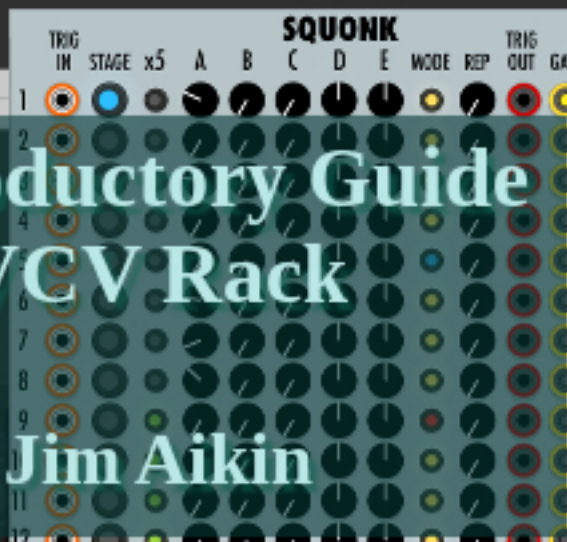


How to Rack



An Introductory Guide to VCV Rack

by Jim Aikin



© 2019 by Jim Aikin. All rights reserved. This book may be freely copied and distributed electronically, but may not be adapted, may not be printed other than as a single copy for personal use, and may not be sold for money.

This is version 1.0 of *How to Rack*. For updated releases and links to supplementary files, consult <https://synthage.wordpress.com/2019/07/09/how-to-rack-the-book/>.

also by Jim Aikin

nonfiction:

Power Tools for Synthesizer Programming, 2nd Edition
Csound Power!
Chords & Harmony

fiction:

The Leafstone Shield
The Rainbow Tree
The HeartSong Fountain
The Firepearl Chalice

1: Introduction

I love modular synthesis. With a decent modular instrument, the possibilities for sound design and musical expression are beyond vast. On the other hand, a modular instrument is not as easy to learn, nor as easy to use effectively, as a pre-configured synthesizer.

This morning (July 11, 2019) the module count in my VCV Rack 1.1 installation stands at 1,141. That tally includes some blank panels, but more than 1,000 of the modules actually do something. There's also quite a bit of redundancy, and some of my modules are paid rather than free, but still, that's a lot of modules. Someone who is new to modular synthesis can surely be forgiven for being both confused and intimidated by such a profusion of possibilities. "How can I possibly figure this stuff out?!"

A plaintive question posted on a VCV Rack Facebook group illustrates this dilemma. In order to avoid embarrassment, I'm going to omit the name of the individual who posted it; by the time this book appears I'm sure he'll be a lot further down the road.

Hi, mates. I just had a first try with VCV. It's nice but here is my first problem.

I have plugged an ADSR and a VCO to a VCA but the envelope is doing nothing. I am sending MIDI notes to a MIDI interface inside VCV. Does anyone know why the ADSR won't function, please?

If you've been using modular synthesizers for a while, you may be tempted to chuckle — but remember, there was a time when you didn't know this stuff either. This question (and others) will be answered in this book.

There are, to be sure, some very good video tutorials on YouTube (a huge tip of the hat to Omri Cohen!) — and many musicians prefer video instruction to print. But video is not free of limitations. Most videos cover only a single topic; few of them provide the kind of background or overview that a newcomer may need. And if you're searching for the answer to one particular question, finding it in a 15-minute video is likely to be harder than searching through a print document.

I'm now retired, but my career (at the late, lamented *Keyboard* magazine and subsequently as a freelance writer and editor) has been devoted to helping musicians learn about technology. Possibly I'm the right person to fill in the knowledge gap by writing a reasonably thorough introduction to VCV Rack — a document that will help newcomers (and maybe a few old hands who are a bit rusty) get up and running with a minimum of head-scratching and hair-pulling.

This book is not intended to serve as a comprehensive owner's manual covering every detail of VCV Rack. For one thing, such a document would be thousands of pages long. I'm not quite that dedicated, and I'm sure most people, no matter how desperate their search for knowledge, would never read it. In addition, VCV is a moving target. By the time I finish writing the first edition of this book, it will be time to rewrite it! I'm thinking of expanding the book in a future edition, in order to cover a number of concepts in sound synthesis, but that may or may not happen.

For now, we'll start with an overview of the Rack system and an explanation of a few key concepts. I've also provided a few basic tutorial patches, which you can assemble step by step in order to see what's going on. The largest chapter of the book is divided into sections covering the various types of modules. I'll explain their typical usages, and also describe briefly some of the stronger modules in each category. It seems to me that newcomers to VCV will probably benefit from these informal recommendations. Which oscillators should I try? Which filters?

The modules covered in this book are my own subjective preferences and those recommended by people in the VCV user community. You might feel I should have included others — and quite possibly you would be right! My goal is to help newcomers select modules that will give them a powerful palette of musical possibilities with a minimum of confusion. Once you've tried out my suggestions, you'll have some hands-on knowledge that will enable you to explore further on your own.

To finish the rather cursory exposition of VCV Rack, I've added a chapter that contains a few patching ideas.

Thanks to Andrew Belt for creating and maintaining VCV Rack, and also to the dozens of developers who have put time and energy into creating sets of modules.

For the foreseeable future, copies of this book will be downloadable from a link on my music blog at <https://synthage.wordpress.com/2019/07/09/how-to-rack-the-book/>. As time goes on, that page may also have links to tutorial patches, to videos demonstrating some of the concepts discussed in this book, and so forth.

2: The Big Picture

VCV Rack is a software-based modular synthesizer. We may as well start by looking at what that means, though it will already be obvious to most readers. To make music — or, if you prefer, sounds that aren't musical at all — with a modular synthesizer, you choose (perhaps from a browser) the modules that you'll use. You then connect the modules to one another in whatever manner seems appropriate. Making connections is called patching, and in many software synthesizer systems it's done using what appear to be cables (patch cords). Signals flow from one module to another along the connections you've set up. Eventually, some of the signals are routed to the audio output, and you listen to them.

To understand how this works, you need to know what patch cords are. I'm sure you're eager to start working with VCV Rack, so please feel free to jump ahead to later parts of this book. You can return to this section later if you need or want to. But some readers may already feel a bit at sea. For their benefit, let's take a closer look.

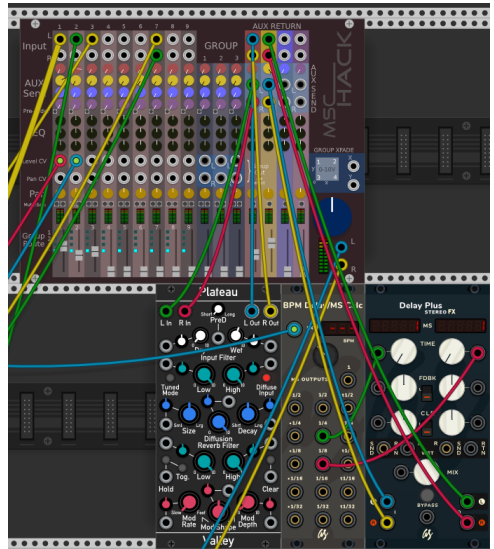


Figure 1. Patch cords in VCV Rack. The jacks, most of which are unconnected in the image above, resemble hardware jacks. To connect an output to an input, click and hold on an empty output and drag the mouse to the desired input, or vice-versa.

Until 20 years ago, all synthesizers were hardware. Computers simply weren't fast enough to generate audio in real time. (This generalization, while true, is a bit oversimplified. The software

synthesis system called Csound debuted in 1985, and its antecedents date back to the late 1950s. However, these systems were not real-time musical instruments. In order to create sound, you had to write code and then wait, perhaps an hour or more, for the computer to compile it and produce an audio file that you could listen to.)

The history of music production using electricity goes back to 1899, and we're not going to get into the details here. For our purposes, the first true synthesizers were built in the 1960s by Robert Moog, Don Buchla, and a few other visionaries. The instruments consisted of banks of hardware modules, which were connected to one another using physical patch cords.

Modular synthesis was largely eclipsed in the 1980s, when commercial instruments with keyboards and pre-configured sets of components became available. Modular synthesis was widely felt to be old-fashioned, the sort of thing that would interest only university music researchers wearing white lab coats. In 1995, however, German hardware designer Dieter Doepfer produced his first modular synthesizer. Other manufacturers soon began producing modules that were compatible with Doepfer's hardware. This was the beginning of what is today called Eurorack, a vigorous revival of hardware-based modular synthesis.

Like earlier modular synthesizers, Eurorack modules are connected to one another using patch cords. Every module has either input jacks or output jacks (and usually some of both types). A cord is patched from an output on one module to an input on another, and the cord then carries one signal from the output to the input.

That is the single most essential point to understand about modular synthesis: A patch cord carries a signal from an output to an input. Generally speaking, if there is no cord connecting an output to an input, the two modules are entirely unaware of and independent of one another. There are some important exceptions to this rule, as we'll see, but linking one module to the next with a patch cord is the default, and it's what you'll be doing a lot as you work with VCV Rack.

VCV Rack is rather closely modeled on Eurorack hardware. In fact, a few VCV modules are direct clones of hardware Eurorack modules. Others, while not clones, have clearly been inspired by Eurorack designs. In a broader sense, VCV borrows the easy-to-understand user interface of Eurorack. The on-screen “rack rails” are part of this interface, but they’re mostly a visual convenience. What’s essential is the process of connecting modules using droopy colored lines that look like physical patch cords.

VCV Rack is not the only software that specializes in modular synthesis. The free systems include Csound, Pd, and Supercollider. Paid systems include Cycling ’74 Max, Native Instruments Reaktor, Softube Modular, and Propellerhead Reason. Each has its own strengths and limitations. Pd, Max, Reaktor, and Reason all use patch cords for connecting modules; Csound and Supercollider are text-based. To me, one of the big advantages of VCV Rack over these competing systems is that its patch cords (with an important caveat that we’ll get to below) carry only one type of signal. Pd is free, and Pd connections are made using graphic patch cords, but connections in Pd can carry many different types of messages. Patching in VCV Rack is a much more straightforward and intuitive process.

As a footnote, one of the extremely cool things about a software modular synth is that you can use as many copies of a module as you need. In a hardware modular, if you want to add an extra oscillator you’ll have to buy one and fit it into the case with the other modules. In software, you can use several copies of any module without spending more money.

Voltages. The signals in hardware modular patching are called voltages. I’m not an electrical engineer, so I’m not quite sure what a voltage is. All I know is that it’s a signal whose strength can vary. The typical range of voltages in a hardware modular is from -10 to +10 volts. The voltages in a hardware modular are analog rather than digital. Without going into detail about what this means, we can say that a digital signal consists of a stream of numbers. An analog signal can vary in a smooth, continuous manner, and there are no

numbers anywhere in it. When you adjust the amount of water coming from the tap in your kitchen sink, you’re making an analog adjustment. The amount of water can be more or less, but it’s not in the form of numbers.

Signals in VCV Rack pretend to be analog voltages. In reality, they aren’t. In reality, they’re streams of numbers, as Figure 2 shows. But the numerical resolution of the system is fine-grained enough that for practical purposes we can view them as if they were analog signals — and indeed, most of the discussion of modules in the world of VCV calls the signals voltages, even though they aren’t. I’ll conform to this usage throughout this book. (For more on voltage signals in VCV Rack, see the chapter “Patch Concepts.”)

In any audio synthesis system, we would like to distinguish between two types of signals — audio signals and control signals. Audio signals are the ones that we’re eventually (perhaps after some processing) going to listen to. Control signals are not listened to directly; instead, they control how the various modules operate.

In some synthesizers, there is a hard and fast distinction between the two signal types. In a system based on voltages, however, there is really only one type of signal: the voltage. Any signal can be either listened to or used to control something. (This is not true of Buchla hardware instruments, by the way: Buchla made a hard distinction between control signals and audio signals by using two different types of patch cords.) We won’t usually want to listen to a control signal; it isn’t likely to produce anything but clicks and pops; but in principle the two signal types are interchangeable in VCV Rack.

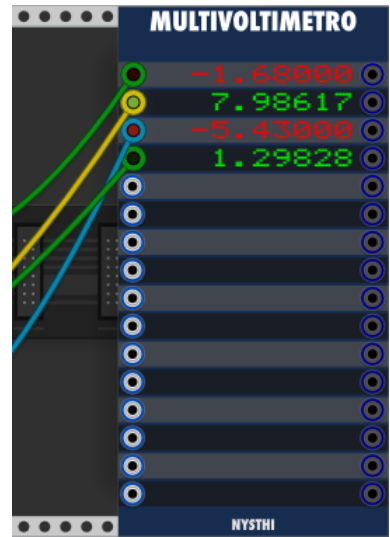


Figure 2. The Nysthi Multivoltmetro module displays VCV “voltages,” which are actually numbers, not true voltages. The values will normally range from -10 to +10; in this module, negative values are displayed in red.

Downloading & Installing. VCV Rack is available for Windows, MacOS, and Linux computers. I don't have access to Mac or Linux machines, so I'm not going to go into the process of downloading and installing. There's a brief description of the process at vcvrack.com/manual/. If you have problems, post a message on the Facebook VCV Rack group or on the VCV Community Forum (<https://community.vcvrack.com/>), and someone will surely be able to help solve the problem.

Installation has been a problem for some people who had previously installed beta version 0.6.x. It's possible to make a copy of your 0.6 installation and keep it somewhere rather than just overwriting it with the 1.x version. A few very cool modules have not (or not yet) made the leap to 1.x compatibility; if you're using any of these modules to make music that's important to you, keeping the beta version around would be a good idea.

Suggested Hardware. To use VCV Rack effectively, you'll need a reasonably fast computer. A decently large screen would be a plus too, though you can certainly run it on a laptop. To the best of my knowledge, there are no plans to port VCV to tablet or phone operating systems.

You'll also benefit from a decent pair of studio-quality monitor speakers, and of course a soundcard or audio interface to send the output to the speakers. If you're not planning to play live, an interface that has a low-latency ASIO driver won't be strictly necessary, but ASIO has become standard. (As a side note, software that captures on-screen video may not be compatible with the ASIO audio driver. You may need to use a higher-latency driver, such as DirectSound in Windows, in order to capture both audio and video at the same time.)

A MIDI hardware controller is, again, not strictly necessary. You can make tons of music directly in VCV without touching an external hardware controller, and your computer's QWERTY keyboard can be used for real-time input if you don't have a MIDI device. Still,

you'll find a 5-octave MIDI keyboard with a bank of sliders that can send MIDI CC messages very useful.

When you add it all up, even though VCV itself is (mostly) free, you can easily spend a couple of thousand bucks on a computer music system. If you already have such a system, though, you'll be ready to go with no added expense.

The User Interface. The VCV user interface is pretty easy to navigate. Here's a disorganized list of things you may want to know:

- To view the Module Browser, right-click in an empty area.
- When you hover the mouse over the image of a module in the Browser, a tooltip text will tell you what it is.
- The gray bar in the upper left corner of the Browser is a search field. You can type in it.
- If you click and hold the image of a module in the Browser, you can drag it to wherever you want in the rack before releasing the mouse button.
- All modules have right-click menus. A few have different menus depending on where on the panel you click.
- You can restore any knob to its default value by double-clicking on it.
- If you right-click a knob, you'll see its current numeric value displayed (see Figure 3). You can type a new number.
- If you hold down Ctrl while dragging a module left or right, it will push other modules along with it.
- You can delete a module by hovering the mouse pointer over it and hitting the Delete or Backspace key.
- You can duplicate a module by hovering the mouse over it



Figure 3. Right-clicking the Bend knob in Vult Trummor 2.

and typing Ctrl-D.

- Only one cable can be connected to an input jack, but many cables can be attached to a single output jack.
- If there's already a cable on an output jack, you can add another one by Ctrl-clicking the jack.
- Thick cables mean the signal is polyphonic; thin cables mean it's monophonic.
- Some modules' outputs can be either mono or poly, depending on what type of input they have.
- Holding Ctrl while scrolling the mouse wheel will increase or decrease the zoom level.
- Patch cords can be made more or less transparent using the Cable Opacity slider in the View menu.

DSP Usage. With any computer program that's designed to do things (usually video or audio) in real time, it's possible to overload the computer's CPU (its main processing chip) by asking it to do too many tasks at once. When this happens, the output will stutter. With audio, you'll hear clicks and pops. In extreme cases the computer may lock up.

Your audio hardware will have a setting for its buffer size. Increasing this setting may help. In my system, for example, the ASIO driver happens to run with a 256 sample buffer. If I need to, I can increase it to 512 or even a higher number. Depending on your audio hardware's driver, you may be able to do this within VCV by clicking on the lower right data field in the Audio-8 module and changing the block size. Or you may need to open the driver's control panel to make the change.

When your patch grows so large that VCV itself is using too much CPU, changing this setting may not help. To check VCV's performance, go to the Engine menu and select CPU Timer. This will cause each module to display, in the lower left corner, approximately how many microseconds are needed for that module to compute an

audio sample. (The word “sample” has several different meanings. Here we’re talking about the time needed to compute one sample “word” — the smallest possible grain of audio.)

Each module will also display a little red meter that shows its CPU consumption, as shown in Figure 4. However, the data display and meter on the Audio-8 output module operate in reverse: They show how much CPU remains available. If you don’t know this, switching on the CPU meter may cause you needless panic when it appears that a simple patch is using up 80% of the CPU.

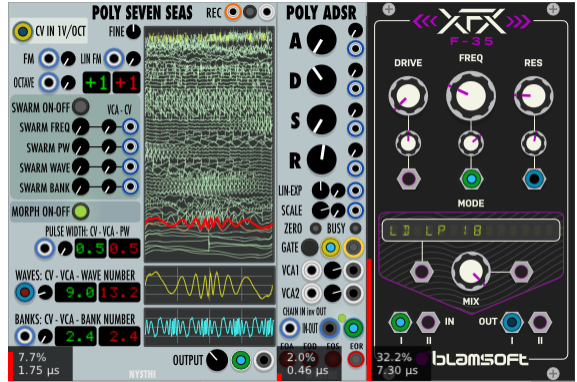


Figure 4. When the CPU Timer is active, each module displays the amount of processor time it’s using, both numerically and with a red bar graph. Here, we can see that the Blamsoft filter is occupying a lot of CPU time.

Running the CPU metering takes some CPU time. When you’re not using this feature to check for overload, it’s better to keep it switched off.

Suggested Module Downloads. If you’re feeling brave or just curious, you can download all of the free plug-ins in the Library on the VCV Rack website. Some are brilliant, some are goofy, and some are both. Some are trivial and can safely be ignored. If you’re new to modular synthesis, you may want to start with a smaller subset of modules rather than downloading everything, so as to get your feet firmly on the ground before you wander off into the jungle.

If you just install VCV Rack without any third-party add-ons, you’ll get a good starter set of modules and enjoy quite a lot of functionality, but if you limit yourself to the installed modules you’ll be missing out on 90% of the fun. To start your VCV experience, I recommend

the module sets from the developers listed below. I don't promise to limit myself to just these modules in the discussion of specific module types, but these will give you a solid foundation.

To add them to your VCV installation, go to the Library page on vcvrack.com. Find these developers in the left column. Then go to the right column. Click the buttons that say +Free so that they change to -Free. Then launch VCV Rack. At this point you'll see a red dot beside the Library item in the top menu. Click the Library menu header and select Update All. When the update is finished, quit and restart VCV Rack.

AS has more than 70 modules, and you certainly won't need all of them, but I find a couple of the utilities very useful.

Alikins is a smaller set, and strictly utilities, but there are a couple of useful ones.

Audible Instruments is a collection based on the hardware Eurorack modules by Mutable Instruments. These are complex modules, but they're brilliant. No rack should be without them.

Befaco offers only a small set of modules, but the Rampage dual slew device is extremely useful.

Bogaudio is essential, if only for the FM-OP (frequency modulation operator) and the sample-and-hold. Some of the others are also worth having around.

The utilities from **HetrickCV** are worth having.

Impromptu Modular has some powerful sequencers, and a lot of people use their Clocked module as a clock source.

ML Modules has a superior pitch quantizer, along with some other useful widgets.

Nysthi has more than a hundred modules, some of them highly esoteric and others useful or even essential. You'll want this set.

Valley has an excellent reverb (called Plateau), and Valley Topograph is useful for percussion sequencing in pop music styles.

The **Vult** modules are available either free or as a paid set. The paid set has a few more modules, but you can start with the free ones. Vult has good filters; also a great oscillator and a great percussion sound generator.

3: Patch Concepts

In this chapter we'll build some basic patches. This discussion will be useful mainly to people who are new to the concept of modular synthesis. Along the way, I'll touch on a few general principles that may also be enlightening to those who are a bit further down the road.

One of the most important concepts in modular synthesis is this: Most modules do nothing unless you tell them to. They just sit there. There are several important exceptions to this rule, but it's the default behavior. If you're approaching VCV Rack from a background in keyboard synthesizers, the concept may not be at all obvious. On a keyboard instrument, when you press a key you hear a sound. It's easy! But behind the front panel, the manufacturer's engineering team has made a lot of internal signal connections for you. In a modular synth, you have to make those connections yourself. That's the challenge, and that's the fun of it.

Since I mentioned important exceptions to this rule, I may as well list them.

1. Oscillators (and LFOs, which are a type of oscillator) are constantly running. They will produce an output signal at their output even when none of their inputs are connected.
2. A few modules have their own expander modules. In a hardware Eurorack system, an expander will be connected to its parent by a cable that's tucked away inside the case. In VCV, an expander automatically connects to its parent when it's positioned directly to the right (or, in one case, directly to the left) of the parent, touching it. Signals pass back and forth between the expander and its parent even though no patch cords are employed.

3. Within a module, some internal connections are “normalled.” For instance, there may be three or more input jacks for various sections of the instrument, all of them designed so as to respond to the same type of signal. If a signal is patched to the first jack in the series, it will be magically passed on to the other jacks unless something is patched into a later jack in the series. Connecting a patch cord to a later jack breaks the normal internal connection.

4. Using the CV-Map module from stoermelder, as shown at right, control signals can be invisibly patched to almost any panel control on any module, even if that module has no associated jack for accepting control signals.

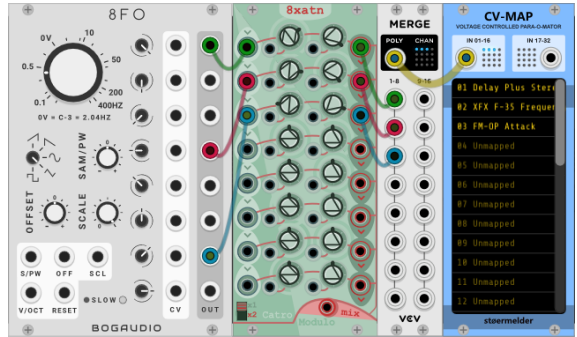


Figure 5. Outputs from the Bogaudio 8FO are shaped by the catronomix 8xatn attenuverter and then passed on to VCV Merge. The Merge module combines the mono signals into a poly signal, which is passed on to stoermelder CV-Map. CV-Map sends the signals on to the parameters of other modules without using patch cords.

Modulation. Modular synth music is at its best when the sound changes, subtly, or radically, from one note to the next. Changes in the sound are made by means of modulation. Most modules have one or more parameters that can be altered. Of course you can change the setting of a parameter by moving the knob or slider manually (with the mouse) while the music plays, but it’s usually more efficient to make the changes using signals. If you use signals (also called control voltages, or CVs), the changes in the sound will be repeatable. Also, you can sit back and listen while the sounds morph or mutate.

CV signals in VCV Rack have a theoretical range from -10 to +10 “volts.” However, some modulation sources may output a range from -5 to +5, from 0 to +5, or from 0 to +10. If you’re not sure

what values are being transmitted by a modulation source, you can use the Nysthi Multivoltmetro module to check the values. If you need to display polyphonic values in this module, attach a poly cable to input 1. The numerical value of a signal can change rapidly, however, so it's often the case that the VCV Scope module will give you better information than Multivoltmetro; Scope shows the shape of the signal over time.

Most oscillators have a V/oct input for controlling their frequency. Technically, frequency is not quite the same thing as musical pitch, but for many purposes the terms are interchangeable. This input jack maps a voltage range from -5 to +5 volts to the pitch of the oscillator in a standard way: Increasing the input by one volt raises the pitch by one octave. To raise the pitch by a semitone, simply divide 1 volt by 12 (the result is 0.08333) and add this amount to the signal being sent to the V/oct input. (In practice, there are modules that will do the math for you.)

Some modules have a manual knob for setting the basic value of a parameter, a CV input jack for the parameter, and also a smaller knob for adjusting the amount of CV signal that is applied. This is the ideal system, because it puts both of the important settings (the basic value and the amount of CV) at your fingertips.

Some modules have a parameter knob and a CV input jack, but no smaller knob for adjusting the amount of CV. In this case, you'll probably want to adjust the amount of CV using a second, external module before it is sent to the input jack. The VCV 8Vert module is useful for this, but I prefer the AS AtNuVrTr. An AtNuVrTr (attenuverter) module has only two sections as opposed to the eight in the 8Vert, but it has an offset knob in each section, which the 8Vert lacks, and also allows you to control both the offset and the attenuation amount from other CVs.

If the module doesn't have an input jack for changing a parameter with a CV signal, all is not lost. Just about any parameter can be mapped to respond to a CV using the stoermelder CV-Map module,

as shown in Figure 5. This module adds cordless modulation to VCV Rack. To use it, you'll also need the VCV Merge module and possibly an attenuator too. The VCV MIDI-Map module does the same thing with external MIDI CC signals, mapping them cordlessly to VCV parameters.

To use CV-Map, patch the CV signal you want to use (such as perhaps the output of an LFO) into an input in VCV 8Vert, AS AtNuVrTr, or catronomix 8xatn. Patch the corresponding output into an input on Merge, and then patch the poly output of the Merge into CV-Map. Click on one of the slots in CV-Map (they will start out by reading “#1 Unmapped” and so on) and then wiggle the knob in any module that you want the CV signal to be applied to. A red square will appear beside the knob, and it will start animating in response to the signal CV-Map is receiving.

Polyphony. Polyphony is the ability to play more than one note at a time. Polyphonic software synths that use patch cords are not unheard-of: the Bazille VST synth from u-he is a good example. But polyphony in a fully modular environment is rather unusual.

The set-up for polyphony in VCV Rack makes sense, but you may occasionally trip over the details. Some modules are fully ready for polyphonic operation; some are strictly monophonic; and some may be either mono or poly depending on how they're patched.

The MIDI-CV module allows you to set the maximum number of poly voices you want to use. When it's patched to other modules further down the line, if they're capable of poly operation they will sense the number of incoming voices and deploy the proper number of their own voices. Other modules that generate signals don't have a polyphony setting; they simply respond to as many voices as appear at their inputs. And bear in mind, developers are still (as of July 2019) ramping up to handle the demands of poly operation. Some modules may not yet be entirely free of bugs.

Polyphonic patch cables are visibly thicker than mono cables, so

you can always tell what's what. When you're using a poly patch, you can split the signal on a poly cable into separate mono signals using the **VCV Split** module. (The Nysthi mixer modules can do this too, with separate control over the level and pan position of the voices.) Mono signals can be combined into a poly signal using the **VCV Merge** module. And the separate voices in a poly signal can be mixed into a single mono signal using the **VCV Sum** module.

If you send a poly signal to a mono module (such as perhaps an effect processor), the most likely result is that you'll hear only the first voice within the cable. The mono module will ignore the other voices. Combining the voices with the Sum module will solve this problem.

Musical Considerations. Let's be honest: VCV Rack 1.x is not the best choice in software if your goal is to produce pop songs. That's because there's no timeline, and thus no convenient way to set up your intro, verse, chorus, and so on. You can probably make a pop song, if you're stubborn and patient — but don't say you weren't warned. You'll be much happier working with VCV if you're producing beats, loops, and textures and exporting the audio into a conventional multitrack DAW, or if you're doing experimental music of some sort.

VCV will happily produce endless streams of slowly changing post-minimalist texture — sustained drones, a repeating chord progression, or tinkling sequencer patterns. VCV is also a fine source for steady electronic dance beats (which are, after all, a form of minimalism). Gradual and sudden changes in the sonority are easy to manage, either automated in a step sequencer or with an LFO, or through live interaction as the music plays.

Sections can begin and end. Audio channels can fade in and out. All sorts of animation can be added to the sound. Complex polyrhythms are quite easy to set up. And the palette of tone colors coming from oscillators and filters is endless. You could easily spend years exploring the possibilities. But produce a pop song? Maybe in the 2.0 release; we'll have to wait and see.

A Starter Patch

VCV Rack comes with a template patch, which will be loaded when you select New from the File menu. This is stored as <Rack user directory/template-v1.vcv>. You can replace this with your own preferred template if you like. To revert to the original template if you've replaced it, simply delete template-v1.vcv, launch Rack, and choose New from the File menu. This will recreate the patch. You can then use the Save Template command in the File menu.



Figure 6. The VCV Rack template patch. In this layout, signals happen to flow from left to right. This is not a requirement for patching, but it's often convenient. I've clicked on the middle field in the Audio-8 module to choose my audio hardware's ASIO driver.

If you're new to modular synthesis, spend some time looking at the template patch; it will help you get started. All of the connections are already made for you; the only thing you have to do is choose your audio output device in the Audio-8 module. Once you've done this, pressing the keys along the bottom row of your QWERTY keyboard (z, x, c, v, and so on) will play a major scale.

You can easily experiment with this patch, to learn more about how a modular synthesizer works. Here are some things to try:

- Wiggle the large Freq knob in VCO-1. The frequency of the oscillator's tone will move up and down.
- Wiggle the Freq knob in the VCF. This will increase or decrease the strength of the higher overtones in the sound.

- The knobs in the ADSR will change the shape of your notes.
- The Freq CV knob in the VCF will adjust the amount of the ADSR envelope that is being used to control the filter's frequency. This knob is set to zero when it's centered, so moving it further to the left will *invert* the signal coming from the ADSR.
- Grab the patch cord connected to the Saw output of the VCO and move it to the Sqr output. A square wave has more of a hollow tone.
- While listening to the square wave, wiggle the P. Width (pulse width) knob in the VCO. This will make the sound thinner.
- With the VCF Freq somewhere in a center position, increase the Res (resonance) knob. This will give the tone a more peak-like character.
- If you have a MIDI keyboard, select it using the top and middle data fields in the MIDI-CV module. Now you can play this basic synth using MIDI.
- After switching to MIDI input, right-click the MIDI-CV module. Go to the Polyphony Channels item and select 4 from the sub-menu. (Notice that the patch cords are suddenly thick rather than thin.) Now you can play four-note chords.

Now we'll look at a few ways to expand this basic patch. For starters, we'll add a second ADSR and a second oscillator.

1. Right-click the ADSR module and choose Duplicate from the pop-up menu, or hover the mouse pointer over the ADSR and type Ctrl-D. This will create a second ADSR. (It will be in the rack row above the first one, so it may not be visible until you scroll the display up.)
2. Drag the mouse from the Gate input of the new ADSR back to the Gate output of the MIDI-CV module.

3. Disconnect the patch cord from the Freq input of the VCF, and attach the output of the new ADSR to this input. Now you have two envelopes, one for amplitude and one for the filter.
4. Next, we're going to add a second oscillator. Use Ctrl-D (or the command in the right-click menu) to duplicate both the VCO and the mixer.
5. Attach the V/oct output of the MIDI-CV module to the V/oct input of the new VCO. (You can do this by Ctrl-clicking on the output to create a new patch cord.)
6. Remove the connection between the first VCO and the filter. Instead, attach the Saw outputs of both VCOs to the inputs (In 1 and In 2) of the new Mixer.
7. Attach the new mixer's Mix out to the In jack of the VCF.
8. At this point, the VCF will be overloading slightly, producing a warmer but more distorted tone. Turn down the Mix knob in the new Mixer so that the line on the knob is pointing to the left.
9. Change the frequency of the second oscillator using its Freq or Fine knob. As the tones of the two oscillators mix, you'll hear

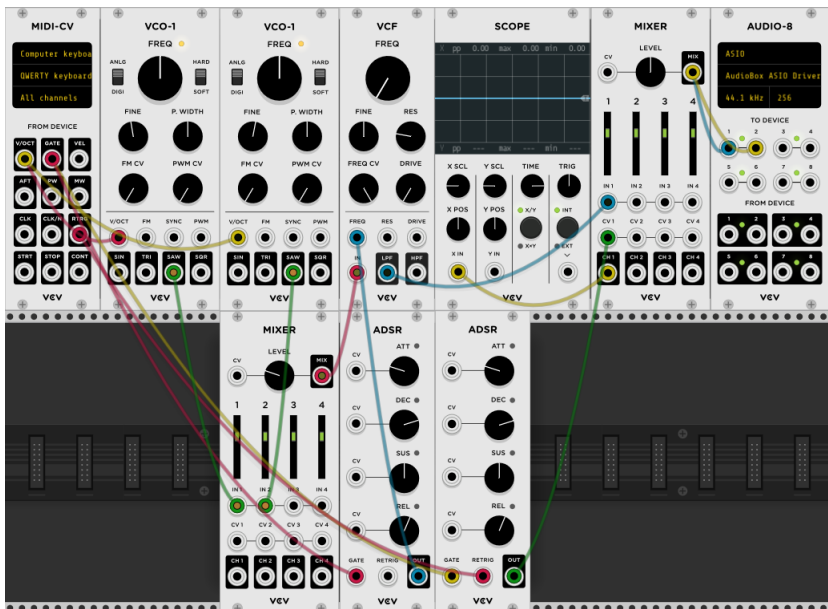


Figure 7. The template patch with two VCOs and two ADSR envelopes.

a sound with some richness, or possibly a tone that combines two pitches, perhaps a major third or a perfect fifth apart.

Next we're going to add an LFO, so as to create vibrato. To do this, you'll need to find the LFO module in the Browser.

1. Right-click in an empty area of the rack to open the Browser.
2. If you've already been poking around in the Browser, you may need to click on Reset Filters to return it to its default state.
3. Choose VCV in the upper list in the Browser. This will bring up a graphic of the included modules.
4. Click and hold on LFO-1. While holding the mouse button down, drag the module into the patch.
5. Attach the Tri (triangle wave) output of the LFO to the FM inputs of your two VCOs.
6. Turn up the FM CV inputs for the VCOs slightly, and use

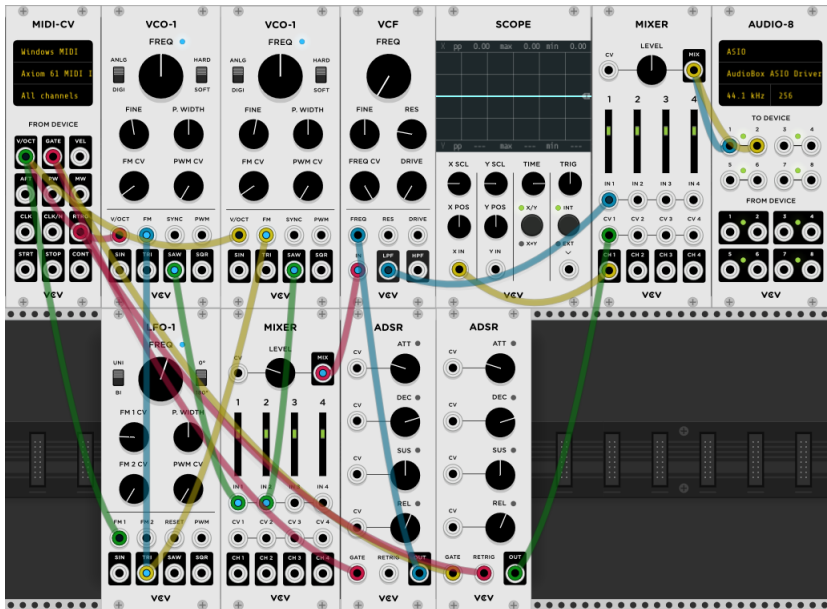


Figure 8. The third version of the template patch has a vibrato LFO and is polyphonic. (Note the thicker patch cords.)

the LFO's large knob to increase its frequency slightly. Now you have vibrato.

7. If you followed the bullet list of suggestions at the start of this section, you'll already have poly input from your MIDI keyboard. If you skipped that step, go back a couple of pages and follow the instructions there.
8. You'll notice that the patch cords running from the LFO to the oscillators are skinny. The LFO is currently monophonic, so all of the voices in a chord you play on your keyboard will have exactly the same vibrato. You may want to give each voice its own vibrato.
9. Attach the LFO's FM1 input to the V/oct output of the MIDI-CV module. Now it's polyphonic.
10. Raise the LFO's FM1 CV knob slightly. Now its frequency will track the keyboard, so higher notes will have faster vibrato.

And how about using the key velocity from the MIDI keyboard? The MIDI-CV module has a Vel (velocity) output. We might like to control the filter cutoff from velocity, so as to make some notes brighter than others. But the VCF cutoff frequency only has one CV input. In order to control this parameter from both the ADSR envelope generator and velocity, we'll have to get a little fancier.

1. Duplicate (Ctrl-D) one of the mixers.
2. If you've been following along, one of your ADSRs is patched to the Freq input of the VCF. Remove this connection, and patch the ADSR output instead to the new Mixer's In 1.
3. Drag the mouse from In 2 of the new mixer up to the Vel output of the MIDI-CV. Now the velocity data is routed to the new mixer.
4. Connect the new Mixer's Mix output back to the Freq input of the VCF.
5. You'll probably want to turn up the Freq CV knob on the VCF and turn the Freq knob down. Now you can adjust the relative

amounts of velocity response and filter envelope using the sliders in the new Mixer.

Adding key velocity to amplitude (loudness) is only slightly more complicated. To do this, we'll need a VCA module (from the VCV palette of modules) and yet another mixer.

1. Open the Browser and drag a VCA module into the patch.
2. Duplicate one of the Mixer modules.
3. Remove the connection between the ADSR and the main Mixer's CV 1 input.
4. Attach this ADSR's output both to the CV input of the VCA and to In 1 of the new Mixer.
5. Attach the Vel output of the MIDI-CV to the In of the VCA.
6. Attach the output of the VCA to In 2 of the new Mixer.
7. Attach the output of the new Mixer to the CV 1 input of the main Mixer.

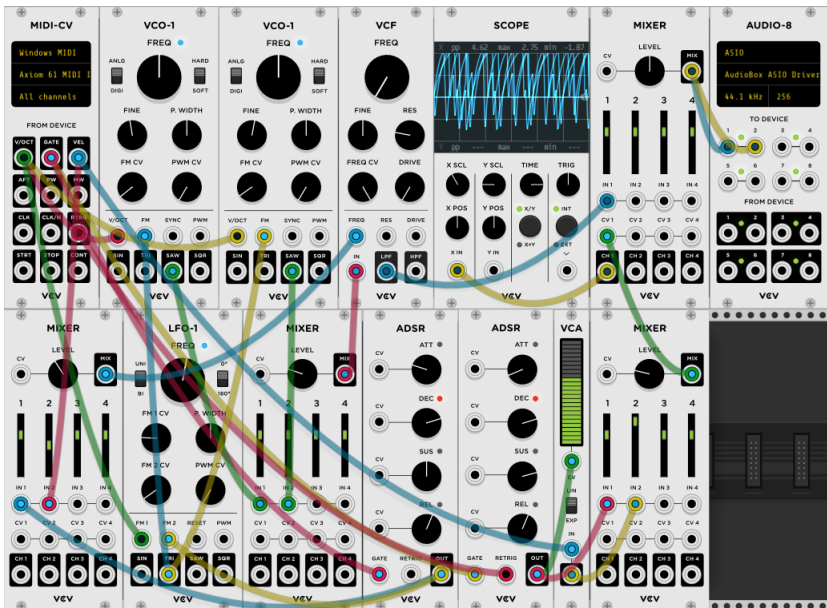


Figure 9. Velocity control of both the filter cutoff and the main amplitude has been added. If you inspect the patch closely, you may notice that I have also added modulation from the VCF envelope to LFO rate.

At this point, the loudness of the audio will be affected by both of the inputs to the new Mixer. You can adjust these inputs to taste.

For the next-to-last step in this tutorial, we're going to replace the MIDI-CV input with a step sequencer. Step sequencing is a vital part of modular synth sound design, so let's give it a try.

1. Open the Browser and drag a Seq-3 from the VCV palette into the rack.
2. Drag the output cables from the MIDI-CV Vel jack over to the Row 3 jack in the sequencer. (They will now be mono, not poly.)
3. Disconnect the V/oct input of the first oscillator, and connect this jack instead to the Row 1 output of the sequencer.
4. Do the same with the second oscillator, connecting its V/oct input to the output of Row 2 of the sequencer.
5. Disconnect the Gate outputs of the two ADSRs, and reconnect them to the Gate output of the sequencer.
6. Delete the MIDI-CV module. We're done with it.

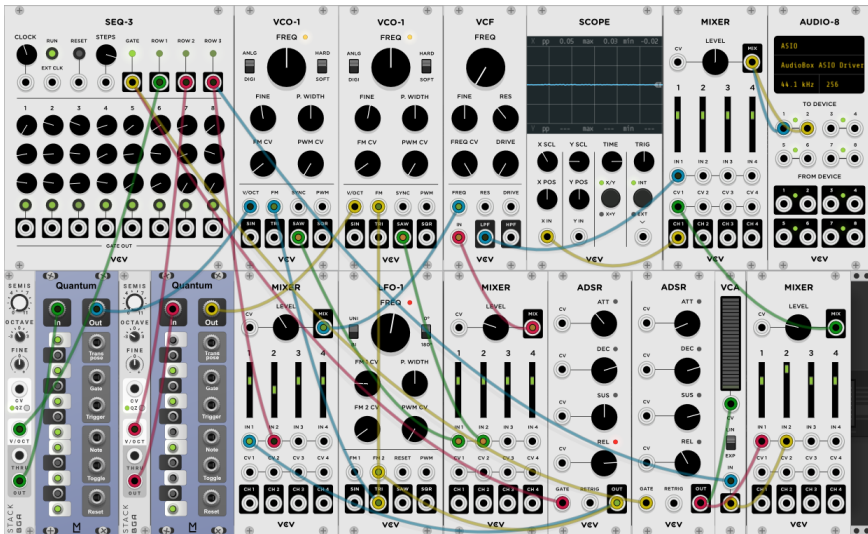


Figure 10. A step sequencer has replaced the MIDI-CV input module, and quantizers from ML Modules and transposers from Bogaudio have been added to the patch.

7. Adjust the knobs in the three rows of the sequencer to taste. The top two rows will control the pitches of the two oscillators. Row 3 substitutes for velocity, so you can use it to accent certain notes.

When you do this, you'll find that tuning the knobs in the first two rows so as to produce standard musical pitches is quite difficult. This is because the outputs of the Seq-3 module are not quantized. If you've downloaded the ML Modules set, you can search for ML Quantum in the Browser. If you haven't, you'll need to navigate to the Library page on vcvrack.com, select the ML Modules set by clicking on its +Free button to change it to -Free, then launch VCV Rack and update its modules from the Library menu tab. Once you've done this, you can make two Quantums, as shown in Figure 10, patch them between the first two Row outputs of the sequencer and the oscillators' V/oct inputs, and activate a few of the steps in the Quantums by clicking on buttons in the left row.

You'll also find that the pitches coming from the oscillators are rather high. This is because the knobs on Seq-3 go no lower than 0. You can adjust the oscillators' frequency downward using their own knobs. A better option is to leave the oscillators tuned to their default pitch and insert Bogaudio Stack modules between the Seq-3 row outputs and the Quantum quantizers.

To finish up, how about adding an audio effect?

1. In the Browser, find the VCV Delay module and drag it into the patch.
2. Remove the connections from the main Mixer to the Audio-8.
3. Patch the output of the main Mixer to the input of the Delay.
4. Patch the output of the Delay to the To Device inputs 1 and 2 of the Audio-8.
5. Experiment with the Delay's Time, Fdbk (feedback), and Mix knobs.

6. When you get bored, click the Run button in Seq-3 to turn it off.

If you add a Delay to the patch without replacing the MIDI-CV with a Seq-3 as described above, and if your MIDI-CV is set to more than one poly voice, you'll find that the Delay module only "hears" voice 1. To remedy this, you'll need to insert a VCV Sum module on the output of the main Mixer. This will sum the poly voices to a single mono signal, allowing the Delay to process all of the voices.

Setting up a patch so that the steps of the sequencer are synchronized with the delay time of the Delay module is getting a bit more advanced, so we won't get into it here.

4: Module Suggestions

In this chapter I'll list a bunch of modules that I feel you'll find useful for creating your own patches. My goal is not to list everything (since that would be both time-consuming and dull to read) but to guide newcomers to a fairly small but manageable and versatile set of modules. I'll omit some extraordinary modules, but I may include a few of my personal favorites, even though they're not precisely beginner-friendly.

Oscillators

The audio signals we're eventually going to listen to are produced by oscillators. Some people may want to quibble with this definition, and it's true that other types of modules can also produce signals that can be listened to. This is true of noise generators and sample playback modules. It's also the case that some filters, when their resonance parameter is set to a high value, will self-oscillate, producing a tone. Sample playback will be covered in a separate section, as will percussion sound sources.

LFOs (low-frequency oscillators) are oscillators, and when set to a high enough frequency will produce audio that we can listen to. LFOs will be discussed in their own section, below.

Oscillators come in several basic flavors. Analog, wavetable, FM, additive, and physical modeled are the most common types. Most of them have a couple of features in common: There will almost always be a 1V/oct input for controlling the frequency (pitch) of the output in a calibrated way, and most have both coarse and fine tuning knobs. The coarse tuning knob may or may not be calibrated so as to "lock" to equal-tempered half-steps. If it's not calibrated, you can patch Bogaudio Stack to the oscillator's V/oct input.

In addition, many oscillators have one or more inputs with which to change the tone color.

For some reason, if you search for “oscillator” in the VCV Browser’s search field, you’ll find only two modules. The correct text to search for is “vco”. This is an abbreviation for “voltage-controlled oscillator.”

As noted earlier, some oscillators can respond to polyphonic inputs by playing multiple notes at different pitches, and some can’t. This distinction is noted below, and is current as of July 2019. More modules may have been updated to polyphonic operation by the time you read this.

VCV VCO-1 (poly). This is a basic analog-type oscillator, with no frills. The output waves (sine, triangle, sawtooth, and square/pulse) are all available simultaneously. As in classic hardware from the ’60s, pulse width modulation is available only on the square wave. The hard/soft switch applies when there’s a signal (presumably from another oscillator) at the sync input jack. The two types of sync have a distinctly different sound, so they’re both worth knowing about. Choosing different waveforms for the FM input and output will have a significant effect on the tone.

Squinky Labs EV3 (mono). Another analog oscillator module, but with three mono oscillators that can be used separately or together. The octave and calibrated coarse tune knobs are useful, and oscillators 2 and 3 can be synced to osc 1. V/oct input 1 is normalised to inputs 2 and 3, so if there’s only one pitch input all three oscillators will track it. Not a fancy VCO, but useful.

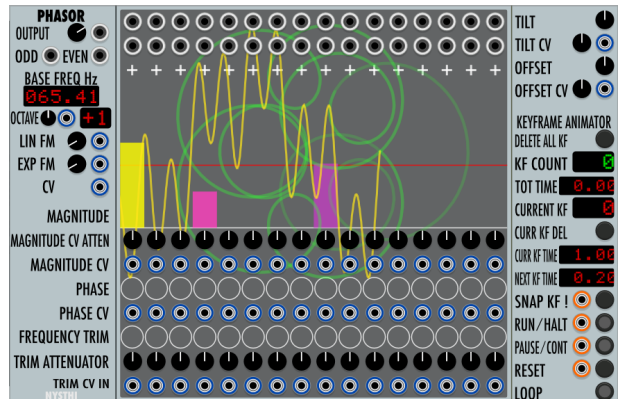
Vult Noxious (poly). Available only in the Vult paid module set, Noxious is perhaps the ultimate analog-style oscillator.



Curiously, Noxious is polyphonic, but (currently, at least) Noxious-Poly is monophonic. There are half a dozen basic tone controls, plus internal FM or FM from an external signal. Also provided: glide, matrix-style modulation, and the ability to respond to a gate input signal by adding chaotic attack transients. My only issue with this oscillator is that the Tune knob can operate in only one mode at a time — fine, coarse, or semitone-calibrated. If you switch to a different mode, the setting you’ve already made disappears.

Nysthi Phasor (mono).

It’s hard to imagine a more comprehensive additive oscillator than Phasor. (Hint: The big right-click menu with all of the editing options opens only when you right-click in the central display area.) Sixteen sine partials can be given their own amplitude, phase, and frequency. Odd and even partials can be sent to separate outputs if desired. An arbitrary number of keyframes can be designed, each with its own mix of partials, and you can move through them as a one-shot, or loop them. There doesn’t seem to be a way to advance one keyframe at a time in response to a trigger input, but setting up an animated timbre by looping slowly through three or four keyframes is quite practical. There’s an individual output for each partial, and also an associated LFO output for each.



Blamsoft XFX Wave (poly). Lots of wavetables, with a position sweep knob and input for moving around. Three wave effect “inserts” with two dozen choices for effect types. Multi-voice spread detuning. All this and a simple additive synthesis panel too, so you can combine your chosen wave at the fundamental with the same wave at any overtone.

Nysthi Seven Seas (poly, mono). Seven Seas lacks the waveform modulation of some other wavetable oscillators. What it has instead is a large two-dimensional bank of waves. CV inputs can be used to modulate both the wave within a bank and the choice of bank. The multi-voice spread detuning can assign different voices to different waves, making for a very rich stereo sound.

Bogaudio FM-OP (mono). Often useful for simple patching chores, such as playing a decorative sequence pattern. FM-OP is only a single operator (the standard term for a sine oscillator with its own envelope generator), so for most FM synthesis situations you'll want to patch together two or more of them, using the output of one (which in this case would be the modulator) as the FM input for another (the carrier). FM-OP doesn't have modulation inputs for its ADSR envelope, but you can easily use a VCV Merge, a stoermelder CV-Map, and if necessary an attenuator such as the AS AtNuVrTr to introduce envelope modulation and get it in the desired range.



Valley Dexter (mono). Though it's a bit CPU-intensive, Dexter provides a solid implementation of four-oscillator FM. We can't call it four-operator FM, because there are no envelope generators; you'll have to patch external envelopes into the Level CV inputs of the oscillators. Each oscillator has its own wavetable and waveshaping choices and lots of modulation inputs. There are two dozen choices for the algorithm (the arrangement of carriers and modulators), a completely separate B voice, which has its own pitch input, and a CV-controllable chord generator.

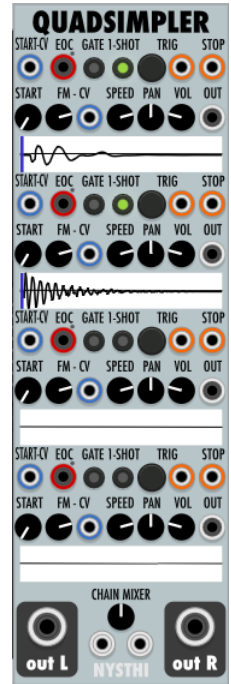
Audible Instruments Modal Synthesizer (mono). Audible's Macro Oscillator and Macro Oscillator 2 are well worth exploring, and so complex that it would take pages to explain them. I'll conclude this quick tour of oscillators with the Audible Modal Synthesizer, which specializes in physical modeling. Its bowed, blown, and struck (mallet) models can be mixed together. The Geometry and Position

parameters have a strong effect on the tone color, in ways that are not easy to predict. The damping, brightness, and contour knobs are also important. Modal Synthesizer needs its own gate input to trigger notes, and it has its own reverb (the Space knob). There are also external audio inputs, so it can be used as a resonant processor. Expressive bells, struck percussion, and other timbres are easy to dial up.

Sample Playback

For many sample playback chores, the right choice will be **Nysthi Quadsimpler** (mono). Each of its four slots can load a sample, and for each slot you can control the frequency, adjust the start point, and either loop the playback or trigger the sample as a one-shot. Using the handy end-of-cycle (EOC) trigger output, you can trigger, for instance, another sample.

If you need more features, the **Nysthi Simpliciter** (mono) will do the job. It will load samples (and append new ones to what's already loaded), record at the audio input, and overdub into the existing audio. Multiple regions can be defined within the sample data. Simpliciter is not a simple module suitable for beginners, it's a powerhouse.



Percussion Modules

Several developers, including AS, Autodafe, and Hora, have released plug-ins with drum sounds, some sampled and some synthesized. However, most of them aren't free. **Vult Trummor 2** is included in the free Vult module set, and its sounds can be shaped in far more ways than the sounds coming from any of the above. Each Autodafe module, for instance, gives you a choice of eight samples, but there are no tone controls at all.

Synthesized kick drums are generally useful for music, but unless

you're doing vintage 808-style beats, sampled sounds are definitely better when it comes to cymbals. With a little hunting around online, you should be able to find free or low-cost samples from a number of vintage drum machines, such as the E-mu Drumulator and the Oberheim DMX. Loading these samples into a VCV Rack sample player will give you plenty of options in the percussion department. The Nysthi QuadSimpler will load and play four samples at a time. (Set the slots to one-shot mode.)



Or, you could drop a few bucks on the sets from AS, Autodafe, and Hora, and save yourself the trouble of hunting for samples.

No discussion of VCV drumming would be complete without a quick mention of **Valley Topograph**. This is not a sound source but rather a rhythm pattern generator. Topograph has outputs for kick, snare, and hi-hat patterns, and a large preset library of standard pop drum rhythms, which can be selected using the Map X and Map Y knobs. The one peculiarity you need to be aware of when adding Topograph to a patch is that its clock input will be ignored unless you rotate the tempo knob all the way to the left, so that it reads “Ext.”

Filters

To find the VCV Rack filters, search in the browser for the text “vcf”. Most of these modules have similar features. The usual configuration includes parameters for cutoff frequency, resonance,

and drive. Cranking up the drive in a filter will probably increase the output level, and will also add a bit of warmth to the tone.

In addition to a lowpass (LP) output, you may find bandpass (BP), highpass (HP), and perhaps notch modes, though notch filters are less common. A choice of cutoff slopes may also be on offer; 6, 12, 18, and 24dB per octave are standard values. Another term for the same thing is 2-pole and 4-pole. A 2-pole filter has a 12dB per octave rolloff slope. Comb filters, formant filters, and state-variable filters are a bit more specialized, but not unusual. (For more on the meanings of these terms, you can do some research online or buy a copy of my book, *Power Tools for Synthesizer Programming, 2nd edition.*)

Some filter modules are stereo, but as with other stereo modules, if you patch a signal into the left channel but leave the right input open, the signal will almost certainly be normalised into the right channel as well.

Valley Feline (mono). A good-sounding stereo filter, lowpass or bandpass, 2-pole or 4-pole. There are two modulation inputs for each parameter — really necessary for the cutoff frequency, at least, and few VCV filters have two. Even more unusual are the Spacing and Target parameters. The cutoff frequencies of the left and right sides can be offset from one another using these. With a single audio input to the left channel the spacing makes for some nice stereo animation.



Vult filters (poly). Vult has a whole line of filters, some of them free and some paid: Ferox (not free), Lateralus, Nurage, Rescomb, Rescomb 2 (not free), Stabile, Tangents, Unstable, and Vortex (not free). Each has a slightly different tone and different features. Lateralus is good for basic lowpass filter work. It's not stereo. For comb filtering, Rescomb 2 is handy, as it has its own built-in LFO for sweeping the cutoff frequency. Cutoff sweeps are an essential part of the sound of

a comb filter. Also worth note: If you wrangle a polyphonic patch that has a separate filter envelope for each voice, Lateralus (and presumably the other Vult filters) will apply envelope modulation separately to the various voices.

Southpole Rakes (mono). For pinpoint tone control, Rakes might be a good choice. It has six resonant bands, each with its own frequency and amplitude controls.

VCAs & Mixers

A mixer is used, obviously, to mix several input signals into a smaller number of outputs (basically, one for monaural sound or two for stereo sound). Some of the VCV mixers have built-in level control for each input channel, which means they can function effectively as VCAs (voltage-controlled amplifiers). Some mixers also have effects send and return features.

A mixer is not a complex device, so there may not be much reason to choose one over another. I like the **mscHack** mixers (available in 9, 16, and 24 channel models) because they have four FX sends and returns, and also four group buses. The Nysthi mixers have no sends, but they do have CV-controllable muting and soloing, which can be handy for developing a song.

For simple VCA work, the **VCV VCA-1** and VCA-2 are dead easy to use. Ditto for the **VCV Mixer**, a four-channel device with no frills, not even a stereo output. It can also be used as a quad VCA.

Crossfading is a useful technique in mixing. The Submarine XF-104 provides four channels of crossfading.

Envelope Generators

Most patches will need at least two envelope generators, one for amplitude (loudness) control and one for tone shaping. The classic ADSR (attack-decay-sustain-release) type needs to receive a gate

signal. The gate is open for some period of time, during which the attack and decay segments are active; the output then settles at the sustain level. When the gate falls, the output falls back to zero during the release stage.

Other envelopes are free-running, and respond to a trigger. A trigger signal is the same as the rising edge of a gate, so you can generally send a gate signal to anything that's expecting a trigger. Once the triggered envelope starts, it continues on its own, without regard to when the gate ends.

For giving notes different articulations, it's useful to have CV control over the time or level of individual envelope stages. You may need an external attenuator to control the amount of CV modulation of the envelope. (See the section on modulation in Chapter 3.)

VCV ADSR (poly). A basic ADSR envelope. With a poly gate input, the output will be poly. The Retrigger input is a nice extra; when this receives a trigger signal, the envelope will return to the attack segment and start over, even if the signal at the Gate input is still high. (The starting over doesn't return the output to zero, however; the new attack starts from the current level, whatever it is.) If the gate signal is not high, the Retrigger input does nothing.

Nysthi Poly DAHD (poly) and **Complex DAHD** (mono). The DAHD is my go-to envelope generator for most musical tasks. It has no sustain stage, and responds to triggers while ignoring the length of the gate. Instead of sustain, it has a hold stage; also a delay stage. All four stages can be controlled with CVs. There's a knob for controlling the curvature of the attack and decay, and also a Scale knob. You might expect that setting a negative scale would cause the output value to drop below zero, but that's not what happens. Instead, the envelope goes high during the delay, drops to zero during the hold segment, and rises back up to a positive



value during the decay. DAHD can loop, can be set (using the Busy button) to ignore new triggers that arrive while it's cycling through its stages, and has a built-in stereo VCA, allowing it to be used as an amplitude envelope with no external VCA module needed. It also has poly trigger outputs for end-of-delay, end-of-rise, end-of-hold, and end-of-cycle. Using these outputs you can create complex envelopes. And then there's the Chain input, with which you can sum two or more envelopes to create complex contours.

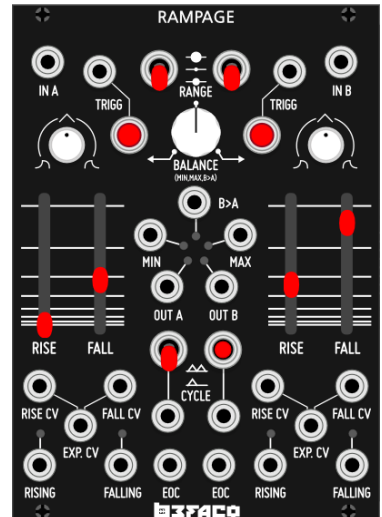
LFOs

Like envelope generators, LFOs are an indispensable tool in the sound designer's arsenal. They're typically used for pitch modulation (vibrato) and amplitude modulation (tremolo), but are possibly even more useful for making changes in parameters like envelope attack time, oscillator waveform, and panning. VCV is well equipped with LFOs.

VCV LFO-1 (poly). A good basic LFO, with the classic analog waveforms. The unipolar/bipolar switch is handy, because sometimes you want a unipolar response (such as for producing tremolo) and sometimes you want bipolar (for vibrato).

Befaco Rampage (mono). Rampage is a multi-purpose dual module. It will do attack/decay envelopes, output a triangle/saw LFO, or serve as a slew limiter, which is useful for smoothing out clicks and creating a pitch glide. The minimum and maximum outputs, which combine what's happening in the left and right sides, can be handy for producing complex modulations, and there are plenty of other ins and outs.

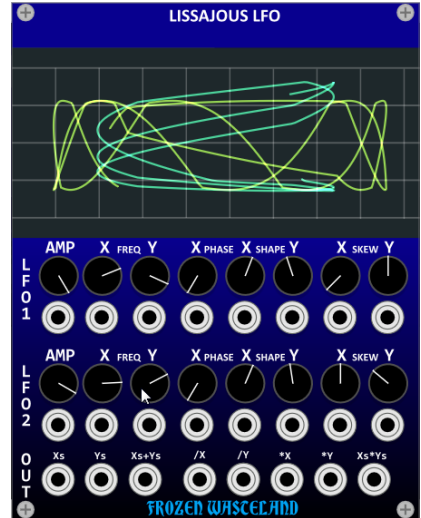
Bogaudio 8FO (mono). Eight simultaneous outputs of the same waveform, by default each of them 45 degrees out of phase with



those above and below it. Phase modulation of any output is allowed, so complex motorboat noises are quite practical. A built-in sample/step feature lets you turn the waveform into a staircase. Also, the frequency can be dialed up to 400Hz, so 8FO can function as an audio oscillator.

Frozen Wasteland Lissajous LFO.

I'm partial to this quirky device, because I find that combining two LFO waves is often more interesting musically than using one by itself. The Lissajous LFO sports two 2-dimensional LFOs, each with X and Y frequency control and an amplitude setting. The outputs are sums, multiplications, and divisions of pairs of waves. Phase, shape, and skew controls are also provided, and everything is CV-controllable.



Frozen Wasteland BPM LFO 2. A useful device for rhythmic modulation contours that are synced to a clock. The module has its own internal clock, which senses the incoming clock signals and adjusts as needed; the result is that it will continue to run even when the clock module is not running. The tempo can be multiplied or divided in integer amounts (for instance, 3/5 of the tempo, or 7/6), so setting up complex polyrhythms is a breeze.

Step Sequencers

At a guess, well over half of the music made in modular synthesizers is produced using step sequencers as the primary or exclusive source of notes (and other phenomena). The importance of step sequencing would be hard to exaggerate. And indeed, the Browser in my VCV installation finds a hundred items when I search for “seq”. (Not all of them are actually sequencers.) Some of the VCV sequencers are classic vanilla designs; some are full-featured; a few are insane.

Most step sequencers have their own internal clocks, which are used for advancing the sequence from one step to the next. To coordinate them with other musical events, though, you'll most likely want to use an external clock source. External clock sources are discussed in the section after this one.

With most sequencers, you can set the length of the sequence (the number of steps it will go through before it repeats). Some allow you to control the direction of the stepping — forward, backward, back-and-forth, random, and so on. Some will let you skip or mute certain steps within the pattern, or set the probability that a given step will play. Some can send a gate or trigger signal from a given step if desired. Some have a pattern memory, allowing you to construct a whole song by stringing patterns one after another. Some can repeat a step several times before moving on to the next step.

Some have outputs that are quantized (calibrated in well-tuned half-steps), but others don't. If the sequencer doesn't quantize its own output, you'll probably want to patch the output through a quantizer module; trying to get an in-tune musical phrase from unquantized knobs is just too painful a process to contemplate.

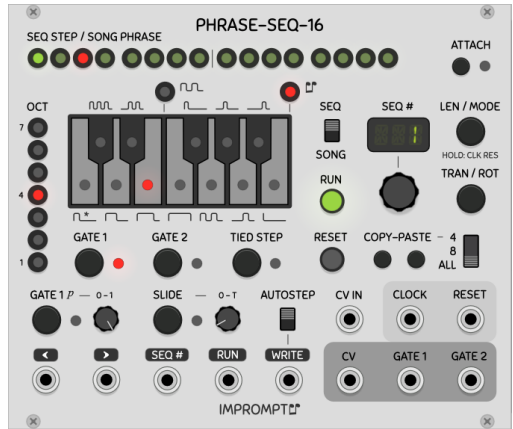
Most step sequencers have several lines, each of which outputs a monophonic signal. Unless otherwise noted below, you should assume that the only way to get polyphonic sequencing with a step sequencer is by sending the output of each row to a different oscillator, or by using the VCV Merge module to combine the separate outputs into a poly signal that can be sent to a single poly oscillator.

With so many options to choose from, any of which may be musically relevant to you today (or not), we're going to have to dig into some detail in this module category. I'm not going to mention every single feature of every sequencer, however. That would take way too much space.

AS 16-Step Seq. This module is very similar to the VCV Seq-3, the

main difference being that it has 16 steps rather than 8. That being the case, there's no reason ever to use Seq-3.

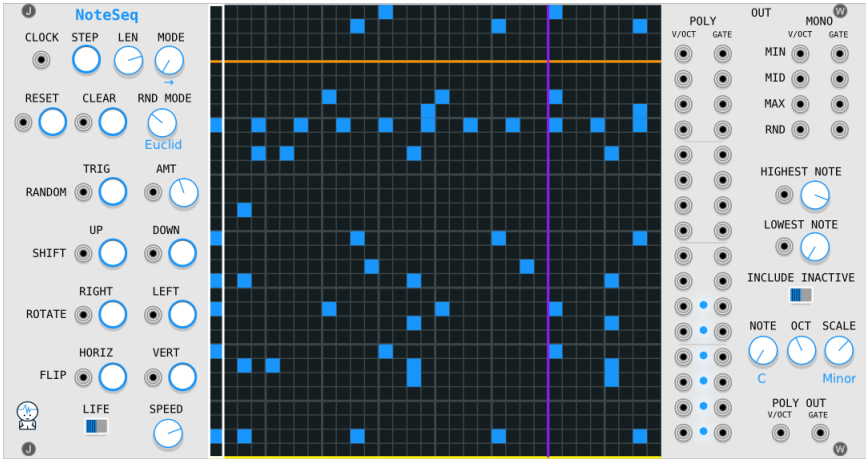
Impromptu Modular Phrase-Seq 16. The Impromptu sequencer line includes Phrase-Seq 16, Phrase-Seq 32, and Foundry, among other items. Foundry is very powerful indeed, but not so easy to learn and use. I recommend that you start by learning Phrase-Seq 16. Most of what you learn will apply directly to the larger modules.



Phrase-Seq 16 can be programmed with up to 16 separate sequences of up to 16 steps each. The sequences can then be strung together to create a song, which again can have up to 16 items (called phrases in the documentation) in its list. Instead of programming a song, you can use the Seq# CV input to select different sequences on the fly, so with a bit of clever patching your song can be as long as you like. There's a handy keyboard on the panel for note entry, and copy/paste commands for copying one sequence to multiple locations.

Steps can be tied to make longer notes, and the pitch of any step can be set to glide from the pitch of the previous step. There's a second gate output for each step, which can be used for accents. A sequence can move forward, backward, back-and-forth, and so on. The rotate editing command can be very useful, because sometimes you'll create a sequence and then discover that the downbeat sounds better on some other step than step 1.

JW-Modules NoteSeq. The first feature that sets NoteSeq apart is that it can play a chord on any step. The big square display lets you see you see the entire sequence. In addition to the poly note and gate outputs, there are mono outputs for each of up to 16 notes. You can



choose a base pitch and scale (major, minor, pentatonic, etc.) for the sequence. CV-controllable knobs for highest and lowest note let you skip the highest or lowest notes in your sequence so as to add or subtract notes from the chords.

You can generate random patterns in several different ways, and then edit them to make something you like. New random patterns can be generated from trigger signals, allowing NoteSeq to play a constantly shifting pattern of notes. You can also move the note pattern up or down in pitch using trigger inputs, while the music plays.

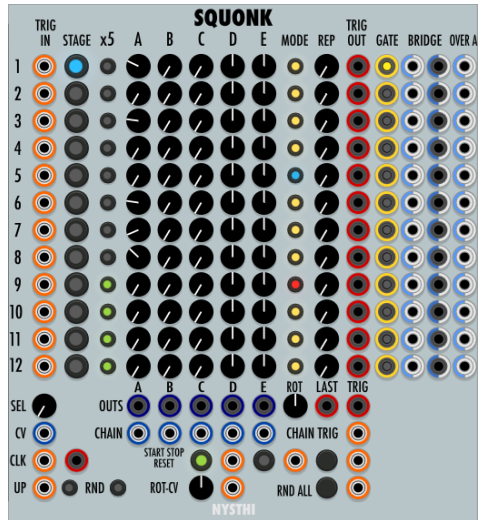
JW GridSeq is another popular option, by the way. Its 4x4 matrix of knobs can be clocked separately in the horizontal and vertical directions. Clocking the two at different speeds will produce various patterns.

Nysthi Squonk. Squonk is a complex and rather peculiar sequencer. At first glance it may seem a strange choice to include in a list of introductory modules. But Phrase-Seq 16 and NoteSeq are also complex, and building complicated sequences is part of the fun of working with a modular synthesizer. So here we are.

Squonk has five rows — or in this case, columns — of knobs. The first three operate unidirectionally, which is suitable for pitch control,

and the last two bidirectionally, which is more suitable for control of expression. Any column can be used for whatever you like, of course. Squonk's outputs are not quantized, so for each column that you're using for pitches you'll need to patch a quantizer between Squonk and the oscillator's V/oct input.

Squonk can have up to 12 steps, which may not seem like enough, but multiple Squonks can be chained, so you can build sequences of arbitrary length. (The patching needed for chaining is a bit tricky.) Instead, you can send a trigger input to any stage; when Squonk sees a trigger at that input, it will jump immediately to that stage. Or you can use the Select CV input to choose Squonk's active step, perhaps from the step output of another Squonk.



Each step has an Override A input. When a signal is present at this input, the signal is passed through to the column A output instead of the setting of the column A knob. Using a couple of these inputs (from another step sequencer whose length is different) is a great way to add a little variation to your sequence.

Impromptu Modular Gate-Seq 64. Gate sequencers, of which VCV has several, don't output note pitch data. They specialize in creating rhythms. The rhythms will be most suitable for drum patterns, but really you can use them for whatever you like.

Gate-Seq 64 is a good choice for drum pattern programming. It can run as four simultaneous 16-step patterns, as two 32-step patterns, or as a single 64-step pattern. Patterns can be strung together into songs. Each step in a pattern can have its own probability of firing, a useful feature for adding variation to a drum pattern.

Clock Sources & Processors

Clock pulses can be sent to a step sequencer from an LFO or from a dedicated clock source. A clock source is usually a better choice, because its output will be calibrated in beats per minute (bpm). On the other hand, if you want a smooth acceleration or deceleration, an LFO square wave would be the right choice.

By far the most popular choice as a clock source is **Impromptu Modular Clocked**. It's easy to use, its features are comprehensive, and if you need more outputs, two or more Clocked modules can be chained together. The pulse width of the gate outputs can be controlled, as can the amount of swing. The most useful divisions and multiplications of the clock tempo are supported; if you need more complex polyrhythms, Nysthi Clock Mult Div can be pressed into service.



I/O Modules

The default VCV module set provides most of the input and output you're likely to need. **MIDI-CV** is for playing VCV Rack from a MIDI keyboard or other hardware controller. **MIDI-CC** can translate up to 16 different MIDI continuous controller (CC) messages to CV form; with this, you can control VCV from hardware knobs and sliders. **MIDI-Gate** will assign 16 different MIDI notes — or

computer QWERTY keys, if you like — to gate outputs, giving you instant command over whatever processes you like. For instance, you could run a clock signal and a MIDI key through a logic AND gate, so that the clock is sent to its destination only while you're pressing the key.

A similar set of modules can translate internal CV signals into MIDI messages, allowing you to control external MIDI devices from within VCV.

The **Audio-8** module is standard for sending VCV's final audio output to your computer's soundcard or audio interface hardware. If eight channels of output aren't enough, you can use Audio-16.

Effects Processors

For simple VCV projects, you may prefer to patch an effect in-line (technically called an insert) between the sound source and the mixer. In this case, you'll need an effect that has a wet/dry amount knob. If you're using a mixer with FX sends, you'll want to turn the wet/dry knob fully clockwise, to 100% wet.

The first choice for reverb seems to be **Valley Plateau**. It's highly controllable, and it sounds terrific.

It's a peculiarity of the current slate of VCV modules that many of the delay units can't be clocked; their internal timers are free-running. Because a clocked delay is so useful, one good choice would be the **mscHack Step Delay**. Alternatively, the **AS Delay Plus Stereo** pairs well with the **AS BPM Delay/ms Calc**; the latter has a variety of outputs for common rhythm

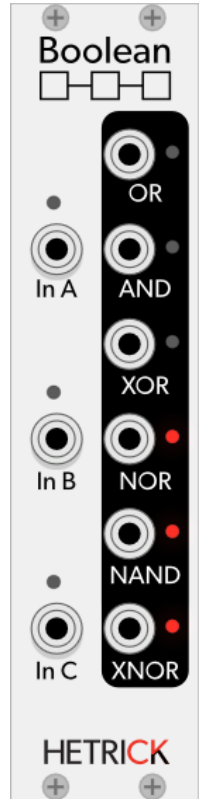


values, which can be patched to the Time inputs of one or more Delay Plus Stereo modules.

Logic

In a modular system, you sometimes want to have the patch make decisions based on certain conditions. This could be for something as simple as opening a VCA so it will pass a signal only when two gate signals are both high. This would be done with an AND gate, a simple logic utility. Or you might want to step through several different signal routing conditions in sequential order. VCV Rack is well supplied with devices that will do this type of thing.

HetrickCV Boolean Logic has three inputs. Its six outputs provide OR, AND, XOR, NOR, NAND, and XNOR result. The OR output is high if any of the three inputs is high. The NOR output is just the opposite — it's high only when none of the three inputs is high. The AND output requires signals on inputs 1 and 2; the output is high only when both 1 and 2 are high. In this configuration, input 3 doesn't have to be connected, but if it is, the output will be high only when all three inputs are high. The NAND (not AND) output is, again, the opposite of AND; it's high when AND is low and vice-versa. The XOR (exclusive OR) output is high when exactly one of the three inputs is high, and low otherwise. You can probably guess what the XNOR output is.



DHE Modules Gator is functionally very similar to HetrickCV Boolean Logic, but it has 16 inputs, inputs can be negated (flipped) with front-panel buttons, and there are outputs for even and odd numbers of true (high) inputs.

Befaco Rampage was mentioned earlier as an LFO and envelope generator. It's also useful as a comparator; whenever the signal in the B section is higher than the signal in the A section, the B>A

jack sends out a gate. The signals can be either internally generated by the rise/fall contour generators, or arriving externally at the In A and In B inputs. The **Bogaudio CMP** module is another good comparator; its A or B level can be set manually if desired, and it can send out a trigger whenever the two signals are equal.

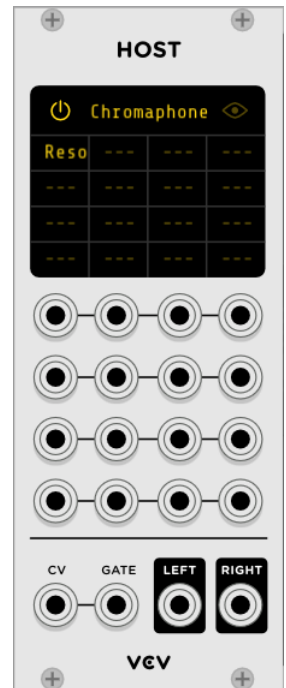
Utilities

This is a grab bag category, so (to mix a metaphor) roll up your trousers and wade in. We'll look at various widgets that didn't fit neatly into earlier categories.

stoermelder Strip. Any serious VCV sound designer will need this module. Strip has no signal outputs. What it does is memorize the module configuration to its left or right, allowing that configuration to be copied and pasted or saved to disk as a multi-module preset. With Strip, you can design a complete synth voice, store it, and later load it into a different patch, something that is not otherwise possible in VCV.

VCV Host. Not a free module, but if you have any VST instruments or effects, you'll want to own it. 64-bit VST plug-ins are required; the 32-bit versions won't load. Input jacks can be used to send CV automation to the VST plug-in's parameters, and if you send poly notes and gates from VCV, your VST will play polyphonically.

Nysthi Master Recorder 2. Several different modules can record your audio to the computer's hard drive. I suggest using the Nysthi module because it has built-in meters and a built-in compressor, as well as triggerable start and stop commands.



Amalgamated Harmonics PolyScope. The input of PolyScope

needs to come from the output of a VCV Merge. With that patch cord in place, PolyScope can display up to 16 waves in real time. The blue traces are frankly too dim to be seen easily, but eight or ten signals can be displayed very nicely at once.

Alikins Momentary On Buttons. This basic module has 13 buttons. Click on a button with the mouse and the corresponding output goes high for as long as you hold the mouse button down. Couldn't be simpler.

Alikins Big Mute Button. If you're worried about ear damage or waking the neighbors, patch the Big Mute Button between your mixer and the audio output module. Clicking the button (and yes, it's big) will mute or unmute the stereo signal.

stoermelder CV-Map. CV-Map is a brilliant tool. With it (and a VCV Merge module to feed its input), you gain CV control over almost any parameter in VCV Rack, even when that parameter has no CV input of its own.

Impromptu Modular Tact. Tact is a two-channel playable mixing surface. Click anywhere in the strip and the CV output rises or falls to the corresponding level, at a rate determined by the Rate knob. Trigger inputs can send the strip up to its maximum or down to its minimum. A value can be stored, and will be recalled by a trigger at the Recall input. By patching the CV output into a VCA or a mixer's level CV input, you can adjust your mix while the music plays, with much more precise control than if you were turning a knob with the mouse.



AS AtNuVrTr. Simple and effective. Two independent stages for attenuating and/or inverting CV signals. Includes an offset knob — and both the offset and the amount of attenuation can be CV-controlled, which is useful for changing things like envelope amount from one sequencer step to the next.

Bogaudio S&H. This is a simple dual module, with two identical stages. Each stage can function as either a sample-and-hold or a track-and-hold. If there's no input, an internal noise source is used, but I find that patching an LFO sawtooth or triangle wave to the input is often more interesting.

Bogaudio Stack. For effortless transposition from one key to another, or just for an octave shift up or down, Stack is ideal. Run your pitch CV through it on the way to the oscillator. Stack is not a quantizer, however; it's just a transposer.

ML Modules Shift Register. A shift register is an excellent resource for machine-generated music. Essentially it's a series of sample-and-hold modules lined up in a row. All of them are triggered by the same clock input. When a clock signal arrives, whatever is at the main input is sampled, and the result — a fixed voltage value — is sent to the first output. On the next clock impulse, this value is passed on to the second output, and a new sample is taken for the first output. And so on. Each sampled value is passed on, one step at a time, to outputs 1 through 8.

Nysthi Jooper. For switching signals from one or more inputs to one or more outputs, nothing beats Jooper. Its eight independent lines each have four inputs and four outputs, which can be configured in any combination. You can store and recall a series of scenes, each with its own routing configuration, and then switch from one scene to another under external control.

VCV Notes. A handy text module for taking notes on what's going on in your patch. For short labels of specific modules or signal chains,



you can use **Submarine TD-202**. Its text is vertical, unfortunately.

Do the Math....

Before we bid farewell to this introductory tour of VCV Rack, we need to consider the possibility that you just won't be able to find a module that does exactly what you need. Wouldn't it be nice if you could write an equation with which to process a signal? Indeed it would — and there are several ways to do this. **Nysthi Const Add Mult** is rather straightforward math processor. Each of its four independent stages has A and B inputs, and can output the sum, the multiplication product, and the division A/B . By chaining one output to an input of another stage, you can do some interesting things. The **AD** series from **Submarine** has six math processors in each module, several CV inputs, and a wide choice of math equations. **Frank Buss Formula** goes even further: You can write your own equation.

5. Ideas

There are surely thousands of useful patching ideas that you could dig into with VCV Rack. In this all-too-brief chapter, I'm going to stick to a few basic ideas. (Maybe in a later edition of this book I'll go further.)

Recording the Audio Output

If you want to use your VCV audio in another program, perhaps a conventional multitrack DAW such as Reason, Logic, or Cubase, you'll need to record its output to your hard drive. There are several ways to do this. The VCV Recorder module is good, as is the Nysthi Master Recorder 2. Both will record stereo .wav files. I like the Nysthi module because it has meters and a built-in compressor. (Note: When using the Pre-Choose Destination File command in Master Recorder 2's right-click menu, the Autosave button must be switched off. If Autosave is on, your filename and destination folder will be ignored.)

1. Attach the recorder's inputs to the outputs of whatever module (such as a mixer) is sending to your Audio-8 I/O module.
2. Right-click the recorder and choose a destination file and a folder on your hard drive in which to put it.
3. Start your patch playing, if necessary. (You may also prefer to start recording first and later trim the silent start of the file in an audio editor.)
4. Click the Start button on the recorder.
5. When you've recorded whatever you want to record, click the stop button on the recorder.

Other factors may be relevant. If you're planning to import your

audio file(s) into a multitrack DAW, you'll want to set the tempo of your clock source before starting the recording. If the output file has clipping, you'll need to re-record after lowering the level of the signal going to the recorder. When stopping VCV at the end of the recording, you may want to switch off the clock that's driving your sequencer(s) but continue recording the audio output until the reverb and delay effects finish decaying back to zero.

Smooth Fades

Operating the tiny knobs on an on-screen mixer using your mouse is easy enough, unless you happen to want a smooth fade-in or fade-out. Smooth fades are desirable both for live performance and for recordings. A great tool for doing smooth fades is Impromptu Modular Tact. You'll also need to be using a mixer that has CV inputs for controlling the levels of your channels — or, alternatively, a VCA module for each channel.

1. Hook the CV 1 or CV 2 output of Tact to a level input jack.
2. Set the rate of the fade-in/out using the Rate knob.
3. Click the vertical strip to start Tact moving and set the endpoint of the fade.

You can also use the store button and the Recall input jack to set an intermediate level on the strip, and the to-max and to-zero input jacks as needed.

Synchronizing Step Sequencer Starts

This is a thorny topic in VCV Rack. Ideally, when you click the Run button on your master clock module, all of the sequencers you're using should start nicely at Step 1 — but that won't necessarily happen.

By default, a step sequencer will move from one step to the next each time it receives a clock signal. If it's on Step 5 when the clocks

stop arriving, the next time it sees a clock, it will proceed to Step 6.

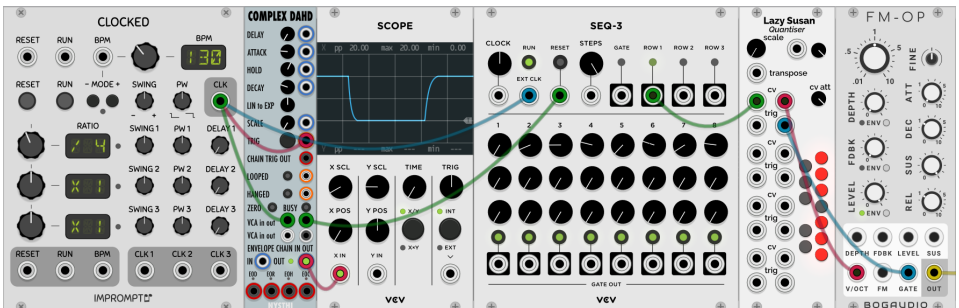
Many sequencers have a Reset jack. When the sequencer sees a pulse at the Reset input, it will jump back to Step 1, ready to go. The Impromptu Modular Clocked module has a right-click-menu command that will cause it to send a Reset when the clock is turned off (that is, when the Run button is deactivated). You'd think this would solve the problem — just send a signal from the Reset output of Clocked to the Reset input of your step sequencer, and the next time you start the clock, the sequence will start playing at Step 1. Unfortunately, this may not be what happens.

Here's the problem: If a sequencer resets to Step 1 when it receives the Reset command, the next clock pulse it receives will probably send it on to Step 2! In my opinion, this is incorrect behavior, but many developers seem to feel it's just fine. If your music requires that all of your sequencers start at the same spot in the phrase, you'll have to do something tricky to manage it.

Impromptu Phrase-Seq 16 resets in such a way that the next clock causes it to play Step 1, as does Nysthi Squonk. They're the exceptions rather than the rule.

So here's a way to get a typical VCV step sequencer to start at Step 1 rather than Step 2 when you click the Run button on your clock source.

1. From the Browser, drag a Nysthi Complex DAHD into your patch.
2. Set the DAHD Attack knob to 0.25, the Hold knob to 0.6,



and the Scale knob clear down to -2. (The values don't need to be precise, but you can set them precisely by right-clicking the knobs.)

3. At slower tempos, the Hold knob will need to be set to a higher value.
4. At fast tempos, you'll need to reduce the Attack time.
5. Do not click the Zero or Busy button in the DAHD; these buttons will prevent the patch from working the way we want it to.
6. Patch your clock input to the DAHD's Trig input, and also to its first VCA input.
7. Patch the first VCA output to the reset input jacks of any misbehaving sequencer.

This patch works because of the way the DAHD interprets a negative setting of its Scale knob. A negative Scale value inverts the envelope, but the envelope still goes from zero to a positive value. As a result, the built-in VCA will be open when your first clock arrives at the Trig input. It will then close down. As long as the Hold segment of the envelope lasts longer than one clock step, the VCA will stay closed until you stop the clock. It will then open up and be ready to transmit the next clock as a reset command.

Randomness

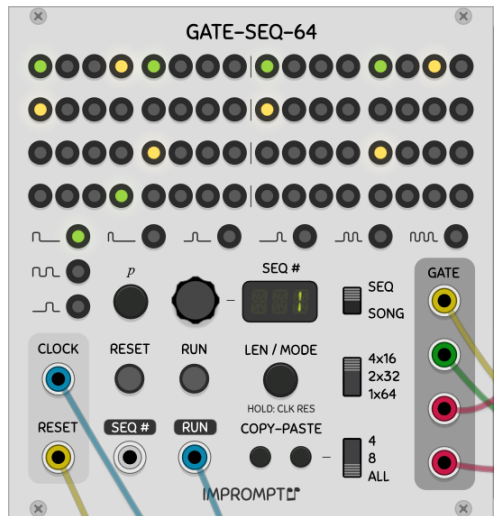
It's often useful to add a bit of controlled randomness to a patch, so as to make it more lively and less predictable. There are many ways to do this in VCV Rack. Below are a few ideas to get you started.

Extra Drum Hits. When setting up a drum pattern, you may want to have certain notes play only sometimes, not every time. To illustrate this, we'll use the **Impromptu Gate-Seq 64** and Clocked, and a Vult Trummer 2.

1. Set one of the Clocked outputs to x12.

2. Patch that clock signal to the Clock input of the Gate-Seq.
3. Click and hold the Len/Mode button on the Gate-Seq until you see x1 on the display.
4. Turn the knob until it says x4.
5. Connect the Reset output of Clocked to the Reset input of Gate-Seq.
6. If the Run button on Gate-Seq is lighted, click it to switch it off.
7. Connect the Run output of Clocked to the Run input of Gate-Seq.
8. Patch the top Gate output of Gate-Seq to the Gate input of Trummor.
9. Patch Trummor's output to your mixer module.
10. In the top row of Gate-Seq, click buttons 1, 5, 9, and 13.
11. Click the Run button on Clocked.

At this point you should hear a nice four-on-the-floor kick drum pattern. Now for some randomness.



12. Click button 4 in the top row of Gate-Seq, and then click the button that says p (for probability).
13. Do the same with button 15.

The probability of steps 4 and 15 is now 50%. You can adjust those values up or down with the knob. Sometimes you'll hear a drum hit a sixteenth-note before beat 2, and sometimes you'll hear one on the "and" of beat 4.

If your drum pattern uses only the top row of Gate-Seq, you can use the other rows, again with probability settings on specific beats, to add occasional accents to the drum sound.

Trigger Routing. Sometimes you want a step sequencer to march along steadily, but sometimes you want it to linger, not every time but occasionally and at random, on some step before it moves on. The easy way to do this is using the **Bernoulli Gate** from Audible Instruments. This module accepts a gate input and passes it on to either the A or the B output, choosing an output at random. The probability of A vs. B is controlled by a knob.

Sample-and-Hold. Sample-and-hold, or S&H for short, is a classic synth effect. The **Bogaudio dual S&H** is perfect for this.

1. Patch the gate output from a clock to the Gate input of the S&H.
2. Patch its output to the cutoff frequency CV input of a filter.
3. Turn up the CV modulation amount, and turn the main Cutoff knob down. Possibly turn up the resonance.
4. Patch the output of an oscillator's sawtooth wave (or anything with a lot of overtones) into the filter.
5. Start the clock.

At this point the filter's cutoff will step up and down randomly, in time with the clock. This will add a rhythm to the oscillator's tone.

Other Modules. A few step sequencers, such as JW NoteSeq and Geodesics Entropia, can jump around randomly in their list of steps. For further experiments in randomness, you may want to try the **Stellare Modular Turing Machine**, the **Nysthi Soy Model SOU**, or the **Audible Instruments Random Sampler**.

...and that's all for now. Happy Racking, everybody!