The effect of the **Phaser** module is to introduce rejection in the spectrum of the input signal depending on the tuning of the filter.

The all-pass filter modifies a signal by delaying its frequency components with a delay which increases with the frequency. This phase variations will introduce a certain amount of cancellation when this “wet” signal is mixed down with the original “dry” signal as shown in Figure 57. The rejection is maximum when the phase delay is equal to 180 degrees and a given component is out of phase with that of the original signal. The amount of effect is determined by the ratio of “wet” and “dry” signal mixed together as shown in Figure 57. As the amount of “wet” signal sent to the output is reduced, the amount of rejection increases. The shape of the frequency of the Phaser module is also influenced by the amount of “wet” signal re-injected into the feedback loop. Increasing the feedback enhances frequency components least affected by the all-pass filter. As the feedback is

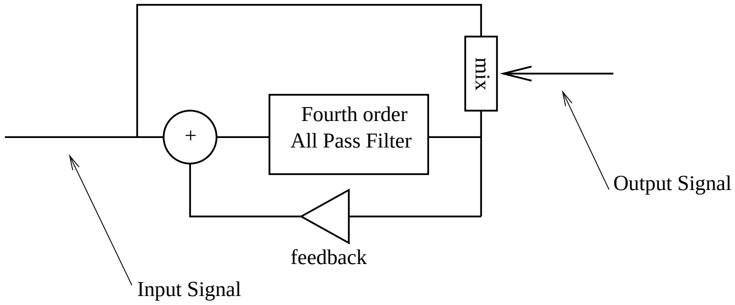


Figure 56: **Phaser** algorithm.

increased, these peaks become sharper. The functioning of the **Phaser** is very similar to that of the **Flanger** module. The filtering effect is different however, since the **Phaser** module only introduces rejection around two frequencies which, in addition, are not in an harmonic relationship.

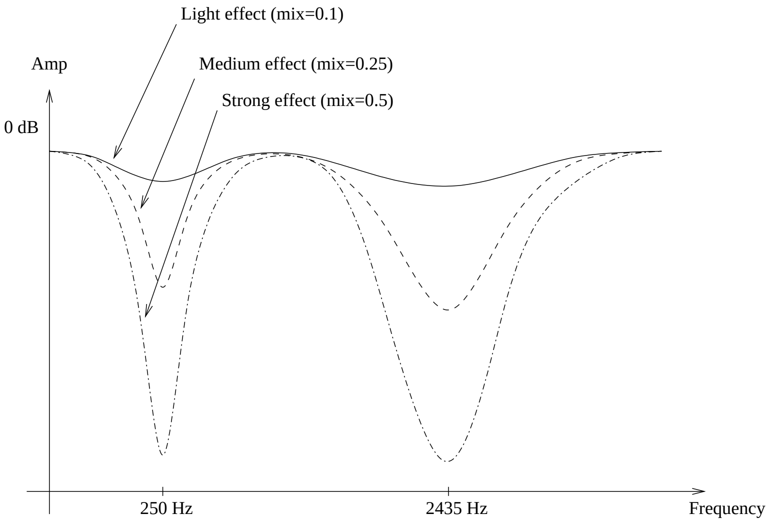


Figure 57: Frequency response of a **Phaser** module. Effect of the mix between “wet” and “dry” signal on the frequency response.

Tuning

The location of the first notch in the frequency response of the module is adjusted with the *frequency* knob and is displayed, in Hertz, in the counter next to the knob. This frequency can be modulated by using the second input of the module, the amount of modulation depending on the adjustment of the *depth* knob. In the left position, there is no modulation and the frequency remains fixed while in the right position, with a modulation signal varying between [-1,1] Volt, the frequency varies between 0 and twice the value set with the *frequency* knob. The *feedback* knob is used to fix the amount of “wet” signal re-injected into the delay. Finally, the mix knob determines the amount of “dry” and “wet” signal sent to the output. When this knob is adjusted in the left position, only “dry” signal is sent to the output, in its center position (green LED On), there is an equal amount of “dry” and “wet” signal in the output and in the right position, only “wet” signal is sent to the output.

Typical Use

The output from a LFO module can be used to control the filtering of a signal (the output of a VCO for example) with a Phaser module as shown in Figure 58.

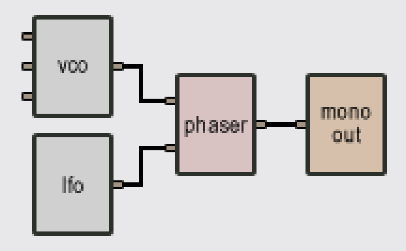


Figure 58: **Phaser** modulated with an **LFO** module.

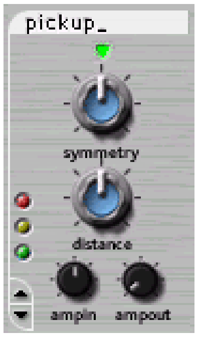
A **wah-wah** effect is obtained by using much feedback. This has the effect of increasing the peak in the frequency response located between the two notches in the curve. In these conditions the module acts as bandpass filter.

The default value of the following parameters is set at construction

- *frequency*: frequency, in Hertz, of the first notch in the frequency response of the filter, [50,2000] Hertz.
- *feedback*: coefficient,[0, 1[, determining amount of “wet” signal re-injected into the filter. If feedback = 0 there is no “wet” signal re-injected while if *feedback* = 0.99, maximum of “wet” signal re-injected.
- *depth*: gain coefficient, [0,1], multiplying the modulation signal.
- *mix*: amount of “dry” and “wet” signal sent to output. If *mix* = 0 there is only dry signal while if *mix* =1, there is only “wet” signal.

6.59 Pickup

The **Pickup** module simulates the function of magnetic pickup coils which are used, for example, in electric guitars and electric pianos. This type of transducer is sensitive to the motion of metallic objects (such as a string or a beam) near the pickup. As such an object vibrates near a pickup, the latter outputs an oscillating signal determined by the varying distance between the object and the pickup. The waveform of the output signal can be varied by adjusting the pickup position relative to the object.



The **Pickup** module has one input and one output. The input can be any oscillating signal that one wants to process through the **Pickup**. The shape of the output signal depends on the settings of the controls on the module front. The *symmetry* and *distance* knobs are used to adjust the position of the pickup relative to a signal source as shown in Figure 6.59. The pickup can be positioned precisely in front of the source by adjusting the *symmetry* knob in the center position (green LED on). The position of the pickup determines both the amplitude of the signal transmitted by the pickup and the amount of "distortion" applied to the input signal. Distortion occurs because the distance between the vibrating source and the pickup does not vary in the same way as the original vibrating motion. The *ampin* and *ampout* knobs on the front panel determine the gain of amplifiers placed respectively at the input and output of the pickup module.

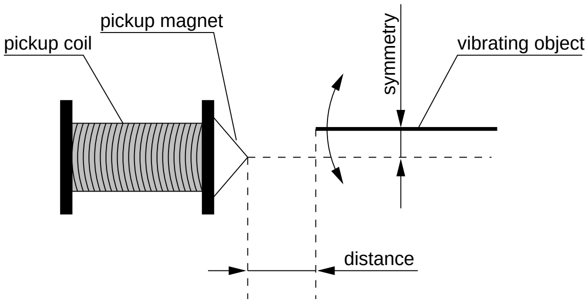


Figure 59: **Symmetry** and distance settings in a **Pickup** module.

Typical Use

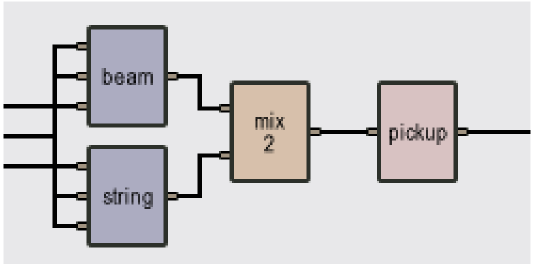


Figure 60: A **Pickup** is used to create an electric piano.

The Pickup module is used in Figure 60 to construct an electric piano. In Figure 61, a Pickup module is used as a distortion. The effect is applied to the signal coming out from a Polyphonic Mixer.

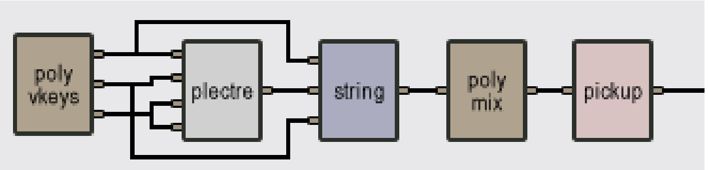


Figure 61: A **Pickup** used as a distortion.

The default value of the following parameters is set at construction

- *symmetry*: vertical position of the pickup relative to the oscillating object. This parameter varies between -2 and 2, 0 is exactly in front of the object.
- *distance*: proportional to the horizontal distance between the oscillating object and the pickup. Varies between 0.1 and 2.
- *ampin*: gain of amplifiers placed at the input of the pickup module.
- *ampout*: gain of amplifiers placed at the output of the pickup module.

6.60 Pitch Wheel

The **Pitch wheel** module reads the signal from the pitch wheel of a keyboard. The output signal lies between -1 and +1, its value depending on the position of the pitch wheel. The output signal is equal to 0 when the pitch wheel is not moving. This module has no input and no front panel control.

The default value of the following parameter is set at construction

- *MIDI channel*: MIDI channel used by the pitch wheel.

6.61 Plate

The **Plate** module simulates sound production by rectangular plates of different materials and sizes. This module first calculates the modal parameters corresponding to plate-shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Multimode** module to simulate the sound. The module has one output, the sound produced by the plate, and three inputs. The first input signal is a damping signal which, depending on its value, lowers or raises dampers on the structure. When the input signal is equal to 0, dampers are lowered on the plate thus shortening the decay time of the sound produced by the structure. When the signal is greater than 0, dampers are raised. Note that this damping adds to the natural damping of the plate itself. If this input is not connected to any other module, the default value is set at 0 which implies that the plate motion will be damped. This input is, therefore, usually connected to a **Constant** module to obtain undamped motion or to a **Damper** module or the gate signal from a **Keyboard** in order to vary the damping while playing. The second input signal is the force signal exciting the plate and the third is a pitch modulation.

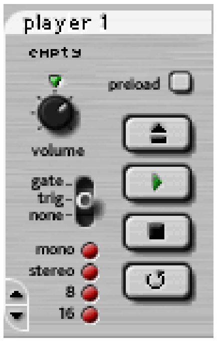
The default value of the following parameters is set at construction

- *Length*: the length, in meters, of the plate.
- *Width*: the width, in meters, of the plate.
- *Frequency*: fundamental frequency, in Hertz, of the plate beam when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the size of the plate. The software automatically calculates the physical parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **Plate** module with a **Keyboard** module.
- *Decay*: proportional to the decay time of the sound produced by the plate.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point-x*: x-coordinate, in meters, of impact point from the lower left corner of the plate.
- *Excitation point-y*: y-coordinate, in meters, of impact point from the lower left corner of the plate.
- *Listening point-x*: x-coordinate, in meters, of listening point from the lower left corner of the plate.
- *Listening point-y*: y-coordinate, in meters, of listening point from the lower left corner of the plate.

Note: For more details on this module and especially the front panel controls, see the **Multimode** module.

6.62 Player

The **Player** module is used to read sound files from a disk. This module has one input and two outputs, the latter being the left and right channel from the signal read from the sound file. When the sound file is mono, both output pins carry the same signal. The outputs from the **Player** can be sent to any other module for processing. The input signal is a gate signal, typically the gate signal from a **Keyboard**, or a **Sequencer** which triggers the **Player** according to the *gate-trig-none* selector. When the selector is at the *gate* position, the **Player** starts whenever a low-to-high transition occurs and stops whenever a high-to-low transition occurs. When the selector is at the trig position, the **Player** starts whenever a low-to-high transition occurs. When the selector is at the *none* position, the gate signal is ignored. Every time the **Player** module is triggered, it starts to play again the file from the beginning even if it has not yet reached the end of the file.



The **Player**module can also be started by pressing on the play button. If the **Player** was already playing a file, pressing on the play button makes the module start playing again from the beginning. The **Player** module can be stopped by pressing on the stop button. The **Player** can also be put in a loop mode by pressing on the loop button. In this mode the **Player** will start to play again the file from the beginning when it reaches the end of the recording and will keep on doing this until the stop button is pressed.

The **Player** first starts to play an empty sound file. One can select a given file by pressing on the *select* button (located above the *play* button) which will make a browse window appear on the screen. Sound files can also be drag'n'dropped on the **Player** module. The name of the file currently playing appears on the module front panel while its format is indicated by the red LEDs on the bottom-left of the module.

Another way to have the **Player** playing a given file is to load a file and then save a preset with the save arrow on the lower left corner of the module panel. The **Player** will play the same file the next time the preset is loaded.

Typical Use

In the example of Figure 75 under **Shifter**, a sample played with the **Player** module is pitch-shifted.

Notes: When small sound files are involved, it may be advantageous to preload the file in memory using the *preload* button.

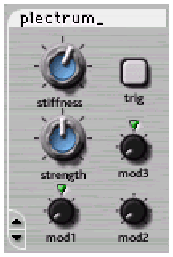
In order to facilitate the exchange of patches and presets between them, users are encouraged to copy their sound files in the folder which name appears in the **General Preferences** dialog box.

File format supported

- PC: mono, stereo 8 or 16-bit wave files. The sampling rate of the file must match the sampling rate of the audio settings.
- Mac: any AIFF, AIFC, Wave or MP3 file readable by QuickTime.

See also the **Recorder** and **Recorder2** modules.

6.63 Plectrum



The **Plectrum** module is used to simulate the excitation of a string when it is plucked by a finger or a pick. The output of this module is the force signal applied by the plectrum on the string. Before a string starts to vibrate, the plectrum moves the string. A force is supplied to the string while the plectrum and the string are in contact. The shape of the force signal is dependent on the stiffness of the plectrum which can be adjusted with the *stiffness* knob. The harder the pick, the sharper the force signal while a soft plectrum results in a smoother signal. The amplitude of this force signal is adjusted with the *strength* knob.

This module has four inputs. The first input triggers the plectrum every time a low-to-high transition is encountered in the input signal. This input is usually connected to the gate signal from a **Keyboard** module. Note that the **Plectrum** can also be triggered manually by using the *trig* button on the front panel. The second input signal modulates the stiffness of the plectrum relative to the value selected with the *stiffness* knob. The amplitude of the modulation is adjusted with the *mod1* gain knob. The greater the amplitude, the greater the stiffness. This modulation input is used, for example, when a variation of the stiffness of the plectrum with the note played is desired. When the knob is adjusted in its center position and when this input is connected to a pitch signal, the stiffness exactly follows the pitch variation so as to ensure that the spectral content (or color) of the sound produced by a structure is uniform when the pitch is varied. The third input also modulates the stiffness but in the reverse manner of the second input which implies that the stiffness decreases when the input signal increases. This input is usually connected to the velocity output from a keyboard module which implies that the plectrum softens as the impact velocity increases. The amplitude of this input is adjusted with the *mod2* gain knob. The last input modulates the amplitude of the force signal relative to the adjustment of the *strength* knob. This input is also generally connected to a velocity signal in order to increase the amplitude of force supplied by the plectrum with the velocity signal. The amplitude of this modulation signal is adjusted with the *mod3* gain knob.

Typical Use

A Plectrum can be used to excite a String module as in the following example.

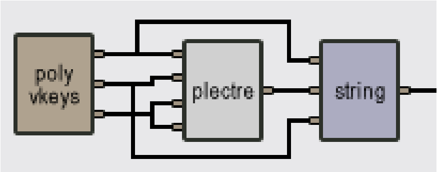
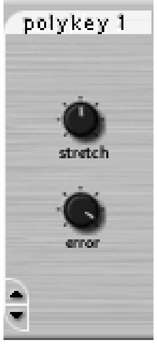


Figure 62: A **Plectrum** module exciting a **String**.

The default value of the following parameters is set at construction

- *Strength*: value of the impact force (value between 0 and 2).
- *Stiffness*: value of plectrum stiffness (value between 1 and 20000).

6.64 Polykey



The **Polykey** module reads signal from a MIDI keyboard and is used to create polyphonic instruments. This module must always be used in combination with a **Polymixer** module. A polyphonic patch is created by inserting modules between a **Polykey** and a **Polymixer** module as shown in Figure 63. *Tassman* will automatically duplicate the modules appearing between the **Polykey** and **Polymixer** module one time for each voice requested during construction in the **Polykey** module edit pop-up menu. The outputs of the different voices of the polyphonic patch are mixed by the **Polymixer** module. The resulting signal, coming out from the **Polymixer** module can then be sent to any other module. The front panel of the different modules included in a polyphonic section of a patch are only mapped once on the Player. This means that every voice of the patch is similarly affected by the settings of the front panel controls. Keep in mind that the computational load inevitably increases with the number of voices. This means that the number of voices that can be played really depends on the complexity of the patch one is playing and on the computer processing power.

The **Polykey** module has no input and two outputs which are similar, for every voice, to that of a **Keyboard** module. The first output is the gate signal. It is equal to 0 Volt when no key is played, and 1 Volt when a key is played. The second output signal is the pitch signal. The pitch signal varies by 1 Volt per octave which implies a change of 1/12 Volt for a pitch variation of 1 semitone. The pitch signal is calculated with respect to the C3 key (middle C) which outputs a value of 0 Volt. This means that, for example, the C2 key signal is -1 Volt and that of the C4 key is +1 Volt. The *stretch* knob on the interface is used to simulate stretched tuning used on instruments such as pianos. Turned to the left, low notes will be tuned higher and high notes lower (inner stretch); turned to the right, low notes will be tuned lower and high notes will be higher (outer stretch). In the center position, the tuning will be equal. The *error* knob introduces some randomness in the pitch signal. Turned to the left, no error is outputted and the pitch signal is perfect; as the knob is turned to the right, errors will start to appear causing small fluctuations in pitch. The effect of this knob is to simulate pitch variations found in analog synths.

Typical Use

A polyphonic instrument is created by inserting modules between a **Polykey** and a **Polymixer** module as shown in Figure 63. The number of voices is determined during construction.

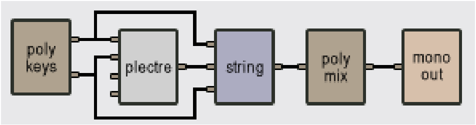


Figure 63: Creating a polyphonic synth with a **Polykey** and **Polymixer** module.

The default value of the following parameters is set at construction

- *pitch wheel range*: determines the range of pitch variation that can be obtained with the pitch wheel. The convention is 1 Volt/octave (maximum value is 2 Volts). A semitone is equal to a 0.08333 value.
- *MIDI channel*: MIDI channel used by the keyboard.
- *number of voices*: number of voices requested for the modules enclosed within a **Polykey** and **Polymixer** module.

Note: see also the **Keyboard**, **Vkeyboard** and **Polyvkey** modules.

6.65 Polymixer

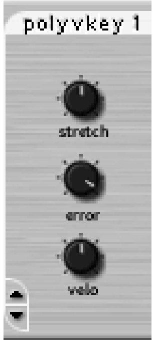
The **Polymixer** mixes the outputs of the different voices of a polyphonic patch. This module is always used in combination with a **Polykey** or **Polyvkey** module. A polyphonic patch is created by inserting modules between a **Polykey** or a **Polyvkey** and a **Polymixer**. *Tassman* will automatically reproduce the modules appearing between the **Polykey** and **Polymixer** module one time for each voice requested during construction in the **Polykey** or **Polyvkey** module edit pop-up menu. The output of the **Polymixer** module is the sum of the outputs of the different voices of the polyphonic patch. The front panel of the different modules included in a polyphonic section of a patch are only mapped once on the Player. This means that all the voices of the patch are similarly affected by the settings of the front panel controls. Keep in mind that the computational load inevitably increases with the number of voices. This means that the number of voices that can be played really depends on the complexity of the patch one is playing and on the computer processing power. You can use multiple **Polymixer** modules in an instrument.

Typical Use

See the example of Figure 63 under **Polykey**.

6.66 Polyvkey

Similar to the **Polykey** module except that there is an additional output which is proportional to the velocity with which the key was pressed. The *stretch* knob on the interface is used to simulate stretched tuning used on instruments such as pianos. Turned to the left, low notes will be tuned higher and high notes lower (inner stretch); turned to the right, low notes will be tuned lower and high notes will be higher (outer stretch). In the center position, the tuning will be equal. The *error* knob introduces some randomness in the pitch signal. Turned to the left, no error is outputed and the pitch signal is perfect; as the knob is turned to the right, errors will start to appear causing small fluctuations in pitch. The effect of this knob is to simulate pitch variations found in analog synths. The *velo* knob adjusts the velocity curve of the **Polyvkey**. In the center position, the curve is linear. Turned to the left, the velocity increases more quickly; conversely, turning the knob to the right results in a slower velocity curve.



The default value of the following parameters is set at construction

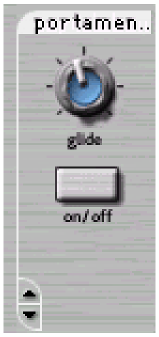
- *pitch wheel range*: determines the range of pitch variations that can be obtained with the pitch wheel. The convention is 1 Volt/octave (maximum value is 2 Volts). A semitone is equal to a 0.08333 value.
- *MIDI channel*: MIDI channel used by the keyboard.
- *number of voices*: number of voices requested for the modules enclosed within a **Polykey** and **Polymixer** module.

Note: see also the **Keyboard**, **Vkeyboard** and **Polykey** modules.

6.67 Portamento

The **Portamento** module is an integrator and is used to smooth a signal. The module has two inputs: a gate signal which turns it on or off, and the signal to be smoothed. The output of the **Portamento** is the input signal smoothed according to the time constant set by the *glide* knob. The higher the time constant, the slower the response of the **Portamento**.

When the front panel *on/off* switch is on, the gate signal is ignored and the input signal is always smoothed. When the switch is off the input signal is only smoothed if the gate signal is greater than 0.1 Volt, otherwise the output is a copy of the input. The following figure shows the **Portamento** behavior.



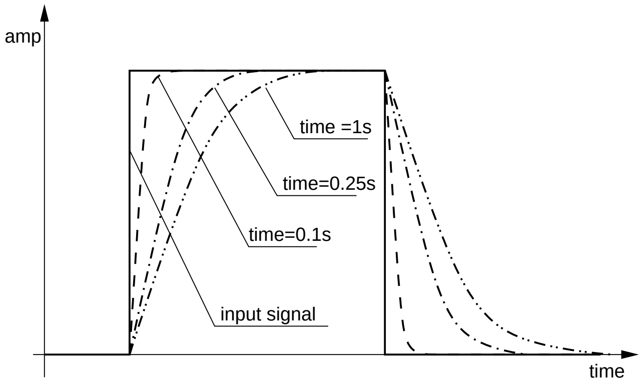


Figure 64: Behavior of **Portamento** as a function of time constant.

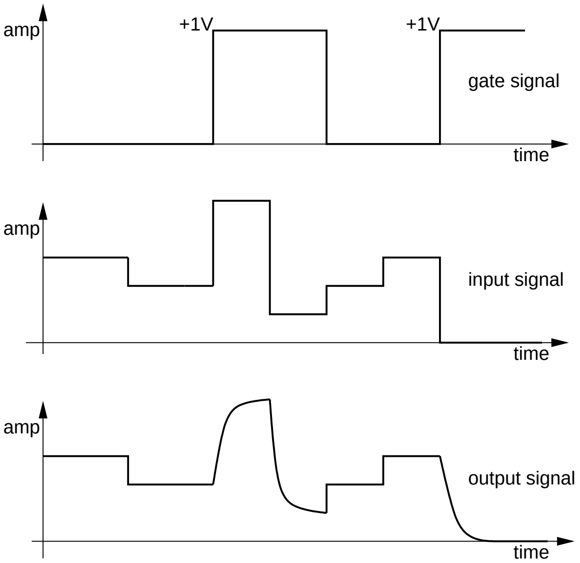


Figure 65: Portamento triggered by gate signal.

Typical Use

The Portamento is often used to create a glissando effect between two notes. Figure 45 shows a complete example of a Portamento triggered by a gate signal (when the on/off switch is off). The Portamento is frequently inserted between a Keyboard module and a VCO to smooth the note output of the Keyboard as shown in Figure 46. It is then switched on or off manually.

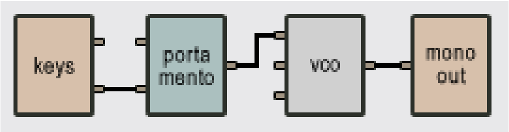
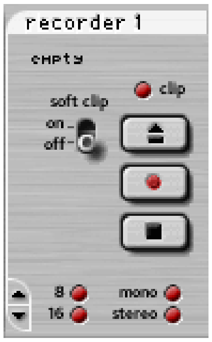


Figure 66: **Portamento** used with **Keyboard**.

The default value of the following parameter is set at construction

- *glide time*: sets the time constant of the integrator (values between 0.01s and 10s); the higher the time constant the slower the response of the integrator.

6.68 Recorder



The **Recorder** module is used to record the output of an instrument to a sound file. This module has two inputs which are respectively the left and right channel signals to be recorded.

Recording is triggered by pressing on the *record* button which has a red LED on it. Recording is stopped by pressing on the *stop* button located below the *record* button. One can record to a specific file by pressing on the *load* button located above the record button which will make a browse window appear on the screen. The name of the file being written onto is displayed above the *select* button. The file format is determined by the settings in the browse window.

The recorded signal is not compressed so that it can be reloaded again without any loss. However, when the amplitude of the signal is too high, it will be clipped according to the “Soft Clip” selector on the panel. When clipping occurs during a recording, the red clip LED is switched ON and remains ON until the record button is pressed again.

Notes:

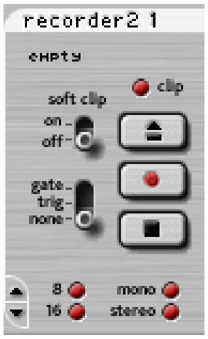
- Because the **Recorder** starts recording at the beginning of the file, it will erase what might already have been recorded. The **Recorder** has the same behavior if you press on *stop* and press on the *record* button again without changing the destination file.
- For the saturation characteristics of the soft clip, refer to the **Audio Out** module.

File format supported

- PC: mono, stereo 8 or 16-bit wave files. The sampling rate will match the sampling rate of the audio settings.
- Mac: any AIFC or AIFF. The specific settings supported are determined by QuickTime.

See also the **Player** and **Recorder2** modules.

6.69 Recorder2



The **Recorder2** module is used to record the output of an instrument to a sound file. This module has 3 inputs which are respectively the gate signal and the left and right channel signals to be recorded.

Recording is triggered from the module front panel or from the *gate* signal according to the *gate-trig-none* selector. When the selector is at the *gate* position, the **Recorder2** module starts recording whenever a low-to-high transition occurs and stops recording whenever a high-to-low transition occurs. When the selector is at the *trig* position, the **Recorder2** module starts recording whenever a low-to-high transition occurs. When the selector is at the *none* position, the gate signal is ignored.

Recording can also be stopped by pressing on the *stop* button located below the *record* button. One can record to a specific file by pressing on the *load* button located above the *record* button which will make a browse window appear on the screen. The name of the file being written onto is displayed above the *select* button. The file format is determined by the settings in the browse window.

The recorded signal is not compressed so that it can be reloaded again without any loss. However, when the signal's amplitude is too high, it will be clipped according to the "Soft Clip" selector on the panel. When clipping occurs during a recording, the red clip LED is switched ON and remains On until another recording is initiated.

Notes:

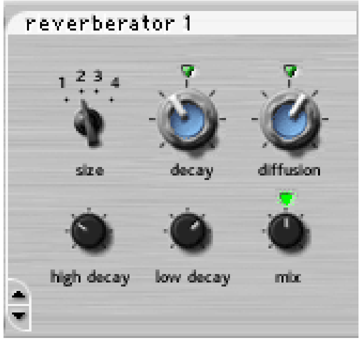
- Because the **Recorder2** starts recording at the beginning of the file, it will erase what might already have been recorded. The **Recorder2** has the same behavior if you press on *stop* and press on the *record* button again without changing the destination file or if the recording is initiated from the gate signal.
- For the saturation characteristics of the soft clip, refer to the *Audio Out* module.

File format supported

- PC: mono, stereo 8 or 16-bit wave files. The sampling rate will match the sampling rate of the audio settings.
- Mac: any AIFC or AIFF. The specific settings supported are determined by QuickTime.

Note: See also the **Player** and **Recorder** modules.

6.70 Reverberator



The **Reverberator** module is used to recreate the effect of the reflexion of sound on the walls of a room or a hall. These reflexions add spaciousness to the sound and make it warmer, deeper, and more "real". This makes sense as we always listen to instruments in a room and thus with a room effect. This module has two inputs, the left and right source signals and its two outputs are these signals as they would be heard in a given room.

Impulse Response of a Room

The best way to evaluate the response of a room is to clap hands and to listen to the resulting sound. Figure 67 shows the amplitude of the impulse response of a room versus time. The first part of the response is the clap itself, the direct sound, while the remaining of the response is the effect of the room which can itself be divided in two parts. Following the direct sound, one can observe a certain amount of echoes which gradually become closer and closer until they can not be distinguished anymore and can be assimilated to an exponentially decaying signal. The first part of the room response is called the early reflexion while the second is called the late reverberation. The total duration of the room response is called the reverberation time (TR).

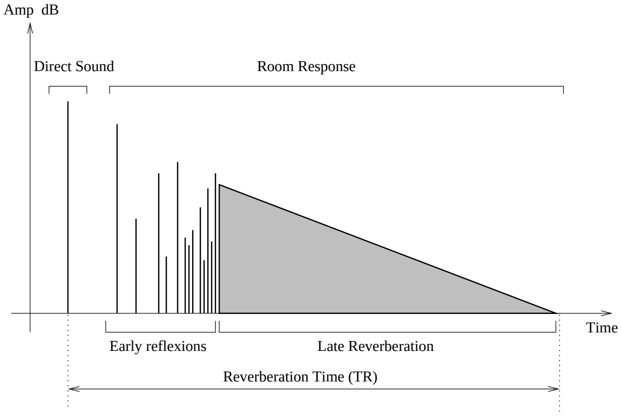


Figure 67: Impulse response of a room.

Adjusting the room effect

The size of a room strongly affects the reverberation effect. The *size* button is used to choose this size from small (1) to large hall (4). The duration of the reverberation time (TR) depends on both the size of the room and the absorption of the walls, which can be adjusted with the *decay* knob. In a real room the reverberation time is not constant over the whole frequency range. As the walls are often more absorbent in the very low and in the high frequencies the reverberation time is shorter for these frequencies. This can be adjusted in the **Reverberator** module with the *low* and *high decay* knobs. Another parameter which affects the response of a room is its geometry; the more complex the geometry of a room, the more reflexion are observed per unit of time. This quantity is known as the time density and can be set trough the *diffusion* knob. In a concert all the time density is supposed to be quite high in order not to hear separate echoes which are characteristic of poor sounding rooms.

The last parameter which affects our listening experience in a room, is the distance between the sound source and the listener. While the room response is quite constant regardless of the position of the source and the listener, the direct sound (the sound which comes directly from the source) depends strongly on the position of the listener. The farther we are from the sound source the quieter is the direct sound relatively to the room response. The ratio between the direct sound and the room response is adjusted with the *mix* knob which in other words is used to adjust the perceived distance between the source and the listener. In its leftmost position, only the direct sound is heard while when fully turned to the right, one only hears the room response.

One important feature taken into account by the **Reverberator** module is the difference between the signal reaching the left and right ear. In the first part of the response, the echoes are panned right and left from the position of the source in the stereo space helping to create the sensation of space and in the second part of the response (the late reverberation) the signals from the left and right channel are uncorrelated which is an important property of the “diffused field” observed in real rooms.

Typical Use

In Figure 68, a panpot is used to position a mono signal in the stereo space before adding a room effect with a **Reverberator** module.

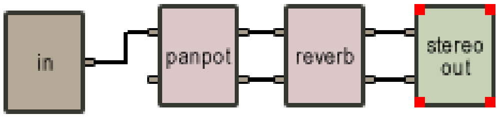


Figure 68: Adding a room effect to a mono signal.

Note: A **Reverberator** is included in the output stage effect stage as explained in Section 4.8. See also **Tube Reverb**

6.71 RMS

The **RMS** (Root Mean Square) module is an envelope follower. Its output is the root mean square of the input signal. The inverse of the integration time ($1/\tau$) is set during construction and determines the response time of the circuit. This module has no front panel control.

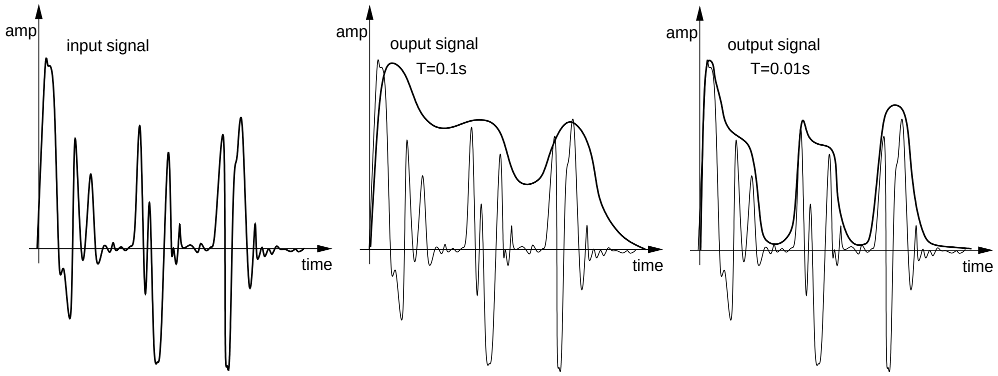


Figure 69: RMS value of a signal.

Typical Use

This module can be used to extract the envelope of a signal as in the following example:

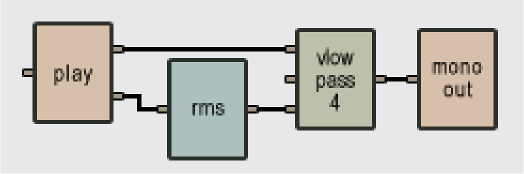


Figure 70: RMS Module Used to Extract Volume.

In the example of Figure 51, **Root Mean Square** modules are used to make a vocoder. The **RMS** modules are used to extract the envelope of different frequency bands of an audio signal from a **Player** (typically a voice recording) which is then used to modulate the amplitude of corresponding frequency bands in the signal from a **VCO**.

The default value of the following parameter is set at construction

- *Cutoff frequency*: Inverse of the integration time (in Hertz, values between 0.2 and 100). Sets the response time of the circuit.

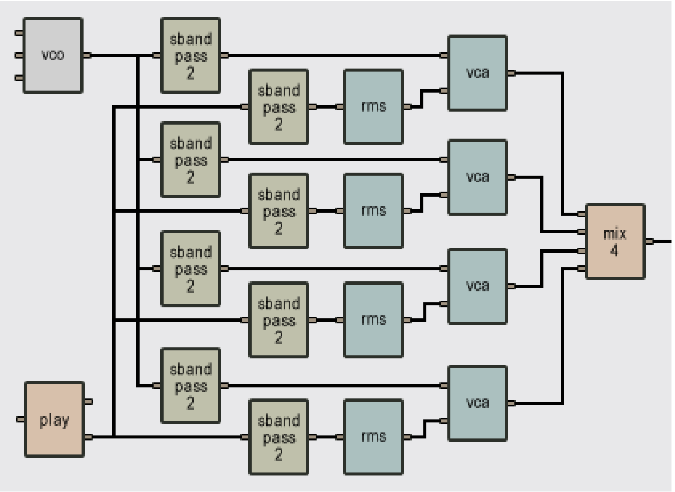
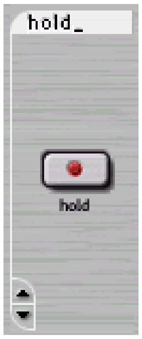


Figure 71: Use of RMS in Vocoder.

6.72 Sample & Hold



The **Sample & Hold** module performs a sample & hold function. It has two inputs, the first a triggering signal and the second the signal to be sampled. The module has one output which holds the last sampled value of the second input. The second input is sampled every time a low-to-high transition is detected in the signal of the first input. The level of a signal is considered to be “high” for amplitudes above 0.75 Volts while it is considered to be “low” for values below 0.25Volts. The signal of the first input must drop again (below 0.25V) before it can trigger another sample. This prevents minor variations (less than 0.5V) in the gate signal, due to noise, from triggering a new sample. The *hold* button on the front panel is used to manually sample the input signal independently of the value of the gate signal.

Typical Use

The **Sample & Hold** can be used with two **LFO**s for generating pseudo-random signals as shown in Figure 73. One **LFO** acts as the trigger of the **Sample & Hold** module while the second generates the signal to be sampled.

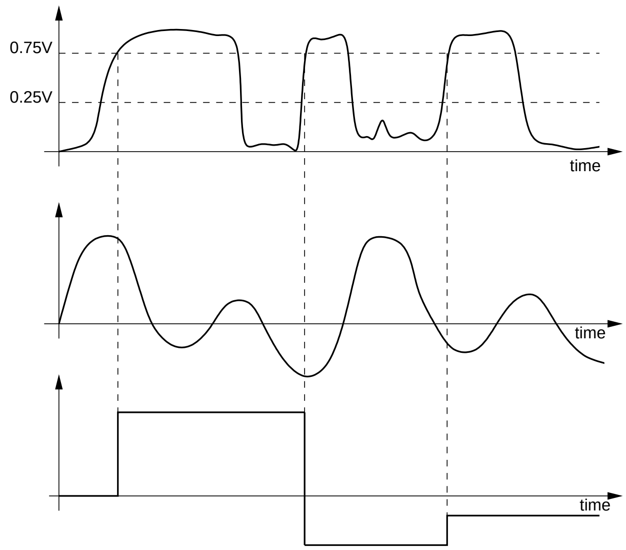


Figure 72: Behavior of **Sample & Hold** module.

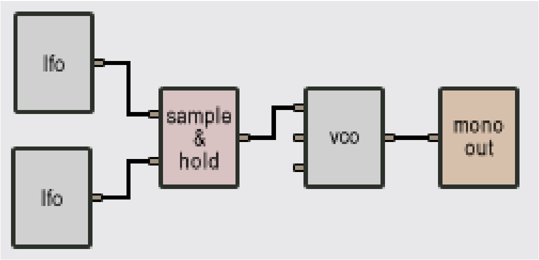


Figure 73: A **Sample & Hold** module is used to generate pseudo random signals.

6.73 Sbandpass2

This module is a static second-order band-pass filter (-6dB/octave). It is the same as the **Bandpass2** module but without the playing interface. The center frequency and resonance are static and set at the time of construction.

Typical Use

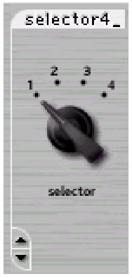
See **Vocoder** example, Figure 71, under **RMS** module.

The default value of the following parameters is set at construction

- *Center frequency*: frequency of the middle of the passing band.

- *resonance*: resonance around the center frequency.

6.74 Selector2, Selector3 and Selector4



The **Selector** module comes in 3 flavors: **Selector2**, **Selector3** and **Selector4**. These modules have 2, 3 or 4 inputs respectively and one output. The purpose of these modules is to connect the input corresponding to the position of the knob on the front panel to the output.

Typical Use

Selector modules can be used when one wants to change the cabling of a synth. It can be useful, for example, to be able to choose between an **ADSR** or a **LFO** to modulate the cutoff frequency of a filter.

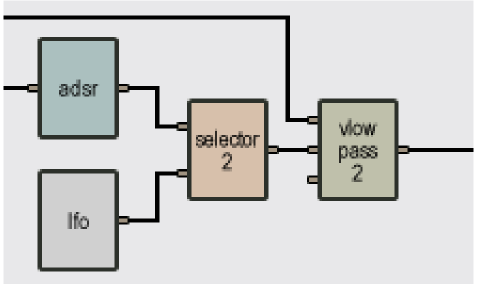
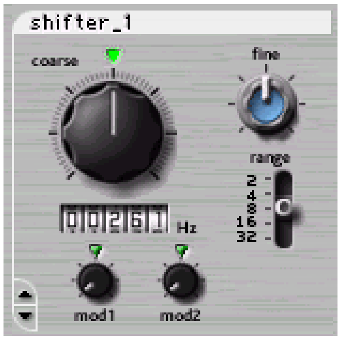


Figure 74: A **Selector2** switch is used to choose the modulation signal for the cutoff frequency of a **Vlowpass2** filter.

6.75 Shifter



The **Shifter** module is used to change the pitch of an audio signal. It functions similarly to analog hardware where a magnetic tape is scrolled in front of a two head system. In this way the pitch of the signal is changed without affecting the duration of the signal (unlike the effect obtained by simply slowing down or accelerating a tape).

This module has one output, the pitch shifted signal, and four inputs. The first input signal is the audio signal to be shifted. The pitch shift is adjusted with the *coarse* and *fine* knobs and the *range* switch. When the two knobs are in their center position (green LEDs on for the *coarse* knob), the *range* switch is set to 8 and there is no modulation signal, the playing frequency has a value of

261.6 Hz, which corresponds to the C3 key on a piano (middle C). The *range* switch transposes the pitch one or two octaves up or down. The reading on the counter gives the frequency of the output signal, in Hertz, when there is no modulation signal. The second input is used to give the module an estimate of the pitch of the original signal in order to minimize the artifacts it introduces. The convention followed here is 1 Volt/octave where a value of 0 Volt is C3 (middle C on the piano). For example, A3 is equivalent to a value of 9/12 Volts since this note is located 9 semitones above the C3. This input is usually connected to a **Constant** module. If you are not certain of the pitch of the original signal, you can use a **Constant** followed by a **Volume** module in order to tune this input signal. Finally the third and fourth input signals are used to modulate the pitch variation relative to the setting of the *coarse*, *fine* and *range* knobs and selectors. The gain of these modulation signals are controlled with the *mod1* and *mod2* knobs. The total modulation signal is the sum of the two inputs each multiplied by the gain corresponding to its respective *mod* knob. When the knobs are in the center position (green LEDs on), the gain equals 1 and the pitch variation is 1 Volt/octave. This position is used to play an equal temperament scale when connecting the output of a **Keyboard**.

Typical Use

In the example of Figure 75, a sample from a **Player** module is pitch-shifted with a Shifter, the final pitch being controlled by the Keyboard. The pitch of the original sample is adjusted with a Constant module.

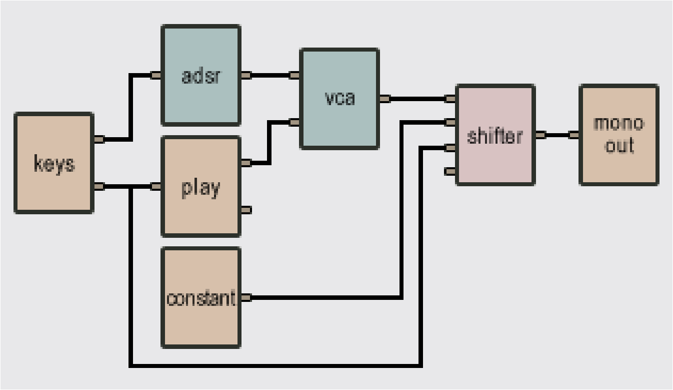
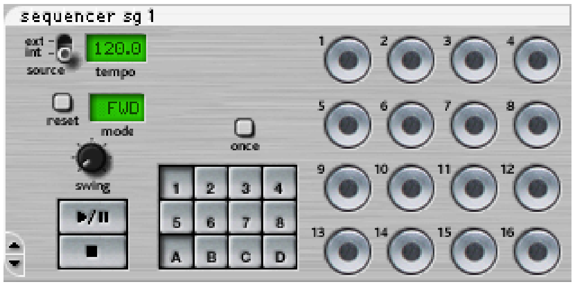


Figure 75: A sample from the **Player** is pitch-shifted with a **Shifter**.

6.76 Single Gate Sequencer

The **Single Gate Sequencer** module enables you to record sequences of gates. This module in itself does not produce sound but is used, usually instead of a **Keyboard** module, to trig other modules such as a **Player**. This module is a 16-step sequencer, which means that it plays sequences or patterns of 16 notes in loop. Sequences can be set to have 1 to 16 steps. Because each sequence represents a bar containing four quarter notes, each step of the sequencer itself represents a sixteenth note. The module can memorize 32 different sequences between which you can switch while playing.



This module has three inputs and four outputs. the first input is a sync signal which controls the tempo from an external source, the second is a start/stop input which will start the sequencer when it goes form 0 to 1 volt and stop it when it goes from 1 to 0 volt. The signal can come from another sequencer or a Keyboard. The third one is a reset input which will restart the sequence from beginning when it goes from 0 to 1 volt. The first three outputs are the same as the inputs (sync, start/stop, reset) and are used to control other sequencers. The fourth output is the gate signal which can be used as control source for other modules.

This sequencer has 16 *gate* buttons. The gate output will generate a square pulse of 1/8 of a quarter note for each active *gate* button.

Creating Patterns

To create a pattern, you must first select its location. You can select it with a combination of letters (A, B, C, D) and numbers (1 to 8), on the front panel, giving you a total of 32 patterns.

The sequencer will loop each time a pattern ends. To make the sequencer stop at the end of a pattern, the *once* button must be clicked.

The patterns can be played following 5 play modes using the *mode* control. Forward (FWD) plays the pattern incrementally. Backward (BWD) plays the pattern decrementally. Pendulum (PEND) plays the pattern forward then backward. Random 1 (RDN1) plays the pattern randomly, the same random sequence is repeated when looping. The *reset* button is used to generate a new random sequence. Random 2 (RDN2) plays the pattern randomly changing the random pattern when looping.

The *tempo* display will adjust the speed of the pattern. The *ext/int* switch will determine if it is the internal clock (int) that sets the tempo or an external source (ext) such as another sequencer or a **Sync Lfo**. The swing knob will introduce a swing feel to the rhythm of the pattern. The *gate* buttons control the gate output. The *gate* output will generate a square pulse of 1/8 of a quarter note for each active gate buttons. To hear a step, the *gate* button must be clicked (green light on). The *loop* buttons are used to set the length of the Pattern from 1 to 16 steps.

Typical Use

In the following example, two **Single Gate Sequencers** are synced together to control two **Player** modules.

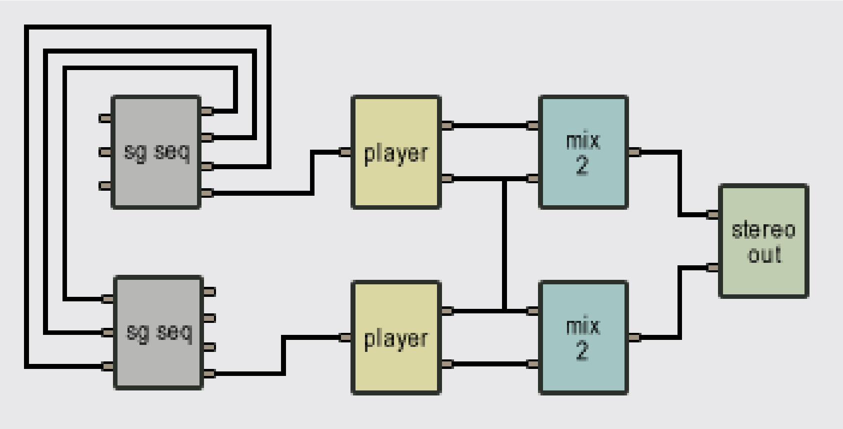
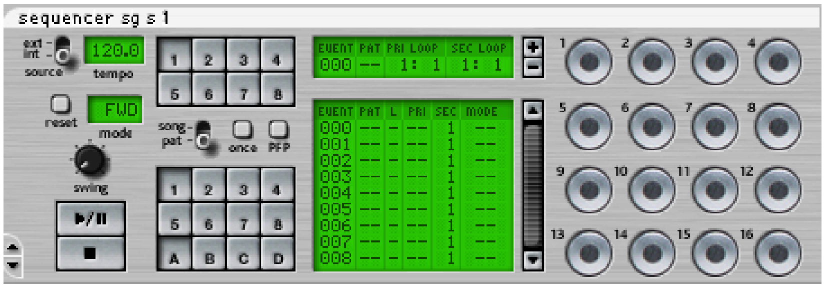


Figure 76: **Single Gate Sequencers** controlling **Player** modules.

Note: see also **Multi-Sequencer**, **Control Voltage Sequencer**, **Control Voltage Sequencer with Songs**, **Single Gate Sequencer with Songs**, **Dual Gate Sequencer** and **Dual Gate Sequencer with Songs**.

6.77 Single Gate Sequencer with Songs



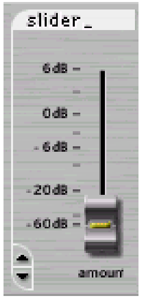
EVENT	PAT	PRI LOOP	SEC LOOP
000	--	1: 1	1: 1

EVENT	PAT	L	PRI	SEC	MODE
000	--	-	--	1	--
001	--	-	--	1	--
002	--	-	--	1	--
003	--	-	--	1	--
004	--	-	--	1	--
005	--	-	--	1	--
006	--	-	--	1	--
007	--	-	--	1	--
008	--	-	--	1	--

This module is the same as the **Single Gate Sequencer** but with song mode added. To read more about song mode, please refer to the **Multi Sequencer** module documentation.

Note: see also **Multi-Sequencer**, **Control Voltage Sequencer**, **Control Voltage Sequencer with Songs**, **Single Gate Sequencer**, **Dual Gate Sequencer** and **Dual Gate Sequencer with Songs**.

6.78 Slider



The **Slider** module is used to adjust the amplitude of a signal. It acts in the same way as the **Knob** module. It has one input and one output. The output signal is the input signal multiplied by a constant varying between 0 and 2 (+6dB).

Typical Use

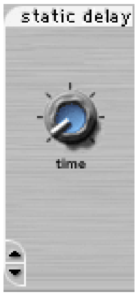
The **Slider** module is used whenever the level of a signal must be adjusted.

Note: See also **Gain**, **Selector**, **Volume**.

The default values of the following parameter is set at construction

- *gain*: default value of the volume gain (value between 0 and 2).

6.79 Static Delay



The **Static Delay** is a simple delay line with no feedback. Its single input is the signal to be delayed and its one output is the delayed signal. The delay time is adjusted with the delay knob on the front panel. The *delay* can have a maximum length of 5 seconds. To get a longer delay time, simply cascade multiple **Static Delay** modules.

Typical Use

The **Static Delay** will often be used to delay the gate signal triggering an **ADSR** module.

Note: See also **Delay**, **Sync Delay** and **Sync Ping Pong Delay**.

6.80 Stereo Audio In

The **Stereo Audio In** module is used to process external audio in *Tassman*. The outputs of this module is the left and right signals received from a track or a bus of a host sequencer where the *Tassman* has been inserted as an effect. This signal can be then be processed on the fly by *Tassman* modules and then sent back to the track or the bus trough the use of an **Audio Out** or **Stereo Audio Out** module.

Note: See also **Audio In**.

6.81 Stereo Audio Out

The **Stereo Audio Out** module is used to output stereo signals. It has two inputs, the first sent to the left audio channel of the sound card, the second to the right channel. For the saturation characteristics of this module, refer to the **Audio Out** module.

Typical Use

A stereo signal can be produced from a mono signal using a **Panpot** module. Here sound sources one and two are transformed to stereo using the **Panpot** modules:

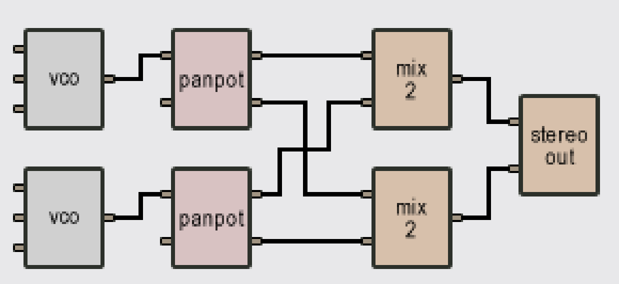
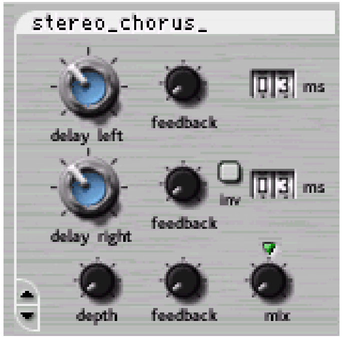


Figure 77: Use of **Panpot** module and a **Stereo Audio Out** to obtain a stereo output.

Note: See also **Audio Out**.

6.82 Stereo Chorus



The **Stereo Chorus** module implements a wide range of stereo effects such as flanging, chorusing and vibrato. This module has three inputs and two outputs. The first and second inputs are respectively the left and right audio signals to be filtered while the third input is a modulation signal. The outputs are respectively the left and right filtered signals.

The **Stereo Chorus** module consists, as shown in Figure 58, of two **Flanger** modules, one for the left channel and one for the right channel, with cross feedback between the two channels.

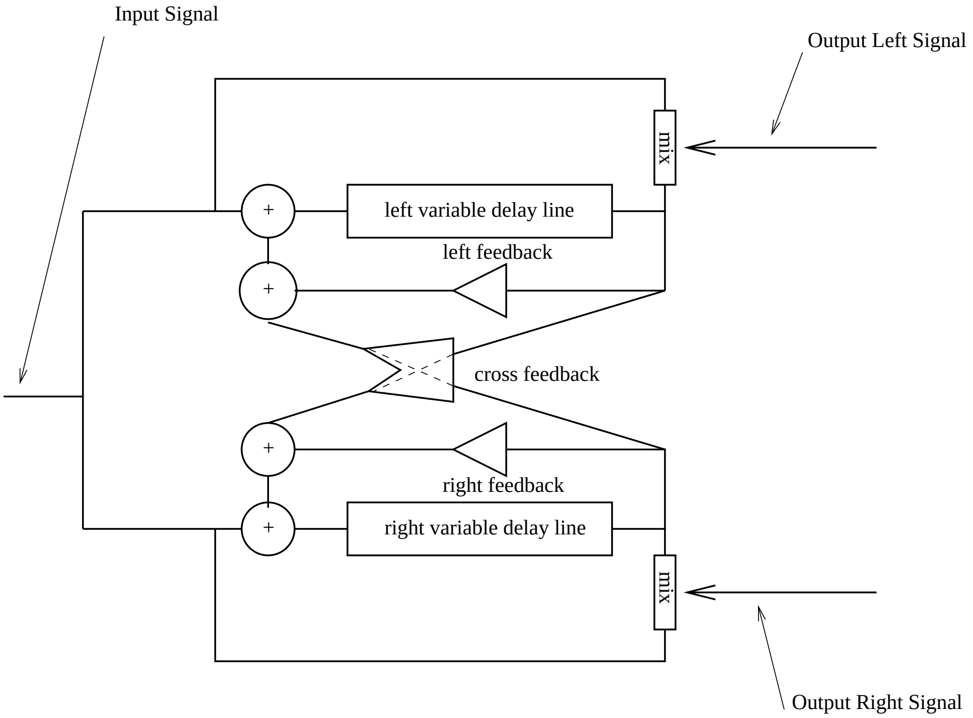


Figure 78: **Stereo Chorus** module.

Tuning

The length of the delay lines associated with the left and right channel, are adjusted with the *delay left* and *delay right* knob respectively. The length of the two lines are displayed, in milliseconds, in the counter next to these knobs. The *feedback* knobs located on the right of the *delay* knobs are used to fix the amount of "wet" signal re-injected into the left and right delay lines. The length of these delay lines can be modulated by using the third input of the module, the amount of modulation depending on the adjustment of the *depth* knob. In the left position, there is no modulation and the delay lines remain fixed while in the right position, with a modulation signal varying between [-1,1] Volt, the delay lines varies between 0 and twice the value set with the *delay* knobs. Note that the modulation on the left and right channels can be set out of phase by pressing the *inv* button on the right channel. The *feedback* knob located on the right of the *depth* knob is used to adjust the amount of cross-feedback between the two channels. Finally, the *mix* knob determines the amount of "dry" and "wet" signal sent to the output. When this knob is adjusted in the left position, only "dry" signal is sent to the output, in its center position (green LED On), there is an equal amount of "dry" and "wet" signal in the output and in the right position, only "wet" signal is sent to the output.

The default value of the following parameters is set at construction

- *delay*: time delay, in seconds, applied to the left and right input signals (values between [0, 92]ms).
- *feedback*: coefficient,[0, 1[, determining amount of "wet" signal re-injected into the delay lines. If *feedback* = 0 there is no "wet" signal re-injected while if *feedback* = 0.99, maximum of "wet" signal re-injected.
- *depth*: gain coefficient, [0,1], multiplying the modulation signal.
- *mix*: amount of "dry" and "wet" signal sent to left and right outputs. If *mix* = 0 there is only dry signal while if *mix* =1, there is only "wet" signal.

Note: For typical uses, see **Flanger** module.

6.83 String

The **String** module simulates sound production by strings of different materials and sizes. This module first calculates the modal parameters corresponding to string-shaped objects according to the value of the different parameters requested at construction time and, next, calls the **Multimode** module to simulate the sound. The module has one output, the sound produced by the string, and three inputs. The first input signal is a damping signal which, depending on its value, lowers or raises dampers on the structure. When the input signal is equal to 0, dampers are lowered on the string, thus shortening the decay time of the sound produced by the structure. When the signal is greater than 0, dampers are raised. Note that this damping adds to the natural damping of the string itself. If this input is not connected to any other module, the default value is set at 0, which implies that the string motion will be damped. This input is, therefore, usually connected to a **Constant** module to obtain undamped motion or to a **Damper** module or the gate signal from a **Keyboard** in order to vary the damping while playing. The second input signal is the force signal exciting the string, and the third is a pitch signal which is multiplied by the gain corresponding to the adjustment of the *mod* knob on the module front panel.

Typical Use

In the following example, the output from a **String** is filtered by a **Plate** module.

The default values of the following parameters is set at construction

- *Length*: the length, in meters, of the string.
- Frequency: fundamental frequency, in Hertz, of the string beam when there is no pitch modulation signal or when its value is equal to 0. Note that the fundamental frequency is independent of the length of the string. The software automatically calculates the physical

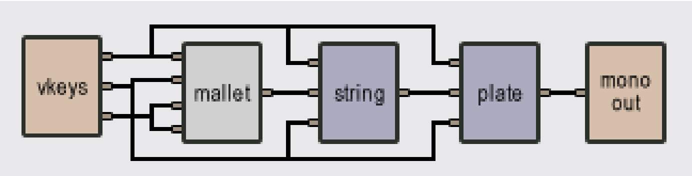


Figure 79: A **String** exciting a **Plate**.

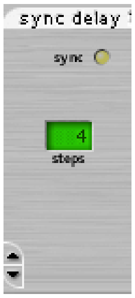
parameters necessary to obtain the required fundamental frequency. The default value of this parameter is 261.62 Hz which corresponds to the middle C (C3) of a piano keyboard. This setting is convenient when controlling a **String** module with a **Keyboard** module. Decay: proportional to the decay time of the sound produced by the string.

- *Inharmonicity*: detunes the partial toward higher frequencies with respect to the fundamental. This parameter has a value between 0 and 1 where 0 represents a perfect string.
- *Number of Modes*: number of modes used to simulate the object. As the number of modes is increased, the number of partials in the sound increases but also inevitably the calculation load.
- *Excitation point*: x-coordinate, in meters, of impact point from the extremity of the string.
- *Listening point*: x-coordinate, in meters, of listening point from the extremity of the string.

Note: For more details on this module and especially the front panel controls, see the **Multimode** module.

6.84 Sync delay

This modules is a simple delay line. Its main feature is that the delay time is adjusted in steps relatively to a sync input. The first input of the module is used to plug the sync input coming from a sequencer or from a **Sync LFO**. The second input if the signal to be delayed. The output of the module is the same as the input signal delayed by the amount of steps displayed in the front panel window, four steps representing a quarter note. The LED light is illuminated when there is a sync signal and the delay time does not exceed the delay line size.

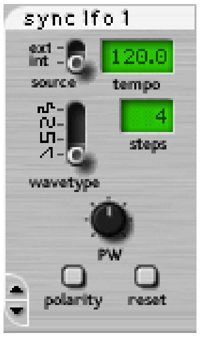


Note: See also **Delay**, **Static Delay** and **Sync Ping Pong Delay**.

6.85 Sync LFO

The **Sync LFO** is a low frequency oscillator that can be synced to an external clock signal. The sync signal can come from a **Sequencer** or another **Sync LFO**. This module generates signals at very low frequencies used as control signals rather than audio ones. It has two inputs and three outputs. The first input is the sync input signal which is used to sync the module to an external source. The second input resets the waveform at the beginning of its cycle each time a signal above 0.1 Volt is received. The first two outputs are the same as the first two inputs and are used to control other **Sync LFO** or **Sequencers**. The third output is the waveform signal delivered by the module.

The module can generate four waveforms: sine, sawtooth, square and random. The waveform is selected with the four position switch on the front panel. The width of the waveform can be adjusted with the *pw* knob. The *polarity* knob inverses the phase of the waveform. The *reset* knob enables one to manually resets the waveform at the beginning of its cycle each time it is pressed. The *ext/int* switch selects if the sync signal comes from another source (ext) or the internal clock of the module (int). The green *tempo* display is used to set the tempo of the module, to adjust it, click-hold on it and drag up or down. The *step* control is used to multiply or divide the sync signal by the number of steps indicated in the display. When the *ext/int* switch is set to *ext* the internal tempo controls have no effect.

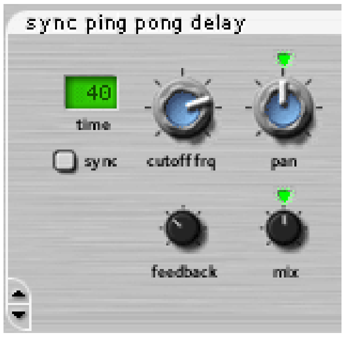


Typical Use

Two **Sync LFO** can be linked to create double synced modulation source for other modules.

6.86 Sync Ping Pong Delay

The **Sync Ping Pong Delay** is a module which generates echoes that can be panned in order to regularly alternate between the left and the right channel. This module has three inputs and two outputs. The first input is a sync signal from a **Master Sync Input**, **Sync LFO** or a **Sequencer** module. The second and third inputs are source signals sent into the left and right channel while the two outputs of the module are the left and right channel output signal respectively.



The algorithm implemented in this module is presented in Figure 80. It is based on two delay lines each including a low-pass filter. The signal at the end of each delay line is fed back into the input of the other line with an attenuation coefficient. This algorithm results in a signal traveling from one channel to the other, each time attenuated and filtered in the high frequencies due to the gain factor and the presence of the low-pass filter.

Tuning the delay

The *time* knob sets the length of the delay line and therefore the time between echoes. When the *sync* button is pressed, the sync signal from the first input of the module is used to determine the

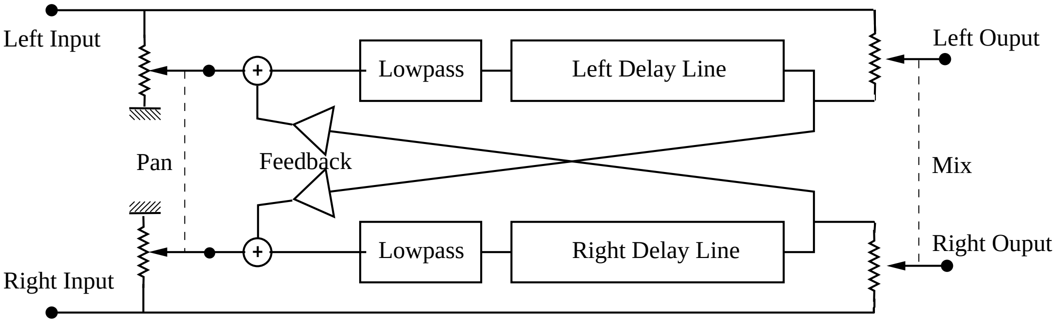


Figure 80: **Ping Pong Delay** algorithm.

length of the delay line which is adjusted to fit the number of steps appearing in the display, four steps representing a quarter note. The *feedback* knob is used to adjust the amount of signal re-injected from the end of one line into the input of the other one. The *cutoff* knob controls the cutoff frequency of the low-pass filter applied to the signal in the feedback loop and is used to reduce the brightness of each echo. The *pan* knob is used to balance the input signals between the left and right channels. In its leftmost position, signal will only be fed into the left delay line and one will hear clearly defined echo first from the left channel and then from the right channel and so on. In its rightmost position, the behavior will be similar but with the first echo coming from the right channel. These two extreme position correspond to the standard ping pong effect but a a less extreme behavior can be obtained by choosing an intermediate position. Finally, the *mix* knob is used to control the amount of “dry” and “wet” signal in the output signal. In its leftmost position only dry signal is sent to the output while when fully turned to the right, only wet signal is heard.

Typical Use

In Figure 81, a **Sync Ping Pong Delay** module is used to add a space effect to a mono signal.

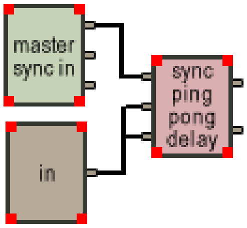


Figure 81: Creating a stereo delay effect with mono signal.

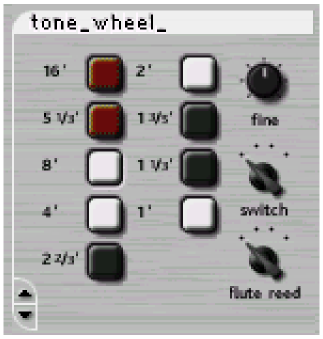
Note: A **Sync Ping Pong Delay** is included in the output stage effect stage as explained in Section 4.8. See also **Delay** and **Sync Delay**.

6.87 Toggle

The **Toggle** module is a clock divider. This module has two inputs and one output. The first input is the clock signal to be divided. The second input is used to reset the circuit. The output is the input signal with a frequency divided by two. To perform this operation properly, this module should only be used with clock signals in the first input.

6.88 Tone wheel

The **Tone wheel** module is used to build combo organs. It is mainly based on a Hammond tone wheel but also allows to generate different tone colors ranging from flute-like sounds (Hammond) to reed-like sounds. This module has two inputs, a gate and a pitch signal (1 Volt/octave), and one output, the signal produced by the tone wheel.



Tuning

The nine buttons on the front panel are used to determine the note played by the tone wheel. The different note possibilities follow harmonic relationships and are labeled in feet (as a reference to pipe lengths in organs). When the 8´button is pressed, the tone wheel will play the note corresponding to the signal received on the pitch input of the tone wheel (usually connected to the pitch output of a **Keyboard** module, convention is 1 Volt/octave). When the 16´button is pressed the output is one octave below. The other buttons, 5 1/3, 4, 2 2/3, 2, 1 3/5, 1 1/3, output a note a fifth, an octave, a twelfth, 2 octaves, 2 octaves and a major third, 2 octaves and a fifth and 3 octaves above the pitch input respectively. The pitch of the output can be fine tuned by plus or minus 3 semitones using the *fine* knob.

Switch effect

In addition to the sound of the tone wheel itself, the timbre of original hardware electric organs also contains very typical noise components. This noise is due to two separate effects. First, when a key is depressed on an organ, mechanical imperfections result in a slight random triggering delay between tone wheels which has the effect of blurring the attack. The second component is a contact noise. The **Tone Wheel** module reproduces this characteristic behavior of organs. When the *switch* selector is in the left position there is no noise and the random delay is minimal. In the right position, the amount of noise and random delay are maximum

Timbre

The timbre of the output of the tone wheel can be varied with the *flute reed* selector. In the left position, the module outputs a sine-like tone. As the selector is turned to the right, the signal gets distorted and evolves toward a triangle-like tone as its harmonic content increases.

Typical Use

Tone wheel modules are usually used in parallel. In the example of Figure 82, three modules are connected to a polyphonic **Keyboard** module. Each of the **Tone wheel** module is tuned to a different note in order to create a complex tone. The amplitude of every component of the tone can be adjusted with a **Volume** slider reproducing drawbars on hardware organs.

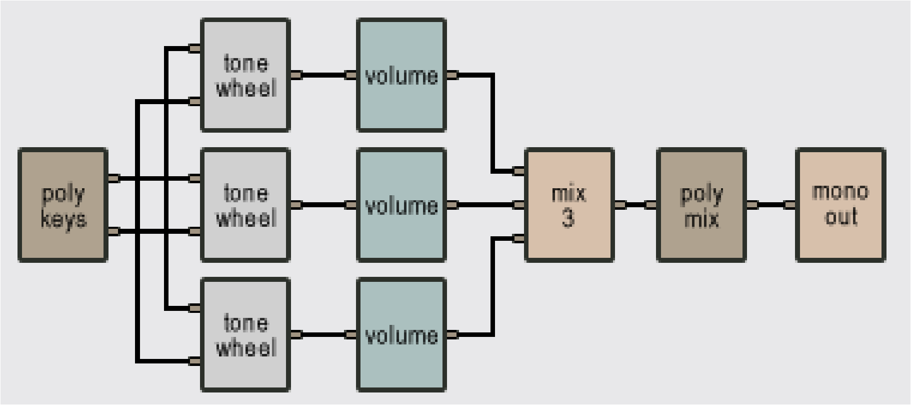
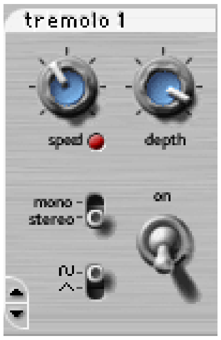


Figure 82: A simple polyphonic organ.

6.89 Tremolo



This module has one input and two outputs. It applies a tremolo effect to the input signal and outputs the resulting signal on the two outputs.

A tremolo effect consists in modulating the amplitude of a signal with a low frequency wave. The *speed* and *depth* knobs of the front panel is used to adjust the speed and depth (frequency) of the modulation wave respectively. The *waveform* selector is used to switch between a triangle and square modulation signal. With the *mono/stereo* switch in stereo position the tremolo will bounce with a 180 degrees phase from left to right. When the switch is on mono the left and right output signals are the same. The *on* switch is used to switch the tremolo on or off. The red LED indicates the speed of the vibrato.

6.90 Tube

The **Tube** module simulates sound propagation in a cylindrical tube of a given length and radius. The effect of a tube is to color an input signal by enhancing frequencies located around its resonance frequencies. When the tube is very long, it produces an echo effect. The source is assumed to be at an extremity of the tube. The output of this module is the signal that would be measured by a microphone placed at the other extremity of the tube. The geometry of the tube is determined during construction.

Tuning a tube

The resonance frequencies of a tube depend on its length and termination. A tube with open extremities has resonances located at harmonic intervals, i.e. located at 1, 2, 3 . . . times any fundamental frequency as can be see in Figure 83. When an extremity is closed, the tube only has resonance for odd harmonics of its fundamental frequency.

A tube can be tuned by adjusting its length in order to obtain a given fundamental resonance frequency. When the tube is open at both of its extremities, the fundamental frequency is given by c/2*length* where *c* is the speed of sound (approximately 344 m/s in air, depending on the temperature) and *length* is the tube length. For a tube with one extremity closed, the fundamental frequency is given by c/4*length*.

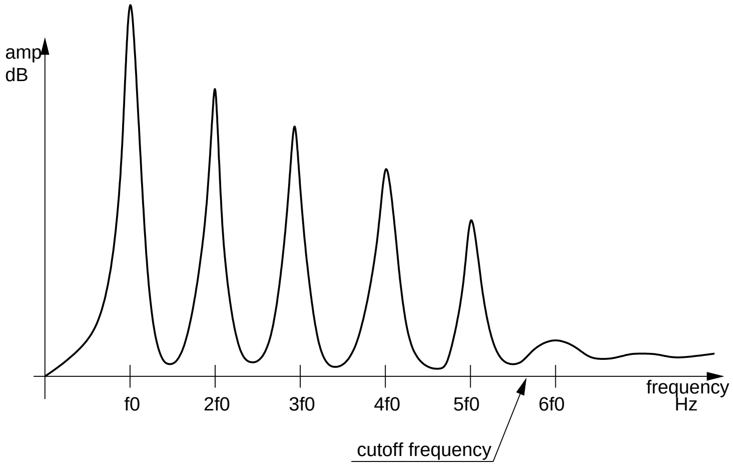


Figure 83: Resonance frequencies of a cylinder with both extremities open.

Amplitude of the tube resonances

The amplitude of the resonances can be adjusted with the radius of the tube when it is open at the listening point. At the extremity of a tube sound energy is radiated toward the exterior, the termination in fact acting like a low-pass filter. Increasing the radius of the tube, both increases the amount of energy radiated (thereby decreasing the amplitude of the tube resonances) and lowers the cutoff frequency of the filtering effect of the termination. In other words, you can obtain very strong resonances and long decay time by using tubes of small radii. Note that changing the radius of a tube has no effect when a closed termination is used. The decay time in the tube can also be adjusted with the *decay* knob on the module front panel.

Typical Use

The **Tube** module can be used to introduce delay in a patch. Tubes can also be used as sympathetic resonators as illustrated in Figure 62. In this example the output of a String module is sent into an array of three tubes each having a different length. The tubes enhance certain notes depending on their resonance frequencies.

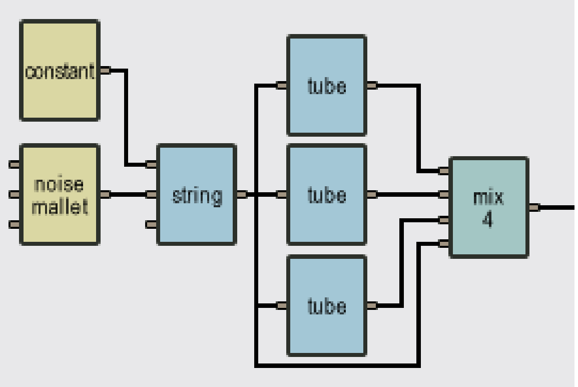


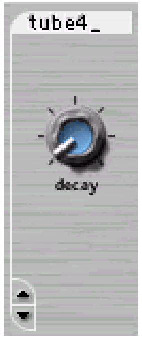
Figure 84: Sympathetic tubes.

The default values of the following parameters is set at construction

- *length*: length, in meters, of the tube (between 0.01 and 1000m).
- *radius*: radius, in meters, of the tube (between 0.001 and 1 m).
- *termination*: specifies whether the tube is open or closed at its extremity. A value of 0 indicates that the tube is closed and a value of 1 that it is open.

Note: See also **Tube4** and **Tube Reverb**.

6.91 Tube4



The **Tube4** module simulates sound propagation in a resonator made from 4 tubes of variable lengths and radii connected in series as shown in Figure 85. The input signal, or source, is assumed to be localized at the extremity of the first tube while the output, or listening point, is placed at the extremity of the fourth tube.

The discontinuities in the resonator scatter acoustic energy at these points and thereby create different standing wave patterns in the different sections of the resonator. This results in a complex filtering effect that varies according to the relative lengths and radii of the different tubes. The *decay* knob controls the decay time of sound wave traveling inside the resonator.

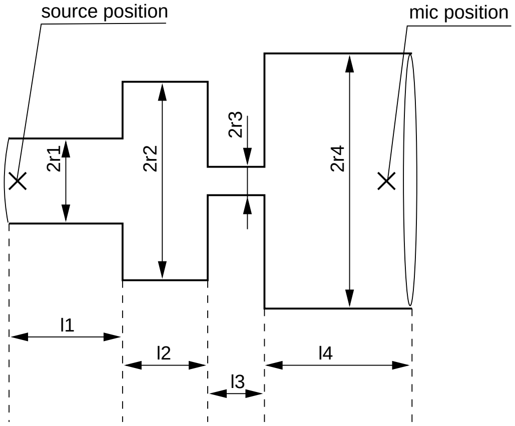


Figure 85: Tube4 module geometry.

Typical Use

Tube4 modules can be used to obtain reverb effects. When the total length of the tube is long, the module will introduce delay in the patch. The discontinuities in the resonator filter the input signal depending on their relative lengths and radii. In the example of Figure 86, under reverb, a **Tube4** module is used in combination with two **Reverb** modules in order to make a stereo reverb. The **Reverb** modules produce early reflections in a room while the **Tube4** module produces the late reflections.

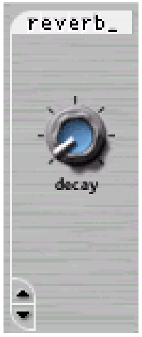
The default values of the following parameters are set at construction

- *length*: length, in meters, of the 4 tubes (between 0.01 and 1000m).
- *radius*: radius, in meters, of the 4 tubes (between 0.001 and 1 m).

- *termination*: specifies whether the final tube is open or closed at its extremity. A value of 0 indicates that the tube is closed and a value of 1 that it is open.

Note: For more details on the filtering effect of tubes see the **Tube** module.

6.92 Tube Reverb



The reverb effect obtained with this module is obtained with an assemblage of three tubes. The tubes are assumed to be connected at one of their extremities and the sound source to be located at this connection point. Each tube of the array filters the source according to its resonance frequencies, but, since the three are connected together, the output of each tube is also distributed to the other tubes which results in interesting reverb effects. This module has one input, the source signal, and its one output is the signal that would be measured by a microphone located at the tube connection point.

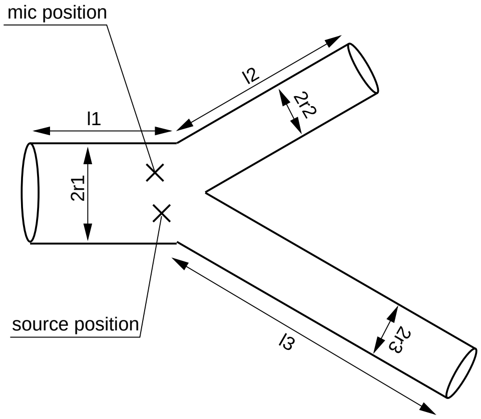


Figure 86: A reverb made out with three tubes.

Tuning the reverb

The reverberation effect obtained with the module can be tuned by adjusting the lengths and radii of the different tubes. The longer the tube, the longer the reflection time and the smaller the radii, the lower the absorption. Small radii will also increase the amount of high frequencies in the sound. As a rule of the thumb, use tubes having a length similar to the size of the room or space you are imagining.

Typical Use

In the example of Figure 87, two **Tube Reverb** modules are used to make a stereo reverb effect. The **Reverb** modules are adjusted with short tubes in order to simulate early reflections in a room. The **Tube4** module is used to introduced a delay and simulate late reflections.

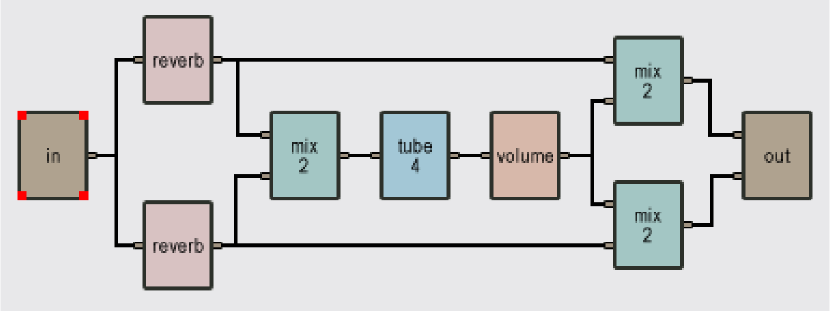


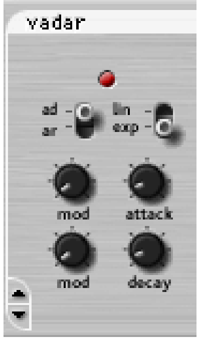
Figure 87: A stereo reverb.

The default value of the following parameters is set at construction

- *length*: length, in meters, of the 3 tubes (between 0.01 and 1000m).
- *radius*: radius, in meters, of the 3 tubes (between 0.001 and 1 m).

Note: For more details on the filtering effect of tubes see the **Tube** module. See also **Reverberator**.

6.93 VADAR



The **VADAR** acts exactly like the **ADAR** module except that the **VADAR** has two additional inputs for controlling the *attack* time and the *decay* time. It also has two more knobs to adjust the gain of those two inputs. The modulation signals affect the duration of the attack and decay stages: the higher is the amplitude of the modulation signal the shorter is the attack or decay time; the lower is the amplitude of the modulation signal the longer is the attack of decay time.

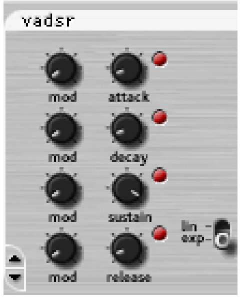
Typical Use

The **ADAR** is typically used for generating amplitude envelopes through a **VCA**, or spectral envelopes by modulating the frequency of the filter modules.

The modulation entries can be connected to the velocity output of a **Vkeyboard** or a **Sequencer** module.

Note: See also the **ADAR**, **ADSR** and **VADSR** modules.

6.94 VADSR



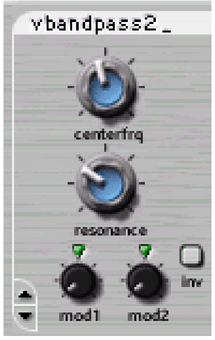
The **VADSR** acts exactly like the **ADSR** module except that the **VADSR** has four additional inputs for controlling each phase of the envelope. It also had four more knobs for adjusting the gain of these four inputs. The first, second and fourth modulation signals affect the duration of the attack decay and release phases: the higher is the amplitude of the modulation signal, the shorter is the attack, decay or release time; the lower is the amplitude of the modulation signal, the longer is the attack, decay or release time. The third modulation affects the sustain level.

Typical Use

The **VADSR** is typically used for generating amplitude envelopes through a **VCA**, or spectral envelopes by modulating the frequency of the filter modules. The modulation entries can be connected to the velocity output of a **Vkeyboard** or a **Sequencer**.

Note: See also the **ADAR**, **ADSR** and **VADAR** modules.

6.95 Vbandpass2



The **Vbandpass2** module is a voltage-controlled second-order band-pass filter (-6dB/octave). This module has one output, the filtered signal, and three inputs. The first input is the signal to be filtered, while the second and third inputs are modulation signals which are used to vary the center frequency of the filter. The amplitude of the two modulation signals is adjusted with the two gain knobs *mod1* and *mod2* respectively.

Tuning the Center Frequency

The *center frq* knob tunes the center frequency to the desired level. The variations in center frequency caused by changes in the modulation inputs are relative to this level. The *resonance* knob is used to emphasize the frequencies near the center frequency as is shown in Figure 24 under **Bandpass2**.

Center Frequency Variation

The amount of variation of the center frequency obtained with the modulation inputs depends on the adjustment of the *mod1* and *mod2* gain knobs. The total modulation signal is the sum of the two inputs each multiplied by the gain corresponding to its respective *mod* knob. When they are in the center position (green LEDs on), the gain equals 1 and the pitch variation is 1 Volt/octave. When the modulation signal is the note output from a **Keyboard** module, this position can be used to make the center frequency follow an equal temperament scale. The modulation signal of the second input can be inverted by pressing the inv button. This can be useful when generating bass sounds, for example, where one wants to close the filter with an upward going envelope such as during the attack of a note.

Typical Use

A **Vbandpass2** and an **ADSR** module can be used to obtain an auto wah wah effect. In this example, the center frequency of the filter is modulated with the output from the **ADSR** module.

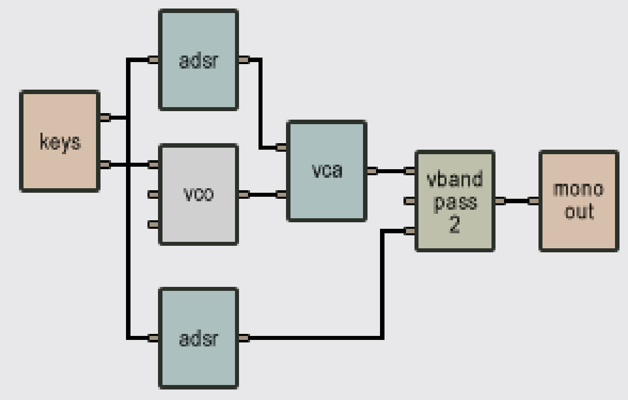


Figure 88: Auto wah wah effect obtained with an **ADSR** and Vbandpass2 module.

Note: See also the **Bandpass2** module.

6.96 VCA (Voltage Controlled Amplifier)

The VCA module has two input signals, a signal to be amplified, and a gain. The output is the product of the two inputs. This module has no front panel control.

Typical Use

A **VCA** is mainly used to apply an amplitude envelope to a signal. An **ADSR** can be used to supply the appropriate gain signal.

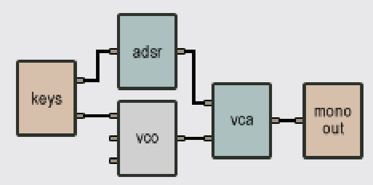


Figure 89: ADSR as Gain Signal to VCA.

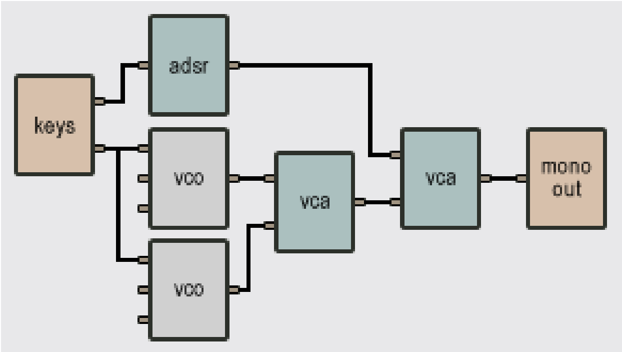


Figure 90: **VCA** in Ring.

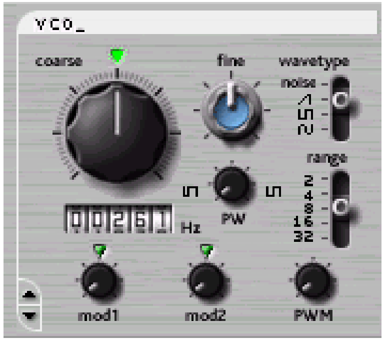
A **VCA** can also be used to obtain a ring modulation effect. In this case the gain signal is a sine wave around 50 Hz.

Note: Since a multiplication operation is involved, the order of the inputs is not important. For the same reason, if one of the two input signals is null, the output will also be null.

6.97 VCO (Voltage Controlled Oscillator)

The **VCO** module is an oscillator used to generate signals of different frequencies and waveforms. This module has 3 modulation inputs and 1 output.

The sum of the first two inputs controls the pitch of the output signal. Each of these two inputs is multiplied by its respective gain knob, *mod1* or *mod2*. The third input controls the pulse width for the pulse wave subject to the setting of the *PWM* knob.



Tuning the Output Pitch

The *coarse* and *fine* knobs and the *range* switch are used to tune the output frequency (or pitch) to the desired level. The variations in output pitch caused by changes in the modulation signals are relative to this level. When the two knobs are in their center position (green LED on for the *coarse* knob), the *range* switch is set to 8 and there is no modulation signal, the playing frequency has a value of 261.6 Hz, which corresponds to the C3 key on a piano (middle C). The *range* switch transposes the pitch one or two octaves up or down. The reading on the counter gives the frequency of the output signal, in Hertz, when there is no modulation signal.

Pitch Variation

The amount of variation of the playing frequency obtained with the modulation inputs depends on the adjustment of the *mod1* and *mod2* gain knobs. The total modulation signal is the sum of the two inputs each multiplied by the gain corresponding to its respective *mod* knob. When the knobs are in the center position (green LEDs on), the gain equals 1 and the pitch variation is 1 Volt/octave. This position is used to play an equal temperament scale when connecting the output of a **Keyboard** module to a modulation input of a **VCO**. The frequency variation with the modulation signal can be increased or decreased by turning the modulation knobs clockwise or anti-clockwise.

Waveform

The *wavetype* switch switches between the four well-known waveforms: noise, sawtooth, pulse and sine. When choosing the pulse wave, the width of the pulses is adjusted with the *PWM* knob. When the *PWM* knob is adjusted in the right position the waveform is square and only includes odd harmonics. The pulse width can be modulated by an external signal through the pulse width modulation input. The *PWM* knob is used to control the amount of modulation applied by this third input signal. When this knob is in the left position, there is no modulation applied to the pulse width while when it is in the right position, the amplitude of the modulation is almost equal to the width of the pulse. Figure 91 shows the result of the modulation of the pulse width by a sine wave

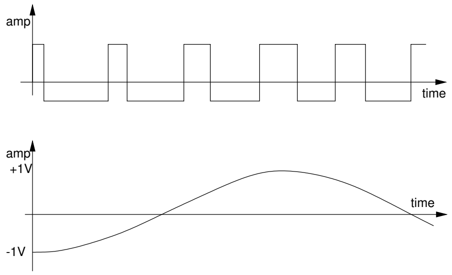


Figure 91: Sine Wave Modulation of Pulse Width.

Typical Use

The **VCO** is used for generating the starting signal of an analog synthesizer. Figure 92 shows a standard patch using this module.

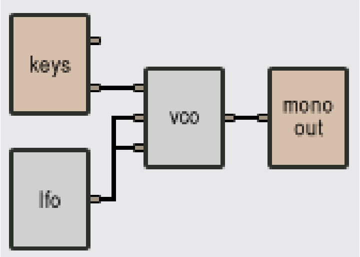
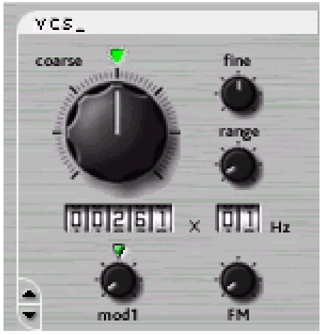


Figure 92: Typical **VCO** Use.

6.98 VCS



The **VCS** module is very similar to the VCO module except that it only generates sine waves. This module has 2 modulation inputs and 1 output.

The first input controls the pitch of the output signal. The signal of this input is multiplied by a value determined by the adjustment of the *mod1* gain knob. The second input signal is used to perform frequency modulation (FM modulation).

Tuning the Output Pitch

The *coarse*, *fine* and *range* are used to tune the output frequency (or pitch) to the desired level. The variations in output pitch caused by changes in the modulation signals are relative to this level. When the two knobs are in their center position (green LEDs on for the *coarse* knob), the *range* knob is in the left position and there is no modulation signal, the playing frequency has a value of 261.6 Hz, which corresponds to the C3 key on a piano (middle C). The *range* knob transposes the pitch by multiplying the frequency appearing in the frequency counter by the number appearing in the range counter. This enables to generate the different harmonics of the fundamental frequency appearing in the frequency counter. The reading on the counter gives the frequency of the output signal, in Hertz, when there is no modulation signal.

Pitch Variation

The amount of variation of the playing frequency obtained with the modulation inputs depends on the adjustment of the *mod1* and *FM* gain knobs. The total modulation signal is the sum of the two inputs each multiplied by the gain corresponding to their respective knob. When the *mod1* knob is in the center position (green LEDs on), the gain equals 1 and the pitch variation is 1 Volt/octave. This position is used to play an equal temperament scale when connecting the output of a **Keyboard** module to this modulation input. The *FM* modulation input is used to apply frequency modulation to the signal generated by the **VCS** module. The amount of frequency modulation applied by the modulation signal is controlled with the *FM* knob.

Typical Use

The FM synthesis technique can be reproduced by using **VCS** modules connected in cascade as in the following example.

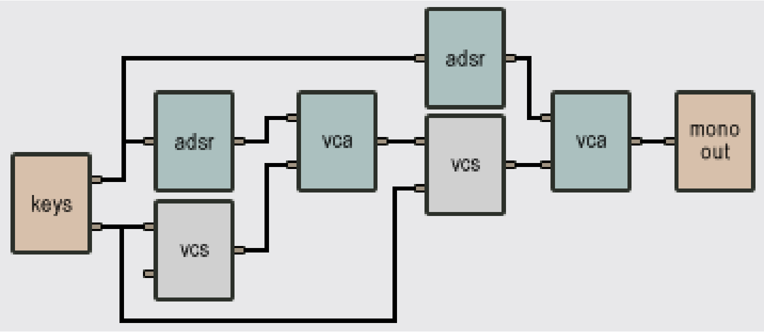
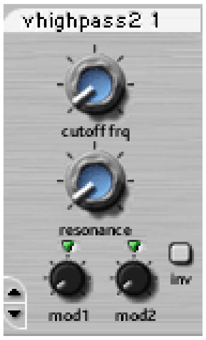


Figure 93: FM synthesis with **VCS** modules.

6.99 Vhighpass2



The **Vhighpass2** module is a voltage-controlled second-order high-pass filter (-12dB/octave). This module has three inputs and one output. The first input is the signal to be filtered, while the second and third inputs are modulation signals which are used to vary the cutoff frequency of the filter. The amplitude of the two modulation signals is adjusted with the two gain knobs *mod1* and *mod2* respectively. The output is the filtered signal.

Tuning the Cutoff Frequency

The *cutoff frq* knob tunes the cutoff frequency to the desired level. The variations in the cutoff frequency, caused by changes in the modulation inputs, are relative to this level. The **resonance** knob is used to emphasize the frequencies near the cutoff frequency as shown in the following figure:

Cutoff Frequency Variation

The amount of variation of the cutoff frequency obtained with the modulation inputs depends on the adjustment of the *mod1* and *mod2* gain knobs. The total modulation signal is the sum of the two inputs each multiplied by the gain corresponding to its respective *mod* knob. When the *mod* knobs are in the center position (green LEDs on), the gain equals 1 and the pitch variation is 1 Volt/octave. When the modulation signal is the pitch output from a **Keyboard** module, this position can be used to make the cutoff frequency follow an equal temperament scale. The modulation signal of the second input can be inverted by pressing the *inv* button.

Typical Use

A **Vhighpass2** can be used as a filter to reduce the low frequencies in a signal, as shown in the following example :

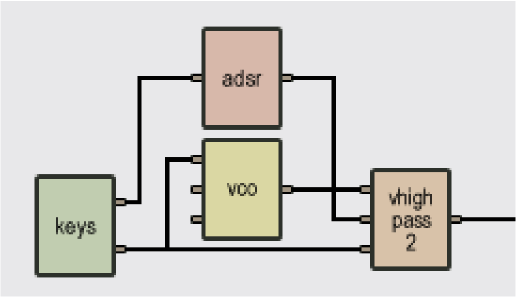
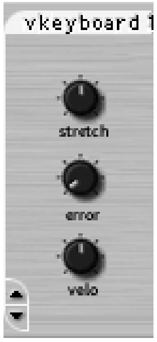


Figure 94: Use of **Vhighpass2** to reduce low frequencies.

6.100 Vkeyboard



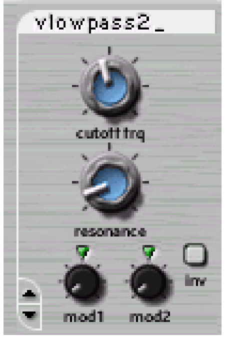
Similar to the monophonic **Keyboard** module except that there is an additional output proportional to the velocity with which a key is depressed. The *error* knob introduces some randomness in the pitch signal. Turned to the left, the pitch signal is correct; as the knob is turned to the right, small fluctuations start to appear in pitch. The effect of this knob is to simulate pitch variations found in analog synths. The velocity knob adjusts the velocity curve of the **Polyvkey**. In the center position, the curve is linear. Turned to the left, the velocity increases more quickly; conversely, turning the knob to the right results in a slower velocity curve.

The default values of the following parameters is set at construction

- *pitch wheel range*: determines the range of pitch variations that can be obtained with the pitch wheel. The convention is 1 Volt/octave (maximum value is 2 Volts). A semitone is equal to a 0.08333 value.
- *MIDI channel*: MIDI channel used by the keyboard.

6.101 Vlowpass2

The **Vlowpass2** module is a voltage-controlled second-order low-pass filter (-12dB/octave). This module has three inputs and one output. The first input is the signal to be filtered, while the second and third inputs are modulation signals which are used to vary the cutoff frequency of the filter. The amplitude of the two modulation signals is adjusted with the two gain knobs *mod1* and *mod2* respectively. The output is the filtered signal.



Tuning the Cutoff Frequency

The *cutoff frq* knob tunes the cutoff frequency to the desired level. The variations in the cutoff frequency, caused by changes in the modulation inputs, are relative to this level. The *resonance* knob is used to emphasize the frequencies near the cutoff frequency as shown in the following figure:

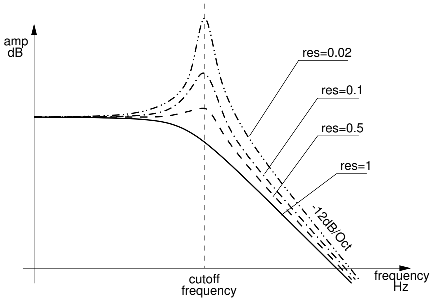


Figure 95: Frequency Response of **Vlowpass2**.

Cutoff Frequency Variation

The amount of variation of the cutoff frequency obtained with the modulation inputs depends on the adjustment of the *mod1* and *mod2* gain knobs. The total modulation signal is the sum of the two inputs each multiplied by the gain corresponding to its respective **mod** knob. When the *mod* knobs are in the center position (green LEDs on), the gain equals 1 and the pitch variation is 1 Volt/octave.

When the modulation signal is the pitch output from a **Keyboard** module, this position can be used to make the cutoff frequency follow an equal temperament scale. The modulation signal of the second input can be inverted by pressing the *inv* button. This can be useful when generating bass sounds, for example, where one wants to close the filter with an upward going envelop such as during the attack of a note.

Typical Use

A **Vlowpass2** can be used as a filter to reduce the high frequencies in a signal, as shown in the following example :

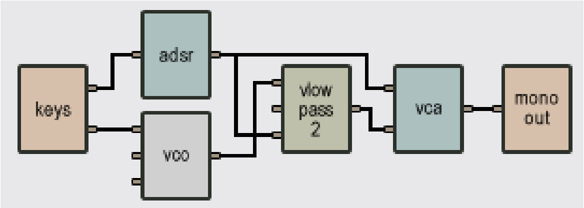


Figure 96: Use of **Vlowpass2** to Reduce High Frequencies.

Or it can be used as a sound source by turning the resonance knob up (clockwise). This gives the filter a more drawn out response (a longer transient), and generates a sound at the cutoff frequency, as shown in the following example :

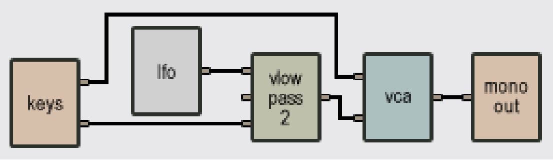
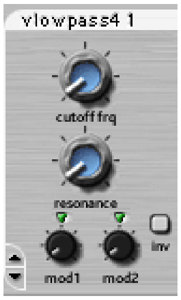


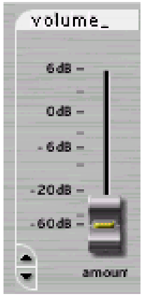
Figure 97: Use of **Vlowpass2** to generate Sound.

6.102 Vlowpass4



The **Vlowpass4** module is a voltage-controlled fourth-order low-pass filter (-24dB/octave). This module has three inputs and one output. The first input is the signal to be filtered, while the second and third inputs are modulation signals which are used to vary the cutoff frequency of the filter. The amplitude of the two modulation signals is adjusted with the two gain knobs *mod1* and *mod2* respectively. The output is the filtered signal. The behavior of this filter module is exactly the same as that of the **Vlowpass2** module except for the attenuation after the cutoff frequency which is steeper.

6.103 Volume



The **Volume** module is used to adjust the amplitude of a signal. It has one input and one output. The output signal is the input signal multiplied by a constant varying between 0 and 2 (+6dB).

Typical Use

The **Volume** module is used whenever the level of a signal must be adjusted. A **Volume** is usually placed just before an **Audio Out** module (see Figure 22.

The default values of the following parameter is set at construction

- *gain*: default value of the volume gain (value between 0 and 2).

6.104 Xor

The **XOR** module performs an XOR logic operation. The one output of this module is either 1 (true) or 0 (false) depending on the values sent to the two inputs. To deliver 1 at the output, To deliver 1 at the output, only one of the two inputs must receive a value of 1, otherwise the output will deliver a value of 0. This module has no front panel. The following diagram shows the output value depending on the values in the two inputs. Input signals are considered False (0) when smaller than 0.1 Volts and True (1) when greater than 0.1 Volts.

Input1	Input2	Output
1	1	0
1	0	1
0	1	1
0	0	0

Table 5: **Xor** module output as a function of its inputs.

7 Quick references to commands and shortcuts

File Menu

Command	PC	Mac OS	Description
New	Ctrl+N	Apple+N	New patch
New Folder		Apple+Shift+N	New Folder in the Browser
Open Instrument	Ctrl+O	Apple+O	Open the selected patch
Close		Apple+W	Close the window and exit the application
Close Current Patch	Ctrl+W	Apple+Shift+W	Close the current patch
Save Instrument	Ctrl+S	Apple+S	Save the current patch
Save Instrument As			Save the current patch under a new name
Save Preset		Apple+Option+S	Save the current preset
Save Preset As	Ctrl+Shift+S		Save the current preset under a new name
Save Midi Links			Save the current MIDI links
Save Midi Links As			Save the current MIDI links under a new name
Save Performance			Save the current performance
Save Performance As			Save the current performance under a new name
Import			Import a .txf file
Export			Export a .txf file
Exit (Quit on Mac)			Quit the application

Edit Menu

Command	PC	Mac OS	Description
Undo	Ctrl+Z	Apple+Z	Undo last command
Redo	Ctrl+Y	Apple+Shift+Z	Redo last command
Cut	Ctrl+X	Apple+X	Cut selected item
Copy	Ctrl+C	Apple+C	Copy selected item
Paste	Ctrl+V	Apple+V	Paste
Delete	Del		Delete selected item
Select All	Ctrl+A	Apple+A	Select all items
Duplicate	Ctrl+D	Apple+D	Duplicate selected item
Info	Ctrl-I	Apple+I	Edit information about a selected item (browser)
Module Settings	Alt-Enter	Apple-J	Edit module settings (builder)
Learn MidiLink			Enter MIDI link learn mode (player)
Forget MidiLink			Drop a MIDI link (player)
Set MidiLink Minimun Value			Limit the value of a MIDI link to a minimum value
Set MidiLink Minimun Value			Limit the value of a MIDI link to a maximum value
Edit Midilinks			Display the Edit MIDI links window to view or edit MIDI links of the current patch
Edit Program Change			Associate performances with MIDI program changes

Edit / Preferences Menu

Command	PC	Mac OS	Description
General			Display the Edit General Preferences window
Audio Settings			Display the Audio Settings window
Midi Settings			Display the MIDI Settings window
Latency Settings			Display the Latency Settings window

Arrange Menu

Command	PC	Mac OS	Description
Align Left Edges	Alt-Left		Align the left edges of the selected modules (builder)
Center Horizontally	Shift F9		Align horizontally the selected modules (builder)
Align Right Edges	Alt-Right		Align the right edges of the selected modules (builder)
Align Top Edges	Alt-Up		Align the top edges of the selected modules (builder)
Center Vertically	F9		Align vertically the selected modules (builder)
Align Bottom Edges	Alt-Down		Align bottom edges of the selected modules (builder)
Simplify Wire			Simplify the selected wire (builder)

View Menu

Command	PC	Mac OS	Description
Show Player/Builder	Ctrl-T	Apple-T	Toggle between the builder and player views
Show/Hide Browser		Apple-B	Show/Hide the browser panel
Show/Hide Help			Show/Hide the help panel (builder)
Locate	Ctrl-L	Apple+	Select and make visible in the browser the current instrument or the module currently selected in the builder
Previous Patch	Ctrl-Shift-Tab	Apple+=	Walk forward in the list of opened instruments
Next Patch	Ctrl-Tab	Apple+-	Walk backward in the list of opened instruments

Help Menu

Command	PC	Mac OS	Description
Manual	F1		Display the user manual
About Tassman			Display the About Tassman window

8 License Agreement

IMPORTANT! CAREFULLY READ ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT BEFORE OPENING THIS PACKAGE. OPENING THIS PACKAGE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS AGREEMENT, PROMPTLY RETURN THE UNOPENED PACKAGE AND ALL COMPONENTS THERETO TO THE PARTY FROM WHOM IT WAS ACQUIRED, FOR A FULL REFUND OF ANY CONSIDERATION PAID.

This software program, any printed materials, any on-line or electronic documentation, and any and all copies of such software program and materials (the "**Software**") are the copyrighted work of Applied Acoustics Systems DVM Inc. ("**AAS**"), its subsidiaries, licensors and/or its suppliers.

1. LICENSE TO USE. The Licensee is granted a personal, non-exclusive and non-transferable license to install and to use one copy of the Software on a single computer solely for the personal use of the Licensee. The Software contains a construction interface (the "**TassBuilder**") that allows Licensee to create, from custom patches, synthesizers or other materials ("**TassPlayer**") for Licensee's personal and/or commercial use in connection with the Software. Use of the TassBuilder and TassPlayer are subject to this Agreement.

2. RESTRICTIONS ON USE. The Licensee may not nor permit third parties to (i) make copies of any portion of the Software, other than as expressly permitted under this Agreement; (ii) modify, translate, disassemble, decompile, reverse engineer or create derivative and/or competitive products based on any portion of the Software; (iii) provide use of the Software in a network, timesharing, interactive cable television, multiple CPU service bureau or multiple user arrangement to users not individually licensed by AAS, other than as expressly permitted by the terms of this license; and, (iv) to use the TassBuilder for commercial purposes including but not limited to, distribution of TassPlayer on a stand alone basis or packaged with other software or hardware through any and all distribution channels, without the express written consent of AAS. The Software is licensed to you as a single product. Its component parts may not be separated for use on more than one computer. Patches resulting from the TassBuilder and presets for the Tassplayer may be used on other computers.

3. OWNERSHIP. AAS retains title to the Software, including but not limited to any titles, computer code, themes, objects dialog concepts, artwork, animations, sounds, audio effects, methods of operation, moral rights, any related documentation and "applets" incorporated into the Software. AAS retains ownership of and title to all intellectual property rights in the Software, underlying technology, related written materials, logos, names and other support materials furnished either with the Software or as a result of this Agreement, including but not limited to trade secrets, patents, trademarks and copyrights therein. Licensee shall not remove or alter any copyright or other proprietary rights notices contained on or within the Software and shall reproduce such notices on all copies thereof permitted under this Agreement or associated documentation.

4. LIMITED WARRANTY. Except for the foregoing, THE SOFTWARE IS provided “AS IS” without warranty or condition of any kind. AAS disclaims all warranties or conditions, written or oral, statutory, express or implied, including but not limited to the implied warranties of merchantable quality or fitness for a particular purpose, title and non-infringement of rights of any other person. AAS does not warrant that THE SOFTWARE will meet the Licensee’s requirements or that the operation of the software will be uninterrupted or ERROR-FREE.

5. LIMITATION OF LIABILITY. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL AAS BE LIABLE TO THE LICENSEE OR ANY THIRD PARTY FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR EXEMPLARY DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO LOSS OF REVENUE OR PROFIT, LOST OR DAMAGED DATA, BUSINESS INTERRUPTION OR ANY OTHER PECUNIARY LOSS WHETHER BASED IN CONTRACT, TORT OR OTHER CAUSE OF ACTION, EVEN IF AAS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, EXCEPT IN RELATION TO GROSS NEGLIGENCE OR WILFUL BREACH OF THIS AGREEMENT BY AAS. NO AAS AGENT, REPRESENTATIVE OR DEALER IS AUTHORIZED TO EXTEND, MODIFY OR ADD TO THIS WARRANTY ON BEHALF OF AAS. THE TOTAL LIABILITY OF AAS FOR DAMAGES, WHETHER IN CONTRACT OR TORT, UNDER OR RELATED IN ANY WAY TO THIS AGREEMENT SHALL BE LIMITED TO THE LICENSE FEES ACTUALLY PAID BY LICENSEE TO AAS, OR IF NO FEES WERE PAID, AAS’ LIST PRICE FOR THE SOFTWARE COVERED BY THIS LICENSE. THE EXCLUSION OF IMPLIED WARRANTIES AND/OR THE LIMITATION OF LIABILITY IS NOT PERMITTED IN SOME JURISDICTIONS, AND SOME OR ALL OF THESE EXCLUSIONS MAY THEREFORE NOT APPLY.

6. TERMINATION. This License also shall extend to the Software and any updates or new releases thereof obtained by the Licensee, if any, subject to any changes to this License made by AAS from time to time and provided to the Licensee, provided AAS is under a separate obligation to provide to Licensee such updates or upgrades and Licensee continues to have a valid license which is in effect at the time of receipt of each such update or new release. This License shall remain in effect until terminated. The Licensee may terminate this Agreement at any time, upon notification to AAS. This Agreement will terminate immediately without notice from AAS if the Licensee fails to comply with any provision of this License. Any such termination by AAS shall be in addition to and without prejudice to such rights and remedies as may be available, including injunction and other equitable remedies. Upon receipt of notice of termination from AAS, the Licensee must (a) immediately cease to use the Software; (b) destroy all copies of the Software, as well as copies of all documentation, specifications and magnetic media relating thereto in Licensee’s possession or control; and (c) return all original versions of the Software and associated documentation. The provisions of Sections 1, 3, and 5 shall survive the termination of this Agreement.

7. GOVERNING LAW. This Agreement shall be governed by and construed in accordance with the laws of the Province of Quebec, without regard to the United Nations Convention On

Contracts for the International Sale of Goods and conflict of laws provisions, if applicable, and the parties hereby irrevocably attorn to the jurisdiction of the courts of that province. Les parties sont d'accord à ce que cette convention soit rédigée en langue anglaise. The parties have agreed that this agreement be drafted in the English language.

8. SEVERABILITY. If any of the above provisions are held to be illegal, invalid or unenforceable, such provision shall be severed from this Agreement and this Agreement shall not be rendered inoperative but the remaining provisions shall continue in full force and effect.

9. ENTIRE AGREEMENT. This Agreement is the entire agreement between AAS and the Licensee relating to the Software and: (i) supersedes all prior or contemporaneous oral or written communications, proposals and representations with respect to its subject matter; and (ii) prevails over any conflicting or additional terms of any quote, order, acknowledgement, or similar communication between the parties during the term of this Agreement except as otherwise expressly agreed by the parties. No modification to the Agreement will be binding, unless in writing and signed by a duly authorized representative of each party.

10. NON-WAIVER. No delay or failure to take any action or exercise any rights under this Agreement shall constitute a waiver or consent unless expressly waived or consented to in writing by a duly authorized representative of AAS. A waiver of any event does not apply to any other event, even if in relation to the same subject-matter.

Index