

# Vectron



creamw@re<sup>©</sup>

fidelity at work.

Version 1.0

Vectron

Synths Page

Main Table of Contents

Contents

Index

1

# Contents

## Introduction

Preface

About The Manual

## Overview Of The Vectron

Overview Of The Vectron Sound

Architecture

Vector synthesis

About Vectron wavetable playback

Overview Of The Wavetables

Wavetable Categories

Modulation options

Types of modulation sources

**Vectron User Interface Overview**

The parameter pages

The graphical ModWheel

Auxiliary surfaces

## The ProgOsc Page

**OscA .. OscD – The Oscillators**

Select Wave

Oscillator tuning controls

**Mix Level**

**Grunge**

**Portamento/Glissando (Porta/Gliss)**

ON

Speed

Porta / Gliss

Norm / Finger (normal / fingered mode)

Porta/Gliss with one voice (legato)

**Chorus**

ON

LFO

Offset

FB (feedback)

Speed

Depth

Level (effect level)

**Delay**

ON

Del L and Del R

MIDI

FB (feedback)

Cross (cross-coupled feedback)

HDamp (high-frequency damping)

Level (effect level)

A special note regarding external MIDI sync

**Vec Disp Mode (Vector Display mode)**

## Vector Mixing Overview

**The vector concept**

**The Vector Display**

Vec Disp Mode (Vector Display mode)

**The graphical joystick controllers**

*Continued on next page*

## The Vector Envelope

### Vector Envelope Editing Basics

- The vector envelope multi-segment display
- Basic multi-segment display operations
- Display and editing of point vector positions

### Vector Envelope Editing

#### Convenience Features

- The Point Step buttons
- Mono mode
- Point Solo mode
- Env Solo mode
- Speed Ctrl DISABLE
- Copy / Paste

### The Vector Envelope Speed Controls

### The Vector Envelope Mix Amount Controls

### The Vector Envelope As Modulation Source

- With or without vector mix modulation
- Vector envelope component signals
- Special capabilities as a modulation source

### Vector Envelope Programming Tips

- Setting envelope point positions
- Creating a loop in the envelope
- Deleting an envelope loop
- Adjusting loop repeat count

## The Amp/Filter Page

### Amplifier

- Att (envelope attack time)
- Dec (envelope decay time)
- Sus (envelope sustain level)
- Rel (envelope release time)
- V Att (attack time velocity mod)
- Speed Track – Center – Slope
- V Lev (amp level velocity mod)
- Additional level modulation

### Filter

- Cutoff
- Res (resonance)
- Env (cutoff envelope mod)
- Vel (cutoff velocity mod)
- Kbd (cutoff keyboard follow)
- Additional cutoff modulation

### Filter Envelope

- Att (envelope attack time)
- Dec (envelope decay time)
- Sus (envelope sustain level)
- Rel (envelope release time)
- Shape A (attack shape)
- Shape D-R (decay-release shape)
- Vel Depth (env depth velocity mod)
- Speed Mod
- Speed Track – Center – Slope

## The LFO Page

### Overview Of The LFOs

#### LFO1 and LFO2

- Waveform select
- MIDI (LFO speed MIDI sync)
- Rate
- 16ths-192nd (16th-notes – 1/192 notes)
- Retr (retrigger)
- Ph (initial phase)
- Dly (delay time)
- F-In (fade-in time)
- Amp Mod (amplitude modulation)
- Var Src (variable-level LFO control source)
- Rate Mod 1-2

### The SineLFO

## The Vec/Pan Page

### Vector Modulation

- A-D Mod (modulation on the A-D axis)
- B-C Mod (modulation on the B-C axis)
- Vec Env (vector envelope amount)
- Joystick (joystick controller amount)

### Pan Modulation

- Base (pan base position)
- Modulation source / amount controls

## The Pitch Page

### Global Pitch Modulation

### Pitch Modulation Per Oscillator

### Pitch Modulation Possibilities

*Continued on next page*

## The Global Page

### Mstr Tune (Master Tuning)

- Coarse (coarse tuning)
- Fine (fine tuning)
- Bend (pitch wheel range)

### Sync (LFO / Delay Sync)

- Ext Clk ON (external sync)
- Tempo
- LFO Retrig

### Velocity (MIDI Velocity Curves)

- Curve (curve type)
- Intensity
- Offset

### Key Scaling Generator

## The Auxiliary Surfaces

### The Joystick Controller

- Opening the Joystick Controller surface
- Operating the joystick
- Effect of the joystick on the vector mix
- AutoReturn
- Changing the AutoReturn position
- Centering the joystick
- What are those faders for?
- Joystick MIDI
- The joystick amount controls
- The joystick as general purpose controller
- If the joystick seems not to work

### The Aux Controllers

- Opening the Aux Controllers surface
- Applying the controller wheels
- Centering a controller wheel
- AutoReturn
- Changing the AutoReturn position
- Controller wheels and MIDI

## Making Waves – The WaveCreate Page

### Introduction

- Important tips for getting started
- The WaveCreate waveform display

### The Direct Audition (DA) Option

- DA Volume
- DA Grunge
- Temporary disable while Sound Preset List is open

### Wave Mixing

- Selecting waves for the mix
- The Harmonic controls
- The Use Waves From Preset button
- Extracting waves from sound presets
- Recovering an overwritten mix

### Waves From Samples – The Wave Extractor Tool

- Getting started
- Some behind-the-scenes info
- Basic Wave Extractor operations
- Using the Wave Extractor tool effectively
- Recovering an overwritten imported wave

### Draw Mode – Sketching Waves

#### Freehand

- A note of caution

### WaveCreate Post-processing Options

- The Crossfade Option (Xfade Depth)
- The Bit-Depth Reduction Option (Bit Depth)
- The Downsampling Option (Downsamp Fac)

### Auditioning Waves Via Sound Presets

- Techniques
- Special handling of the Wave Edit Buffer in presets
- Wave creation tools as performance controls

### Storing User Wavetables

- The basic steps
- Additional details
- A user wave location is not permanent storage!

### User Wave Sets

- Creating user wave sets
- No connection to sound presets!
- A user wave set is not permanent storage!

### Permanent Storage Of User Waves And Wave Sets

- User wave set files
- Saving the device
- Storage of waves within sound presets

# Introduction

## Preface

Welcome to the CreamWare Vectron!

As you may already know, the inspiration for the design of the Vectron was drawn in large part from a classic hardware synth from the early days of digital wavetable-based synthesis – namely, the Sequential Circuits Prophet VS. The Vectron encompasses essentially all of the capabilities of the original Prophet VS – it even incorporates a set of wavetables which are equivalent to the Prophet VS factory wavetable set. Thus, it can faithfully reproduce sounds produced by the Prophet VS.

However, the Vectron doesn't stop there. Its capabilities extend well beyond those of the original hardware instrument from years ago and include functions which would have been unrealizable at that time. Furthermore, thanks to its implementation as a virtual, software-based device which uses the resources of your computer to advantage, the

Vectron is also much more flexible and user-friendly friendly than typical synths from that era.

But enough of comparisons and historical background. The Vectron is an extremely powerful, versatile and above all *professional* synthesizer designed for today's electronic musician. The *vector synthesis* technique – dynamic mixing of wavetables – which is at its core also represents a departure from existing CreamWare synthesizers and ensures that the Vectron will bring fresh, new sounds to your tracks. Last but not least – the Vectron is a modulation *monster*. Its modulation possibilities alone are guaranteed to set your imagination on fire – and make the Vectron quite interesting for live performance as well as studio use.

## About The Manual

The Vectron is a deceptively deep, complex device. It incorporates a number of unusual functions, techniques and concepts. It may take a little extra time for you to grasp the essence of particular features and to understand how to work with them most effectively. However, the effort you invest in learning about the Vectron will be rewarded in equal measure. The more you realize what it is capable of, the wider is the spectrum of possibilities which it opens up to you.

The Vectron User's Manual has been designed with an eye toward making the process of getting acquainted with the Vectron as efficient and as stimulating as possible. Admittedly, it's not always light reading! Detailed explanations and extra background information have been provided where necessary to ensure that you have access to the knowledge you need to get the most out of each feature. This, combined with the sheer depth of features which the Vectron provides, has led unavoidably to the result that the Vectron manual is rather large.

Don't let the size of the manual worry you! Although every part of the manual is "recommended reading", we don't expect anyone to sit down and read the whole thing from start to finish. Instead, we've assumed the opposite – that you will read up on individual topics as your "need to know" dictates – and focused on making each section of the manual stand on its own, as far as possible. This means that more than the bare minimum amount of relevant information is included at each point. Rather than forcing you to constantly flip pages, we've even repeated information in some places – so the manual isn't really as large as it appears.

In order to support the "random access" approach, the manual is organized in large part according to the user interface structure of the Vectron itself. Individual chapters are devoted to each parameter page and to each additional major element of the user interface. Each chapter is likewise broken down into sections

corresponding to the layout of the associated portion of the device. The sections are in turn comprised of subsections which describe individual parameters or functions and are named accordingly. The detailed Table Of Contents mirrors this organization and lets you find your way quickly to the precise information you need, in addition to providing a compact overview of the topics covered in each chapter.

Nevertheless, we *do* recommend that you start by reading the brief *Overview Of The Vectron* chapter which immediately follows this *Introduction* before jumping in too deep. This will "raise your awareness" of the Vectron and provide you with the basic background you need in order to begin your explorations.



# Overview Of The Vectron

## Overview Of The Vectron Sound Architecture

### Vector synthesis

The Vectron produces its sounds using four wavetable oscillators per voice. Each of the four oscillators plays back a separately selectable static wavetable. The oscillator outputs are then mixed to create a monophonic signal (per voice)

which is passed through a more or less conventional synthesizer voicing structure (resonant low-pass filter, amplifier and pan control), followed by a stereo chorus and a stereo delay.

The mix of the four oscillators can be varied dynamically via a dedicated two-dimensional (XY or “vector”) multi-point envelope and a long list of other modulation sources, both internal and external. Furthermore, each of the four oscillators can be individually tuned and pitch-modulated (likewise via a long list of modulation sources). The dynamic mixing possibilities, along with the wavetables themselves, are the most significant distinguishing aspects of the Vectron sound.



*The outputs of the four wave oscillators are dynamically mixed using a vector technique.*

## About Vectron wavetable playback

Each Vectron wavetable consists of 128 16-bit words. The Vectron incorporates a set of permanent wavetables which are sonically equivalent to the factory wavetable set found in the Sequential Circuits Prophet VS, the synthesizer which was taken as the starting point for the Vectron design. In the Vectron, these wavetables are effectively 12-bit wavetables in terms of the data they contain, even though they're stored in a 16-bit format.

A significant feature of the Vectron is its ability to mimic the *non*-interpolated wavetable playback of the Prophet VS. Decreasing or eliminating interpolation adds a noticeable, characteristic and often musically interesting “edge” to the basic wavetable sound – in contrast to the “clean” interpolated playback which is standard in modern samplers.

As it turns out, with the Vectron (as with the original Prophet VS), less is more. The resulting roughness of the sound turns out to have a lot to do with its unique and interesting character. You can dial in a variable amount of interpolation and smooth out the sound by turning the appropriately-named **Grunge** control down –but most of the time, you probably won't want to.

One additional tip: to come as close as possible to the original “lo-fi” sound of the Prophet VS, set the sample rate of your system to 32kHz.



## Overview Of The Wavetables

Each Vectron sound is constructed around a selection of four wavetables (also referred to as *waves*) – one for each of the four oscillators **OscA**, **OscB**, **OscC**, and **OscD**. When creating or editing Vectron sounds, you can choose from roughly one hundred waves which are built permanently into the Vectron (the *fixed wavetable set*) as well as up to 128 waves in the *user wavetable list*, into which unlimited additional user wave sets can be loaded at any time.

Furthermore, each sound preset contains its own copies of the four waves it uses. Sound presets are thus completely independent of the fixed and user wavetable lists. They can always be loaded into the Vectron and played, even when the waves they use are not present in the fixed or user wavetable lists.

On the **ProgOsc** page (and the **WaveCreate** page), waves can be selected either by number or by name. The available selections (explained later in further detail) are the same for all four oscillators. These selections, as they

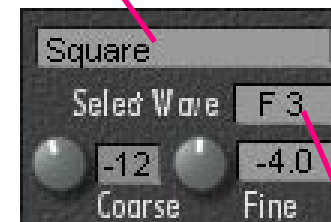
appear in the wave number field, and presented here in the same order in which they appear there, include:

- **WBuf**: The Wave Edit Buffer.
- **PreA .. PreD**: The waves contained in the *most recently loaded* sound preset for oscillators **OscA** .. **OscD** respectively.
- **Noise**: A white noise generator.
- **F 1 .. F 127**: Fixed wavetables.
- **U 0 .. U 127**: User wavetables.

The various wavetable categories are described in more detail in the following section.



Wavetable selection by name



Wavetable selection by number

## Wavetable Categories

### **WBuf: the Wave Edit Buffer**

The *Wave Edit Buffer* is the working wavetable space for the wave you are working on in **WaveCreate** page or for the last wave you extracted using the **Wave Extractor** tool. The *WBuf* selection is available in the **ProgOsc** page so that you can audition the wave-under-construction immediately within any desired existing preset, without the need to first store it into a user wave location.

However, in addition to this very practical option, it is also possible to use the **WaveCreate** page as an additional set of performance controls which give you the ability to modify a waveform in realtime via various techniques while playing it via a preset.

For this reason, it is also possible to create and store presets in which *WBuf* is selected as the source wave for one or more oscillators. However, it should be remembered that the contents of the Wave Edit Buffer are completely temporary. The next time you load such a pre-

set, it may sound substantially different than it did before, depending upon what happens to be in the Wave Edit Buffer at the moment.

Finally: as a form of “safety net” for wave editing, in the event that you save a preset using *WBuf* as a source wave but forget to also save the wave you were working on, the stored preset will contain a copy of whatever was in the Wave Edit Buffer at the time that the preset was stored, with the name *unidentified mix*. The next time you load this preset, this wave will thus appear as one (or more) of the *PreA .. PreD* selections and can be accessed accordingly.

### ***PreA .. PreD*: preset wavetables**

In order to guarantee that sound presets always load correctly and that they can easily be transferred intact from device to device, copies of each of the waves used in a sound preset are contained directly within the preset itself.

This is especially important for sound presets which use user wavetables, since user wavetables can be freely swapped in and out of the Vectron, and the user waves which a particular sound preset uses may or may not be present in the user wavetable list the next time the preset is loaded. This also ensures that sound presets which you give to or obtain from another Vectron user also include the correct waves (even if not accompanied by user wavetable presets).

After a sound preset is loaded, its four waves (and their respective names) are copied into temporary holding spaces. These are designated as *PreA .. PreD* and can subsequently be directly selected as the source waves for any of the oscillators (as well as in the **WaveCreate** page).

However, this aspect of presets is normally transparent, since the Vectron always attempts to locate matching fixed or user waves which are already present within the device when a sound preset is loaded, and automatically adjusts the **Select Wave** fields accordingly whenever it finds a match. (If a particular wave in a preset is one of the fixed waves, this matching attempt will *always* succeed.) Therefore, you will generally not see *PreA*, *PreB*, *PreC* or *PreD* as a **Select Wave** value immediately after loading a preset. This automatic association of preset waves with existing device waves helps to keep the usage of waves by presets from becoming confusing and unmanageable.

Nevertheless, these holding spaces for the preset waves provide distinct benefits:

- Regardless of any changes you make to the **Select Wave** fields in a sound preset after loading it, these holding spaces remain unaffected and provide a reliable reference point if you want to restore the wave selections to their original state and don't remember exactly what they were.
- Furthermore, you can use any or all of these waves directly in modified versions of a preset (which can then be stored without any other special action being necessary). The wave holding spaces are overwritten *only* when a new preset is loaded.
- Finally, you can access these waves directly in the **WaveCreate** page and, from there, store them into user wave locations. This provides a simple method of importing individual wavetables obtained from elsewhere in the form of sound presets. Once you've imported a wave in this fashion, you can use it in other sound presets, or to create completely new presets.

## Noise

The exception to the “static wavetable” rule – unlike all other **Select Wave** options, this setting introduces a true dynamic noise generator which takes the place of a static wavetable when used in a sound preset. Special feature: the noise generator responds to oscillator tuning adjustments! Check it out.

Naturally, it is not possible to use this selection in the **WaveCreate** page, where only static wavetables are applicable.

## F 1 .. F 127: the fixed wavetables

These wavetables are built directly into the device and are always available. It is not possible to delete them from the device.

Currently, only selections *F 1 .. F 94* and *F 127* are used. The fixed wavetable set may later be expanded to include some or all of selections *F 95 .. F 126*. The existing fixed wavetables are not likely to be changed or renumbered.

*F 127* contains a “silence” waveform (don’t laugh – this is also an entirely legitimate and useful option as an element in a dynamic wavetable mix). As such, it constitutes a special-purpose waveform and is for this reason placed at the very end of the space assigned to fixed wavetables (at the opposite end from the Noise selection, appropriately enough). Sound presets which use this wave will remain unaffected in the event that additional fixed wavetables are added into the *F 95 .. F 126* selection range.

### ***U 0 .. U 127*: user-created wavetables**

This set of wavetables is also referred to as the *User Wavetable List*. It consists of 128 locations which are empty by default. Wavetables you create via the **WaveCreate** page (or import via the **Wave Extractor** tool) can be stored in these locations.

A *user wavetable preset* stores the contents of all 128 of these locations. Correspondingly, when a user wavetable preset is loaded, it overwrites all 128 user wavetable locations.

## **Modulation options**

The Vectron features very extensive modulation options. Not only the vector mix, but most other adjustable parameters as well, can be modulated in real time. The complete list of modulation sources is a long one, and all of these sources are available at all times, without restrictions.

For virtually all modulatable parameters or modulation *destinations*, you can select the modulation signal from a list of sources. The modulation amount – as well as direction, in most cases – is separately adjustable for each modulation destination. Furthermore, many parameters can be modulated by two or more sources at the same time – and the LFOs can even modulate one another! Finally, even “dedicated” modulation sources such as the vector and filter envelopes can generally be used as modulation sources for other parameters, regardless of whether or not they’re also being used for their “intended” purposes.

## Types of modulation sources

### Dynamic vs. event-driven modulation sources

The Vectron features two basic types of modulation sources:

*Dynamic* sources include signals such as LFOs and envelopes. These sources are essentially waveform generators which automatically produce time-varying modulation signals. These signals are updated at the system sample rate and can therefore vary themselves quite rapidly – the multi-waveform LFOs can produce “clean” waveforms at up to 400 Hz! – and in quite complex ways. In this sense, they can be considered to be audio signals – hence, the designation *dynamic*.

*Event-driven* sources include all MIDI sources – not only modwheels and such, but also note velocities and note numbers – and all “manual” sources such as the graphical controller wheels and joysticks. In contrast to the dynamic sources, and despite all appearances to the contrary, these sources operate on the basis of individual events – either MIDI messages or your own movements while operating a graphical controller wheel or joystick controller onscreen. Each such event relays a new level value for a particular source – whose level then stays constant until the next event. These sources thus change their values only occasionally, unlike the dynamic sources, which change constantly.

It may appear that onscreen controllers such as graphical controller wheels and joysticks can be adjusted continuously. MIDI controllers are even referred to officially as “continuous controllers”. In fact, however, value changes for these sources are always conveyed via individual events. When you move a controller wheel, for example, you generate an event stream in which consecutive events may be separated by many milliseconds. The Vectron incorporates smoothing functions where necessary to keep the “stepped” character of these sources from becoming noticeable in the sound.

## Polyphonic vs. monophonic modulation sources

Vectron modulation sources can also be grouped into two categories according to a different criterion: whether the source is *polyphonic* or *monophonic*.

A *polyphonic* modulation source is available separately per voice. Such a source is thus actually a *set* of sources. Although it is referred to as a single source and is controlled by a single set of controls, there is actually an independent generator for each synthesizer voice (according to the setting of the **Voices** control in the sound preset list dialog). Thus, at any instant in time, the effect of such a source can be different in each note which is currently playing.

A *monophonic* source is a single signal which affects all voices in common. Such modulation sources are therefore also referred to as *global* modulation sources.

Both the dynamic and event-driven modulation source groups described above include polyphonic as well as monophonic signals.

For purposes of sound programming, the distinction between polyphonic and monophonic sources is the more important one, because it has more to do with what the actual *effect* of the source is. The categories are as follows:

*Polyphonic* sources include all envelopes, MIDI note velocity and MIDI note number. Also included in this group is the **Key Scaling Generator**, which is driven by MIDI note number, and the envelope **Speed Tracking** generators, which are driven by either note number or note-on velocity. These sources always produce a polyphonic modulation of whatever parameter they are applied to.

*Monophonic* sources include the LFOs, all graphical controllers, MIDI controller messages (modwheel, volume etc.), MIDI pressure (channel aftertouch) messages and MIDI pitch bend messages. These sources always produce a monophonic modulation of whatever parameter they are applied to.



# Vectron User Interface Overview

## The parameter pages

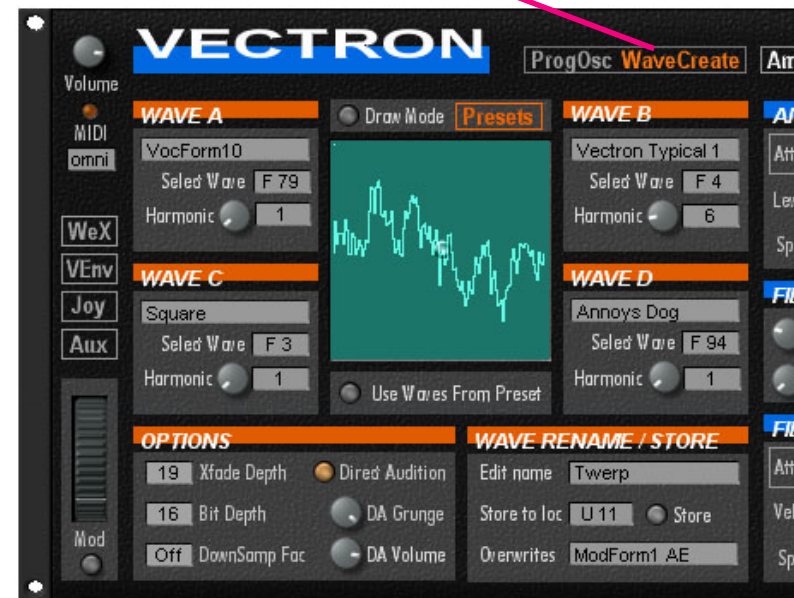
The Vectron interface or surface is divided into two main sections, each of which has two or more pages. Only one page is visible at any one time in each section. The pages are called up by clicking on the corresponding labels in the switch strips located above each section.

## The ProgOsc and WaveCreate pages

In the section at left, you can choose between the **ProgOsc** (program/oscillator) page and the **WaveCreate** page.

The **ProgOsc** page can be viewed as the “main” page for editing sound presets. All settings on this page (excepting **Vec Disp Mode**) are stored in sound presets.

*The surface consists of multiple pages - you can switch between them as needed via the Page Select buttons.*



The **WaveCreate** page provides access to the Wave Mix and Wave Drawing functions used to create and edit user wavetables. Additional audition and “post-processing” options on this page affect not only these wavetables, but also those imported from samples via the **Wave Extractor** tool. Functions for naming and storing new waves are also found on this page, as is the button for accessing the **User Wave Sets** dialog for creating wavetable presets. In contrast to the **ProgOsc** page, *none* of the settings on the **WaveCreate** page are stored in sound presets.

## Additional preset parameter pages

In the section at right, you can call up several additional pages containing a wealth of additional sound preset controls. With the exception of a few controls on the **Global** page, all of these settings are stored in sound presets. Each of these pages is discussed separately in its own chapter of the manual.

Amp/Filter LFO Vec/Pan Pitch Global

Vectron

## The graphical ModWheel

This functional equivalent of a “real” physical modwheel is located in the lower left corner of the main surface. It can be applied quite freely to modulation of Vectron sound parameters, and also features full MIDI functionality.

The characteristics and use of the graphical *ModWheel* are identical to those of the four additional graphical controller wheels on the **Aux Controllers** surface. Full details are provided in the *Aux Controllers* section of the chapter *The Auxiliary Surfaces*.



## Auxiliary surfaces

Via the set of switches at far left (above the ModWheel), you can open and close the **Wave Extractor** tool (**WeX**), the **Vector Envelope Edit** drawer (**VEEnv**), the graphical **Joystick Controller** (**Joy**) and the **Aux Controllers** surface (**Aux**). Each of these is discussed separately in a later section.



Contents

Index

16

# The ProgOsc Page

## OscA .. OscD – The Oscillators

### Select Wave

The **Select Wave** field for each oscillator on the **ProgOsc** page (for editing sound presets) and for each source wave on the **WaveCreate** page (for wave mixing) consists of two parts: a small text-fader field in which waves can be selected by number (according to the number ranges described in the section

*Wavetable Categories* above) and a wider name field which opens up into a scrollable name list when clicked. You can select the wave for each oscillator or mix source wave via either of these fields – the other field adjusts itself accordingly. Both fields offer alternative methods for selecting waves.



### The Wave Number (Text Fader) fields

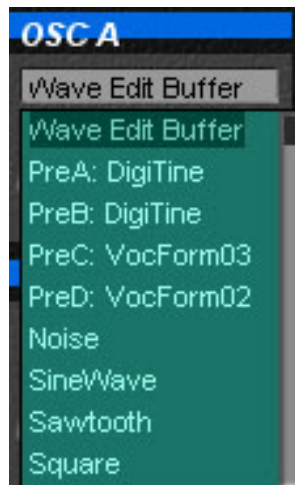
In addition to scrolling using the mouse (left-click and hold, then drag up/down), the wave number text fader field can also be scrolled via the PageUp and PageDown keys on your computer keyboard. You can use either of these techniques while playing a sound – each new wave selection will be heard immediately as soon as the selection changes. This is the easiest and fastest method for scrolling through the list when you're merely browsing and not trying to find a specific wave.

Selected Wave F 58

## The Wave Name (Combo Box) fields

The wider wave name field or “combo box” offers a few options for making selections via its drop-down name list:

- Left-click on the name field to open the name list, then navigate through the list using any combination of the scroll bar and the PageUp, PageDown and Up/Down (arrow) keys on your computer keyboard. Confirm your selection by clicking on it or by hitting the <Enter> key. Alternatively, cancel the selection process (and close the name list) by clicking anywhere outside the list or by hitting the <Esc> key.



- If you already know the name of the wave you want to select, left-click on the name field to open the name list and then begin typing the name on the keyboard. The name list springs to the first matching wave name in the list with each new character you type, so in many cases you need type only the first few. Confirm or cancel as described in the preceding paragraph.

Note that the wavetable name list is not sorted into alphabetical order, but corresponds to the numbering order of the wave number field. Therefore, typing the first few letters of a wave name will not always bring the name of the desired wave into view.

- You can also work the combo box like a text fader. Left-click *and hold* on the name field to open the name list, then drag up/down to select a wave name in the list, and finally release the mouse button to confirm the selection and close the name list. Alternatively, if you don't wish to confirm a selection, move the mouse pointer anywhere outside the list before releasing the mouse button. You can then continue using either of the previously described combo box techniques.

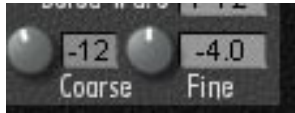
Note that combo box selections must be confirmed – they only become effective after you click them or hit the <Enter> key. For browsing the list “by ear”, the wave number field is a better option.

The wave selection list is quite long. It is often easiest to select waves using the keyboard options described above for the wave number (text fader) field instead of using the mouse or selecting waves via the wave name field.



## Oscillator tuning controls

The base oscillator tuning can be set separately for each of the four oscillators. Both coarse and fine tuning controls are provided for each oscillator.



### Coarse

Sets the tuning in semitone steps over the range -72 .. +72 semitones. The text field next to the knob is a text fader which can be scrolled using the mouse and which also accepts direct entry of values from the computer keyboard.

### Fine

Permits fine tuning over the range -100 .. +100 cents with a resolution of 0.1 cent. Values can be entered directly into the text field from the computer keyboard.

The extremely wide – almost exaggerated – tuning range of the Vectron not only offers great tonal flexibility, but also permits you to take advantage of certain effects which can be produced with wavetables:

- At the lower end of the range, certain wavetables – especially very overtone-rich ones – can reveal tonalities which are normally hidden from the human ear. Simpler waves, such as the square wave and sawtooth wave, degenerate into a rhythmic pulse train – while *Noise*, which also responds to the tuning controls, begins to resemble the sound of the lead-out groove on a phonograph record.
- At the upper end, *aliasing* can be deliberately introduced into the sound. This works especially well with wavetables which have a very broadly distributed overtone content. As the tuning is pushed upward past the limits of fidelity which are determined by sample rate, the overtone spectrum is gradually inverted. High-frequencies are transformed into low frequencies and become audible again. In digital audio systems, aliasing is normally something to be avoided at all costs. The Vectron turns aliasing into a musical tool which will raise the pulse of all true noise freaks! (Not to be used in the presence of dogs, cats and other household pets.)

## Mix Level

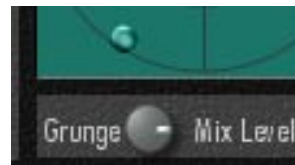
Adjusts the volume level of the sound preset. This setting is stored and recalled with each sound preset and can be used to balance the levels of various presets against one another.



By contrast, the **Volume** control in the upper left corner of the Vectron is a master volume control which sets the sound level for the entire device in all modes and is independent of sound presets.

## Grunge

Adjusts the amount of *interpolation* (waveform smoothing) performed by the Vectron during wavetable playback. Turning this control down *increases* the amount of interpolation, gradually eliminating the gritty edge (the “grunge”) from the sound.



High-quality interpolation is important in instruments such as samplers, where a high premium is placed upon achieving utmost fidelity in the reproduction of recorded sounds. With the Vectron, however, interpolation is somewhat less interesting. The Vectron is not a sampler, but a sound *generator*, and the “grunge” which results from reducing or altogether eliminating interpolation actually adds a great deal to the character of the sound, especially at the low end of the keyboard.

Therefore, in most of the sounds which come with the Vectron, you’ll find the **Grunge** control turned all the way up. By the way, the maximum-grunge setting corresponds to the characteristic playback performance of the Prophet VS and permits the Vectron to most closely emulate the sound of the Prophet VS.



## Portamento/Glissando (Porta/Gliss)

These are the controls for the portamento (smooth pitch-gliding from note to note) and glissando (pitch glide in semitone steps) functions of the Vectron.



### ON

Activates the portamento/glissando functions. The remaining controls in this section are effective only when this switch is on.

### Speed

Controls the speed of the glide. At high settings, the glide is quite rapid and may be noticeable only when notes which are separated by large note intervals are played. At the maximum setting, the glide essentially disappears – which can be useful for single-voice legato playing (see below).

## Porta / Gliss

Either portamento (smooth glide) or glissando (stepwise glide) can be selected.

### Norm / Finger (normal / fingered mode)

In *normal* mode, gliding occurs on every note played.

In *fingered* mode, the glide occurs only when a note is played *before* the previous key is released. This most is mainly appropriate for single-note playing.

## Porta/Gliss with one voice (legato)

When the **Voices** parameter of the Vectron (on the sound preset list dialog) is set to 1, the behavior of the portamento/glissando functions changes in a way which can be very useful for single-note soloing: namely, a *legato* mode is introduced.

In this mode, note attack is triggered only on the first note played. Amplifier, filter and vector envelopes are triggered only on this note. The same applies to LFO retriggering if enabled per note.

As long as you hold down the keys you play after playing them, any additional new key you play will merely change the pitch of the note by way of a portamento or glissando glide. When you release this key, the pitch will again glide back to the most recently played key which is still being held down. The glide can be turned into an instantaneous pitch change from one note to the next – useful for fast playing – by setting the **Speed** control to maximum.

## Chorus

The Vectron features a rich-sounding chorus/flanging effect with separate left and right delay lines and extensive programmability. The settings of all Chorus controls – including the on/off switch – are stored and recalled separately for each sound preset.

The Vectron chorus effect features separate left and right delay lines which are both modulated by the common LFO. However, modulation of the left delay line is phase-shifted with respect to that of the right delay line, resulting in a thicker and more complex sound.



### ON

Activates the chorus effect. The remaining in this section are effective only when this switch is on.

## LFO

Permits the choice between a sine wave or triangle wave LFO for modulation of the internal delay time under control of the **Speed** and **Depth** parameters.

### Offset

Adjusts the base delay time for both left and right delay lines. Since the base delay times at the lower end of the range reach down below 1 msec, the chorus effect can also function as a flanger.

### FB (feedback)

Controls the amount of feedback of the delay line output signals back to their inputs. The feedback can be either positive or negative and can be increased in either direction almost to the point where the chorus begins to self-oscillate.

### Speed

Sets the speed of the delay-modulation LFO.

## Depth

Sets the amount of delay-modulation produced by the LFO.

Note: When this control is set to minimum, the **Speed** control has no effect and the delay times of both left and right delay lines are identical.

### Level (effect level)

Controls the amount of effect signal which is mixed with the unprocessed (dry) input signal. The effect signal can be mixed in non-inverted (+) or inverted (-). The unprocessed input signal is always passed through the chorus effect at full level and is not affected by this control.

## Delay

The Vectron features a high-quality delay/echo effect with separate left and right delay lines and extensive programmability, including optional synchronization of delay times to a MIDI clock. The settings of all Delay controls – including the on/off switch – are stored and recalled separately for each sound preset.



### ON

Activates the delay effect. The remaining in this section are effective only when this switch is on.

## Del L and Del R

Delay time controls for the left and right delay lines. These can be adjusted fully independently of one another.

The manner in which these controls work depends upon whether the MIDI sync option is active:

- If the MIDI sync option is not active, delay times are set directly in terms of time. Delays can range from a minimum of 4 msec to over five seconds. Delay time settings (in milliseconds) can optionally be entered directly into the text fields.
- If the MIDI sync option is active, delay times are set in terms of note durations. These durations (1/1 = whole note, 1/2 = half-note, 1/4 = quarter-note, and so on) are based on the standard MIDI clock timing of 24 clocks per quarter note. The actual delay times are automatically adjusted according to the tempo of a MIDI clock signal in order to fit the specified note durations. This MIDI clock may come from an external source (via the Vectron's **MCik** input)

or from the Vectron's own internal tempo generator. Selection of an internal or external clock source as well as setting of the internal clock tempo is done in the **Sync** section of the **Global** page. Refer to the corresponding section of the manual for a full description of these settings. See also *A special note regarding external MIDI sync* below for details relevant to MIDI sync operation of the delay effect.

Note that glitching may occur in the delay effect sound if the delay time settings are changed while sounds are passing through the delay. This is normal. To avoid this, simply wait until the echoes die away before changing delay settings. Tip: temporarily set the feedback (**FB**) control to 0 (center).

## MIDI

Activates or deactivates the MIDI sync option (see **Del L** and **Del R** above).

## FB (feedback)

Controls the amount of feedback of the delay line output signals back to their inputs – and therefore, the number of audible echoes produced by the delay. The feedback can be either positive or negative and can be increased in either direction to the point where virtually infinite repeats are obtained.



## Cross (cross-coupled feedback)

Cross-couples the feedback connections of the left and right delay lines. When this option is switched off, the output of each delay line is fed back to its own input according to the setting of the feedback (**FB**) control. When the option is switched on, the output of the left delay line is fed to the input of the right delay line, and vice versa. This results in denser and more complex echoes which bounce from side to side in the stereo field.

## HDamp (high-frequency damping)

This control adjusts the cutoff frequency of a low-pass filter which is wired inline in the feedback path of each delay line. Turning this control up lowers the filter cutoff frequency, so that an increasing amount of the feedback signal's high end is “shaved off” on each pass through the delay line. This “rounds the edges” of the feedback signal, producing echoes which become progressively softer and more “muffled”.

Note that this setting affects *only* the feedback signal. The **HDamp** control will have no effect if the feedback (**FB**) control is set to 0 (center).

Conversely, turning **HDamp** all the way up will lower the cutoff frequency of the low-pass filters to the point where no signal gets through and feedback is effectively eliminated. In this case, the feedback (**FB**) control will appear to have no effect.

## Level (effect level)

Controls the amount of effect signal which is mixed with the unprocessed (dry) input signal. The unprocessed input signal is always passed through the delay effect at full level and is not affected by this control.

## A special note regarding external MIDI sync

When the MIDI sync option of the delay effect is active, the delay times adapt themselves automatically to the speed of the MIDI clock signal. To minimize the possibility of delay-line glitching resulting from the tracking of minor variations in the speed of the MIDI clock, the delay lines adjust themselves *only* when no notes are being played – i.e., when no keys are being held down on the keyboard.

At higher feedback (**FB**) settings, however, sounds may continue to circulate through the Delay well after all keys have been released, especially at longer delay time settings. Under these circum-

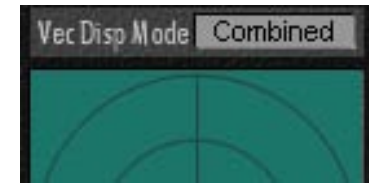
stances, glitching can be introduced into the delay effect output, and it becomes more noticeable the longer the sound is permitted to recirculate through the delay – the quality of the echoing sound will gradually deteriorate. For this reason, you may wish to avoid the combination of *external* MIDI sync, long delay times and high feedback settings.

An excellent alternative, when working with a non-varying tempo, is to switch the Vectron over to *internal* clock operation and set the desired tempo directly in the **Tempo** field of the **Sync** options section on the **Global** page. In contrast to an external MIDI clock signal, where minor variations are unavoidable, the internal timing of the Vectron is intrinsically fully synchronized, and the glitching problem described above does not arise.

Selection of an internal or external clock source as well as setting of the internal clock tempo is done in the **Sync** section of the **Global** page. Refer to the corresponding section of the manual for a full description of these settings.

## Vec Disp Mode (Vector Display mode)

This control (which is not stored in presets) sets the mode of the **Vector Display**. The **Vector Display** is described fully in the chapter *Vector Mixing Overview*.

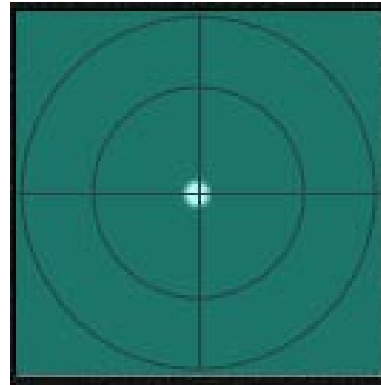


# Vector Mixing Overview

## The vector concept

The dynamic mix of the four active waves in a sound preset is responsible for much of the animated character and tonal complexity of a Vectron sound. To make it easier to work with this mix and to visualize the effects of its various components, the Vectron employs a “joystick” model as the basis of the mix.

This joystick can be visualized as moving freely within a square space. It sets the levels of all four waves according to a method which is intuitively easy to grasp. Each corner of the square is associated with one of the waves. The closer the “stick” is to a particular corner, the greater is the proportion of the associated wave in the mix. The total of all four wave levels always adds up to 100%, thus yielding optimal level at all times without overloading.



For example:

- When the stick is exactly in the center, all four waves are mixed at the same level (25%).
- If the stick is moved all the way into one corner, the wave associated with that corner constitutes 100% of the mix.
- If the stick is positioned along one edge of the square, the mix consists of only the two waves whose assigned corners are at the ends of that edge. The mix levels of these waves vary according to the position of the stick along this edge, but the sum of these *two* mix levels still always adds up to 100% – because the mix levels of the *other* two waves are both zero.

And so on.

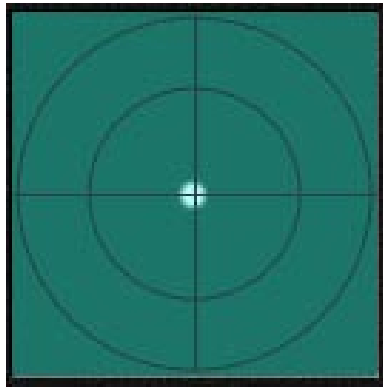
You can think of the Vectron as having an internal “master” joystick which directly controls the vector mix. The movements of this “master” joystick are influenced by a combination of “forces” which include the vector envelope and a variety of other modulation sources, both internal and external.

Although you can’t get at this “master” joystick directly, the **Vector Display** (described next) shows you its “movements” and makes it easy to understand the effect produced by each vector mix modulation source.



## The Vector Display

The bright “vectorscope beam” spot in the square XY display in the middle of the **ProgOsc** page provides a dynamic display of the vector mix coordinates of the note currently playing.



The **Vector Display** has some limitations:

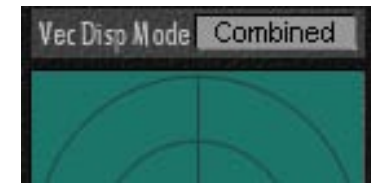
- The **Vector Display** is quite accurate. However, the actual vector mix can move *much* faster than the **Vector Display** is able to follow – literally at audio frequencies.
- The Vectron is polyphonic. The **Vector Display**, by contrast, can show the vector mix for only a single voice –

namely, voice #1. Normally, voice #1 gets used when the “first” note is played (i.e., following a period of silence), so the **Vector Display** *usually* provides a visible indication of the sound you’re hearing. However, this doesn’t necessarily occur with every note you play, even when you play single notes. Particularly with sounds in which the amplifier and/or filter envelope is programmed with a long release time, notes may continue to play well after you release a key. Therefore, you may actuate voices other than voice #1, and the **Vector Display** may appear not to be responding, or may appear to be out of sync with the sound. The same thing can happen if you play chords, or if you merely play extremely fast.

This is an inherent limitation of the **Vector Display** and is completely normal. The **Mono** option in the **Vector Envelope Edit** drawer (described in a later section) offers a method for keeping the **Vector Display** in sync while editing the vector envelope.

## Vec Disp Mode (Vector Display mode)

Via the **Vec Disp Mode** control directly above the **Vector Display**, the display supports three different viewing modes. Each has its own area of usefulness:



- Selecting *Combined* produces a display of the combined sum of the vector envelope, the AD-BC modulation signals and the output of the graphical **Joystick Controller**. In this view, both the vector envelope and the joystick signal have already been passed through their respective vector mix amount controls (on the **Vec/Pan** page) by the time they become part of this signal. Thus, the *Combined* setting shows you the final signal which actually drives the vector mix and determines what you hear. This is essentially the “default” viewing mode.

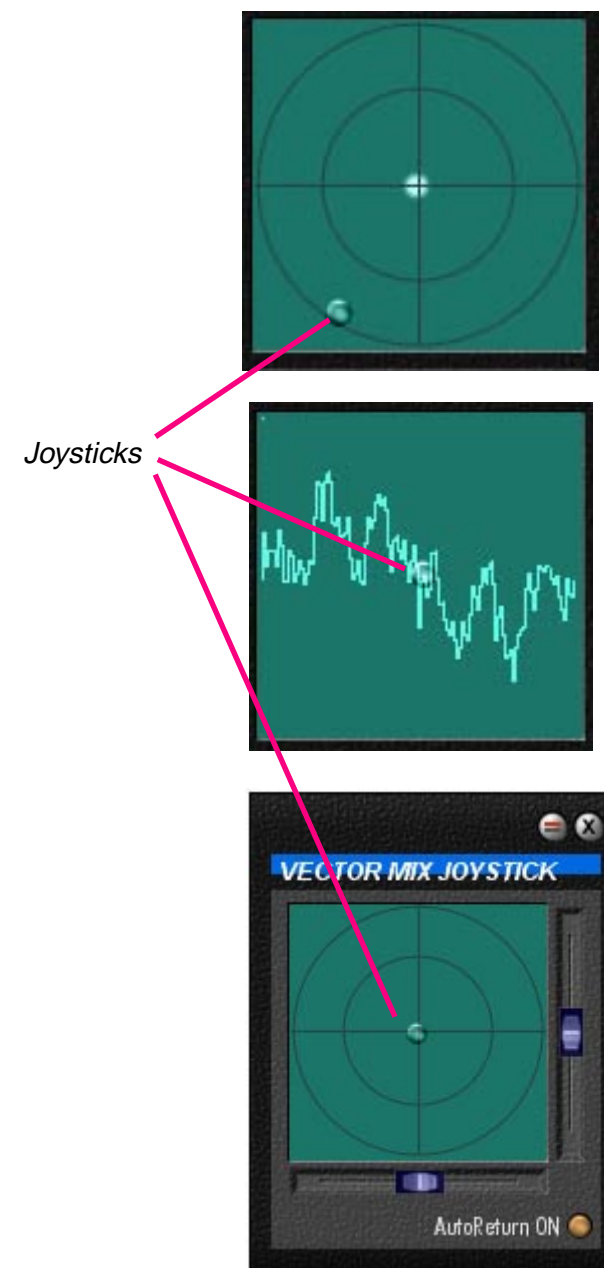
- Selecting *Vec Env* lets you see the vector envelope “solo”, before it is mixed with the other vector modulation sources. In this mode, the vector envelope is displayed uninverted and at full amplitude – the settings of the vector envelope mix amount controls (the **X** and **Y** controls in the **Vec Env** section on the **Vec/Pan** page) have no effect upon this display. Note that the **Vector Display** is forced into *Vec Env* mode whenever the **Vector Envelope Edit** drawer is open – the actual **Vec Disp Mode** setting is temporarily overridden to ensure that the display corresponds directly to the actual state of the vector envelope while you’re editing it.
- Selecting *AD-BC Mod* causes the AD-BC vector modulation signal (as derived via the associated settings on the **Vec/Pan** page) to be displayed “solo”.

## The graphical joystick controllers

The *Vectron* does include not just one, but three (count them!) separate graphical joystick controllers which you *can* operate directly. Each one is provided for a different purpose:

- One joystick appears overlaid upon the **Vector Display** when the **Vector Envelope Edit** drawer is open and is used to view and adjust the vector “positions” of individual points in the vector envelope.
- One joystick is provided on the **WaveCreate** page and is used for adjusting mix levels while creating new waves via wave mixing.
- One joystick – the **Joystick Controller** – is provided as a realtime controller which can be used to “play” the vector mix live, or as a general-purpose two-dimensional controller. It can send and respond to MIDI controller messages assigned to its X and Y axes.

The use of each of these joysticks is described in full detail in corresponding sections of the manual.



# The Vector Envelope

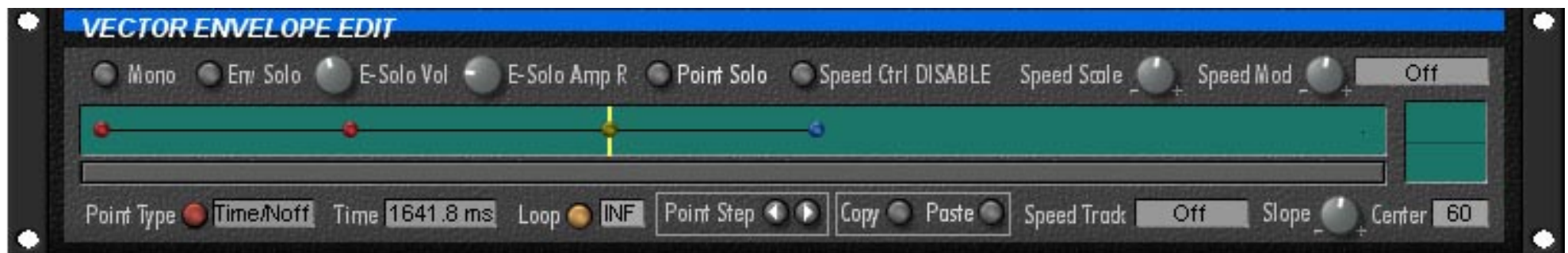
The vector envelope is a *multi-segment* envelope which can be programmed for anywhere from 1 to 64 segments. For each segment there is an adjustable envelope *point*. The shape of the envelope is formed by the sequence of envelope point values in “connect-the-dots” fashion – segments can be visualized as the lines connecting the points. The envelope glides smoothly from point to point over a period of time which is adjustable separately for each segment from a maximum of five seconds down to less than one millisecond.

You may already be familiar with multi-segment envelopes from other Pulsar devices (for example, the Common Pitch Envelope and Common Pan Envelope of

the FM One). The vector envelope differs from these other envelopes in one basic respect: each of its envelope points has not just a single level, but instead two: the X and Y coordinates which together specify a single vector point *position*.

As its name implies, the vector envelope is primarily intended as a modulation source for the vector mix in a sound preset. However, it is also provided in a few different forms as a general-purpose modulation source which can be applied to numerous other sound parameters. These parameters can thus be modulated in tight synchronization to a particular component of the vector mix.

Furthermore, via the vector envelope mix amount controls (the **X** and **Y** controls in the **Vec Env** section on the **Vec/Pan** page), the influence of the vector envelope on the vector mix can be reversed, reduced or entirely eliminated in the X axis, the Y axis or both axes at once. It can thus be used as a programmable complex modulation source (or possibly, *pair* of sources) which is completely independent of the vector mix – while the vector mix, in turn, can be driven by LFOs, the graphical **Joystick Controller** or various other modulation sources as desired.



# Vector Envelope Editing Basics

## The vector envelope multi-segment display

The vector envelope is edited by opening the **Vector Envelope Edit** drawer. This drawer drops down from the bottom of the main device surface when the **VEnv** button at far left is clicked.

The **Vector Envelope Edit** drawer can be opened only when the **ProgOsc** page is displayed. The drawer is closed automatically whenever the **WaveCreate** page is opened, and the **VEnv** switch is disabled as long as this page is displayed.

The multi-segment display which appears in the **Vector Envelope Edit** drawer shows the vector envelope in time overview. Via this display, you can add and remove envelope points, shift them around in time, and change their modes. You can also set up loops in the envelope.

All of the points in the Vectron's multi-segment display are located on a common horizontal line. This display is thus a simple timeline which indicates only the relative time duration of each envelope segment (by means of the horizontal spacing of the points) as well as the type of each segment (via the color of the associated point). The envelope point levels, which are actually two-dimensional *vector positions*, are displayed one point at a time superimposed upon the **Vector Display** in the center of the **ProgOsc** page, as described further below.



## Basic multi-segment display operations

The operations which are possible in the multi-segment display are described here. All vector envelope editing operations are done via this display, except for setting the vector positions of individual points.

### Selecting envelope points

Select an envelope point by single-clicking directly on it or near it.

### Deleting envelope points

Delete an envelope point by double-clicking directly on it or near it.

### Creating new envelope points

Add an envelope point by double-clicking on empty space in the multi-segment display at the *horizontal* position where you wish to add a point. You do not need to click directly on the timeline itself – above or below it is also okay.

To avoid accidentally deleting an existing point while trying to add a new point between two existing points which are close to one another, you may find it helpful to first expand the display scale as described below (see *Zooming and scrolling the display*). This spreads the points apart from each other in the display and makes it easier to double-click between them without “nailing” either of them.

## Changing segment duration

You can shift the position of a point in time – or, expressed in other words, change the duration of the associated envelope segment – by clicking and dragging the point. Note that the duration display is horizontally compressed per segment – that is, a large increase in segment duration appears as a relatively small increase in the graphical segment length. This makes it easier to display combinations of long and short segments within the same envelope.

In some cases, you may find it easier to enter time values directly from the computer keyboard. The segment duration for the selected point appears in the **Time** field below the multi-segment display (this is the time which separates this point from the preceding point). After selecting the desired point, you can click on this field and type in a new time value for it. The value should be in milliseconds and can be anywhere in the range *0.1 msec .. 5000.0 msec*.



## Changing point mode

You can change the mode of the selected point by repeatedly clicking on the **Point Type** button. There are three possible modes, each of which has an associated color:



- **RED** – *Time/Noff* (time / note-off): As long as the key which played this note is still being held down (i.e., a matching note-off has not yet been received), the associated segment lasts for the amount of time specified for this point, after which the next segment begins. If the key is released during this type of segment, the envelope skips immediately to the next segment associated with a **BLUE Time** point.
- **BLUE** – *Time*: The associated segment lasts for the amount of time specified for this point, after which the next segment begins. This type of point always runs its full course and is not affected by note-off events.
- **YELLOW** – *Sustain*: Used to create a sustain point in the envelope. This is additionally indicated by a vertical yellow line cutting through the point.

As long as the key which played this note is still being held down (i.e., a matching note-off has not yet been received), the associated segment lasts for the amount of time specified for this point – the envelope then pauses indefinitely at this point. If the key is released during this type of segment, the envelope skips immediately to the next segment associated with a **BLUE Time** point.

Typically, when a *Sustain* point is used in the vector envelope, it is preceded by *Time/Noff* points and followed by *Time* points.

## Envelope loops

Envelope loop operations are discussed below, in the section *Vector Envelope Programming Tips* (see the items *Creating a loop in the envelope*, *Deleting an envelope loop* and *Adjusting loop repeat count*).



Zoom/scroll bar

## Zooming and scrolling the display

The time scale of the multi-segment display can be changed by clicking and dragging on the extreme left or right end of the zoom/scroll bar directly below the display. This changes the size of the zoom/scroll bar. The new size of the bar relative to the “rail” in which it slides indicates the proportion of the full envelope display which is currently visible.

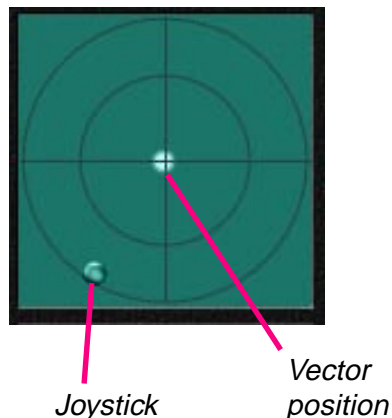
When the display is zoomed in, the zoom/scroll bar can be dragged left and right to view different parts of the timeline – click anywhere *except* at the extreme ends of the bar.

You can switch quickly back and forth between a fully zoomed-out view and a particular zoomed-in view by repeatedly double-clicking the zoom/scroll bar.



## Display and editing of point vector positions

The envelope point vector positions are displayed in the XY (vector) display in the center of the **ProgOsc** page. A round “joystick handle tip” appears superimposed over this display when the **Vector Envelope Edit** drawer is open. The position of this “stick” in the XY display shows the vector position of the *currently selected* envelope point – namely, the point which is highlighted in the multi-segment display in the **Vector Envelope Edit** drawer. To edit the vector position for this point, simply click and hold on the stick and drag it around in the XY space as desired.



## Vector Envelope Editing Convenience Features

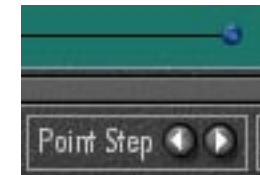
A variety of features and options have been included to make the process of vector envelope editing more effective. These are described in this section. Note that the switchable editing options described here are effective *only* while the **Vector Envelope Edit** drawer is open. However, any or all of these options can be left switched on if desired – they will reactivate themselves automatically every time the drawer is reopened. As a rule, these option settings are *not* stored in sound presets.

### The Point Step buttons

The “stick” springs from place to place in the **Vector Display** to indicate the vector position of the envelope point currently selected in the multi-segment display. Selection of a specific point in the multi-segment display is most directly done by simply clicking on it. However, this is not always the easiest or most useful method. Direct clicking can become difficult when an envelope contains many points. In addition, it is often useful

to step through the points in sequence, allowing you view the succession of positions in the order in which they will occur during playback (as well as hear them in this order – more below).

The **Point Step** buttons enable you to easily move point-by-point in either direction. These buttons select the point directly to the left or right of the currently selected point. If no point is currently selected, the selection “jumps in” at one end of the envelope or the other – at far left, if you’re stepping to the right, and vice versa. Particularly when a lot of envelope points are crowded close together in the multi-segment display, this is often the most effective method for orienting yourself within the envelope or locating a specific desired point.



## Mono mode

This setting is not stored in presets.

Activating this option forces the Vectron temporarily into single-voice mode. The effect is equivalent to that produced by setting the **Voices** parameter (in the device **Preset List** dialog) to 1.

Use of this option can make it much easier to hear the effects of your vector envelope edits, particularly when the sound you're working with has a long amp envelope release time. When this option is active, only one voice is available at all times. Therefore, this voice – voice #1 – is “recycled” *every* time you play a new note and is *always* the only voice playing. This effectively prevents polyphonic “audio clutter” while editing the envelope and guarantees that you clearly hear the complete vector envelope from the beginning on each new note, so you can easily evaluate the effect of every change you make to the envelope.

Perhaps just as importantly – and not only when long amp envelope release times are involved – single-voice mode guarantees that the **Vector Display** is always synchronized with the notes you play, since voice #1 is used for every note. This can be very effective in helping you to audibly correlate the sound of the vector envelope at various stages with individual envelope points by observing the path of the display.

Since the **Mono** option, when enabled, automatically activates and deactivates itself each time you open the **Vector Envelope Edit** drawer, it can slow things down a bit whenever you do this, while Pulsar loads and unloads the additional Vectron voices according to your current **Voices** setting. When doing intensive editing of a vector envelope, you may find it preferable to simply temporarily dial the device polyphony down to 1.



## Point Solo mode

This setting is not stored in presets.



Activating this option effectively “freezes” the vector envelope at the currently selected envelope point – and simultaneously disables all other vector mix modulation sources. Therefore, you hear a static mix corresponding to the selected point. With **Point Solo** active, you can continue to select envelope points as desired. This provides an easy and quick method of locating individual envelope points by ear. **Point Solo** is therefore very useful in conjunction with the **Point Step** buttons described above.

Note: whenever you edit the position of a vector envelope point by clicking and dragging the “stick” in the XY display, **Point Solo** mode is temporarily automatically activated – even if not already acti-

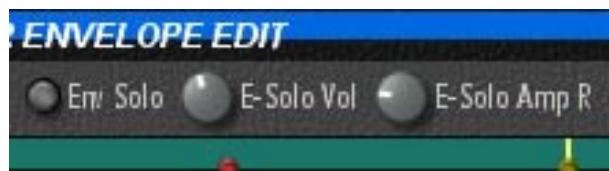
vated via the **Point Solo** button – so that you can clearly hear the vector mix you’re setting for that point. This *temporary* point solo lasts for as long as you hold the mouse button.

**Point Solo** can be used in combination with **Env Solo** (see below) or independently.

## Env Solo mode

This setting is not stored in presets.

Activating this option makes it easier to directly hear (and see) the vector envelope. It disables all other vector mix modulation sources – and simultaneously disables or bypasses an assortment of other device functions which could obscure the effects of the vector envelope, including: pan modulation, the low-pass filter, amplifier modulation, the chorus and the delay.



In addition, the settings of the vector envelope mix amount controls (the **X** and **Y** controls in the **Vec Env** section on the **Vec/Pan** page) are temporarily overridden, so that the envelope is delivered at full level and without inversion of either the X or Y component. Correspondingly, to prevent visual confusion, the **Vector Display** is forced to display the pure vector envelope (without the effects of other vector mix modulation sources), regardless of the current **Vec Disp Mode** setting.

Finally, the **E-Solo Vol** and **E-Solo Amp R** controls become active in **Env Solo** mode:

- **E-Solo Vol** provides an alternate volume control which is effective only in **Env Solo** mode and which is not affected by preset recall. With some presets, disabling of amplifier modulation and the low-pass filter can result in substantial volume changes – usually louder! – and the unfiltered waves can even become a bit irritating to the ear after a long enough time. The **E-Solo Vol** control permits adjustment of a separate volume level for **Env Solo** mode.

- The **E-Solo Amp R** control is the release-time control for the *default* amplifier envelope which is effective in **Env Solo** mode (attack/decay time are fixed at minimum, sustain level at maximum). The **E-Solo Amp R** control automatically takes on the setting of the amplifier **Rel** control in the current preset each time the **Vector Envelope Edit** drawer is opened. In presets which feature a long amplifier release time, this long release time is thus automatically carried over into **Env Solo** mode, so that the portion of the vector envelope which occurs after key release continues to be heard normally and can be conveniently edited if desired. Once the **Vector Envelope Edit** drawer is open, this control can be adjusted as desired. The **E-Solo Amp R** setting is not stored in presets. However, this control does take on the amplifier **Rel** setting of presets which are recalled while the **Vector Envelope Edit** drawer is open.

**Env Solo** can be used in combination with **Point Solo** (see above) or independently.

## Speed Ctrl DISABLE

This setting is not stored in presets.

This switch permits disabling of all vector envelope speed controls – the controls located to the right of this switch, both above and below the multi-segment display – without the need to change their settings.

Disabling of vector envelope speed controls (whose settings *are* stored in presets) while editing the vector envelope itself is highly recommended. It guarantees that “what you see is what you get” – the displayed segment times and sizes will correspond directly to what you’re actually hearing. More importantly: disabling all speed modulation sources eliminates a potential source of (possibly extreme!) confusion while editing the envelope.



## Copy / Paste

Use these buttons to transfer complete vector envelopes from one preset to another. Click **Copy** to capture a copy of the current preset’s vector envelope in a special envelope “clipboard”. Then load another preset and click **Paste** to transfer the captured envelope into the new preset.

Only the envelope itself is copied and pasted. This includes segment times, point positions and segment modes (including loop settings). Envelope speed control settings (see below) are not transferred, nor are the envelope edit option settings affected.

## The Vector Envelope Speed Controls

The vector envelope features a full set of envelope speed modulation controls similar to those provided for the amp and filter envelopes. To learn how these controls work, refer to the *Speed Mod* and *Speed Track – Center – Slope* sections of the *Filter Envelope* description (in the chapter *The Amp/Filter Page*).

Unique to the vector envelope is the **Speed Scale** control. This control permits the entire vector envelope to be sped up or slowed down independently of modulation sources and is useful for tweaking the overall envelope speed without altering the basic “rhythm” of the existing envelope.

When working with the speed controls, keep in mind that the vector envelope, like the amp and filter envelopes – but *unlike* the LFOs – does not respond continuously to speed control changes, but only whenever it makes a transition from one envelope segment to the next. Therefore, if you at any time slow the envelope down radically, it may take a while before the next speed change “registers”.

All speed control settings are stored in sound presets. This includes **Speed Scale**, **Speed Mod** and **Speed Track** including **Slope** and **Center**. By contrast, the setting of the **Speed Ctrl DISABLE** switch (discussed above) is *not* stored in presets and this switch is effective only while the **Vector Envelope Edit** drawer is open.

## The Vector Envelope Mix Amount Controls

The vector envelope is not “hard-wired” to the vector mix. Instead, its effect upon the vector mix is regulated via the **X** and **Y** controls in the **Vec Env** section on the **Vec/Pan** page.

By means of these controls, the influence of the vector envelope on the vector mix can be reversed, reduced or entirely eliminated in the X axis, the Y axis or both axes at once. The vector envelope can thus be used as a programmable complex modulation source (or possibly, *pair* of sources) which is completely independent of the vector mix – while the vector

mix, in turn, can be driven by LFOs, the graphical **Joystick Controller** or various other modulation sources as desired.

If it appears that your changes to the vector envelope are having no effect upon the vector mix, or if the vector envelope seems to produce only vertical or horizontal movements in the **Vector Display**, check the settings of the **X** and **Y** controls in the **Vec Env** section. For normal routing of the vector envelope to the vector mix, both controls should be set fully clockwise.

For a complete description of these controls, refer to the *Vec Env* section in the chapter *The Vec/Pan Page*. This chapter also includes descriptions of the other available vector mix modulation sources.





## The Vector Envelope As Modulation Source

Although it is specially equipped to serve as the primary vector mix modulation source, the vector envelope is also entirely suitable for use as a general purpose modulation source for sound parameters having no connection to the vector mix. Correspondingly, it has been made available – in component form – as a selectable modulation source at many other points in the Vectron. This aspect of the vector envelope is described here.

### With or without vector mix modulation

As mentioned in the preceding section (*The Vector Envelope Mix Amount Controls*), the effect of the vector envelope upon the vector mix is not fixed, but is determined by the settings of the **X** and **Y** controls in the **Vec Env** section on the **Vec/Pan** page. Setting both of these controls to zero eliminates the effect of the vector envelope upon the vector mix. The vector envelope can thus be used as a programmable complex modulation

source (or possibly, *pair* of sources) which is completely independent of the vector mix.

However, doing so is entirely optional. The vector envelope is available *at all times* for use as a general-purpose modulation source, regardless of the settings of the **X** and **Y** controls.

### Vector envelope component signals

Because the vector envelope is in fact a two-dimensional envelope, it cannot be used directly as a general-purpose modulation source – apart from the vector mix itself, all other modulation destinations in the Vectron are one-dimensional (or for you propellerheads out there: *scalar* instead of *vector*). Instead, therefore, various one-dimensional *component* forms of the vector envelope are provided for this use.

### About signal value ranges

As you read the following descriptions of the component signals, keep in mind that all of these signals have a *bipolar* range – they can potentially vary over the full range of negative as well as positive values. In practical terms, this means that they can potentially produce modulation effects in both directions.

Whether these signals actually *are* bipolar in any particular case depends, of course, on how *you* program the vector envelope. With the vector envelope – in contrast to the LFOs, for example – you have direct, point-for-point control over the actual envelope values which occur. You can use the full value range, or restrict the range in any way you wish. Just remember – whenever you set the vector position of a point, you're setting both its **X** and **Y** values at the same time.

As a point of reference, the exact center point of the **Vector Display** (and of the vector coordinate space itself) corresponds to zero in all directions – not only vertical and horizontal but *diagonal* as well.



## ***Vec Env X* and *Vec Env Y***

These signals are simply the horizontal and vertical components, respectively, of the vector envelope (with respect to the vector envelope as it appears in the **Vector Display**).

For the *Vec Env X* component signal, values right of center are positive and those left of center are negative. For the *Vec Env Y* component signal, positive and negative values are above and below center, respectively.

Because the *Vec Env X* and *Vec Env Y* component signals are relatively easy to grasp conceptually and also relatively easy to program directly – and furthermore because each of them can be individually “dialed out” of the vector mix (see the above section *With or without vector mix modulation*) – these signals lend themselves most easily to use as modulation sources which are independent of the vector mix.

Naturally, there’s no requirement that both of these signals always be applied. It’s always possible to use – and therefore, to program your envelope points for – the *Vec Env X* or *Vec Env Y* component signal alone, if one of them is all you need.

## ***Vec Env AD* and *Vec Env BC***

These signals are the *diagonal* components of the vector envelope as measured along the lines which connect opposite corners of the **Vector Display**.

You can visualize what this means most easily by imagining that the X and Y (horizontal and vertical) *measurement* axes are both rotated counterclockwise by 45 degrees. The zero point in both dimensions is still represented by the exact center of the **Vector Display**. However, the rotated Y axis now cuts through the **OscA** and **OscD** corners and is therefore called the *AD* axis, while the rotated X axis cuts through the other two corners and is correspondingly called the *BC* axis.

The value of the *Vec Env AD* component signal at any point in time is nothing more than the instantaneous “position” of the vector envelope as projected (diagonally, that is!) onto the AD diagonal axis. The maximum positive value is obtained when the envelope wanders completely into the **OscA** corner, while the maximum negative value is correspondingly found in the **OscD** corner. Likewise, the maximum positive and negative values for the *Vec Env BC* component signal are found in the **OscB** and **OscC** corners, respec-

tively. Thus, this is just an alternative way of looking at the same vector envelope.

What does all of this mean in practical terms? Simply this: you can use these modulation signals to modulate a Vectron sound parameter in sync with the vector envelope in such a way that the modulation tracks the loudness of a particular oscillator. For example, if you apply the *Vec Env AD* signal to filter cutoff with a positive modulation amount setting, then the filter will “open up” more and more as the vector envelope causes **OscA** to become louder. If you instead use a *negative* modulation amount setting, the filter will open up further as **OscD** becomes louder – and so on.

Therefore, it makes sense to use the *Vec Env AD* and *Vec Env BC* signals when the vector envelope *is* being used to modulate the vector mix – i.e., when it has not been “disconnected” from the vector mix by means of the **X** and **Y** controls in the **Vec Env** section on the **Vec/Pan** page (see “The Vector Envelope Mix Amount Controls” above). However, as with virtually everything else in the Vectron, this is merely an option and not a rule which must be consistently applied.

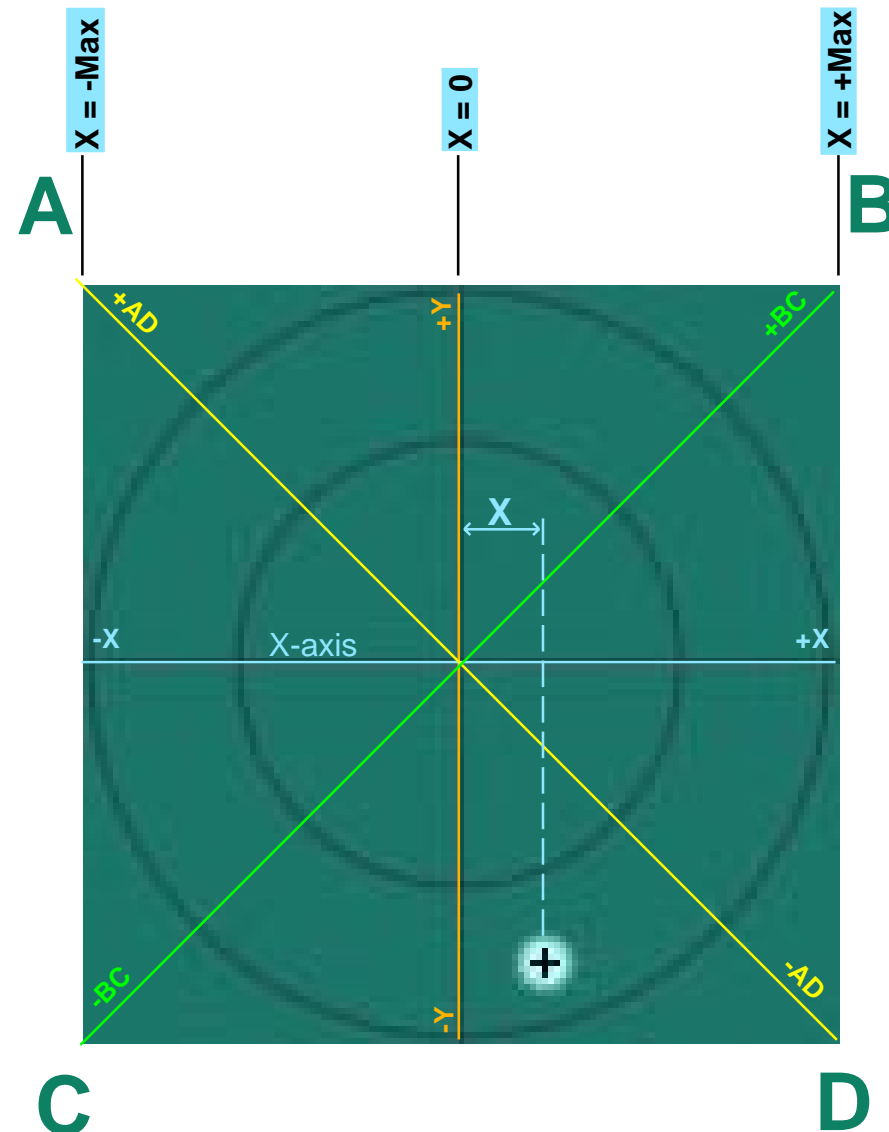
## Component signal diagrams

To help clarify the concept of vector envelope component signals, the following few pages present diagrams depicting the various component signal values for a single vector envelope (or joystick) point position. The same point position is used in all diagrams to emphasize the concept that all of the component values are just different ways of measuring the same signal.

At right, the X component signal value is illustrated. This is the displacement of the point from the (vertical)  $X = 0$  line as measured along the X-axis (blue).

(Note that the  $X = 0$  line (orange) is labelled "-Y .. +Y", since it is actually also the Y-axis.)

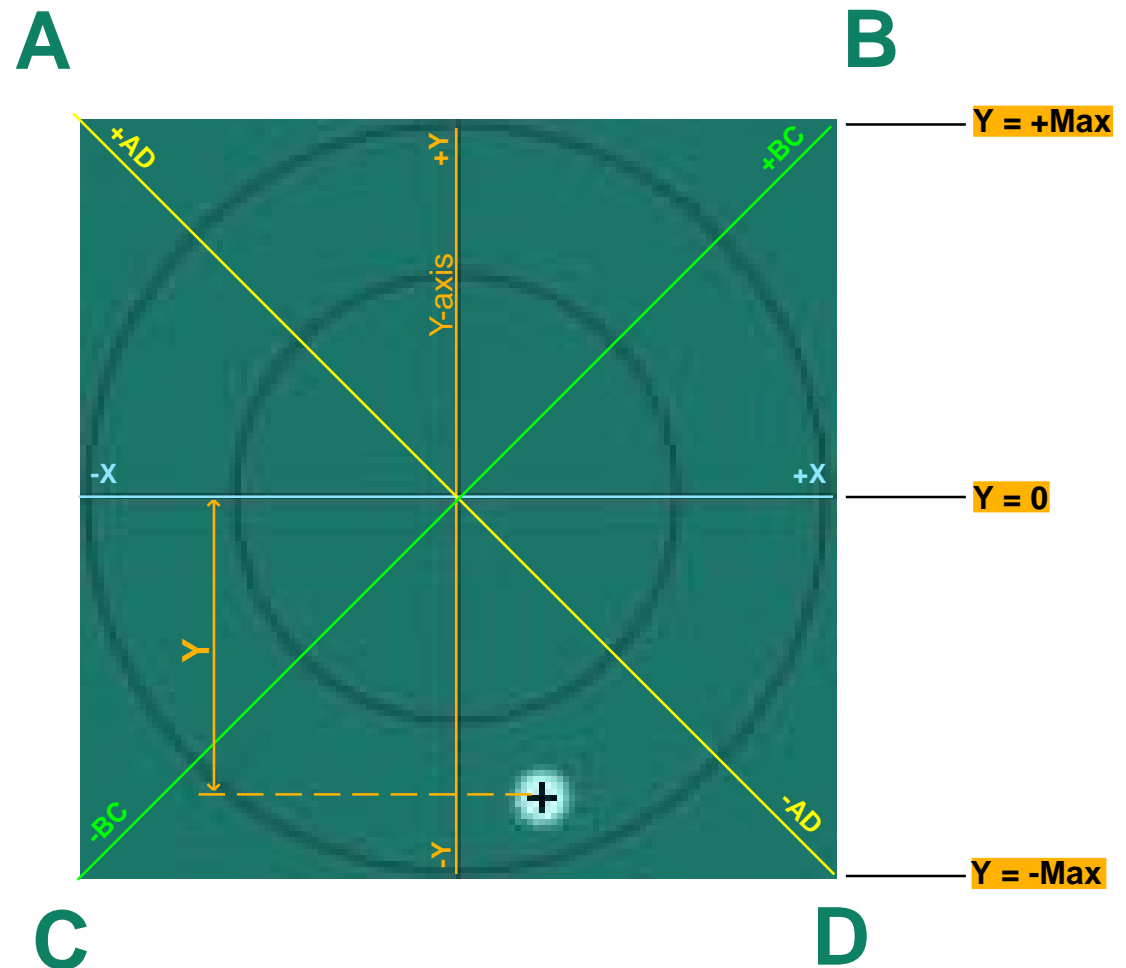
The X component value is indicated by the double-ended arrow labelled "X". In this example, the value is positive, since the displacement is in the +X direction, and is roughly 20% of the maximum positive value for the X component signal.



At right, the Y component signal value is illustrated. This is the displacement of the point from the (horizontal)  $Y = 0$  line as measured along the Y-axis (orange).

(Note that the  $Y = 0$  line (blue) is labelled "-X .. +X", since it is actually also the X-axis.)

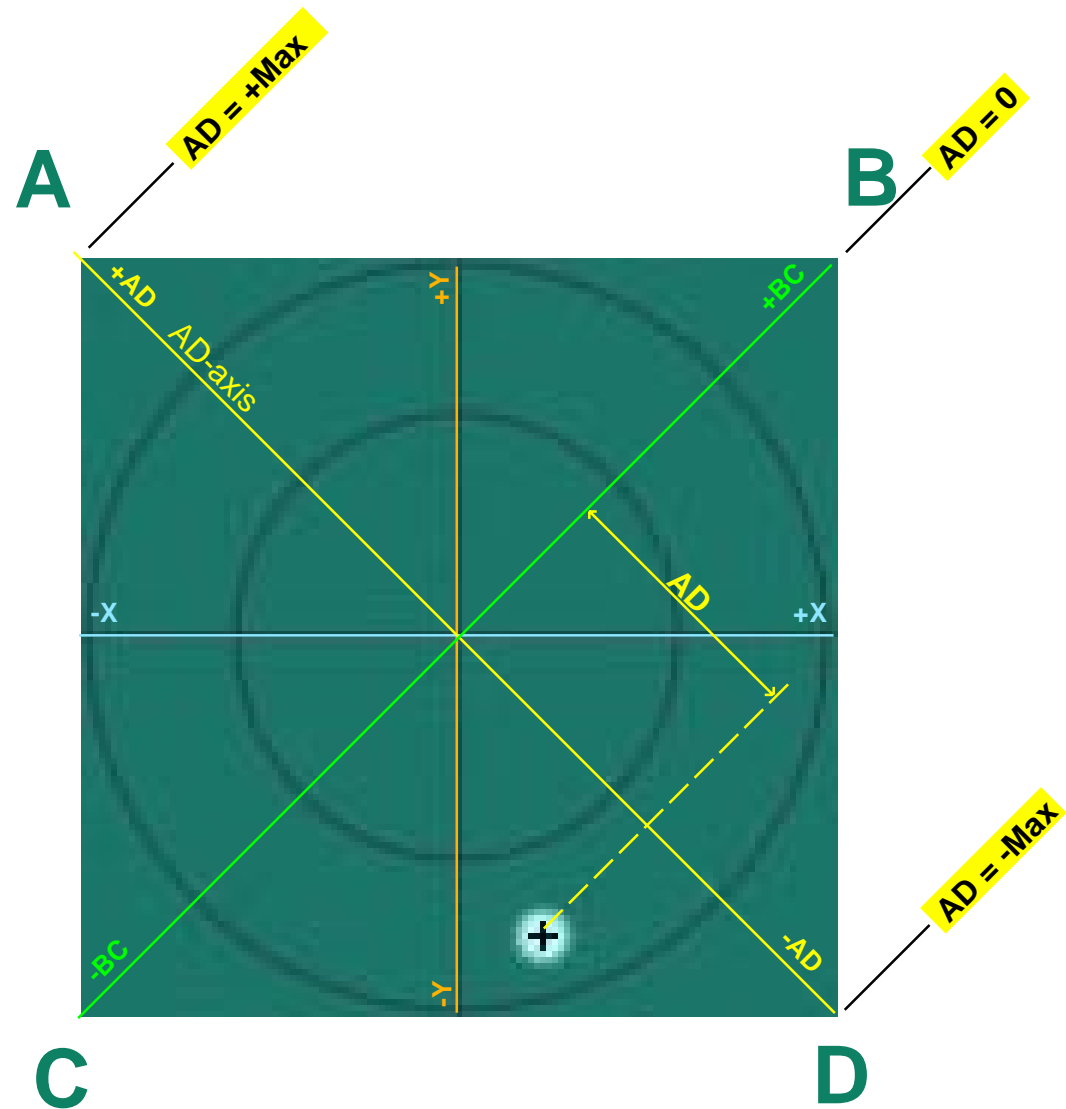
The Y component value is indicated by the double-ended arrow labelled "Y". In this example, the value is negative, since the displacement is in the -Y direction, and is roughly 80% of the maximum negative value for the Y component signal.



At right, the AD component signal value is illustrated. This is the displacement of the point from the (diagonal)  $AD = 0$  line as measured along the AD-axis (yellow).

(Note that the  $AD = 0$  line (green) is labelled "-BC .. +BC", since it is actually also the BC-axis.)

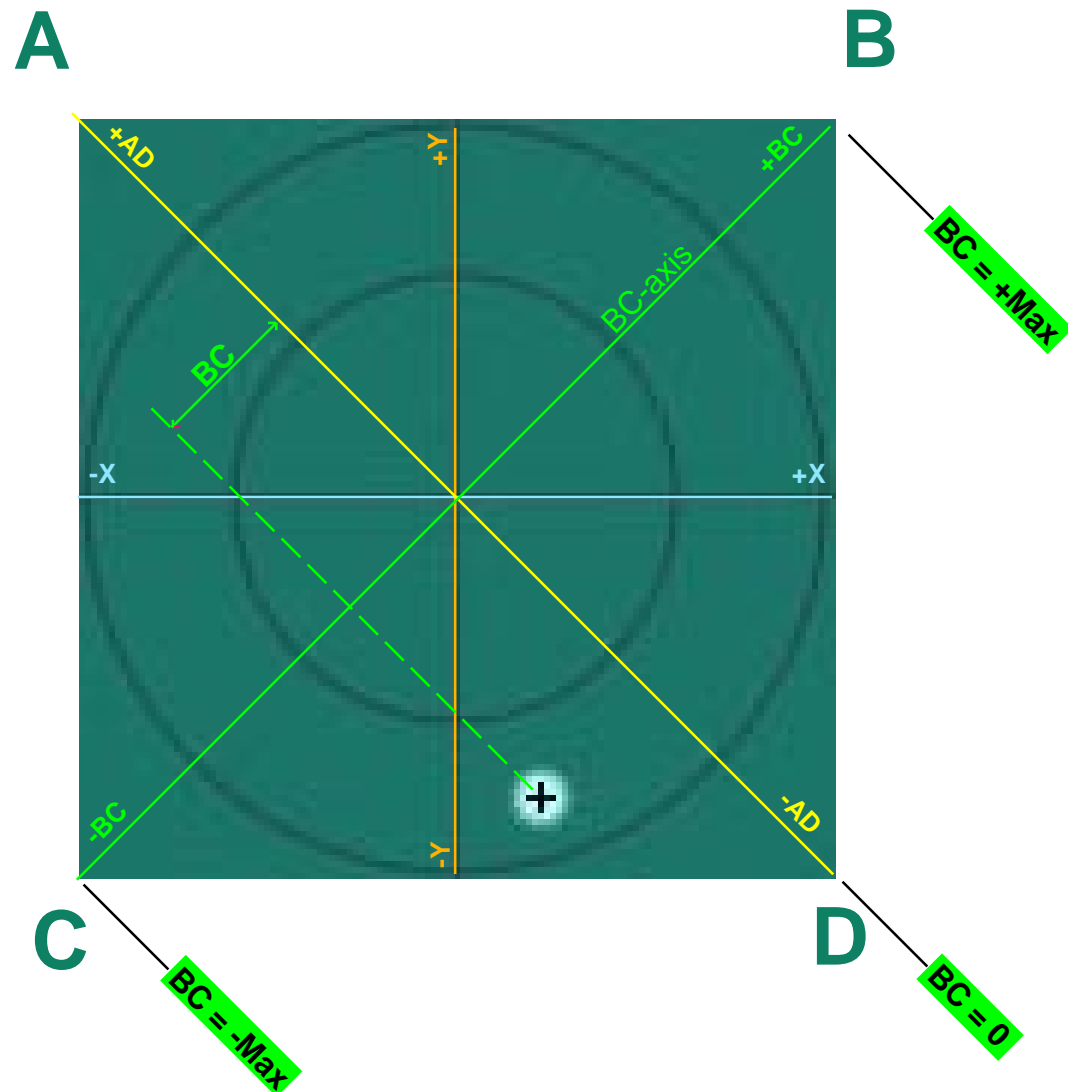
The AD component value is indicated by the double-ended arrow labelled "AD". In this example, the value is negative, since the displacement is in the -AD direction, and is roughly 50% of the maximum signal.



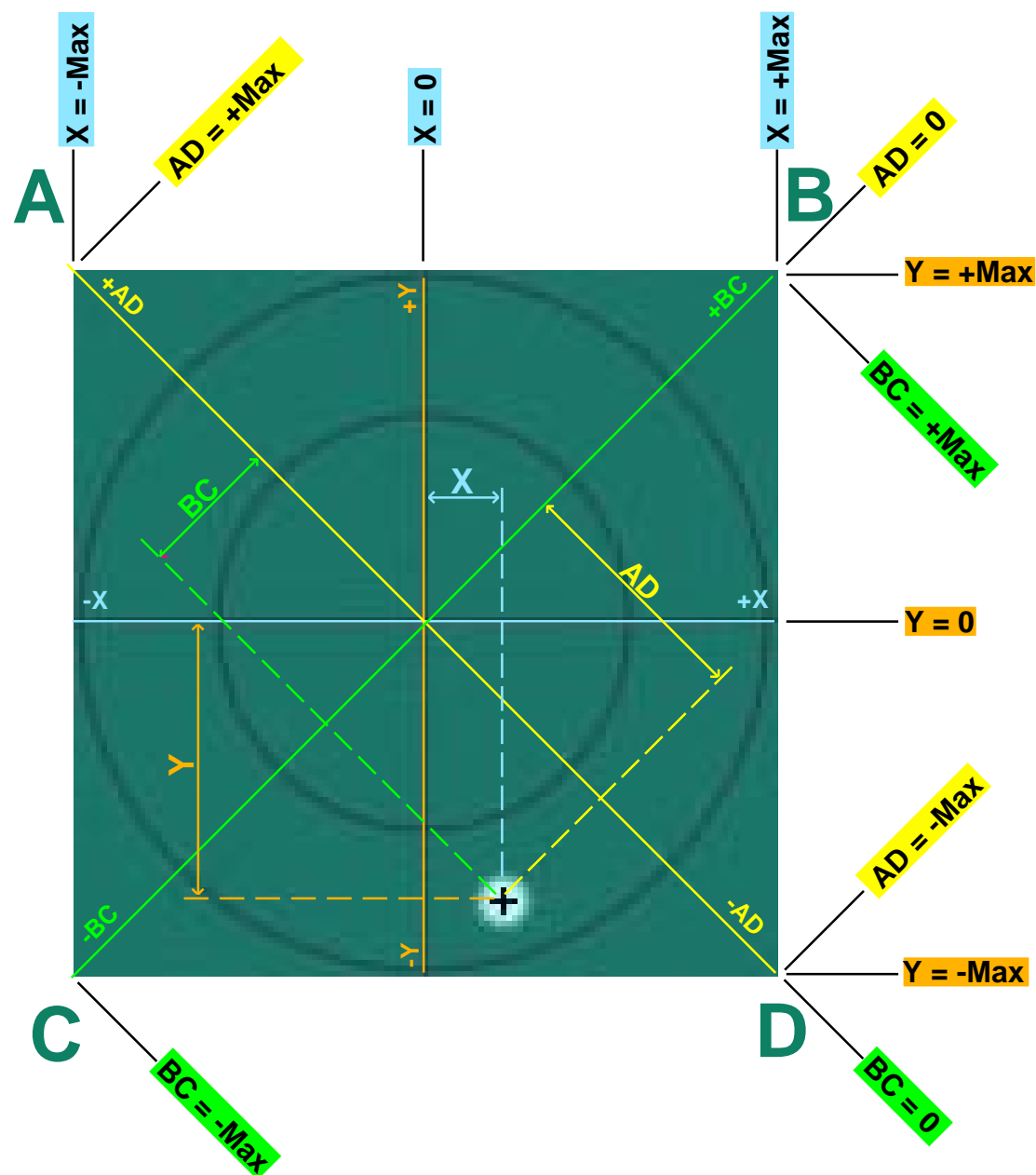
At right, the BC component signal value is illustrated. This is the displacement of the point from the (diagonal)  $BC = 0$  line as measured along the BC-axis (green).

(Note that the  $BC = 0$  line (yellow) is labelled "-AD .. +AD", since it is actually also the AD-axis.)

The BC component value is indicated by the double-ended arrow labelled "BC". In this example, the value is negative, since the displacement is in the -BC direction, and is roughly 30% of the maximum negative value for the BC component signal.



The somewhat complicated-looking diagram at right is merely a combination of the diagrams from the four preceding pages and graphically illustrates how four different modulation signals can be derived from a single vector envelope or joystick.





## Special capabilities as a modulation source

Among other possibilities, its wide operating speed range and ability to execute loops allow the vector envelope to be used as a point-by-point *programmable*, complex-waveform modulation generator or LFO.

In this application, the vector envelope even offers a sound-design advantage – or at least a distinction – as compared to the “real” LFOs in the Vectron: while the Vectron’s LFOs are monophonic, the vector envelope is polyphonic. It is separately generated and triggered *per note* – and its speed can correspondingly be modulated independently per note in response to note-on velocity or note-number. It can therefore be used to produce stunningly complex modulation textures in polyphonic performances in which the LFOs, if applied to the same modulation destinations – for example, pan position – would modulate all notes in lockstep.

Of course, it’s not necessary to set up loops, program complex waveforms or apply polyphonic speed modulations to the vector envelope in order to make good use of it as a general-purpose modulation source – but the possibilities are there.

Furthermore, the vector envelope is – of course – two-dimensional. It can therefore be used to create separate, but *coupled* modulations of separate parameters. Or, for those with a mathematical mind and a bit of patience, who enjoy a geometrical challenge – as well as for those who aren’t afraid to experiment – there is also the option of removing only *one* of the vector envelope components from the vector mix, and using the vector envelope in one dimension to drive the vector mix, while the other dimension serves as a modulation source.

Finally, it should be pointed out (once again) that the vector envelope can be used as a modulation source *regardless* of whether it is incorporated into the vector mix control signal. When the vector envelope is also being used to modulate the vector mix, it can be interesting to use the *Vec Env AD* and *Vec Env BC* component signals to modulate other parameters, whose variations then track the loudness of a particular oscillator (see *Vec Env AD and Vec Env BC* above). However, as with virtually everything else in the Vectron, this is merely an option and not a rule which must be consistently applied.

# Vector Envelope Programming Tips

## Setting envelope point positions

A newly-created vector envelope point takes on the position of the point whose position was most recently *adjusted*. You can use this to advantage when creating envelopes from scratch.

For example, when you add points sequentially to the end of the envelope, each new point will initially have the same position as the point added just before it, assuming that you adjust the position of each new point after adding it. This makes it fairly easy to plot the course of an envelope point by point.

Alternatively, if you set the position of a single point and then add a series of new points *without* adjusting their positions, *all* of the new points will initially have the same position. This works regardless of where the points are added – they need not be added to the end of the envelope. Thus, you can apply this technique in two passes to create a series of envelope points which alternate between two positions – or in three passes with three positions, etc.

## Creating a loop in the envelope

Select a point which is to be one end of the envelope, click the **Loop** button, and then click the point which is to be the other end. The points will be highlighted with loop icons accordingly.



Each envelope can have only one loop at a time. In order for a loop to function, the envelope must contain at least one point following the loop endpoint.

Looping is done in the forward direction only. Note that the segment time setting for the first point in the loop is applied when the loop “wraps around” from the last point back to the first. If your loop seems to have an abrupt “skip”, check whether the segment time for the first point is much smaller than for the other points in the loop. This can easily occur if the first loop point is also the very first point in the envelope.

In order to create an envelope loop which continues to loop unmodified after key release, all of the points in the loop must be set to *Time* mode (blue).

It is also possible to create a loop which loops *differently* after key release. An envelope loop will continue to function after key release if the loop endpoint and at least one of the other points within the loop is set to *Time* mode. Points within the loop which are set to other modes – typically *Sustain* (red) – will “drop out” of the loop following key release, and the loop will speed up accordingly.

## Deleting an envelope loop

Select either of the envelope loop points and click the **Loop** button. The loop icons will vanish, indicating that the loop has been deleted. Deleting an envelope loop does not affect the existing envelope *points* in any way – no points are deleted.

## Adjusting loop repeat count

By default, a newly-created loop will have a loop repeat count setting of 1. Unfortunately, the loop repeat count display (the small text fader next to the **Loop** button) does not immediately reflect this. You must once again select one of the loop points in order to bring the actual setting into the display.

Note that the the loop repeat count display always displays *INF* when a non-loop envelope point is selected. Changes made to this setting have no effect when non-loop points are selected.



# The Amp/Filter Page

## Amplifier

The amplifier controls the instantaneous volume level of a voice. It is under direct control of the amplifier envelope generator, which is a “classic” ADSR envelope generator featuring exponential curves. Most of the controls in this section affect the envelope generator.

The controls described in the *Additional level modulation* section below permit additional modulation of voice volume level by other modulation sources.

All amplifier settings are stored in (and restored from) sound presets.

## Att (envelope attack time)

Adjusts the duration of the attack phase of the envelope, which begins when a note is played. The maximum attack time is approximately 45 seconds, while the minimum time is less than 1 msec. The envelope rises from zero to its peak value over the specified amount of time.

## Dec (envelope decay time)

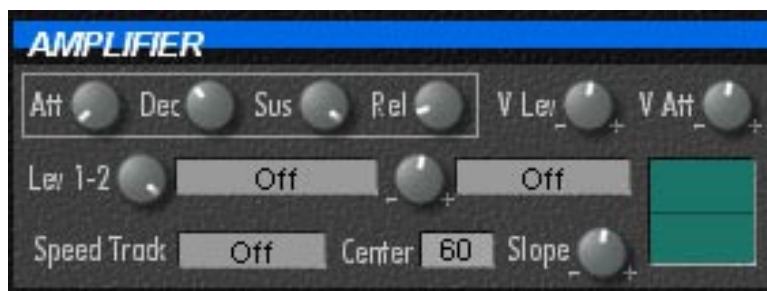
Adjusts the duration of the decay phase of the envelope, which begins when the attack phase is complete. The maximum decay time is approximately 45 seconds, while the minimum time is less than 1 msec. The envelope falls from its peak value to the specified sustain level over the specified amount of time.

## Sus (envelope sustain level)

Adjusts the sustain level. The envelope reaches this level at the end of the decay phase and remains at this level until the key is released.

## Rel (envelope release time)

Adjusts the duration of the release phase of the envelope. This phase begins at the moment of key release, regardless of what phase the envelope is currently in. The maximum release time is approximately 45 seconds, while the minimum time is less than 1 msec. The envelope falls from its instantaneous value to zero over the specified amount of time.



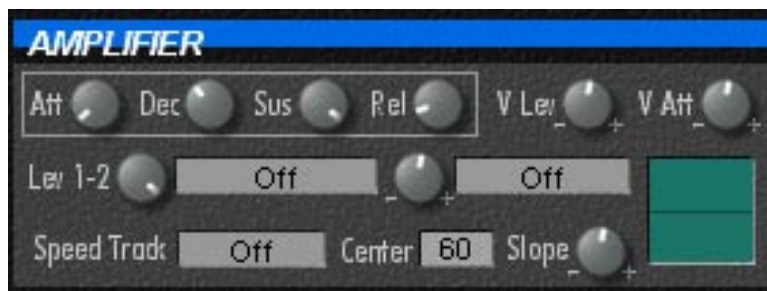
## V Att (attack time velocity mod)

This control adjusts the amount of modulation of amplifier envelope attack time in response to note-on velocity. The modulation works by producing a varying degree of attack time *reduction* with respect to the unmodulated attack time set by the **Att** control. It can be either positive or negative and functions as follows:

- At positive settings, higher velocities produce longer attack times. The maximum attack time is set by the **Att** control. It occurs at maximum velocity and is the same at all **V Att** settings. Larger positive **V Att** settings enlarge the attack time *range* by lowering the *minimum* voice attack time which is produced at minimum velocity.

- At negative settings, the modulation effect is the same, but the velocity response is reversed: *lower* velocities produce longer attack times. The maximum attack time is set by the **Att** control. It occurs at *minimum* velocity and is the same at all **V Att** settings. Larger negative **V Att** settings enlarge the attack time *range* by lowering the *minimum* attack time which is produced at *maximum* velocity.

Velocity modulation of attack time can be used at the same time as envelope speed tracking (see *Speed Track – Center – Slope* below) – the effect upon attack time is cumulative.



## Speed Track – Center – Slope

These controls permit the speed of the amplifier envelope as a whole to be varied as a function of note-on velocity or keyboard position (note number). This is therefore a *polyphonic* modulation whose effect can vary from note to note.

With speed tracking, attack, decay and release times are all increased or decreased in the same proportion. The envelope can be sped up or slowed down by up to a factor of ten with respect to the original **Att**, **Dec** and **Rel** time settings. Note that at higher settings of these controls, speed tracking can potentially produce attack, decay and release times of up to several minutes.

The audible effect of speed tracking is minimal with very short time settings. This fact can often be exploited to focus the effect on a particular part of the envelope. For example, assume that attack time (**Att**) is set to minimum (approximately 1 msec) and decay time (**Dec**) to 100 msec. Speed tracking will cause attack time to vary between 0.1 and 10 msec. These are all extremely short times which sound approximately like an “in-

stant” attack – speed tracking produces little noticeable variation. By contrast, the decay time will vary between 10 msec and one second – a much more pronounced variation in the “contour” of the sound. Thus, in this example, speed tracking effectively produces a variation in the decay time only.

Speed tracking can be used at the same time as velocity modulation of attack time (see *V Att* above) – the effect upon attack time is cumulative.

## Speed Track

Selects the speed tracking source parameter – the note parameter which envelope speed is to track. Choices include *Velocity* (note-on velocity) and *Note Num* (note number) as well as *Off* (no speed tracking).

## Center

Sets the *value* of the selected note parameter (see **Speed Track** above) at which speed tracking produces no net effect. Above and below this value, speed tracking produces a change in envelope speed.

The amount of speed change depends upon the **Slope** setting (see below) as well as on the difference between the parameter value of the current note and the specified **Center** value.

The direction of the change – whether the envelope speed is increased or decreased – depends upon whether the **Slope** setting is positive or negative, as well as whether the note parameter value is higher than or lower than the **Center** value.

A nominal value for the **Center** setting is 60. This is middle C when interpreted as a note number, or a medium velocity value. However, the **Center** value can be set anywhere in the range 0 .. 127.

## Slope

Sets the direction and intensity of the speed tracking effect. With positive settings, envelope speed will increase as you play further up the keyboard or at higher velocities, while with negative settings, a speed decrease will result.





## The Speed Tracking Display

In the small square window to the right of the **Slope** control, the effective speed tracking curve is displayed. This curve cannot be directly edited – instead, it graphically indicates the curve produced by the settings of the **Slope** and **Center** controls.

In this display, the horizontal axis represents (from left to right) the full 0 .. 127 value range of the selected speed tracking parameter, while the corresponding speed change is indicated in the vertical dimension. The horizontal center line corresponds to zero speed change – the point where the tracking curve crosses this line is the **Center** value. Where the curve reaches the bottom of the window, envelope speed is reduced to one-tenth of its original (**Att** – **Dec** – **Rel**) speed, while at the top of the window, the envelope is ten times faster than original speed.



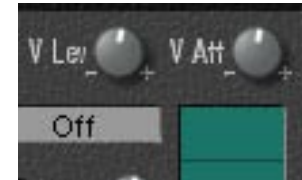
## Extending envelope time ranges

The speed tracking feature can be used to effectively expand the time range of the **Att**, **Dec** and **Rel** controls by a factor of ten for *all* notes. This allows you to achieve extremely long envelope times of up to several minutes.

To do this, select *Note Num* as the **Speed Track** source parameter. Then set **Slope** to full negative and **Center** to 0 (or, alternatively, set **Slope** full positive and **Center** to 127). As the Speed Tracking Display then will clearly show, the speed curve is “bottomed out” for all but a small handful of note numbers at the extreme lower (or upper) end of the note number range. For this handful of note numbers, the factor-of-ten slowdown will not apply. However, these note numbers are not found on most keyboards. Chances are you’re not using this part of the note number range, or can easily work around it (or, if necessary, you can use the alternative **Slope** and **Center** settings to shift it to the other end of the keyboard).

## V Lev (amp level velocity mod)

This control adjusts the amount of voice level modulation in response to note-on velocity.



This modulation works by producing a varying degree of volume *reduction* with respect to the unmodulated voice volume level, which is primarily determined by the amplifier envelope. It can be either positive or negative and functions as follows:

- At positive settings, higher velocities produce higher voice volume levels. The maximum voice volume level occurs at maximum velocity is the same at all settings. Larger positive settings enlarge the voice volume *range* by lowering the *minimum* voice volume level which is produced at minimum velocity.

- At negative settings, the modulation effect is the same, but the velocity response is reversed: *lower* velocities produce higher voice volume levels. The maximum voice volume level occurs at *minimum* velocity is the same at all settings. Larger negative settings enlarge the voice volume *range* by lowering the *minimum* voice volume level which is produced at *maximum* velocity.

## Additional level modulation

In addition to the voice volume modulations described above, voice volume can optionally be further simultaneously modulated via one dynamic modulation source and one event-driven source, each with its own amount control.

These modulations are monophonic – they affect all voices in common. They work by producing a varying degree of volume *reduction* with respect to the unmodulated voice volume level, which is primarily determined by the amplifier envelope. Volume level is never increased via these modulations, which therefore cannot cause overloading.

The *dynamic* source can be one of the fixed-level or variable-level LFOs. The modulation amount setting for this source is positive only. At lower modulation amount settings, the volume level will periodically swing from maximum to a slightly lower level, while at maximum modulation, the volume level goes through a full swing from maximum to zero and back with each LFO cycle.

The *event-driven* source can be the *ModWheel*, *MIDI Pressure* or one of the **Aux Controllers** *Aux Ctl A* .. *Aux Ctl D*. The modulation amount setting for this source can be positive or negative and works in a similar fashion to the amount setting for the dynamic source – larger positive or negative settings permit a greater range of volume *reduction* relative to the unmodulated maximum volume level. Negative settings merely reverse the direction of the change.

**Note:** At maximum positive (or negative) modulation amount settings, this modulation will suppress all sound output from the Vectron whenever the modulation controller is at minimum (or maximum, respectively). The synth may appear to have gone dead – but sound will appear as soon as the modulation controller is actuated.

## Filter



The Vectron features a warm-sounding, fat 4-pole low-pass filter which perfectly complements the Vectron's complex wavetable sounds. The filter incorporates variable resonance which can be adjusted right to the edge of self-oscillation – without reduction of sound level in the filter passband at any point – as well as a smooth overload characteristic.

All filter settings are stored in (and restored from) sound presets.

### Cutoff

Sets the base filter cutoff frequency. The cutoff frequency can be modulated by a variety of sources, as described in the following sections.

**Turning the Cutoff control all the way down will suppress all sound output from the Vectron, if the filter cutoff frequency is not additionally modified by the Filter Envelope or one of the other filter cutoff modulation sources described here.**

Note that modulation of filter cutoff frequency (as described following) works by raising or lowering filter cutoff frequency with respect to the **Cutoff** setting. Therefore, at high **Cutoff** settings, positive modulations may appear to have no effect. Correspondingly, negative modulations may appear to have no effect at low **Cutoff** settings.

### Res (resonance)

Sets the filter resonance amount. At high settings, the filter may begin to self-oscillate.

### Env (cutoff envelope mod)

Adjusts the amount of modulation of filter cutoff frequency by the **Filter Envelope** (described in the next section). This modulation can be positive or negative – the **Filter Envelope** can thus either raise or lower the filter cutoff frequency.

### Vel (cutoff velocity mod)

Adjusts the amount of modulation of filter cutoff frequency in response to note-on velocity. This modulation can be positive or negative – note velocity can thus either raise or lower the filter cutoff frequency.

Velocity modulation of filter cutoff frequency is independent of the modulation produced by the **Filter Envelope** (described below). It produces a velocity-dependent shift in cutoff frequency which is constant for the entire duration of a note.

The **Filter Envelope** also features a **Vel Depth** control. However, this control affects the envelope itself, and does not affect the filter directly. It will produce no effect on the filter if the **Env** control (above) is set to zero. The two velocity controls should not be confused with one another.



## Kbd (cutoff keyboard follow)

Adjusts the tracking of filter cutoff frequency with respect to keyboard position (note number).

The default (center) setting of this control produces an approximately parallel or “timbre-neutral” tracking of filter cutoff frequency and note frequency. This setting can be easily attained by double-clicking the **Kbd** knob.

At **Kbd** settings to the right of center, filter cutoff frequency changes more rapidly than note frequency as you play up and down the keyboard.

As the **Kbd** setting is moved to the left of center, filter cutoff frequency changes more slowly in response to changes in keyboard position. At a **Kbd** setting roughly midway between exact center and full left – indicated by a small “0” – , changes in keyboard position produce *no* filter cutoff change. Further to the left than this, filter cutoff tracks negatively, becoming lower as you play up the keyboard.

## Additional cutoff modulation

In addition to the filter cutoff modulations described above, filter cutoff can optionally be further simultaneously modulated via two dynamic modulation sources and one event-driven source, each with its own amount control.

Each of the *dynamic* sources can be one of the fixed-level or variable-level LFOs, one of the derived components of the vector envelope, or one of the vector modulation signals **AD Mod** or **BC Mod**. Modulation via one of the LFOs is monophonic – all voices are affected in common. By contrast, modulation via a vector envelope component signal is polyphonic (like the vector envelope itself). The **AD Mod** and **BC Mod** signals can be derived from a mixture of monophonic and polyphonic modulation sources – therefore, modulation via one of these signals may be either monophonic or polyphonic or a combination of both. The modulation amount settings for these sources can be positive or negative. However, since the modulation signals themselves are bipolar, these modulation sources produce both increases and decreases in filter cutoff frequency.

The *event-driven* source can be the *ModWheel*, *MIDI Pressure*, one of the **Aux Controllers** *Aux Ctl A .. Aux Ctl D* or one of the **Joystick Controller** component signals *Joystick X*, *Joystick Y*, *Joystick AD* and *Joystick BC*. The modulation amount setting for this source can be positive or negative. Since the joystick component signals are bipolar, the joystick can both raise and lower filter cutoff with either positive *or* negative modulation amount settings. The other sources are all unipolar and can therefore either raise *or* lower the cutoff (but not both at one time), depending upon the modulation amount setting.

## Filter Envelope

The filter envelope generator is a “classic” ADSR envelope generator featuring curves whose shape can be continuously adjusted between linear and exponential.

The effect of the filter envelope generator upon filter cutoff is regulated by the **Env** control in the **Filter** section (see above). When this control is set to zero (center), the filter envelope has no effect upon the filter.

The filter envelope generator can also serve as a modulation source for a number of other Vectron parameters, regardless of whether it is being used to modulate filter cutoff. It produces a polyphonic unipolar modulation signal. The **Env** control in the **Filter** section (see above) af-

fects only the amount of filter envelope signal applied to the filter and has no effect upon the filter envelope itself.

All filter envelope settings are stored in (and restored from) sound presets.

### Att (envelope attack time)

Adjusts the duration of the attack phase of the envelope, which begins when a note is played. The maximum attack time is approximately 45 seconds, while the minimum time is less than 1 msec. The envelope rises from zero to its peak value over the specified amount of time.







## Dec (envelope decay time)

Adjusts the duration of the decay phase of the envelope, which begins when the attack phase is complete. The maximum decay time is approximately 45 seconds, while the minimum time is less than 1 msec. The envelope falls from its peak value to the specified sustain level over the specified amount of time.

## Sus (envelope sustain level)

Adjusts the sustain level. The envelope reaches this level at the end of the decay phase and remains at this level until the key is released.

Note that the sustain level can be affected by the setting of the **Shape D-R** control. In effect, the **Sus** control takes on an increasingly exponential character as the

**Shape D-R** control is adjusted further in the direction of an exponential curve, and the actual sustain level produced by a particular **Sus** setting becomes lower. For this reason, it is recommended practice to adjust the **Shape D-R** control to the desired setting *before* adjusting the **Sus** control.

## Rel (envelope release time)

Adjusts the duration of the release phase of the envelope. This phase begins at the moment of key release, regardless of what phase the envelope is currently in. The maximum release time is approximately 45 seconds, while the minimum time is less than 1 msec. The envelope falls from its instantaneous value to zero over the specified amount of time.

## Shape A (attack shape)

Permits adjustment of the shape of the attack portion of the envelope over a continuous range from linear (constant rate of change) at full left to exponential (rate of change continuously becomes slower) at full right.

The envelope attack time set by the **Att** control (above) is not affected by this setting.

## Shape D-R (decay-release shape)

Permits adjustment of the shape of the decay and release portions of the envelope over a continuous range from linear (constant rate of change) at full left to exponential (rate of change continuously becomes slower) at full right.

The envelope decay and release times set by the **Dec** and **Rel** controls (above) is not affected by this setting. However, changing this setting *will* modify the effective envelope sustain level if the sustain level is not set to its minimum or maximum value. In effect, the **Sus** con-



trol takes on an increasingly exponential character as the **Shape D-R** control is adjusted further in the direction of an exponential curve, and the actual sustain level produced by a particular **Sus** setting becomes lower. For this reason, it is recommended practice to adjust the **Shape D-R** control to the desired setting *before* adjusting the **Sus** control.

## Vel Depth (env depth velocity mod)

This control adjusts the amount of modulation of filter envelope amplitude or intensity in response to note-on velocity.

This modulation works by producing a varying degree of amplitude *reduction* with respect to the unmodulated filter envelope amplitude. It can be either positive or negative and functions as follows:

- At positive settings, higher velocities produce higher filter envelope amplitude. The maximum amplitude occurs at maximum velocity is the same at all settings. Larger positive settings enlarge the amplitude *range* by lowering the *minimum* amplitude which is produced at minimum velocity.
- At negative settings, the modulation effect is the same, but the velocity response is reversed: *lower* velocities produce higher filter envelope amplitude. The maximum amplitude occurs at *minimum* velocity is the same at all settings. Larger negative settings enlarge the amplitude *range* by lowering the *minimum* amplitude which is produced at *maximum* velocity.

The Vel Depth control should not be confused with the Vel control in the Filter section, which adjusts filter cutoff velocity response independent of the filter

envelope. Likewise, the **Env** control in the **Filter** section (see above) affects only the amount of filter envelope signal applied to the filter and has no effect upon the velocity response of the filter envelope itself.

## Speed Mod

These controls permit modulation of the speed of the filter envelope as a whole. Attack, decay and release times are all increased or decreased in the same proportion. The envelope can be sped up or slowed down by up to a factor of ten with respect to the original **Att**, **Dec** and **Rel** time settings. Note that at higher settings of these controls, speed modulation can potentially produce attack, decay and release times of up to several minutes.

The modulation source is of the *event-driven* type. The *ModWheel*, *MIDI Pressure*, one of the **Aux Controllers** *Aux Ctl A .. Aux Ctl D* or one of the **Joystick Controller** component signals *Joystick X*, *Joystick Y*, *Joystick AD* and *Joystick BC* can be selected as the source. Modulation via these sources is monophonic – all voices are affected in common.



The modulation amount setting for this source can be positive or negative. Since the joystick component signals are bipolar, the joystick can both increase *and* decrease filter envelope speed by up to a factor of ten, with either positive *or* negative modulation amount settings – and therefore also permits a speed control range of up to 100:1 using a single controller! The other sources are all unipolar and can therefore either increase *or* decrease filter envelope speed (but not both at one time), depending upon the modulation amount setting.

Speed modulation can be used at the same time as speed tracking – the effect upon envelope speed is cumulative. Refer to the *Speed Track – Center – Slope* section below, which also includes additional discussion of envelope speed modulation.

## Speed Track – Center – Slope

These controls permit the speed of the filter envelope as a whole to be varied as a function of note-on velocity or keyboard position (note number). This is therefore a *polyphonic* modulation whose effect can vary from note to note.

With speed tracking, attack, decay and release times are all increased or decreased in the same proportion. The envelope can be sped up or slowed down by up to a factor of ten with respect to the original **Att**, **Dec** and **Rel** time settings. Note that at higher settings of these controls, speed tracking can potentially produce attack, decay and release times of up to several minutes.

The audible effect of speed tracking is minimal with very short time settings. This fact can often be exploited to focus the effect on a particular part of the enve-

lope. For example, assume that attack time (**Att**) is set to minimum (approximately 1 msec) and decay time (**Dec**) to 100 msec. Speed tracking will cause attack time to vary between 0.1 and 10 msec. These are all extremely short times which sound approximately like an “instant” attack – speed tracking produces little noticeable variation. By contrast, the decay time will vary between 10 msec and one second – a much more pronounced variation in the “contour” of the sound. Thus, in this example, speed tracking effectively produces a variation in the decay time only.

Speed tracking can be used at the same time as speed modulation (see *Speed Mod* above) – the effect upon envelope speed is cumulative.



## Speed Track

Selects the speed tracking source parameter – the note parameter which envelope speed is to track. Choices include *Velocity* (note-on velocity) and *Note Num* (note number) as well as *Off* (no speed tracking).

## Center

Sets the *value* of the selected note parameter (see *Speed Track* above) at which speed tracking produces no net effect. Above and below this value, speed tracking produces a change in envelope speed.

The amount of speed change depends upon the **Slope** setting (see below) as well as on the difference between the parameter value of the current note and the specified **Center** value.

The direction of the change – whether the envelope speed is increased or decreased – depends upon whether the **Slope** setting is positive or negative, as well as whether the note parameter value is higher than or lower than the **Center** value.

A nominal value for the **Center** setting is 60. This is middle C when interpreted as a note number, or a medium velocity value. However, the **Center** value can be set anywhere in the range 0 .. 127.

## Slope

Sets the direction and intensity of the speed tracking effect. With positive settings, envelope speed will increase as you play further up the keyboard or at higher velocities, while with negative settings, a speed decrease will result.



## The Speed Tracking Display

In the small square window to the right of the **Slope** control, the effective speed tracking curve is displayed. This curve cannot be directly edited – instead, it graphically indicates the curve produced by the settings of the **Slope** and **Center** controls.

In this display, the horizontal axis represents (from left to right) the full 0 .. 127 value range of the selected speed tracking parameter, while the corresponding speed change is indicated in the vertical dimension. The horizontal center line corresponds to zero speed change – the point where the tracking curve crosses this line is the **Center** value. Where the curve reaches the bottom of the window, envelope speed is reduced to one-tenth of its original (**Att** – **Dec** – **Rel**) speed, while at the top of the window, the envelope is ten times faster than original speed.

## Extending envelope time ranges

The speed tracking feature can be used to effectively expand the time range of the **Att**, **Dec** and **Rel** controls by a factor of ten for *all* notes. This allows you to achieve extremely long envelope times of up to several minutes.

To do this, select *Note Num* as the **Speed Track** source parameter. Then set **Slope** to full negative and **Center** to 0 (or, alternatively, set **Slope** full positive and **Center** to 127). As the Speed Tracking Display then will clearly show, the speed curve is “bottomed out” for all but a small handful of note numbers at the extreme lower (or upper) end of the note number range. For this handful of note numbers, the factor-of-ten slowdown will not apply. However, these note numbers are not found on most keyboards. Chances are you’re not using this part of the note number range, or can easily work around it (or, if necessary, you can use the alternative **Slope** and **Center** settings to shift it to the other end of the keyboard).

# The LFO Page

## Overview Of The LFOs

The Vectron includes three LFOs (Low Frequency Oscillators) which can be used as modulation sources for virtually every basic synthesis parameter in the Vectron – including some parameters of the LFOs themselves!

It is important to be aware that the Vectron's LFOs are *monophonic*. Each of the LFOs is available only as a single signal source which affects all voices in common. Thus, differing LFO-based modulations per voice – such as varying speeds or phase angles – are not possible. In some cases, this constitutes a limitation. Keep in mind, however, that polyphonic LFO modulation can also quickly produce sonic chaos and is in fact not always desirable!

**LFO1** and **LFO2** are complex LFOs featuring selectable waveforms, adjustable Delay and Fade-in times, rate and amplitude modulation options, optional note-event retriggering with adjustable starting phase, and speeds of up to 400 Hz.



The **SineLFO** is a simple sine-wave only LFO with speeds of up to 100 Hz.

**All three LFOs** feature the ability to synchronize themselves to an internal or external (MIDI) tempo clock. This enables them to produce rhythmic modulation effects in time with the music.

In addition, each LFO is available as either a *fixed-level* or *variable-level* (var) modulation signal at each modulation destination. The control source for the variable-level version of each LFO is an event-driven MIDI controller-type signal – the *ModWheel*, *MIDI Pressure* or one of the **Aux Controllers** *Aux Ctl A* .. *Aux Ctl D*. It can be selected independently for each LFO.

All LFO settings are stored in (and re-stored from) sound presets.



## LFO1 and LFO2

The features of the complex LFOs **LFO1** and **LFO2** are identical. The following function descriptions apply to both of them.



### Waveform select

This small waveform window is actually a *graphical* text fader. Click-hold and drag with the mouse to select the LFO waveform. Options include *Sine*, *Square*, *Sawtooth (up)*, *Sawtooth (down)*, *Triangle* and *Sample-Hold (random)*.

### MIDI (LFO speed MIDI sync)

This button activates the LFO speed MIDI sync option.

When the MIDI sync option is disabled, LFO speed is set directly via the **Rate** control. LFO speed modulation is possible and is determined by the settings of the **Rate Mod 1-2** controls.

When the MIDI sync option is enabled, LFO speed is set in terms of *16th-notes* and *1/192 notes* and is thus relative to the tempo clock being used. The **16ths-192nd** fields for adjusting these settings appear in place of the **Rate** control, whose setting is no longer effective. Likewise, the **Rate Mod 1-2** controls disappear when MIDI sync is activated, and LFO speed modulation is disabled.

Note that the “MIDI” clock can be either an actual MIDI clock signal which is fed into the Vectron from an external source (via the **MCik** input) or the Vectron’s own

internal tempo clock generator. The MIDI sync behavior of the LFOs is the same in either case.

Selection of an internal or external clock source as well as setting of the internal clock tempo is done in the **Sync** section of the **Global** page. Refer to the corresponding section of the manual for a full description of these settings.

### Rate

This control permits direct setting of LFO speed in terms of *frequency*. It is visible (and active) only when the LFO speed MIDI sync option (above) is disabled.

An extremely wide frequency range of *0.01 Hz .. 400.00 Hz* is available. Frequency values can be entered directly into the **Rate** text field from the computer keyboard.



## 16ths-192nd (16th-notes – 1/192 notes)

These controls permit LFO speed to be set in terms of *rhythm*. They are visible (and active) only when the LFO speed MIDI sync option (above) is enabled.



With the MIDI sync option enabled, LFO speed is automatically synchronized to the tempo clock currently being used. The **16th-192nd** fields permit the speed to be set directly to a particular rhythmic interval relative to this tempo. The LFO will track tempo changes and retain the specified rhythmic relationship to the tempo.

The **16th** field sets the number of 16th-notes over which the LFO goes through a complete waveform cycle. For example, at a setting of *1*, the LFO waveform will go through four complete cycles per quarter-note beat. When **16th** is set to *4*, the waveform repeats once per beat, while a setting of *16* produces one full waveform cycle per 4/4 bar – and the maximum setting of *256* causes the LFO

to cycle once every *sixteen* 4/4 bars. (All of these examples assume that the **192nd** field is set to *0*.)

The **192nd** field permits LFO speed to be set in very fine fractions of a beat. In terms of time duration, one 1/192-note is *one-twelfth* of a 16th-note. Among other things, this field can be used (with the **16th** field set to zero) to create LFO rhythms which run quite rapidly even when synchronized to a slow tempo, or (in conjunction with non-zero settings in the **16th** field) to set up LFO rhythms which slowly „drift“ in and out of time. For most conventional rhythms, the **16th** field is adequate and the **192nd** field can be left set to *0*.

It's important to be aware that both the **16th** and **192nd** fields can be set to zero, but *not* both at the same time. If one of these fields is set to zero, the other one cannot be set lower than *1*. In order to be able to set either field to zero, the other field must first be set to a non-zero value.

Note that the „MIDI“ clock can be either an actual MIDI clock signal which is fed into the Vectron from an external source (via the **MCik** input) or the Vectron's own internal tempo clock generator. The MIDI sync behavior of the LFOs is the same in either case.

Selection of an internal or external clock source as well as setting of the internal clock tempo is done in the **Sync** section of the **Global** page. Refer to the corresponding section of the manual for a full description of these settings.

## Retr (retrigger)

This button controls the retriggering option.



When retriggering is disabled, the LFO runs continuously. The smooth repetition of LFO waveform cycles proceeds without interruption at all times.

When retriggering is enabled, the LFO waveform is *restarted* when notes are played – the waveform skips instantly to a specific point in its cycle and resumes from there. The point to which the LFO waveform skips is set by the **Ph** (initial phase) control (below).

In essence, retriggering is used when it is desired that the LFO waveform is synchronized to note starts and that the LFO-modulated parameter – for example, filter cutoff – varies in the same way for each note. Remember that the LFOs are monophonic – when an LFO is retriggered, the effect will also apply to any previous notes which are still playing.

On the other hand, retriggering is *disabled* when it is desired that the LFO-modulated parameter varies over the course of many notes, and that the flow of this modulation – which may be synchronized to the rhythm of the music – not be disrupted each time a new note is played (“techno mode”).

### **Every Note and First Note retriggering modes**

Note that there are *two* LFO retriggering modes available. LFO retriggering can occur on *every* new note which is played, or only on “first” notes – i.e., notes which are played when no keys are being held down.

With the latter mode, you can progressively add new notes to a chord or drop them out of a chord without restarting the LFO. As long as at least one key is being held down – it need not be the first key played – the LFO will continue to run uninterrupted when new notes are played. Once *all* keys are released, the *next* new note will again retrigger the LFO.

Selection of *First Note* vs. *Every Note* LFO retriggering is done in the **Sync** section of the **Global** page and applies to both **LFO1** and **LFO2** in common.

## Ph (initial phase)

This setting adjusts the starting phase of the LFO – i.e., the point in its waveform to which the LFO skips when it is retriggered. It has no effect upon the LFO at other times – and naturally, no effect at all if retriggering is disabled.

“Good” values for this setting depend upon the selected waveform (see *Waveform select* above). A setting of  $0^\circ$  is not always the most useful:

- When the *Sawtooth (up)* and *Sawtooth (down)* waveforms are retriggered with a **Ph** setting of  $180^\circ$  or  $-180^\circ$ , the waveform is restarted at one “end”, so that the modulation “ramps” up or down starting from one extreme of its range.
- The *Triangle* waveform is resynchronized to positive and negative “peaks” with **Ph** settings of  $90^\circ$  and  $-90^\circ$ , respectively.

- With the *Sine* waveform, on the other hand, resynchronization to positive and negative “peaks” is obtained with **Ph** settings of  $45^\circ$  and  $-135^\circ$ , respectively.
- The *Square* waveform has only two values: full positive and full negative. Therefore, it doesn’t “ramp” at all, but instead produces simple “two-state” rhythmic modulation. The **Ph** setting merely affects the “positioning” of this rhythm relative to the moment of retriggering. Settings of  $0^\circ$  and  $180^\circ$  are most often useful. A setting of  $0^\circ$  produces “even” resynchronization to the beginning of the positive half of the wave cycle. Increasing the setting shifts the LFO retrigger point further and further into this half of the wave

cycle, until finally – at  $180^\circ$  – the LFO skips over it completely and begins at the start of the negative half. Similarly, adjusting **Ph** over the range  $-180^\circ \dots 0^\circ$  shifts the retrigger point progressively further through the negative half of the wave cycle, until it “wraps back around” to the start of the positive half at  $0^\circ$ .

- The *Sample-Hold (random)* waveform maintains a single constant random value over the duration of each LFO waveform cycle. With this waveform, a **Ph** setting of  $180^\circ$  produces “even” resynchronization to the beginning of a wave cycle. Other settings result in a “syncopated” resynchronization relative to the moment of retriggering.

## Dly (delay time)

This control permits adjustment of the LFO delay time. The delay begins or is restarted whenever an LFO retriggering event occurs – that is, upon any note-on event which matches the currently selected *Every Note / First Note* filtering condition (see *Every Note and First Note retriggering modes* above). Note that delay triggering is always enabled and is not affected by the setting of the **Retr** (retrigger) switch, which affects only re-starting of the LFO *waveform*.

The level of the LFO remains at zero during the delay period and then rises smoothly to its full level over the time period specified by the **F-In** (fade-in time) control (below). The delay can be effectively disabled by setting the **Dly** control to 0.



## F-In (fade-in time)

This control permits adjustment of the LFO fade-in time. The fade-in period begins at the end of the delay period (see above) which occurs whenever the delay is triggered or retriggered – even if the **Dly** control is set to 0.

Following the delay period (if any), the level of the LFO rises smoothly from zero to its full level over the time period specified by the **F-In** (fade-in time) control. The fade-in can be effectively disabled by setting the **F-In** control to 0.

## Amp Mod (amplitude modulation)

Each of the complex LFOs **LFO1** and **LFO2** can be amplitude-modulated by the other complex LFO or by the **SineLFO**.

Amplitude modulation produces an LFO signal whose level varies over time in accordance with the modulation signal. The **Amp Mod** controls permit selection of the modulation source LFO and the modulation amount.

Note that the speed of the modulating LFO need not be lower than that of the modulated LFO. It can also be run at nearly the same speed, or even faster, resulting in a complex and/or chaotic LFO waveform.

LFO amplitude modulation should not be confused with LFO level control, which is a completely separate function (see *Var Src*).

## Some technical details

(It's not necessary to know this stuff in order to use LFO amplitude modulation – but you may find it useful, or at least interesting.)

The effect produced by amplitude-modulation consists of a varying level *reduction* relative to the normal LFO level. The *maximum* LFO level is always the same as the level of the original LFO *without* amplitude modulation. Adjusting the **Amp Mod** amount control changes only the lower end of the LFO level range. Thus, the *peak* depth of the effect which an LFO *produces* is always the same, regardless of whether or how much the LFO is being amplitude-modulated.

To further ensure that the peak LFO level stays constant, amplitude modulation is done using a full-level (unmodulated) LFO as the *modulation source* signal. Although amplitude modulation of an LFO affects the LFO signal which is delivered to all *other* parts of the Vectron, it does *not* apply to the signal used to amplitude-modulate the other LFO.

As an example, consider the following situation: **LFO2** is being amplitude-modulated by the **SineLFO**, and **LFO1** is being amplitude-modulated by **LFO2**. In this scenario, **LFO1** is actually modulated by an unmodulated, full-level version of **LFO2**. Therefore, the effect of the **SineLFO** upon **LFO2** does not carry over into the amplitude modulation of **LFO1** by **LFO2**. However, **LFO2** appears in amplitude-modulated form everywhere *else* in the device.

In addition, the “(var)” or *variable-level* versions of the LFOs (see below) are not available as source signals for LFO amplitude modulation. This similarly guarantees that the peak LFO level remains unchanged.

## Var Src (variable-level LFO control source)

Each of the Vectron’s LFOs is provided in two versions as a modulation source – either *fixed-level* or *variable-level*. At every point where an LFO can be selected as a modulation source, you can choose either version. The variable-level version is indicated in modulation source selection lists by “(var)” – for example, **LFO1 (var)**.

The **Var Src** parameter lets you select the control source for the variable-level version of an LFO. This source is an event-driven MIDI controller-type signal – the *ModWheel*, *MIDI Pressure* or one of the **Aux Controllers** *Aux Ctl A .. Aux Ctl D*.

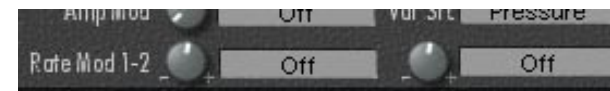
The selected controller can be different for each of the three LFOs. When *Off* is selected, the variable-level version of an LFO is identical to the fixed-level version.

Note that *level control* of an LFO is different from the amplitude modulation described above (see *Amp Mod*), and is also completely separate from it. Level control is a “live” control over the effective level of an LFO. It works in the same way regardless of whether the LFO is being amplitude-modulated.

## Rate Mod 1-2

Each of the complex LFOs **LFO1** and **LFO2** can be rate-modulated by one dynamic modulation source and one event-driven source, each with its own amount control.

The rate modulation controls are visible (and active) only when the LFO speed MIDI sync option (above) is disabled. LFO rate modulation is disabled under MIDI sync.



The dynamic modulation sources include the fixed-level and variable-level LFOs. Note that it is entirely possible – and sometimes quite useful – to use an LFO to modulate its own speed. If the technique is applied subtly – by keeping the modulation amount within limits – it can produce novel variations upon the basic LFO waveforms. A more heavy-handed approach can produce total LFO chaos! Likewise, **LFO1** and **LFO2** can modulate each others’ speeds at the same time.

The event-driven modulation sources include the event-driven MIDI controller-type signals – the *ModWheel*, *MIDI Pressure* or one of the **Aux Controllers** *Aux Ctl A .. Aux Ctl D*.

## The SineLFO

The SineLFO is a simple sinewave-only LFO. It can be used as a general-purpose modulation source in the same way as the complex LFOs **LFO1** and **LFO2**, but does not offer all of the functions provided by those LFOs.

The functions which the SineLFO does offer work in exactly the same way as the corresponding functions of the other LFOs (except that the maximum **Rate** setting is *100.00 Hz* instead of *400.00 Hz*). These functions include **MIDI sync**, **Rate**, **16ths-192nd** and **Var Src**. For descriptions of these functions, refer to the corresponding function descriptions in the section *LFO1 and LFO2* above.





# The Vec/Pan Page

This page contains the vector modulation and pan modulation control sections. All of the settings on this page are stored in (and restored from) sound presets.

## Vector Modulation

This section is the “main control center” for vector mix modulation in a Vectron sound preset. Here, you determine the composition of the **A-D Mod** and **B-C Mod** “diagonal axis” vector mix modulation signals. In addition, you control how and to what extent the vector envelope affects the vector mix – and likewise for the **Joystick Controller**. The final vector mix is the *sum* of all of these components. The **Vector Modulation** section is thus in effect a master mixer for the vector modulation signal.



## A-D Mod (modulation on the A-D axis)

The **A-D Mod** signal is a modulation source primarily intended for modulating the vector mix. The name of this signal comes from the effect it has upon the vector mix. Very roughly, it causes the mix to “swing” between **OscA** and **OscD**.

## What the A-D Mod signal actually does

Described in “joystick” terms, the **A-D Mod** signal alters the vector mix in a *diagonal* direction from upper left to lower right – thus, along the path between the **OscA** corner and the **OscD** corner, or parallel to that path.

This can be seen directly in the **Vector Display** by setting **Vec Disp Mode** to *AD-BC Mod* (the **BC Mod** source select controls should all be set to *Off*, so that a “pure” **A-D Mod** signal can be displayed).

Thus, the effect is actually not as simple as a “swing” between **OscA** and **OscD**. At its positive and negative extremes, the **A-D Mod** signal – acting alone – would produce a mix consisting 100% of **OscA** or **OscD**, respectively. In between these extremes, the mix would contain corresponding proportions of **OscA** and **OscD** – and varying amounts of **OscB** and **OscC** would also be present in the mix.

It would therefore be more accurate to say that the **A-D Mod** signal acts by varying the *balance* between **OscA** and **OscD** in the vector mix. In the process, it

also changes the *levels* of **OscB** and **OscC** – but without changing the *balance* between them.

Of course, not even *that* description is truly complete, because when the other vector mix modulation sources are again taken into consideration, the picture can become a good deal more complex. (And that's the beauty of the Vectron: things seem simple, but they're not – but it doesn't matter! Just go ahead and *use* it!)

## Components of the A-D Mod signal

The **A-D Mod** signal is comprised of a mix of anywhere from one to four individual modulation source signals (each with its own positive/negative amount control) and is completely independent of the vector envelope.

- Each of the two dynamic modulation signals can be any one of the fixed-level or variable-level LFOs, or the filter envelope.
- A third source is an event-driven MIDI controller-type signal – the *ModWheel*, *MIDI Pressure* or one of the **Aux Controllers** *Aux Ctl A .. Aux Ctl D*.
- The fourth source offers the choice between note-on velocity and the output of the **Key Scaling Generator** (described in the chapter *The Global Page*).

The **A-D Mod** signal thus potentially consists of monophonic components only, polyphonic components only, or a mixture of both, and its effect upon multiple voices is correspondingly variable – the vector mix modulation which it produces may affect all voices identically or may be different for each voice.

The *ModWheel*, *MIDI Pressure* and the **Aux Controllers** *Aux Ctl A .. Aux Ctl D* are applied to the **A-D Mod** signal in a slightly different way as elsewhere in the Vectron. Normally, they are *unipolar* sources which produce zero modulation at the bottom end of their range and otherwise produce modulation in one direction only. As applied to the **A-D Mod** signal, however, they are *bipolar* sources which produce zero modulation at the midpoint of their range and can produce both positive and negative modulation. The same applies to note-on velocity as it affects the **A-D Mod** signal.

## Using the A-D Mod signal

The **A-D Mod** signal can be used in a huge variety of ways. It can be combined with the **B-C Mod** signal and the vector envelope to produce quite complex vector mix modulations or to add a subtle extra dimension to the mix produced by the vector envelope.

Also, as described later in this chapter, the vector envelope can optionally be “disconnected” from the vector mix (and optionally used as a freely-assignable modulation signal elsewhere in the Vectron). The vector mix can then be controlled completely by the **A-D Mod** and/or **B-C Mod** signals alone. The large variety of modulation sources which can be selected and combined to create these signals make this alternative likewise quite interesting. Among the possibilities which this affords is that of modulating the vector mix – and thus timbre – in response to note-on velocity or keyboard position.

Note that it is *not* possible to “disconnect” the **A-D Mod** signal from the vector mix – the vector mix is always affected by this signal.

Finally, the **A-D Mod** signal is itself available for use as a modulation source signal in various places in the Vectron which have nothing to do with the vector mix. This permits other parameters in the Vectron to be modulated in synchronization with the vector mix modulation produced by the **A-D Mod** signal and is especially useful when the **A-D Mod** signal is the dominant (or only) component of the vector mix.

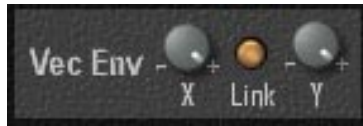
## B-C Mod (modulation on the B-C axis)

It will probably not surprise you to read that the **B-C Mod** signal is in practically all respects exactly the same as the **A-D Mod** signal. It is an equivalent but completely separate signal comprised of an independent mix of modulation source signals. Naturally enough, it produces vector mix modulation along the *other* diagonal – the one between **OscB** and **OscC**.

Thus, whereas the **A-D Mod** signal acts primarily by varying the balance between **OscA** and **OscD** in the vector mix, the **B-C Mod** signal primarily varies the balance between **OscB** and **OscC**. Apart from this crucial but quite basic difference, the two signals are the same. The preceding description of the **A-D Mod** signal is thus applicable to the **B-C Mod** signal as well – all you need to do is substitute the oscillator letters accordingly!

## Vec Env (vector envelope amount)

These controls permit the influence of the vector envelope upon the vector mix to be adjusted. In effect, they are the “gain controls” for the vector envelope as it applies to the vector mix.



### The X, Y and Link controls

Separate **X** and **Y** controls are provided for the corresponding components of the vector envelope signal. Normally, these controls are linked – adjusting one of them sets the other one to the same position. If the **Link** option is switched off, it becomes possible to adjust the **X** and **Y** controls separately. Note that the settings of all of these controls – including the **Link** switch – are stored in (and restored from) sound presets.

The “default” position for the **X** and **Y** controls is full positive (clockwise). This causes the vector envelope to be mixed “full strength” into the vector mix control signal, and without any inversion of direction.

Turning these controls down reduces the amount of vector envelope fed to the vector mix control signal. If only one of the controls is turned down, the amount of the corresponding envelope *component* (X or Y) is reduced fed to the vector mix control signal. At a setting of zero (center – attained by double-clicking on the knob), the corresponding vector envelope component has no effect upon the vector mix.

Either control can furthermore be turned past zero into the negative range, in which case it begins feeding an increasing amount of an *inverted* envelope component signal to the vector mix. This reverses the effect of that component upon the mix. Turning the **X** control all the way to the left thus causes the effect of the vector envelope upon the mix to “mirror” itself from left to right around the vertical center line, while doing so with the **Y** control produces a similar mirroring from top to bottom around the horizontal center line – and adjusting *both* controls to full left effectively “rotates” the effect of the vector envelope by 180°.

### Disabling the vector envelope

As mentioned above, setting the **X** and **Y** controls to zero (by double-clicking on either one of them, when the **Link** option is active) eliminates their influence upon the vector mix.

This can be handy for programming purposes. It’s much easier to pull the vector envelope out of the mix with this method than to actually delete the entire vector envelope, which is the only other way to eliminate it.

At other times, you may wish to keep the vector envelope in place and merely temporarily disable it in order to isolate the **A-D Mod** and/or **B-C Mod** signals and thus be able to program them more easily (but don’t forget that the settings of the **Vec Env** controls are stored in sound presets).

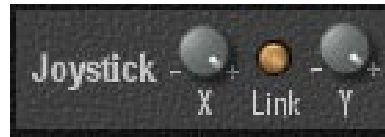
## Other uses for the vector envelope

It should be emphasized once again that the **X** and **Y** controls have no effect upon the vector envelope itself, but control only its *influence* upon the *vector mix*. The vector envelope is available *at all times* for use as a modulation source affecting many other parameters in the Vectron, regardless of the settings of the **X** and **Y** controls.

This aspect of the vector envelope is discussed fully in the chapter *The Vector Envelope* (see the section *The vector envelope as modulation source*).

## Joystick (joystick controller amount)

These controls permit the influence of the **Joystick Controller** upon the vector mix to be adjusted. In effect, they are the “gain controls” for the **Joystick Controller** as it applies to the vector mix.



As with the vector envelope, the **Joystick Controller** is intended primarily as a control source for the vector mix – in this case, a manually-operated one – but can also be used to control many other parameters in the Vectron. The joystick amount controls permit the effect of the **Joystick Controller** upon the vector mix to be modified – including reversal of the X and/or Y components, if desired – and also permit the **Joystick Controller** to be completely disconnected from the vector mix (by setting both controls to exact center), so that it can be used to control other sound parameters without affecting the vector mix.

The joystick amount controls are similar to the vector envelope amount controls described in the preceding section. Please refer to that section for details.

## Pan Modulation

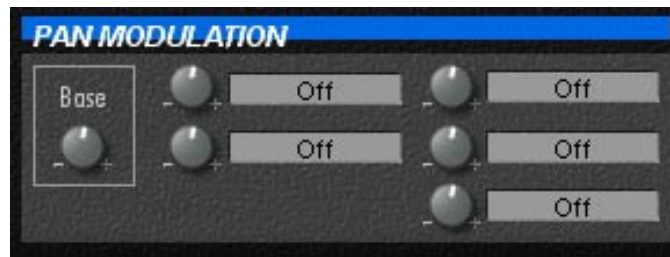
These settings control dynamic panning effects. Up to five modulation sources can be applied to pan position simultaneously, permitting complex panning effects to be produced. In addition, the base or initial pan position can be adjusted here. The pan position for each voice is the sum of all active pan modulation sources and the base pan position setting.

Panning in the Vectron is done independently per voice. However, some of the available pan modulation sources are monophonic. Thus, panning effects can be either monophonic or polyphonic or a mixture of both.

## Base (pan base position)

Adjusts the “initial” unmodulated or *base* pan position.

Typically, this control is set to dead center. With specific modulation sources, it can be useful to set it off-center – for example, when using the filter envelope to create a pan sweep from one side to the other.





## Modulation source / amount controls

These controls permit selection of up to five modulation sources as well as adjustment of the modulation intensity and modulation direction for each source.

The “net” pan modulation signal is the sum of all active modulation sources. Since some of these sources are polyphonic, the pan modulation signal can be different for each voice.

- The first source offers the choice between note-on velocity and the output of the **Key Scaling Generator** (described in the chapter *The Global Page*). These are polyphonic modulation sources.
- A second source is an event-driven MIDI controller-type signal – the *ModWheel*, *MIDI Pressure*, one of the **Aux Controllers** *Aux Ctl A .. Aux Ctl D* or one of the **Joystick Controller** component signals *Joystick X*, *Joystick Y*, *Joystick AD* and *Joystick BC*. All of these modulation sources are monophonic.

- Each of the three dynamic modulation signals can be any one of the fixed-level or variable-level LFOs, the filter envelope, or one of the vector envelope component signals *Vec Env X*, *Vec Env Y*, *Vec Env AD* or *Vec Env BC*. The LFOs are monophonic, while the other sources are polyphonic.

The *ModWheel*, *MIDI Pressure* and the **Aux Controllers** *Aux Ctl A .. Aux Ctl D* are applied to pan modulation in a slightly different way as elsewhere in the Vectron. Normally, they are *unipolar* sources which produce zero modulation at the bottom end of their range and otherwise produce modulation in one direction only. As applied to pan modulation, however, they are *bipolar* sources which produce zero modulation at the midpoint of their range and can produce both positive and negative modulation. The same applies to note-on velocity as it affects pan modulation.

# The Pitch Page



This page contains the pitch modulation control section, which includes global pitch modulation affecting all four oscillators in tandem as well as pitch modulation for each oscillator separately. Both global and per-oscillator pitch modulation can be applied at the same time – the resulting pitch modulation for each oscillator is the sum of the global modulation and the individual modulation.

All of the settings on this page are stored in (and restored from) sound presets.

## Global Pitch Modulation

Two freely selectable *dynamic* modulation sources can be applied. Each source can be any one of the fixed-level or variable-level LFOs, the filter envelope, or one of the vector envelope component signals *Vec Env X*, *Vec Env Y*, *Vec Env AD* or *Vec Env BC*. The LFOs are monophonic, while the other sources are polyphonic.

Both coarse and fine pitch modulation *amount* controls are provided for each source. Both settings indicate the pitch change produced when the associated modulation signal is at its maximum *positive* level. Thus, the coarse control provides adjustment of maximum pitch modulation depth in semitone steps over the range -63 .. +63, while the fine control provides a corresponding adjustment in cents over the range -100 .. +100.

Negative pitch modulation settings produce the same amount of change as corresponding positive settings, but in the opposite direction – e.g., lowering pitch instead of raising it. Note that bipolar modulation sources such as the LFOs “swing both ways” and will therefore both raise and lower pitch in alternation with *either* positive or negative modulation amount settings.

## Pitch Modulation Per Oscillator

The pitch modulation options per oscillator are the same as the global pitch modulation options described above, except that only one of the two possible sources is a dynamic source. The second source is an event-driven MIDI-type signal whose possible selections include the *ModWheel*, *MIDI Pressure*, *Velocity*, the *Key Scaling Generator*, one of the **Aux Controllers** *Aux Ctl A* .. *Aux Ctl D*, or one of the **Joystick Controller** component signals *Joystick X*, *Joystick Y*, *Joystick AD* and *Joystick BC*. *Velocity* and the *Key Scaling Generator* are polyphonic modulation sources. All of the other modulation sources in this list are monophonic.

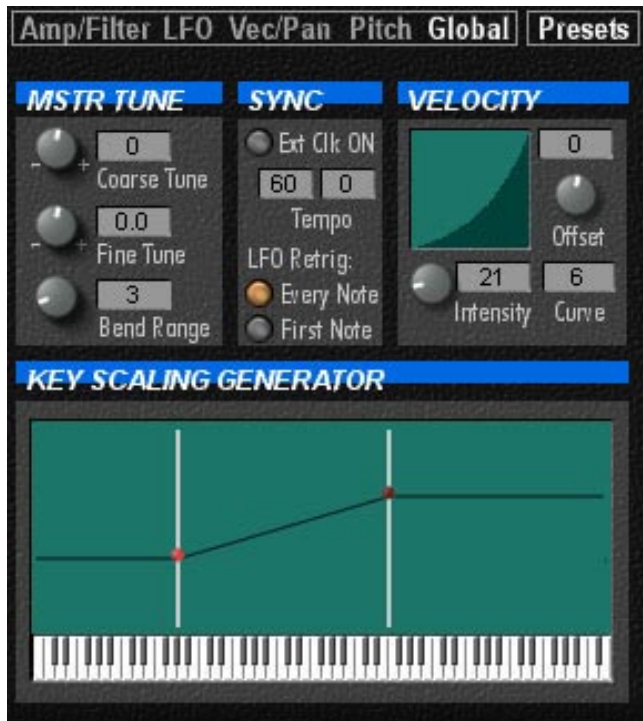
## Pitch Modulation Possibilities

The pitch modulation capabilities of the Vectron offer extremely fertile ground for experimentation. Following are just a few suggestions:

- Apply one of the vector envelope component signals as a global pitch modulation source. By carefully tweaking the modulation amount controls (and possibly the envelope itself), you can produce melodies and arpeggiations on a single key.
- Tune the oscillators to play chords on a single note. Apply different pitch modulations to individual oscillators to cause individual pitches in the chord to vary. These modulations can come from dynamic sources such as the vector or filter envelopes, or from “manual” sources such as the main and auxiliary mod wheels, or combinations of both.
- Use a single modulation source to modulate two different oscillators slightly – one with a positive amount and one with a negative amount. This produces a modulated detuning or chorusing effect.
- Take advantage of the “ultra-clean” LFOs to add a touch of FM to your vector synthesis sounds by way of pitch modulation (which is just another name for FM).
- Run an oscillator at constant pitch across the entire keyboard as follows: Set the two points of the **Key Scaling Generator** to far left bottom and far right top, then select the **Key Scaling Generator** as the pitch modulation source for the desired oscillator and set the coarse and fine modulation amount controls to -63 (semitones) and -50 (cents) respectively. Now, the pitch of this oscillator is determined by the **Coarse** tuning setting for this oscillator on the **ProgOsc** page and is the same for any note you play.
- Explore and exploit the limits of the very wide pitch modulation range and the effects produced by extreme slow-down or speed-up of wavetable playback.

# The Global Page

This page contains a variety of sections: **Master Tuning** (including **Pitch Bend Range**), **LFO / Delay Sync**, **MIDI Velocity Curves**, and the **Key Scaling Generator**.



Despite the name “Global”, many of the settings on this page are not global in the usual sense, but are stored and recalled from sound presets and can therefore differ from one sound to the next. This is indicated for each setting in the control descriptions which follow.

**Vectron**

## Mstr Tune (Master Tuning)

### Coarse (coarse tuning)

Adjusts the overall tuning of the Vectron in semitone steps over the range -24 .. +24. This setting is global and affects all sound presets.

### Fine (fine tuning)

Permits fine adjustment of the overall tuning of the Vectron over the range -100.0 .. +100.0 cents, with a resolution of approximately 0.1 cent. This setting is global and affects all sound presets.

## Bend (pitch wheel range)

Sets the range of MIDI Pitch Bend (Pitch Wheel) control in semitone steps over the range 0 .. 24. The setting indicates the maximum upward *and* downward pitch bend which the pitch wheel can produce. This setting is stored in (and recalled from) sound presets.



[Contents](#)

[Index](#)

78

## Sync (LFO / Delay Sync)

The settings in this section control the Vectron's tempo clock. This clock can be used to synchronize **LFO1** and **LFO2** to an internal or external tempo source, and also enables the **Stereo Delay** to adapt its delay time to match the selected tempo source. The use of the LFOs and the delay effect in synchronized mode is discussed in the corresponding sections of the manual.



## Ext Clk ON (external sync)

When the **Ext Clk ON** option is active, the Vectron's internal clock source will synchronize itself to an external MIDI clock signal. The external MIDI clock source should be connected to the **MCik** input of the Vectron device. If the MIDI note source being used to play the Vectron is also carrying the MIDI clock stream to which you wish to synchronize the Vectron, simply connect this source to both the **MIDI** and **MCik** inputs of the Vectron. The current tempo of the external clock source is displayed in the **Tempo** field (see below). This display reverts to a default value of *60.00* (BPM) if the external clock is currently not running, as does the Vectron's internal tempo clock.

When the **Ext Clk ON** option is not active, the Vectron's internal clock source will run continually at the last tempo set by the user via the **Tempo** field (see below).

This is a global setting which is not stored in sound presets.

## Tempo

This is both a display and value-entry field whose use and meaning depend upon the setting of the **Ext Clk ON** button (see above). The two-part field displays whole BPM (left) and hundredths of a BPM (right).

When the **Ext Clk ON** option is active, the **Tempo** field is a display which shows the current tempo of the external clock source connected to the Vectron's **MCik** input. This display reverts to a default value of *60.00* (BPM) if the external clock is currently not running, as does the Vectron's internal tempo clock.

When the **Ext Clk ON** option is not active, a tempo value can be entered directly into the **Tempo** field, thereby directly setting the tempo of the Vectron's internal clock source. The Vectron will remember this setting if the **Ext Clk ON** option is subsequently activated and will restore the operating tempo to the user-specified value the next time the **Ext Clk ON** option is deactivated.

This setting is stored in (and recalled from) sound presets.



## LFO Retrig

This option offers the choice between *Every Note* and *First Note* retriggering modes for **LFO1** and **LFO2**. These modes are described in the section *Every Note and First Note retriggering modes* in the chapter *The LFO Page*.

This setting is stored in (and recalled from) sound presets.

## Velocity (MIDI Velocity Curves)

The settings in this section control the processing of MIDI note-on velocity values, which can optionally be transformed before they reach the Vectron's synthesis circuits. This can be done in order to change the “action” or velocity response of a keyboard, or to produce specific effects such as constant velocity, inverted velocity etc.



The transformation curve is generated on the basis of three settings and is presented graphically in the square display field in the upper left corner of this section. This display cannot be directly edited.

Note: To defeat velocity value transformation, use the following settings: **Curve** = 4 or 6, **Intensity** = 0, **Offset** = 0.

These settings are stored in (and recalled from) sound presets.

### Curve (curve type)

Selects the basic velocity curve type. The available selections are:

- 1: Linear (increasing)
- 2: Linear (decreasing)
- 3: Constant
- 4: Logarithmic (increasing)
- 5: Logarithmic (decreasing)
- 6: Exponential (increasing)
- 7: Exponential (decreasing)

## Intensity

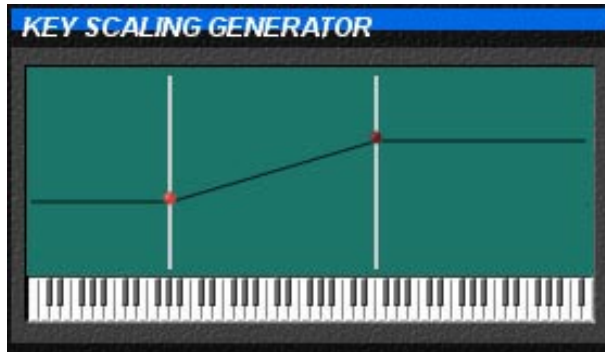
Adjusts the slope of the linear curves and the shape of the logarithmic and exponential curves. Higher settings produce steeper slopes with the linear curves and more extreme “bending” of the logarithmic and exponential curves. This setting has no effect upon curve type 3 (constant).

## Offset

Shifts the entire velocity curve upward or downward, raising or lowering the velocity values at all points along the curve by the same amount. With curve type 3 (constant), this setting alone determines the velocity value for the entire curve.



# Key Scaling Generator



This graphical controller is used to generate a modulation *function* based on keyboard position (MIDI note number). The generated function can be applied as a bipolar (positive and negative) modulation source at various points throughout the device. Since this modulation signal is derived from note numbers, it is a polyphonic modulation source which can affect each note differently.

The **Key Scaling Generator** is adjusted via its two red points. Each of the points can be positioned independently on the keyboard (the keyboard graphic along the bottom corresponds to the 0 .. 127 MIDI note number range) by clicking and dragging the vertical white lines upon which the red points ride up and down. Note that you must click on the white lines *above* their midpoints in order to move them.

The two red points are coupled to one another vertically. Moving one of them upward moves the other one downward, so that they are always roughly the same distance away from the horizontal zero line at center, but on opposite sides of it. Thus, one point represents a positive modulation value and the other one a negative value of the same magnitude. Finally, the three-part black line indicates the resulting modulation curve, which is linear between the two points and then remains constant for note numbers which are beyond the points.

By playing with the vertical and horizontal positions of the points, it is possible to create modulation functions ranging anywhere from an extremely subtle gradual change to a large and abrupt “step” at one point on the keyboard. The latter can be used to emulate split-keyboard setups – for example, by applying it as a modulation source to the vector mix (via the *A-D Mod* and *B-C Mod* signals on the **Vec/Pan** page) or to filter cutoff.

The settings of the **Key Scaling Generator** are stored in (and recalled from) sound presets.

# The Auxiliary Surfaces

The Vectron features a number of additional interface surfaces in addition to its main surface. Two of these surfaces provide additional graphical controller elements which serve as modulation sources in the Vectron and are useful during performance. These surfaces are described in this chapter.

Note: The additional surfaces *not* discussed in this chapter include the **Vector Envelope Edit** drawer (described in the chapter *The Vector Envelope*) and the **Wave Extractor** tool (described in the chapter *Making Waves – The WaveCreate Page*).

## The Joystick Controller

This small auxiliary surface provides a graphical joystick which is intended for use as a performance controller – primarily to permit direct live control of the vector mix, but usable for other control purposes as well.

The **Joystick Controller** is also a MIDI device. Its movements result in the transmission of MIDI controller messages from the Vectron's **MIDI** output – which naturally can be recorded into a sequencer – and it responds to corresponding MIDI controller messages sent to the Vectron's **MIDI** input.

The state of the joystick – its current or initial position, its *AutoReturn* point, and the state of its **AutoReturn ON** switch – are stored in (and recalled from) sound presets.

### Opening the Joystick Controller surface

The surface can be opened (and closed) by clicking on the **Joy** button at far left



on the Vectron main surface. Once open, the surface can be positioned as desired on the screen. The much larger Vectron main surface can then be closed if it is not needed, without affecting the joystick surface. A **Keep On Top** button prevents the joystick surface from being covered by other onscreen objects. The surface also features its own **Close Surface** button.

## Operating the joystick

Working the joystick is simple. Just click-hold on the joystick “handle tip” – the small semi-transparent disc which can be found somewhere within the square control space – and drag it around as desired.

If you do this while playing notes – which is normally the idea! – you will immediately hear the effects which these movements produce. In addition, if the Vectron main surface is open, you will normally see these effects as well, via the **Vector Display** (the **ProgOsc** page must be displayed).

In fact, the **Joystick Controller** can be applied differently from one sound to the next, as described in the following sections. Thus, you could chance upon a sound in which the joystick has no effect upon the vector mix, and therefore also does not register in the **Vector Display**. With such sounds, however, the joystick will usually produce some *other* noticeable effect instead.

## Effect of the joystick on the vector mix

With most Vectron sounds, it will be immediately obvious that the joystick does not “take over” control of the vector mix when it is used. The vector envelope continues to function normally – as do the additional vector modulation signals **A-D Mod** and **B-C Mod**, if they are being used.

The joystick is rather an *additional* element affecting the vector mix – it adds a (two-dimensional) offset to the existing mix. The exact center of the joystick control space is the “neutral” point – when the joystick is positioned at this point, it has no net effect upon the vector mix. When the joystick is positioned elsewhere, it shifts the mix according to its position – that is, corresponding to both its distance and direction with respect to the center point. Naturally, the joystick is a monophonic modulation source which affects all voices in common.

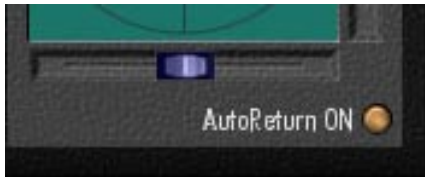
Of course, it is possible to completely eliminate both the vector envelope and the additional vector modulation signals, so that the **Joystick Controller** *alone* determines the vector mix. In this situation, the effect of the **Joystick Controller** upon the vector mix is completely analogous to that of the vector envelope. As with vector envelope editing, the upper left and right corners of the control space correspond to **OscA** and **OscB**, respectively, while the lower left and right corners likewise correspond to **OscC** and **OscD**. Note that this is the *usual* situation – non-typical settings of the joystick amount controls (see below) can produce quite different behavior.

Finally, it’s also possible to “disconnect” the joystick from the vector mix. This is described a bit further on.

## AutoReturn

As soon as you begin using the **Joystick Controller**, you will most likely quickly notice that the joystick springs back to its original position as soon as you release the mouse button. This feature is called *AutoReturn*.

*AutoReturn* is normally quite handy. It ensures that the joystick returns precisely



to a known position when “released” – typically to dead center, where it produces no change in the vector mix. This is important, because the **Joystick Controller** need not be visible in order to produce its effects.

However, you may at times prefer to have the joystick “stay put” when it is released. The *AutoReturn* function is controlled by the **AutoReturn ON** button and can be disabled by switching this button off.

## Changing the AutoReturn position

In some situations, you may wish to change the position to which the *AutoReturn* function returns the joystick – for example, so that it springs back to one corner, rather than to exact center. The *AutoReturn* position can be changed quite simply by moving the joystick to the desired position in the usual way and then holding down the <Ctrl> key on your computer keyboard *while you release* the mouse button. Note: You must have *AutoReturn* activated while doing this.

*AutoReturn* can be switched off and on as often as desired without affecting the *AutoReturn* position. The most recently set position is retained while the function is disabled and becomes effective again when the function is reactivated.

## Centering the joystick

The joystick can be set to exact center by double-clicking the joystick handle tip. In effect, this “zeroes” the joystick, so that it has no effect upon the vector mix.

If *AutoReturn* is active, you must hold down the <Ctrl> key on your computer keyboard while double-clicking. Doing this will also reset the *AutoReturn* position to exact center.



## What are those faders for?

You will no doubt have noticed that the **Joystick Controller** also features a horizontal and a vertical fader. These faders are coupled directly to the corresponding axes of the joystick itself. Via these faders, separate MIDI controller numbers are assigned to each joystick axis. For more information about the **Joystick Controller** and MIDI, read the following section.

## Joystick MIDI

As mentioned earlier, the **Joystick Controller** is also MIDI-capable. Via its horizontal and vertical faders, which are coupled directly to the corresponding axes of the joystick, separate MIDI controller numbers are assigned to each axis.

Default assignments have already been made for you (see below). Thus, when you operate the joystick, MIDI controller messages are correspondingly transmitted via the Vectron's **MIDI** output. These messages can even be used to control external MIDI devices if desired.

The joystick can also be moved via MIDI controller messages sent to the Vectron's **MIDI** input. Modulation destinations driven by the joystick will respond to received MIDI controller messages exactly as if the joystick was being moved directly. Thus, you can use a MIDI sequencer to record your joystick meanderings and later play the sequence back into the Vectron in order to reproduce the original joystick movements. As with direct movement of the joystick, the effect of these MIDI messages depends upon which sound parameters have been programmed for modulation by the joystick in the current sound and can thus vary from sound to sound.

## Routing MIDI controllers via the joystick

Naturally, controller messages sent to the joystick need not originate from the joystick, as long as the controller numbers match. Furthermore, the **Joystick Controller** need not be displayed in order to respond to received MIDI controller messages. Sound parameters which have been programmed for modulation by the joystick will respond to MIDI controller messages assigned to the joystick, even when the **Joystick Controller** surface is hidden. In this sense, the joystick can serve simply as a means of routing MIDI controller messages to sound parameters (with routings programmable per sound) – you need never actually use the joystick itself or even display it.

Because it is a *graphical* controller, the joystick cannot match the speedy real-time performance of a high-quality hardware controller. However, this limitation does not apply to the modulation of sound parameters in response to MIDI messages routed via the joystick. Response to these messages is independent of the graphical representation of the joystick and is essentially instantaneous.



## MIDI controller assignments

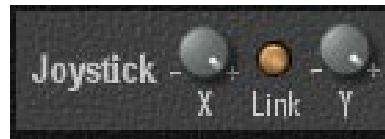
The default MIDI controller assignments for the joystick are *Controller Number 16* (or *10 hex*) for the horizontal axis and *Controller Number 17* (or *11 hex*) for the vertical axis. These assignments can be changed if desired by using the **MIDI Ctrl Assignment** dialog or the **Controllers** dialog in the usual way. However, note that you must modify the assignments to the horizontal and vertical faders on the **Joystick Controller**, since the controller numbers are actually assigned to these faders and not to the joystick itself.

## The joystick amount controls

The **Joystick Controller** is intended first and foremost as a live controller for the vector mix. A permanent connection therefore exists between the joystick and the vector mix circuit.

However, the signal which actually reaches this circuit from the joystick is adjusted by the joystick amount controls in the **Vector Modulation** section on the **Vec/Pan** page. These controls permit reduction, reversal or complete elimination of the joystick horizontal and/or ver-

tical signals being fed to the vector mix control circuit. This makes it possible, among other things, to “tone down” the effect of the joystick upon the vector mix, or to reverse the effect of joystick motion upon the vector mix with respect to another modulation effect driven by the joystick. These controls can also be used to effectively “uncouple” the **Joystick Controller** from the vector mix circuit. The joystick can then be applied to other modulation uses *without* affecting the vector mix.



The settings of the joystick amount controls are stored in and recalled from sound presets. Therefore, the effect of the **Joystick Controller** upon the vector mix *can* vary from one sound to the next. In most sounds, however, the joystick is set to affect the vector mix in exactly the way that one would tend to expect.

Read the next section for more information regarding the use of the **Joystick Controller** as a general-purpose controller in the Vectron. For more information regarding the joystick amount controls, refer to the section *Joystick (joystick controller amount)* in the chapter *The Vec/Pan Page*.



## The joystick as general purpose controller

The **Joystick Controller** is intended first and foremost as a live controller for the vector mix. However, it is by no means limited to this use. You can also apply it to myriad other modulation destinations in the Vectron – and, by means of the MIDI controller messages it generates, to other Pulsar devices or even external MIDI devices as well (see *Joystick MIDI* above). Furthermore, this can be done in tandem with joystick modulation of the vector mix, or the joystick can be “dialed out” of the vector mix and used as a freely-assignable controller (see *The joystick amount controls* above).

Because the **Joystick Controller** is in fact a two-dimensional envelope, it cannot be used *directly* as a general-purpose modulation source – apart from the vector mix itself, all other modulation destinations in the Vectron are one-dimensional (or for you propellerheads out there: *scalar* instead of *vector*). Instead, therefore, various one-dimensional *component* forms of the joystick are provided for this use, as described following.

### *Joystick X and Joystick Y*

These signals are simply the horizontal and vertical components, respectively, of the joystick position.

For the *Joystick X* component signal, values right of center are positive and those left of center are negative. For the *Joystick Y* component signal, positive and negative values are above and below center, respectively.

Because the *Joystick X* and *Joystick Y* component signals are relatively easy to grasp conceptually – and furthermore because each of them can be individually “dialed out” of the vector mix (see *The joystick amount controls* above) – these signals lend themselves most easily to use as modulation sources which are independent of the vector mix.

### *Joystick AD and Joystick BC*

These signals are the *diagonal* components of the joystick position as measured along the lines which connect opposite corners of the joystick control space.

You can visualize what this means most easily by imagining that the X and Y (horizontal and vertical) *measurement* axes are both rotated counterclockwise by 45 degrees. The zero point in both dimensions is still represented by the exact center of the joystick control space. However, the rotated Y axis now cuts through the upper left and lower right (“**OscA**” and “**OscD**”) corners and is therefore called the *AD* axis, while the rotated X axis cuts through the other two corners and is correspondingly called the *BC* axis.

The value of the *Joystick AD* component signal at any point in time is nothing more than the position of the joystick as projected (diagonally, that is!) onto the AD diagonal axis. The maximum positive value is obtained when you move the joystick completely into the upper left (“**OscA**”) corner, while the maximum negative value is correspondingly found in the lower right (“**OscD**”) corner. Likewise, the maximum positive and negative values for the *Joystick BC* component signal are found in the upper right and lower left (“**OscB**” and “**OscC**”) corners, respectively. Thus, this is just an alternative way of looking at the same joystick position.

What does all of this mean in practical terms? Simply this: you can use these modulation signals to modulate a Vectron sound parameter in sync with the joystick position in such a way that the modulation tracks the loudness of a particular oscillator. For example, if you apply the *Joystick AD* signal to filter cutoff with a positive modulation amount setting, then

the filter will “open up” more and more as the joystick causes **OscA** to become louder. If you instead use a *negative* modulation amount setting, the filter will open up further as **OscD** becomes louder – and so on.

Therefore, it makes sense to use the *Joystick AD* and *Joystick BC* signals when the joystick is being used to modulate the vector mix – i.e., when it has not been “disconnected” from the vector mix by means of the joystick **X** and **Y** amount controls in the **Joystick** section on the **Vec/Pan** page (see *The joystick amount controls* above). However, as with virtually everything else in the Vectron, this is merely an option and not a rule which must be consistently applied.

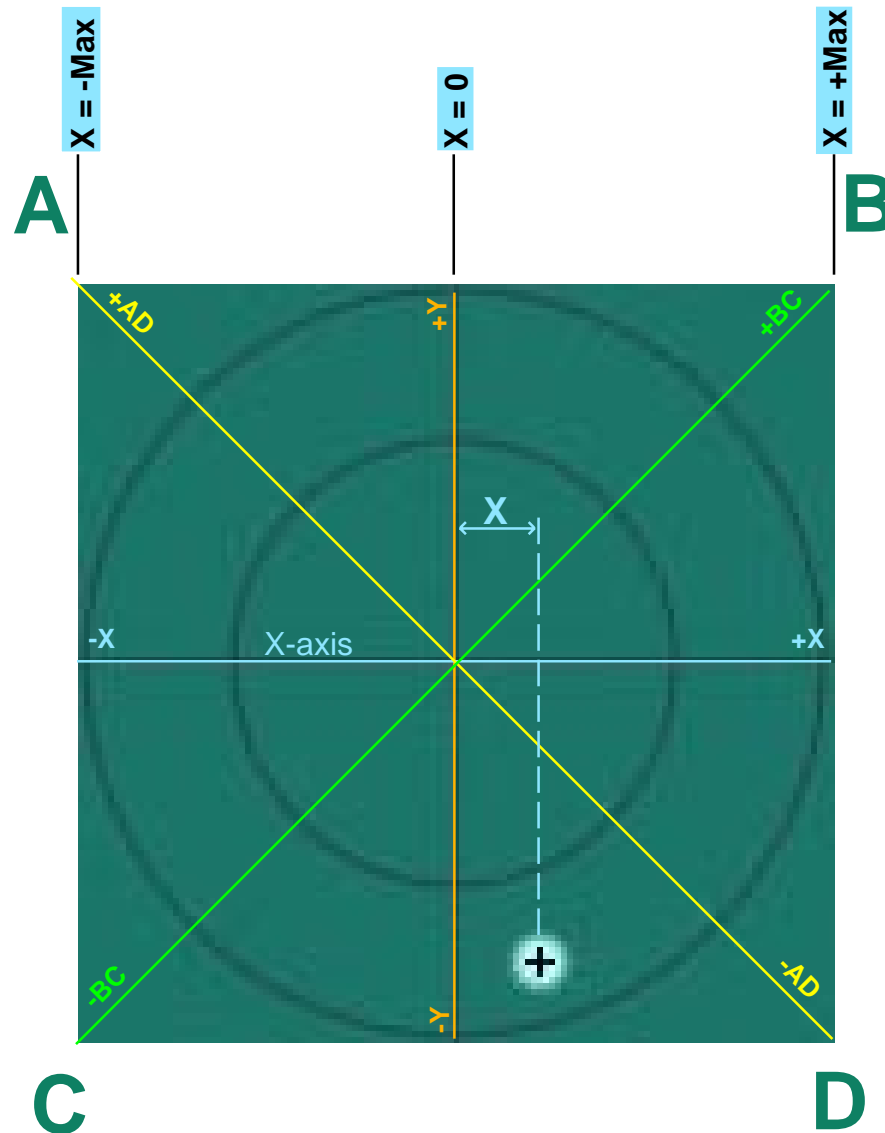
## Component signal diagrams

To help clarify the concept of vector envelope component signals, the following few pages present diagrams depicting the various component signal values for a single vector envelope (or joystick) point position. The same point position is used in all diagrams to emphasize the concept that all of the component values are just different ways of measuring the same signal.

At right, the X component signal value is illustrated. This is the displacement of the point from the (vertical)  $X = 0$  line as measured along the X-axis (blue).

(Note that the  $X = 0$  line (orange) is labelled " $-Y .. +Y$ ", since it is actually also the Y-axis.)

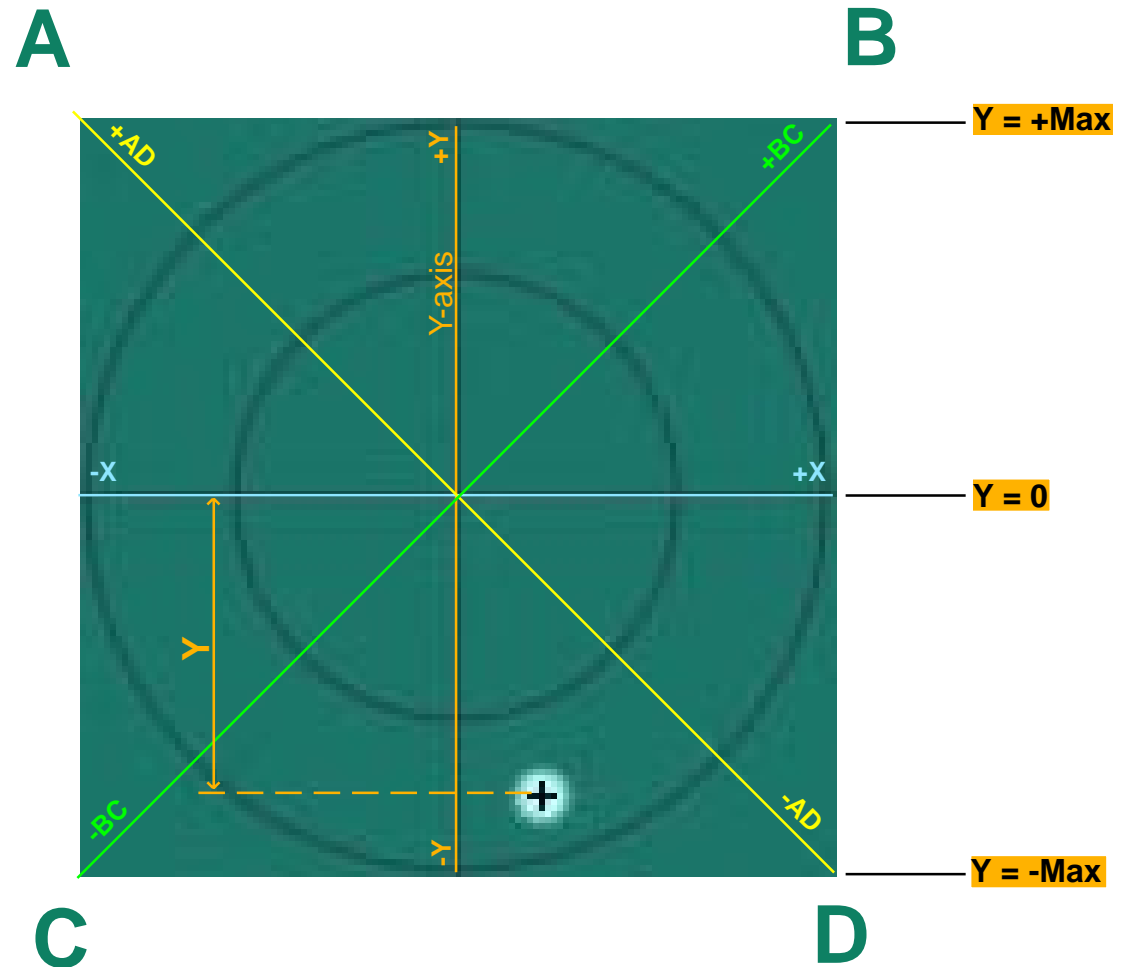
The X component value is indicated by the double-ended arrow labelled "X". In this example, the value is positive, since the displacement is in the  $+X$  direction, and is roughly 20% of the maximum positive value for the X component signal.



At right, the Y component signal value is illustrated. This is the displacement of the point from the (horizontal)  $Y = 0$  line as measured along the Y-axis (orange).

(Note that the  $Y = 0$  line (blue) is labelled "-X .. +X", since it is actually also the X-axis.)

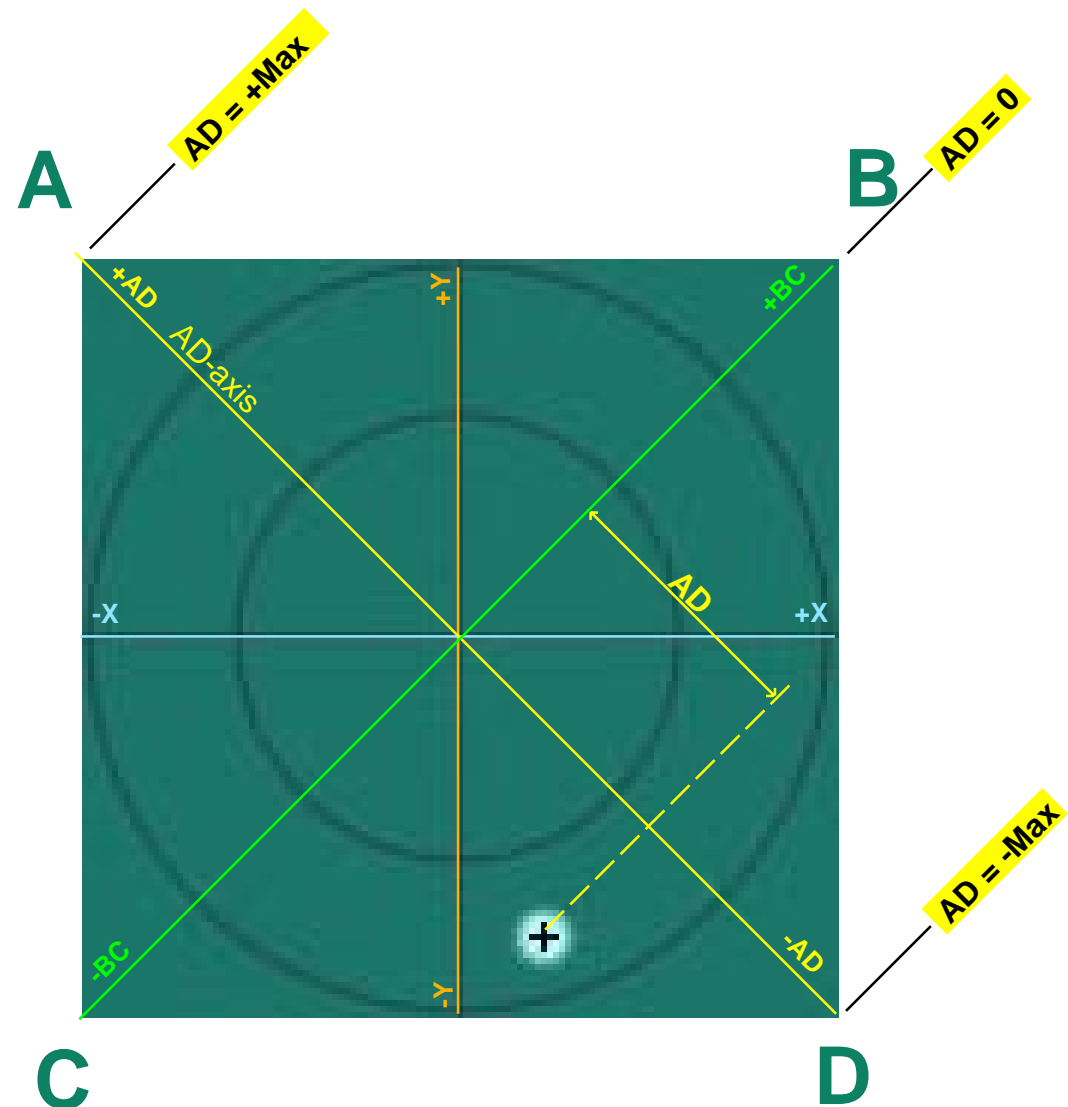
The Y component value is indicated by the double-ended arrow labelled "Y". In this example, the value is negative, since the displacement is in the -Y direction, and is roughly 80% of the maximum signal.



At right, the AD component signal value is illustrated. This is the displacement of the point from the (diagonal)  $AD = 0$  line as measured along the AD-axis (yellow).

(Note that the  $AD = 0$  line (green) is labelled "-BC .. +BC", since it is actually also the BC-axis.)

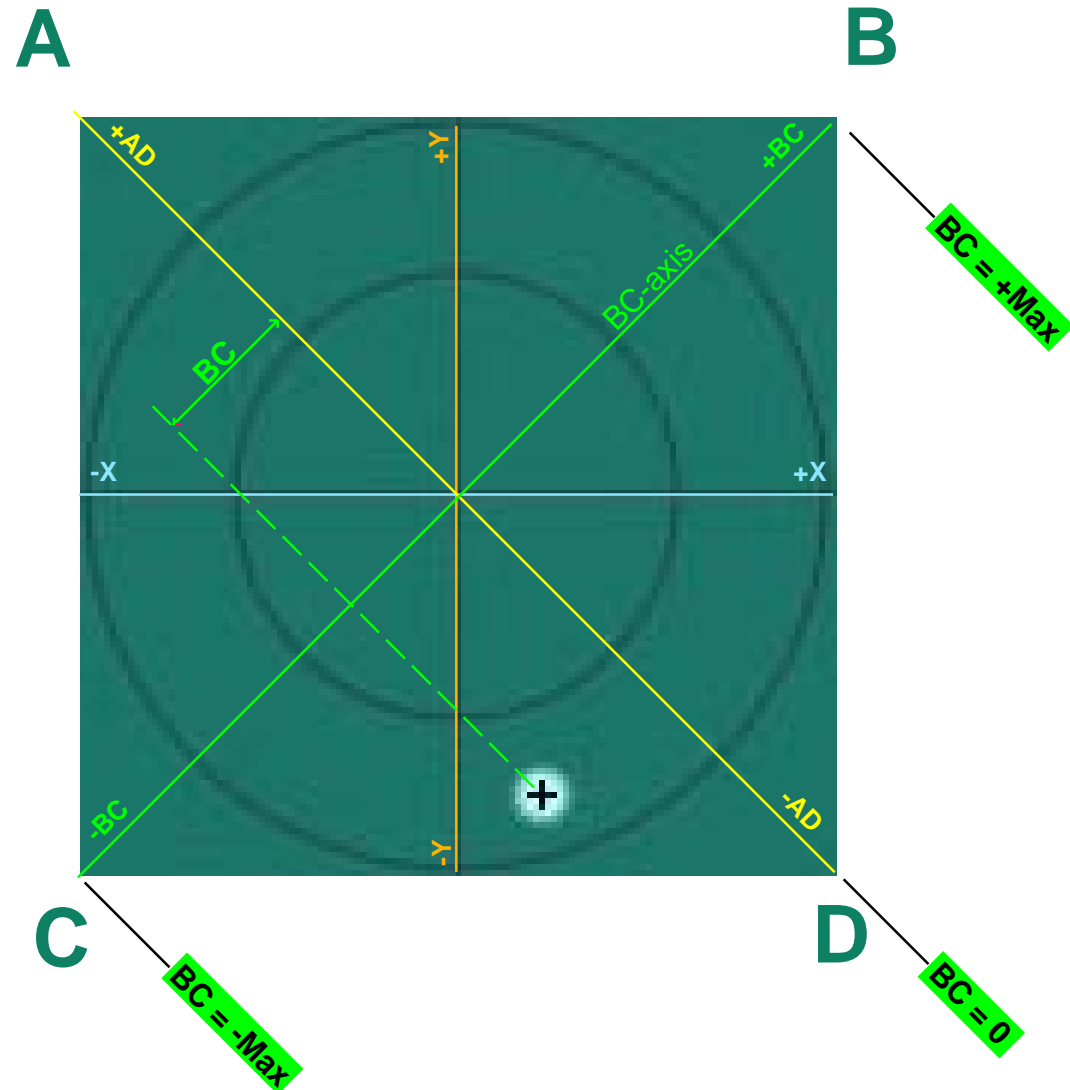
The AD component value is indicated by the double-ended arrow labelled "AD". In this example, the value is negative, since the displacement is in the -AD direction, and is roughly 50% of the maximum negative value for the AD component signal.



At right, the BC component signal value is illustrated. This is the displacement of the point from the (diagonal)  $BC = 0$  line as measured along the BC-axis (green).

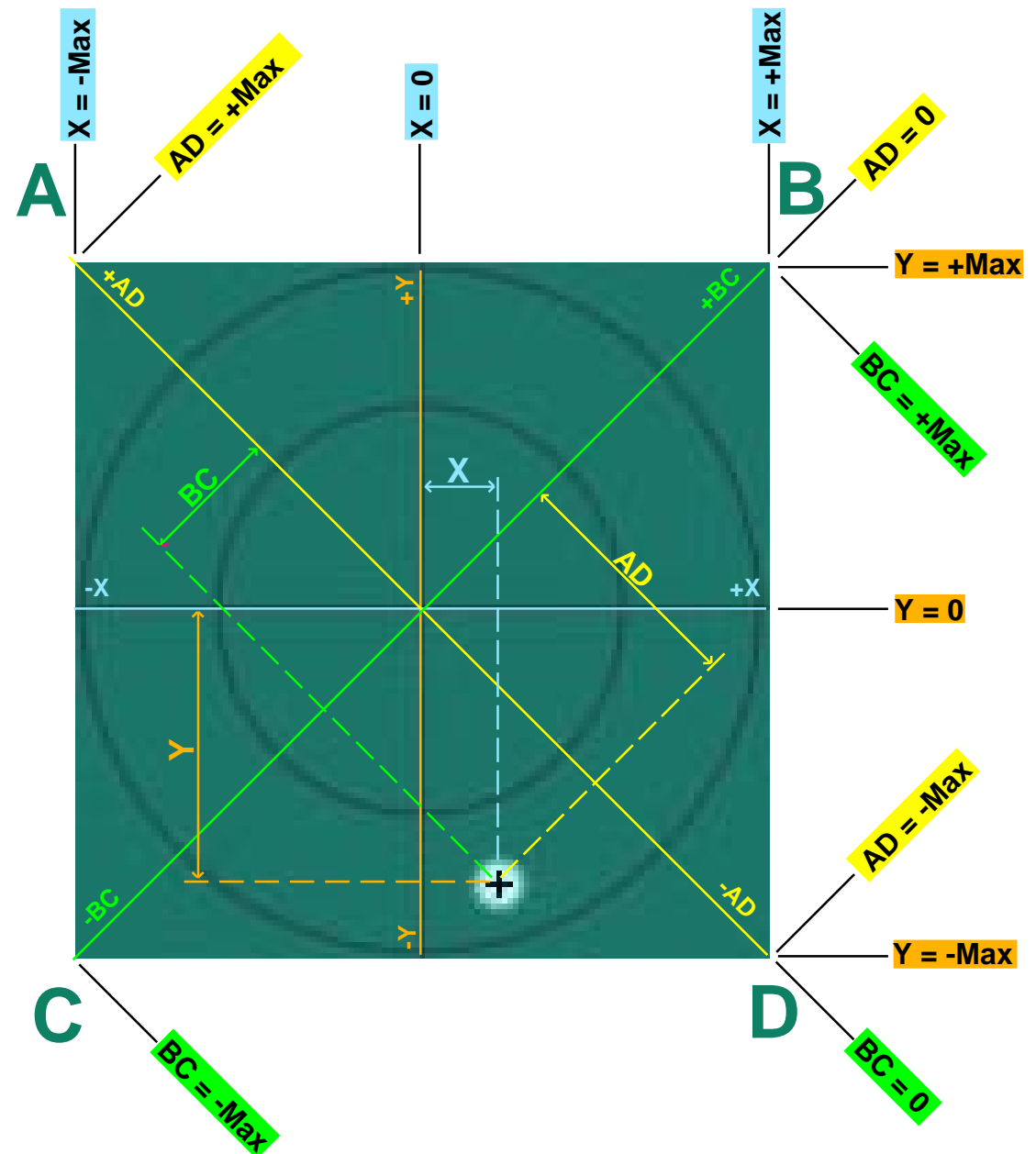
(Note that the  $BC = 0$  line (yellow) is labelled "-AD .. +AD", since it is actually also the AD-axis.)

The BC component value is indicated by the double-headed arrow labelled "BC". In this example, the value is negative, since the displacement is in the -BC direction, and is roughly 30% of the maximum negative value for the BC component signal.





The somewhat complicated-looking diagram at right is merely a combination of the diagrams from the four preceding pages and graphically illustrates how four different modulation signals can be derived from a single vector envelope or joystick.



## If the joystick seems not to work

If the **Joystick Controller** appears not to produce any effect, check the following:

- Make sure that the joystick amount controls (the **X** and **Y** controls in the **Joystick** section on the **Vec/Pan** page) are *not* set to center – the center setting reduces the direct effect of the **Joystick Controller** upon the vector mix to zero. The “standard” position for both of these controls is full clockwise.
- While the **Vector Envelope Edit** drawer is open, activation of either the **Point Solo** or **Env Solo** options will defeat any direct effect of the **Joystick Controller** upon the vector mix.
- While the **WaveCreate** page is displayed, activation of the **Direct Audition** option disables normal vector mixing completely. The **Joystick Controller** has no effect upon the **Wave Mixing** function on this page.

## The Aux Controllers



Given the extensive modulation possibilities in the Vectron, it's clear that the ModWheel alone is not enough. Therefore, the **Aux Controllers** surface is also provided. This small auxiliary surface features four additional graphical controller wheels which are intended for use as general-purpose performance controllers.

The description of the **Aux Controllers** in this section also applies to the graphical **ModWheel** at far left on the main Vectron surface. Apart from their locations, their default MIDI controller number assignments, and of course, their names, all five wheels are identical in function.

The graphical controller wheels are also MIDI devices. Their movements result in the transmission of MIDI controller messages from the Vectron's **MIDI** output – which naturally can be recorded into a sequencer – and they respond to corresponding MIDI controller messages sent to the Vectron's **MIDI** input.

The state of each of the controller wheels – its current or initial position, its *AutoReturn* point, and the state of its **AutoReturn ON** switch – are stored in (and recalled from) sound presets.

## Opening the Aux Controllers surface

The surface can be opened (and closed) by clicking on the **Aux** button at far left on the Vectron main surface. Once open, the surface can be positioned as desired on the screen. The much larger Vectron main surface can then be closed if it is not needed, without affecting the **Aux Controllers** surface. A **Keep On Top** button prevents this surface from being covered by other onscreen objects. The surface also features its own **Close Surface** button.

## Applying the controller wheels

Since they're intended for general-purpose use, the controller wheels are not "hard-wired" to anything in the Vectron. Instead, the effect each wheel produces depends upon its modulation *routings* – that is, the set of parameter(s) for which it is selected as a modulation source. This can be different from one sound preset to the next.

In the Vectron, modulation routings are created at each modulation *destination* by selecting the desired modulation *source* signal. In these selection lists, the output signals of the four **Aux Controllers A .. D** appear as *Aux Ctl A .. Aux Ctl D* respectively (the **ModWheel** signal, naturally enough, is designated as *ModWheel* in these lists). Since all of the wheels are functionally the same, all of them are available for selection at every point where a controller wheel can be used as a modulation source.

Any of the wheels can be selected as the modulation source for any desired number of the possible destinations. There is no fixed upper limit. However, it's not at all unusual to use a particular wheel to

modulate only a single parameter in a sound. The modulation *amount* is adjusted separately at each modulation *destination*, even when two or more destinations are modulated by the same source.

In general, the controller wheels function as *unipolar* sources which produce modulation in one direction only. The output of a wheel is zero when the wheel is at the bottom of its travel and reaches the maximum positive value at the other extreme. For a few exceptional *destinations*, however, the wheels function as *bipolar* sources which produce zero at the center position, and full-scale negative as well as positive values at the bottom and top positions, respectively. These exceptions are **Vec Mod A-D**, **Vec Mod B-C** and **Pan**. A controller wheel which is routed to one of these destinations will nevertheless produce its normal unipolar modulation with any *other* destination to which it is also routed.

Naturally, the wheels are monophonic modulation sources which affect all voices in common.

## Centering a controller wheel

The graphical controller wheels can be set to exact center by double-clicking them. This can be useful with a wheel which is routed to one of the exceptional destinations **Vec Mod A-D**, **Vec Mod B-C** or **Pan**, for which the wheel functions as a bipolar modulation source with zero at center.

## AutoReturn

The graphical controller wheels normally mimic the standard behavior of a “real” controller wheel – they stay in the position you move them to after you let go of them.

However, you may in some cases prefer to have a particular wheel spring back to its original position when it is released – more like a pitch-bend wheel. The *AutoReturn* function is provided for this purpose. It is controlled by the small **AutoReturn ON** button directly below each wheel and can be activated separately for each wheel.

## Changing the AutoReturn position

In some situations, you may wish to change the position to which the *AutoReturn* function returns a controller wheel – for example, so that it springs back to a centered position, rather than to one end of its travel. The *AutoReturn* position can be changed quite simply by moving the wheel to the desired position in the usual way and then holding down the <Ctrl> key on your computer keyboard *while you release* the mouse button. Note: You must have *AutoReturn* activated while doing this.

The centering function described above can be used as follows to set the *AutoReturn* position of a wheel to exact center: first, center the wheel in the normal way by double-clicking it, then press and hold the <Ctrl> key and single-click the wheel again *without* moving the mouse.

*AutoReturn* can be switched off and on as often as desired without affecting the *AutoReturn* position. The most recently set position is retained while the function is disabled and becomes effective again when the function is reactivated.

## Controller wheels and MIDI

As mentioned earlier, the graphical controller wheels are also MIDI-capable. They can both send and respond to MIDI controller messages.

Default MIDI controller assignments have already been made for you (see below). Thus, when you operate a controller wheel, MIDI controller messages are correspondingly transmitted via the Vectron's **MIDI** output. These messages can even be used to control external MIDI devices if desired.

The controller wheels can also be moved via MIDI controller messages sent to the Vectron's **MIDI** input. Modulation destinations driven by a particular wheel will respond to received MIDI controller messages exactly as if the wheel was being moved directly. Thus, you can use a MIDI sequencer to record your controller wheel movements and later play the sequence back into the Vectron in order to reproduce the original movements. As with direct movement of the wheels, the effect of these MIDI messages depends upon which sound

parameters have been programmed for modulation by the wheels in the current sound and can thus vary from sound to sound.

### Routing MIDI controllers via the wheels

Naturally, controller messages sent to the wheels need not originate from the wheels, as long as the controller numbers match. Furthermore, the wheels need not be visible onscreen in order to respond to received MIDI controller messages. Sound parameters which have been programmed for modulation by a wheel will respond to MIDI controller messages assigned to that wheel, even when the **Aux Controllers** surface is hidden. In this sense, the wheels can serve simply as a means of routing MIDI controller messages to sound parameters (with routings programmable per sound) – you need never actually use the wheels themselves or even display them.

Because they are *graphical* controllers, the wheels cannot match the speedy real-time performance of a high-quality hardware controller. However, this limitation does not apply to the modulation of sound parameters in response to MIDI messages routed via the wheels. Response to these messages is independent of the graphical representation of the wheels and is essentially instantaneous.

### MIDI controller assignments

The default MIDI controller assignments for the graphical controller wheels are *Controller Number 20 .. 23* (or *14 .. 17 hex*) for **Aux Controllers** *Aux Ctl A .. Aux Ctl D* and *Controller Number 1* for the *ModWheel*. These assignments can be changed if desired by using the **MIDI Ctrl Assignment** dialog or the **Controllers** dialog in the usual way.

# Making Waves – The WaveCreate Page



## Introduction

With the Vectron, you are not limited to working with the set of wavetables which is built into the device. You can easily create new waves via a variety of methods:

- Existing wavetables from the fixed wavetable set, the user wavetable set or the current sound preset can be combined and mixed as desired.
- Wavetables can be imported directly from sample (.WAV) files.
- Waves can be drawn “freehand” using the mouse.

The result of any of these methods is a wavetable in the Wave Edit Buffer. It is possible to use the new wave directly in sound presets while it is still sitting in the buffer. More typically, however, new waves are given names and then stored to locations in the user wavetable list –

and from there, exported into user wavetable list preset files – thus becoming a permanent part of your wavetable library.

Each of the above-mentioned methods of creating new waves is described below. This is followed by descriptions of the various options and functions on the **WaveCreate** page which apply to all of these methods, including the audition and post-processing controls, the functions for naming and storing of new wavetables, and the User Wave Sets dialog used to create wavetable presets and export them into wavetable preset list files.



## Important tips for getting started

### About Wave Edit Buffer sharing

There's no mode switch for selecting which method of creating waves is active – all three methods are available at all times. You can use any method at any time by simply adjusting one of the controls associated with that method.

All of the methods write their results into the Wave Edit Buffer. This means that each method, when it is used, overwrites whatever was in the Wave Edit Buffer before. Fortunately, if a wave you previously created using the wave importing or wave mixing method is inadvertently overwritten (or if you aren't satisfied with the results of your freehand **Draw Mode** modifications to it) you can easily recover it by simply “twiddling” one of the controls associated with that method again (this is explained in more detail below).

However, you must be more careful with waves which you have modified or created via **Draw Mode** freehand drawing. There's no way to recover a hand-modi-

fied wave once the Wave Edit Buffer has been overwritten, if you haven't already stored it to a user wave location as a new user wave.

### Use Direct Audition mode

All of the wave creation techniques described here operate via the Wave Edit Buffer. There are only two ways to hear what's in this buffer:

- You must use a sound preset in which the Wave Edit Buffer is selected as the source wave for one or more of the oscillators, or
- You must activate the **Direct Audition** option (in the **Options** section on the **WaveCreate** page).

For getting acquainted with wave creation, the **Direct Audition** option is strongly recommended. As its name implies, it lets you hear the wave directly. The entire sound preset apparatus – fil-

tering, envelopes, vector modulation, chorus, delay, etc. – is bypassed. Furthermore, this option guarantees that you *will* hear the wave while editing it, no matter what sound preset is currently selected.

Note that the **Direct Audition** option features its own dedicated volume control – **DA Volume**, in the **Options** section. You can use this control to adjust volume level when working in **Direct Audition** mode. (The **Mix Level** control on the **ProgOsc** page affects only the current sound preset and is therefore bypassed in this mode.)

The Direct Audition option is described more fully below – see *The Direct Audition (DA) Option*.

## Disable the post-processing options

To avoid confusion while learning how the wave creation functions work, the post-processing options should initially be adjusted to neutral settings. In the **Options** section of the **WaveCreate** page:

- Set **Xfade Depth** to *Off*.
- Set **Bit Depth** to 16.
- Set **Downsamp Fac** to *Off*.

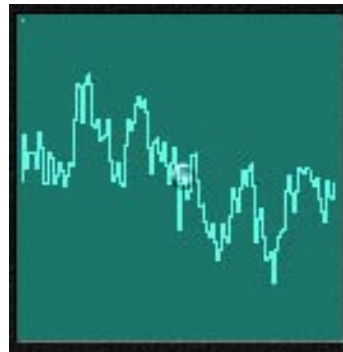
These options are described fully in a later section – see *WaveCreate Post-processing Options*.



## The WaveCreate waveform display

The background of the joystick controller on the **WaveCreate** page is a waveform display which shows the contents of the Wave Edit Buffer. What you see is what you get: the display is continually updated as you adjust wave mix or wave import controls or change any of the wave post-processing settings, providing visual feedback which helps you understand the effects of your actions. And when **Draw Mode** is activated, you can use the mouse to draw wavetables “freehand” directly in the waveform display itself. You can also use the display to view any existing fixed or user wavetable (by setting the mix joystick all the way into the appropriate corner and setting the corre-

sponding **Harmonic** control to 1). The waveform display is not only informative and educational, but makes creating waves with the Vectron quite entertaining as well!



## The Direct Audition (DA) Option



The Direct Audition option is recommended for use with all methods of wave creation.

Direct Audition mode is available *only* when the **WaveCreate** page is displayed. If you leave the **Direct Audition** button (in the **Options** section) switched on, the Vectron will automatically switch Direct Audition mode on whenever you switch to the **WaveCreate** page – and switch it back off again when you switch back to the **ProgOsc** page.

In Direct Audition mode, all sound preset functions are completely bypassed. All that remains is a simple “on-off” amplifier envelope which responds to notes played on the keyboard.

Correspondingly, the vector mix is defeated – or at any rate, “cornered”: Direct Audition playback is done via Oscillator A only, and the Wave Edit Buffer is automatically selected as the source wave for Oscillator A.

This means that you hear *only* the Wave Edit Buffer wave, and that you hear it “dry” – without filtering, effects, envelopes or modulations or even tunings of any kind, other than the basic pitch determined by the notes you play. With only one oscillator in the mix, possible phase-cancellation effects which might mask the true sound of the wave are avoided. All of these measures combine to give you the most faithful possible audio representation of the wavetable in the Wave Edit Buffer.

But the most basic benefit of Direct Audition mode is simply this: it ensures that you *will* hear the wave in the Wave Edit Buffer, regardless of which sound preset is currently loaded and what its selected source waves may happen to be.

You may experience some sound “glitching” if notes are playing when Direct Audition mode is switched on or off. This is normal – a large number of (virtual) switches is being thrown all at once!

## DA Volume

The **Direct Audition** option features its own dedicated volume control – **DA Volume**, in the **Options** section. You can use this control to adjust volume level when working in Direct Audition mode. (The **Mix Level** control on the **ProgOsc** page affects only the current sound preset and is therefore bypassed in this mode.)

## DA Grunge

This control does not affect the contents of the Wave Edit Buffer in any way, but merely modifies the way in which they are heard in Direct Audition mode. It is the counterpart to the **Grunge** control on the **ProgOsc** page, which is bypassed while Direct Audition mode is active. The **DA Grunge** control is active only in Direct Audition mode.

The **DA Grunge** control (like the **Grunge** control on the **ProgOsc** page) controls the amount of *interpolation* (waveform smoothing) performed by the Vectron during wavetable playback. There is no specific recommended setting for this control, other than to adjust it to a setting similar to the settings you typically use for the **Grunge** control in your sound presets. This provides you with the most realistic representation of a wavetable with regard to how it will sound when you use it in your own sound presets.

## Temporary disable while Sound Preset List is open

For reasons of device architecture, sound presets cannot be stored while the **Direct Audition** option is active. Therefore, opening the sound preset list dialog will temporarily deactivate **Direct Audition** if it is enabled. The setting of the **Direct Audition** switch is not affected, and the original status of the **Direct Audition** option is restored as soon as the sound preset list dialog is closed.

This exclusion does not apply to *user wavetable* presets. These are saved independently of sound presets via a separate preset list dialog which is accessed from the **WaveCreate** page.

## Wave Mixing

To create waves from a mix of existing waves, open the **WaveCreate** page. Here you will find a layout whose upper section is largely similar to the **ProgOsc** page and which is operated in a very similar manner.

As with the synthesizer itself, wave mixing is done in “vector style”. You select any four desired existing wavetables and mix them using the “joystick” controller in the middle of the page. The resulting mix is guaranteed to be free of clipping.

**Important:** The **Direct Audition** option on the **WaveCreate** page should normally be switched **ON** when mixing wavetables. If this option is switched off, you will be unable to hear the wave you are mixing, unless the current sound preset happens to include the **Wave Edit Buffer** as one of its source waves. Please refer to the detailed description of the **Direct Audition** option (below).



Joystick

## Selecting waves for the mix

Selection of source waves for wave mixing on the **WaveCreate** page is done in exactly the same way as selection of waves for oscillators on the **ProgOsc** page. Refer to the *Select Wave* section in the chapter *The ProgOsc Page* for details.

As mentioned above, existing wavetables from the fixed wavetable set, the user wavetable set or the current sound preset can be selected for use in a wave mix – in other words, any selection which can be used in a sound preset, except for two:

- **Noise:** In a sound preset, this selection introduces a signal from a digital noise generator. Since this signal is not – and cannot be – derived from a static wavetable, it is not possible to provide *Noise* as a source wavetable for wave mixing. Therefore, this selection is not available on the **WaveCreate** page.

## The Harmonic controls

- **Wave Edit Buffer:** While it might be an interesting problem for “recursion freaks”, we chose not to grapple with the philosophical questions which would be raised by permitting the Wave Edit Buffer to be mixed into itself.



In place of the coarse and fine tuning controls found on the **ProgOsc** page, the **WaveCreate** page features **Harmonic** controls. Rather than tuning the waves in increments of semitones and cents, these controls multiply the “frequency” of a source wavetable by a whole-number value. There are two main reasons for this difference:

- When mixing single-cycle wavetables to create a new single-cycle wavetable, “tuning” of the source waves in standard musical intervals is meaningless. The pitch obtained when the resulting wavetable is used in a sound preset will still depend entirely upon the preset’s oscillator tuning settings.

- Single-cycle wavetables are essentially extremely short sample loops. With sample loops in general, it’s desirable to have a good match between the start and end of the loop. With single-cycle loops in particular, the sonic “trash” which results from a start/end mismatch produces not merely an occasional glitch but instead becomes an integral part of the tone – it’s built into every cycle of the sound – and often completely dominates it. “Tuning” of source wavetables in semitones/cents during wave mixing tends to produce such a mismatch, because the wavetable generally does not repeat a whole number of times and is still “somewhere in the middle” at the end of the mix. However, source wavetables which have a good start/end match can be safely tuned in *harmonic* intervals during mixing. With these intervals, the *entire* wavetable is repeated a whole number of times and the good start/end match is preserved.



The effect of the **Harmonic** control on a source wave can be understood quite easily by viewing the changes it produces in the waveform display (set the mix joystick fully into the corner associated with the source wave whose **Harmonic** control you're adjusting).

## The *Use Waves From Preset* button

The **Use Waves From Preset** button provides a convenience function which copies the wave selections from the current sound preset into the **WaveCreate** page. It also resets the **Harmonic** controls back to the default value of 1 for all four mix source waves.



## Extracting waves from sound presets

The Wave Mix utility can be used to extract “alien” wavetables from imported sound presets – i.e., wavetables not already present in the device fixed wavetable set or available in a user wavetable list. To do this, select one of the preset’s waves (*PreA*.. *PreD*) and mix it to 100% (by moving the mix joystick all the way into the corresponding corner), and then store the resulting mix as a new user wavetable.

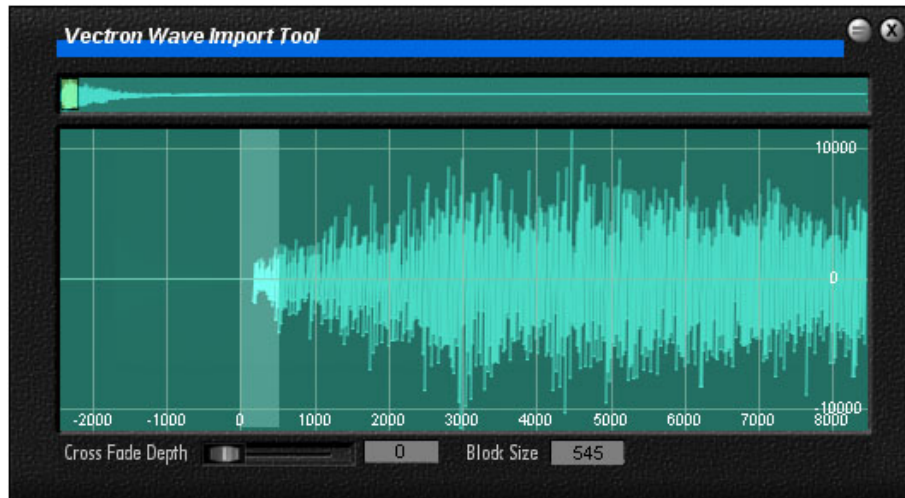
To ensure that you obtain an exact copy of the wavetable data, you must make sure to first disable the post-processing options: in the **Options** section of the **WaveCreate** page: set **Xfade Depth** to *Off*, set **Bit Depth** to *16*, and set **Downsamp Fac** to *Off*. Otherwise, these options will have an effect upon the copied wavetable, which will thus differ from the original, even if only slightly. To furthermore obtain a true exact copy of the complete wave, you must also enter the original name of the wave into the **Edit name** field before storing the copy.

The name should be entered exactly as it appears in the **Select Wave** field – but *without* the prefix (“PreA:“, “PreB:“, etc.). If you have done both of these things correctly, the Vectron will be able to recognize and match the new wave if it appears in any additional presets you load.

## Recovering an overwritten mix

If you inadvertently “wipe out” a wave mix – by using **Draw Mode** or the **Wave Extractor** tool – you can recover it again by simply “twiddling” one of the wave mixing controls away from and then back to its current setting. For example, you can adjust any *one* of the four **Select Wave** (wave number) or **Harmonic** controls one step up and then back down again. Or, you can click on a **Select Wave** (wave name) field and then hit <Enter> on your computer keyboard *without* selecting a new wave, thus reselecting the same wave. Any of these actions causes the mix to be created anew and written into the Wave Edit Buffer.

## Waves From Samples – The Wave Extractor Tool



**Important:** The Direct Audition option on the WaveCreate page should normally be switched ON when using the Wave Extractor tool. If this option is switched off, you will be unable to hear the waves you import, unless the current sound preset happens to include the Wave Edit Buffer as one of its source waves. Please refer to the detailed description of the Direct Audition option (below).

Furthermore, note that the Wave Extractor tool can be used at all times, regardless of whether the WaveCreate page is displayed. However, the Direct Audition option is available *only* when the WaveCreate page is displayed. For the sake of simplicity, it is recommended that the Wave Extractor tool be used in conjunction with the Direct Audition option (and therefore also with the WaveCreate page displayed), especially when learning how to use it.

## Getting started

Click on the **WeX** button (at far left on the main surface, above the graphical ModWheel) to open the **Wave Extractor** tool, which is similar in form and function to the Sample Editors found in the STS series of samplers from CreamWare.

Samples (.WAV files) can be loaded into the **Wave Extractor** tool by dragging them from the File Browser and dropping them onto the tool's large display area. To begin the wave importing process, hold down the <Alt> key on your computer keyboard and double-click on this display area. This defines an *import block* at the location in the sample where you double-clicked. The import block – which is nothing more than a selected section of the sample – appears highlighted in pale green in the Wave **Extractor** display.

The size of the import block is initially determined by the setting in the **Block Size** field on the **Wave Extractor** tool, but can subsequently be changed as described below. It can range anywhere from 2560 sample words down to a minimum of 128 sample words. Thus, it is al-

ways quite small compared to the typical sample file – and therefore usually appears merely as a thin vertical line if you haven't yet zoomed the display in – but the idea is, after all, to extract single wave cycles from the sample.

The sample data contained within the import block is immediately transferred into the Wave Edit Buffer when you first create the import block (you'll see it in the waveform display on the **WaveCreate** page) and is transferred anew into the buffer every time you move the import block or change its size using the operations described below.

## Some behind-the-scenes info

As mentioned above, the sample data within the import block is immediately transferred into the Wave Edit Buffer when you first create the import block and again every time you move it or change its size.

In fact, however, the sample data is not merely copied into the Wave Edit Buffer, but is also simultaneously *resampled* in order to fit the selected section of the sample into exactly 128 words – the wavetable size required by the Vectron.

At the same time, the wave data is *normalized* order to achieve the maximum possible waveform level without clipping, even when the level of the sample data is substantially lower. As a result, the wavetables you create using this tool will automatically match each other quite well with regard to volume level.

Automatic level normalization and the ability to freely adjust the size of the import block are important factors which make it not only possible, but also easy to quickly import *usable* single cycles from virtually any sample file into the Vectron. A further helpful tool is the crossfade option (**Xfade Depth** – described below).

## Basic Wave Extractor operations

Descriptions of the most frequently-used operations in the **Wave Extractor** tool are presented here.

### Resizing the Wave Extractor tool

Resize the Wave Extractor tool by clicking and dragging on its left, right or bottom edge or on either of its lower corners.

### Loading a sample into the tool

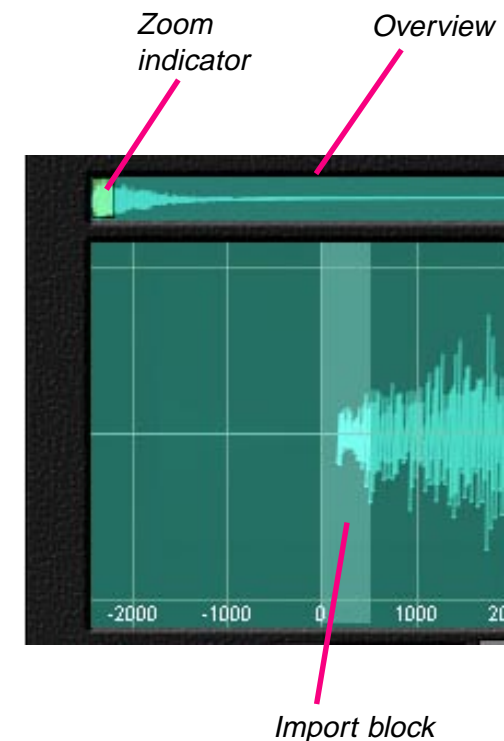
Load a sample (.WAV file) into the tool by dragging it from the File Browser and dropping it onto the tool's large display area.

### Zooming in on a section of the sample

Zoom in on a section of the sample by right-clicking and dragging across the section to be enlarged. This can be done either in the large main sample display area or in the smaller overview display above the main display. Only the main sample display is zoomed. In the overview display, the zoomed section is indicated by a yellow-green rectangle. Since the overview display always shows the entire sample file, you can also use this technique there to zoom the main sample display *out* as desired.

### Zooming the display all the way out

Zoom all the way out by double-clicking on either the main sample display (outside the pale-green highlighted import block) or anywhere in the overview display.



## Scrolling through the sample

Scroll left and right through the sample by clicking and dragging on the main sample display or on the yellow-green zoom indicator block in the overview display. (Scrolling is possible only when the main sample display is not zoomed all the way out.)

## Creating or repositioning the import block

Create the import block by holding down the <Alt> key and double-clicking in the tool's large display area at the location where you wish to create the block. If an import block already exists, it can be quickly moved to any desired location in the sample using the same technique. The new (or repositioned) import block is transferred immediately to the Wave Edit Buffer.

## Shifting the import block left and right

Shift the import block left and right by clicking and dragging in the middle of the pale-green highlighted block in the main sample display. To do this effectively, you may need to zoom the main sample display in until the import block highlight is large enough that you can easily click in the middle of it. The size of the import block remains unchanged.

For fine position adjustments, you can also click once on the import block to select it and then use the cursor-left and cursor-right (arrow) keys on your computer keyboard to shift the import block left or right one sample word at a time – likewise without changing its size.

The import block is transferred immediately to the Wave Edit Buffer each time its position is changed.

## Moving the start or end of the import block

Move the start or end of the import block by clicking and dragging on the corresponding edge of the pale-green highlighted block in the main sample display. To do this effectively, you may need to zoom the main sample display in until the import block highlight is large enough that you can easily click on the desired edge.

The resulting changes in the size of the import block are reflected in the **Block Size** field. Note that it is not possible to make the import block smaller than 128 words or larger than 2560 words.

The import block is transferred immediately to the Wave Edit Buffer each time its start or end is moved.



## Directly setting the size of the import block

Directly set the size of the import block by entering the desired size value in the **Block Size** field. In practice, the **Block Size** field itself is not typically used to set the size of the import block, because it is much more effective to adjust the import block graphically to fit the waveform being imported.

## Using the Wave Extractor tool effectively

The basic strategy for using the Wave Extractor tool is simple: Load a sample file, zero in on an interesting-looking portion of the sample, zoom the view in until individual cycles are clearly visible, create an import block and then tweak its length and/or position until a good-sounding imported wave results. Because you can immediately see and hear the results of every adjustment you make, the actual process requires only slightly more time than it takes to read this paragraph!

The tactic for consistently obtaining usable waves is almost as simple – it's the same tactic which applies to creating clean sample loops: the transition from the end of the sample or wavetable back to the start should be a smooth one, without a large level jump and without introducing an abrupt “bend” in the waveform. To put it in more technically correct terms: the transition from end back to start should not produce an abrupt change in the *slope* of the waveform. The following techniques will help maximize your chances of success:

- Try to recognize complete waveform cycles in the sample file and adjust the import block accordingly.
- For purposes of visual orientation, set both the start and end of the import block at zero-crossing points. This is not necessary for the Vectron – it just makes your task easier.
- Avoid setting the start and end at steeply-sloped points in a waveform cycle – this makes it harder to achieve a smooth end-to-start transition.

- When you're pretty close – magnify the display until the sample scale along the bottom reads individual sample words, and then use your *ears* to find the absolute best location for the start and/or end point. In general, this will be quite easy to recognize.
- The crossfade option (**Xfade Depth** – described below) can be helpful in “making ends meet” – it literally forces the result by modifying the extracted wavetable. However, it can therefore also significantly alter the sound of the wavetable. Quite often, a wavetable which already sounds good “raw” (because the start and end points have been well-chosen) will actually end up sounding *worse* when crossfading is applied. If you choose your wave cycles carefully and consistently apply the above techniques, you'll generally be able to get good results without resorting to crossfading.

## Recovering an overwritten imported wave

If you inadvertently “wipe out” an imported wave – by using **Draw Mode** or the wave mixing functions – you can recover it again by simply clicking on the **Block Size** field and then hitting <Enter> on your computer keyboard *without* typing in a new value, thus reentering the existing value. This causes the wave to be imported anew and written into the Wave Edit Buffer exactly as it was previously.

## Draw Mode – Sketching Waves Freehand

This option allows you to draw wavetables directly in the **WaveCreate** waveform display using the mouse. You can use it to create completely new waves or to tweak existing waves created via wave mixing or imported using the **Wave Extractor** tool – and in the process, you can put both your manual dexterity and your knowledge of acoustics to the test!

To begin drawing, all you need to do is switch the option on via the **Draw Mode** button directly above the waveform display. Then click and drag in the display and start drawing!

## A note of caution

Unlike imported or mixed waves, a free-hand-modified wave is “unprotected”. If you inadvertently “wipe out” such a wave – by using the **Wave Extractor** tool or the wave mixing functions – there is no way to recover it again, if you haven’t already stored it to a user wave location as a new user wave.



## WaveCreate Post-processing Options



### The Crossfade Option (Xfade Depth)

This is a “post-processing” option whose effect is adjusted via the **Xfade Depth** control. It is applied to the results of a wave mix, wave extraction or wave drawing operation before the final result ends up in the Wave Edit Buffer and the waveform display.

The crossfade function forces the start and end of the wavetable to match one another in level by “tilting” the respective ends of the waveform toward one another over the indicated number of sample words. This can be quite helpful in quickly obtaining usable results with wave drawing and extraction of wavetables from

samples. The effect of this function can be seen quite clearly in the waveform display as you vary the **Xfade Depth** setting.

However, note that crossfading is not a magical cure-all. Often, a wavetable created via the **Wave Extractor** tool which already sounds good “raw” (because the start and end points have been well-chosen) will actually end up sounding *worse* when crossfading is applied. When working with the **Wave Extractor** tool, it is recommended that **Xfade Depth** normally be set to *Off*. When you’re done fine-trimming the import block, you can try a bit of crossfading and decide whether you prefer the crossfaded or “raw” version of the extracted wavetable.

With wave *drawing*, on the other hand, crossfading usually brings about a noticeable improvement. It’s pretty difficult to consistently hand-draw a good start/end match into a wavetable!

### The Bit-Depth Reduction Option (Bit Depth)

This is a “post-processing” option whose effect is adjusted via the **Bit Depth** control. It is applied to the results of a wave mix, wave extraction or wave drawing operation before the final result ends up in the Wave Edit Buffer and the waveform display (but *after* crossfading, if the crossfading option is active).

The Bit-Depth Reduction function lets you add a touch of “lo-fi” to your wavetables. It simulates the effect of decreasing the bit resolution of the wavetable data from 16 bits down to any desired lower number of bits – all the way down to one bit, if that’s what you want! The effect of this function can be seen quite clearly in the waveform display as you vary the **Bit Depth** setting.

## The Downsampling Option (Downsamp Fac)

This is a “post-processing” option whose effect is adjusted via the **Downsamp Fac** control. It is applied to the results of a wave mix, wave extraction or wave drawing operation before the final result ends up in the Wave Edit Buffer and the waveform display (but *after* crossfading, if the crossfading option is active).

The Downsampling function lets you add a touch of “lo-fi” to your wavetables. It simulates the effect of decreasing the effective sample rate at which the wavetable is captured – as if the wavetable no longer contained 128 words, but some smaller number instead. The effect of this function can be seen quite clearly in the waveform display as you vary the **Downsamp Fac** setting.

## Auditioning Waves Via Sound Presets

Although Direct Audition mode is generally a good option when creating and editing wavetables, it isn’t the only option. There are situations in which you may prefer to give up the neutrality of Direct Audition mode and instead hear the wave you’re editing in the context of a preset – but without having to go through the steps of storing the wave to a user location and updating the sound preset each time you tweak the wave.

### Techniques

The Wave Edit Buffer can be selected on the **ProgOsc** page as the source wave for any or all oscillators. Assuming that you’ve created such a sound, there are a few different ways to make use of it:

- Leave the **Direct Audition** button switched on. The Vectron will operate in Direct Audition mode as long as the **WaveCreate** page is open. Whenever you switch to the **ProgOsc** page, Direct Audition mode will be disabled and the Vectron will return to preset

mode. Naturally, you can’t go any further with wave mixing or wave drawing until you return to the **WaveCreate** page (although the **Wave Extractor** *can* still be used, since it is independent of the WaveCreate page).

- Use the **Direct Audition** button to switch between Direct Audition mode and preset mode while keeping the **WaveCreate** page displayed. You can thus alternate freely between hearing the wave “dry” and hearing how it sounds in the current preset, without interrupting the flow of your wave editing.
- Leave the **Direct Audition** button switched off. The Vectron will operate continuously in preset mode, even while the **WaveCreate** page is displayed. All wave editing operations are possible, and you can also edit the preset as desired, without encountering the sound “glitch” which occurs when Direct Audition mode is switched on or off.

## Special handling of the Wave Edit Buffer in presets

The first two of the above-mentioned techniques for auditioning waves via sound presets are appropriate for the typical wave “development” process. In this connection, it’s useful to be aware of the special way in which the Vectron handles sound presets which use the Wave Edit Buffer.

When you recall such a preset, the oscillators which were set to use the Wave Edit Buffer as the source wave will continue to do so. Therefore, if you recall the preset in a different session than the one in which you created it, chances are good that you won’t hear the same wave that you were hearing then.

However, the preset *does* contain an embedded copy of the wavetable which was in the buffer at that time – as one (or possibly more) of the *PreA .. PreD* wave selections which every preset contains. So you *can* get it back if you want it. This is nice to know, if you’re working fast – or just feeling lazy – and don’t want to bother with naming and storing your new waves and tweaking your presets. (Naturally, we don’t mean to encourage this sort of approach!)

## Wave creation tools as performance controls

The third of the above-mentioned techniques for auditioning waves via sound presets opens up the possibility of “misusing” the wave creation tools as an additional set of *performance* controls. Thus, you not only have the full synthesis capabilities of the Vectron at your disposal – including vector mixing of up to four wavetables – but you can also *modify* one of these wavetables while playing it, in a variety of interesting ways. Some suggestions:

- Use the Wave Edit Buffer as the source wave for two, three or all four oscillators in your sound. Tune the oscillators in interesting (or weird) intervals.
- Vary the **Harmonic** control settings of one or more mix source waves (this one’s quite nice!)
- Vary the **Downsampling Factor** setting (also very nice!)

- Turn up the **Xfade Depth** control and then, in the **Wave Extractor** tool, *slowly* vary the start or end position of the import block (this one may *really* surprise you!)
- Use **Draw Mode** to make subtle (or not-so-subtle) changes to the waveform while playing it (crossfading is also helpful here).

Experiment! Admittedly, live wave editing as a performance technique may not be for everyone. But there’s no reason *not* to misuse the Vectron in this manner, if you like the sounds you’re getting. No, this will *not* void your warranty!



## Storing User Wavetables

When you come up with a wavetable you like, you can store it as a user wavetable in one of the 128 wave locations reserved for this purpose. It thus becomes part of your working wavetable set and can be used in sound presets just like any other wave.

### The basic steps

Storing a wavetable is done in three simple steps in the **Wave Rename/Store** section of the **WaveCreate** page:

- In the **Edit name** field, enter a new name for the wave.
- Via the **Store to loc** field, select a location to store it to. You can choose any of the user wavetable locations *U 0 .. U 127*. The name of the wave which currently occupies a given location (if any) is displayed in the **Overwrites** field at bottom, to let you know what wave you'll be overwriting if you store the wave to that location.

You can also select the storage location by name via the **Overwrites** field, which is a combo box containing a list of the names of all user wavetables currently present in the Vectron.

- Click on the **Store** button to store the wave.

### Additional details

#### Certain names are not allowed

The Vectron reserves certain wavetable names, which cannot be applied to user wavetables. These include “Noise”, “Wave Edit Buffer”, “—— empty ——”, “unidentified mix”, and all fixed wavetable names.

#### Duplicate names *are* allowed

Duplicate names are allowed with user waves, just as they are with sound presets. Of course, it is not recommended practice to store multiple sound presets *or* wavetables with the same name. But with wavetables, you have a slightly better chance of being able to avoid total confusion, because the wavetables are indexed according to their location numbers (U xxx).



## Store to locations with the name “-empty -”

This name is reserved – you can’t use it to name one of your own wavetables. Therefore, a user wavetable location which shows this name is pretty well guaranteed not to contain a wavetable, and is a good choice for storing a new wavetable.

## If there are no more empty locations

If there are no more empty locations, you can still store a wave to any location you wish – you’ll merely overwrite whatever wave is already stored there. In this sense, the user wavetable list is never truly “full”.

Another option is to build a new list from scratch by loading the user wave set *Default (All Empty)* which comes with the Vectron. This clears *all* user wavetable locations. But before you do this, you should store the existing set of user wavetables as a new user wave set – and ideally, save this wave set to disk in its own wave set file. For information on how to do this, refer to the sections *User Wave Sets* and *Permanent Storage Of User Waves And Wave Sets* which follow.

## Sound presets are *not* automatically updated

Existing sound presets are not automatically updated when you store a wave to a user wavetable location – not even if the wave which was in that location was being used by one or more sound presets, and not even if the new wave is stored under the same name. As explained earlier, each sound preset carries its own private copies of the waves it uses, and there exist no direct links from sound presets back to the wavetable lists.

The wavetables in the user wavetable list (as well as those in the Vectron’s built-in fixed wavetable set) are directly accessed *only* when the **Select Wave** settings on the **ProgOsc** or **WaveCreate** page are modified, and *not* when sound presets are recalled. Therefore, if you modify a user wave and wish to use the modified wave in an existing sound preset, you must recall the sound preset and select the modified wave *anew* as appropriate (even if the name hasn’t changed, the wave data stored in the preset still needs to be updated). If you want to make this change permanent, you must additionally save the updated sound preset again.

Don’t be fooled! If you load a sound preset which uses user wave *U xxx*, and then store a new wave to this user wave location, the new wave *will* appear in the sound when you switch back to the **ProgOsc** page. However, this change is only temporary – the new wave will *not* be present in this sound preset the next time you load it. To make the change permanent, you must resave the sound preset.

## A user wave location is not permanent storage!

As with sound presets, stored user wavetables are not *permanently* preserved until they're somehow committed to disk. Refer to the section *Permanent Storage Of User Waves And Wave Sets* (below) for information on how to do this.

## User Wave Sets

**User wavetable presets** can be created, recalled and managed via a separate preset list dialog, the **User Wave Sets** dialog. This dialog is opened by clicking the **Preset** button directly above the waveform display on the **WaveCreate** page.

A user wavetable preset, or user wave set, is a snapshot of the current contents of *all 128* user wavetable locations *U 0 .. U 127*. Likewise, when a user wave set is recalled, it overwrites all 128 user wavetable locations.



## Creating user wave sets

Creation, recall, renaming and deletion of user wave sets is done in the same way as for sound presets:

- Create a user wave set by clicking on the diskette symbol at bottom, below the *Preset* label (ToolTip: "Save Pre-set").

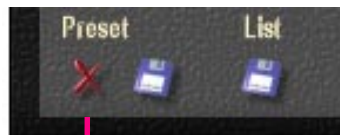


Save preset

If you select an existing user wave set before doing this, you will be asked if you want to overwrite it. If you answer yes, the existing wave set will be replaced with the new one under the same name. If you answer no – or if no user wave set was initially selected – a new user wave set is created with the name *Untitled*. You are immediately given the opportunity to name it, but you can also do this at any later time.

If a user wave set is selected, you can deselect it by clicking in the list window directly to the left of its name.

- Recall a user wave set by double-clicking on its name, or by selecting it and hitting <Enter>.
- Rename a user wave set by selecting it and hitting <F2>. Type in the new name, followed by <Enter> – or hit <Esc> to keep the existing name.
- Delete a user wave set by selecting it and then hitting <Delete> or clicking the red X icon at bottom, below the *Preset* label (ToolTip: “Delete Preset”).



Delete preset

## No connection to sound presets!

Note that existing *sound* presets, including those which use user waves, are *not affected* by changes to the user wavetable list – not even by the loading of a complete new user wave set. As explained earlier, each sound preset carries its own private copies of the waves it uses. The wavetables in the user wavetable list (as well as those in the Vectron’s built-in fixed wavetable set) are directly accessed *only* when the **Select Wave** settings on the **ProgOsc** or **WaveCreate** page are modified – and *not* when sound presets are recalled.

## A user wave set is not permanent storage!

As with sound presets, user wave sets are not *permanently* preserved until they’re somehow committed to disk. Refer to the section *Permanent Storage Of User Waves And Wave Sets* (below) for information on how to do this.

## Permanent Storage Of User Waves And Wave Sets

As with sound presets, a wave which is stored to a user wavetable location is initially only data in your computer’s memory. Nothing is saved to disk via this action – and all of the data will be lost if you exit Pulsar or turn off the computer. The same applies to user wave sets. To store this data permanently, further action is required.

## User wave set files

User wave sets can be exported into files which are similar to the preset list files which you can create for sound presets, but independent of them. This is strongly recommended! Wavetables are the basis of Vectron sounds, and user wave set files can serve the basis of a wavetable library, as well as a medium for backup of your wavetable data and for exchange of wavetables with other Vectron users.

### Saving user wave sets into files

There are two basic ways to save your user wave sets into a file:

- Create a new user wave set file (or overwrite an existing file) by clicking on the diskette symbol at bottom, below the *List* label (ToolTip: “Save Preset List”). All user wave sets in the list will be written into this file.

- Add user wave sets to an existing file: Open the **Preset List Files** drawer of the **User Wave Sets** dialog. Click on the folder icon at bottom, below the *List* label, to open an existing user wave set file, and add individual user wave sets to this file by dragging them one at a time from the device list (left) into the file list (right).

This also works in reverse: you can load individual user wave sets into the device from an open user wave sets file by dragging them from the file list into the device list, where they can be recalled in the usual manner.

Alternatively, you can recall user wave sets directly from the file list by double-clicking on them (or selecting them and then hitting <Enter>) – it’s actually not necessary to load them into the device first.



## A tip for working with user wave set files

Regardless of which method you use to save your user wave sets into files, consider that a single user wave set contains 128 wavetables. This is quite a few! Therefore, although you can save as many user wave sets as you wish into a single file, you may find it easier to keep track of your wavetables if you instead create and maintain a separate file for each user wave set.

It wouldn't make much sense to do this with *sound* presets – normally, you want to have a large number of these presets loaded at one time and instantly accessible. But instant access is not as important with wavetables, which are directly involved only in sound editing and wave mixing.

In practical terms, you sacrifice very little by adopting this approach. The resulting files are relatively small. It's also safer – it's harder to accidentally delete several files than to accidentally delete just one (which hopefully didn't contain your *entire* wave set collection!)

Finally, keep in mind that there's no need to fill up all 128 user wavetable locations with new waves (or even with old waves) before you create a wave set and a wave set file for them – empty wave locations in a wave set or wave set file are no problem. So you can feel free to create a new wave set just for the handful of new waves you've created in a single wave-editing session, and store this wave set into its own wave set file.

## Saving the device

This is another alternative which is available for those who don't wish to export their user wave sets (or their sound presets) into independent external files. Simply click on the diskette symbol directly below the *Device* label at the bottom of the Vectron *sound* preset list dialog. The Vectron device file *Vectron.dev* will be updated – the current contents of the sound preset list, the **User Wave Sets** list and the working user wavetable list (*U 0 .. U 127*) – as well as the current state of the device, even if it hasn't been stored in a preset – are all stored directly into this file.



Save Device

This method has the obvious advantage of being thorough and relatively foolproof. There's more or less no way you can accidentally leave new data unsaved, and it's all accomplished with a single button click. If you suddenly find yourself needing to shut down your project and run out the door, it can be a real lifesaver.

What are the drawbacks of this method?

- It's possible to inadvertently destroy data. If you've deleted sound presets or user wave sets or overwritten user wavetables since you loaded the device, these will be permanently lost when you save the device. Which is not a problem – *if* that was what you intended (or if you have stored them in a separate file).

- All of your data is bundled inside the device. Therefore, this method does not serve very well as a basis for exchange or organized archiving and backup of your preset data.

Whether the advantages of this method outweigh the disadvantages depends entirely upon the way in which you prefer to work with the Vectron.

## Storage of waves within sound presets

As explained earlier, each sound preset carries its own private copies of the waves it uses. You might well find that your wavetable storage needs are adequately met by simply taking advantage of this fact and keeping track of your wavetables according to the sound presets you use them in. Just remember – sound presets must also be saved to disk by one method or another in order to become truly permanent.



# Index

## Symbols

16ths-192nd 63

## A

A-D Mod 69  
Adjusting 47  
Amp/Filter Page 48  
Amplifier 48  
attack 48, 55  
AutoReturn 84, 96  
Aux Controllers 94  
Auxiliary Surfaces 82  
Auxiliary surfaces 16

## B

B-C Mod 71  
Bend 78  
Bit-Depth Reduction 113

## C

Center 49, 58  
Chorus 22  
Coarse 19, 78  
Combo Box 18  
controller wheels 95  
Copy 36  
Cross 24  
Crossfade 113  
Cutoff 53

## D

DA Grunge 102  
DA Volume 102  
decay 48  
Delay 23  
delay 65  
Deleting 47  
Depth 22

Direct Audition 101  
Direct Audition mode 99  
Display 33  
Downsampling 114  
Draw Mode 112

## E

Editing 30, 33  
effect level 22  
Env Solo mode 35  
envelope 53  
envelope amount 72  
envelope points 31  
Ext Clk ON 79  
external MIDI sync 25  
Extracting 106

## F

F 1 .. F 127 11  
fade-in time 66  
feedback 22, 24  
Filter 53  
Filter Envelope 55  
Fine 19, 78  
Finger 21  
fixed wavetables 11  
Freehand 112

## G

Glissando 21  
Global Page 78  
Global Pitch Modulation 76  
graphical joystick controllers 28  
graphical ModWheel 16  
Grunge 20

## H

Harmonic controls 104  
HDamp 24

## I

initial phase 64  
Intensity 80  
Introduction 5

## J

joystick amount controls 86  
Joystick Controller 82  
joystick controller amount 73  
Joystick Controller surface 82  
joystick controllers 28  
Joystick MIDI 85

## K

Key Scaling Generator 81  
keyboard follow 54

## L

legato 21  
Level 22, 25  
LFO 22  
LFO Page 61  
LFO Retrigger 80  
LFO speed MIDI sync 62  
Loading 109  
loops 32

## M

Manual 6  
MIDI 24, 62  
MIDI Velocity Curves 80  
Mix Amount Controls 37  
Mix Level 20  
Modulation options 12  
modulation source 45  
modulation sources 13  
ModWheel 16  
Mono 34

monophonic 14  
Moving 110  
Mstr Tune 78  
multi-segment display 30

## N

Noise 11  
Norm 21

## O

Offset 22, 80  
OscA .. OscD 17  
Oscillator tuning controls 19  
Overview 7

## P

pan base position 74  
Pan Modulation 74  
parameter pages 15  
Paste 36  
Permanent Storage 119  
Pitch Page 76  
pitch wheel range 78  
point mode 32  
point positions 46  
Point Solo mode 34  
Point Step buttons 33  
Polyphonic 14  
Portamento 21  
Post-processing Options 113  
PreA .. PreD 10  
Preface 5  
Preset List 102  
ProgOsc 9, 15, 17  
Programming Tips 46

## R

Rate 62  
Rate Mod 1-2 67  
Recovering 106, 112  
release 48  
Resizing 109  
resonance 53  
retrigger 64

## S

Scrolling 110  
scrolling 32  
Select Wave 17  
Shifting 110  
signal diagrams 40, 89  
SineLFO 68  
Slope 49, 58  
Sound Architecture 7  
Sound Preset List 102  
sound presets 106  
Speed 21, 22  
Speed Controls 36  
Speed Ctrl DISABLE 36  
Speed Mod 57  
Speed Track 49, 58  
Storage 119  
Storing 116  
sustain 48  
Sync 79  
synthesis 7

## T

Tempo 79  
Text Fader 17  
tuning controls 19  
Types of modulation sources 13

## U

U 0 .. U 127 12  
Use Waves From Preset 105  
User Interface 15  
user-created wavetables 12

## V

Vec Disp Mode 25, 27  
Vec/Pan Page 69  
vector concept 26  
Vector Display 27  
Vector Envelope 29  
Vector Mixing 26  
Velocity 80  
velocity 53

## W

Wave Edit Buffer 99  
Wave Extractor Tool 107  
Wave Mixing 103  
Wave Name 18  
Wave Number 17  
WaveCreate 15, 98  
Waveform 62  
Wavetable Categories 10  
wavetable playback 8  
Wavetables 9  
WBuf 10

## X

Xfade 113

## Z

Zooming 32, 109