Circuit Design

Return To Main Table Of Contents

Forward

Sine Synth Tutorial

Basics

- Signal Types
- Range and Resolution
- Help Window and Tool Tips
- Some Important Pads

Procedures in Detail

- Resolutions
- Making Connections
- Device/Module Reference
- Envelopes
- Oscillators
- Modulators

Dialogs

- The Project Explorer Dialog
- Context Sensitive Popup Menus in the Project Window
- Module Exchanger
- Control Ranger

Miscellaneous

- Error Messages

Keyboard Commands

Forward

This section would quickly become unreasonably long were we to try to cover the almost infinite possibilities offered by the large number of modules available in Scope. Instead, this section will use relatively few modules to illustrate clearly the basic elements of circuit design.

Before beginning, activate the Tool Tips in the Help menu. This will of assistance when using the Pads, as the Tool Tips display the various Pads' long names. Also, the current value of certain Pads, such as the 'f-Pad', appears in the Tool Tip display.

You can find specifications for the Modules and Pads in the online Help. Important information such as the types and ranges of the values is detailed here.

In addition to the information found in the online help, this chapter discusses several basic concepts required to understand the usage of modules in Scope circuit design.

This section will also familiarize you with handling the dialogs and using keyboard shortcuts.

Sine Synth Tutorial

Before proceeding on with basic circuit design we'll use the following quick example to acquaint you with the Scope development environment (and create a synthesizer while we're at it!).

First, start the Scope software. When the default project has loaded delete all the modules except the Scope MIDI Source A and the Scope ADAT 1 Destination from the Project Window.

Connect a MIDI keyboard MIDI Out to the Scope MIDI Input. For this example we will use the ADAT 1 Destination channel 1 to monitor the audio.

The remaining modules required for this example are found in the following directory:

\\Scope\Data\Circuit Design

Now we'll verify that we have MIDI input and audio output. In the File browser open the MIDI folder (\\Circuit Design\MIDI) and drag the MIDI Monitor into the Project window. Connect the **Scope MIDI Source A** MIDI output pad to the **MIDI** input pad on the **MIDI Monitor.** In the folder 'Noise' (\\Circuit Design\Oscillators\Noise) you'll find a **Pink Noise** module. Connect it to pad **in 1**. Now we can start designing your first synthesizer on the Scope platform.

\\Scope\Data\Circuit Design**Modules**

Verifying the MIDI connection:

Verifying the audio output:

Delete the MIDI Monitor, and the Pink Noise generator.

Drag the **MIDI Voice Control 16** module (\\Circuit Design\MIDI) into the Project window and connect the **MIDI** pad to the **MIDI** pad on **Scope MIDI Source A**.

Now drag in the **Sine Oscillator** (\\Circuit Design\Oscillator) and connect the two **f** pads. The MIDI Voice Control sends frequency information according to the MIDI note number.

In the 'Envelope' directory you will find an integrated Envelope generator and Amplifier named **Envelope+Amp**. The envelope generator is triggered by information arriving at the **Gate** pad from the voice control. The voice control, in turm requires information from the envelope generator to handle polyphony properly (**ESyn pad** - for **E**nvelope **S**ync). Therefore, both the **Gate** and **Esyn** pads must be connected on the two modules.

Connect the Oscillator **out** pad to the envelope generator input at the **in** pad.

An audio signal is now present at the envelope generator's **out** pad, but in order for the synth to be polyphonic, a final module, **Poly Mix 16** (\\Circuit Design\Basic) must be added. Connect the **out**

pad of the envelope generator to the **in** pad of the Poly Mix 16, and the Poly Mix 16's **out** pad to **in 1.**

You can now play this synthesizer.

In the next step we'll combine the four modules that make up this synthesizer into a single module. It is often desirebale to do this, especially as circuits become more and more complex.

Select the four inner modules with the wireframe selector (drag on the surface with the left mouse button). When they are highlighted, press **<Ctrl><F>** (for Fold) and the four modules will vanish, but a new, 'untitled' module will appear in the upper left corner of the Project Window. Only the pads with which the synth connects to the two external modules are visible now.

Make sure you are in Move mode (**<Ctrl><M>**) and double click on the 'untitled' module. Voila - the four floded modules appear. You are now in the first sublevel of this module.

Note the indicators at the pads which you know to be available at the next level up. Pads connected to modules that aren't folded are automatically **exported**.

Doubleclick anywhere in the Project window (but not on a module) to return to the next higher level.

Before you save this module, you should give it a name. Open the **Project Explorer** from the **Design** menu, and open its **Modules** drawer.

Click on the entry 'untitled' in the Project window or directly in the Module tree.

pad

Click twice slowly over the 'untitled' entry, or use the **<F2>** key to enter edit mode for a tree entry. Type the name 'Sine Synth' and confirm with the **<Return>** key. The name of the module in the project will change appropriately.

The Module Tree provides a hierarchical representation of your project. If you click on one of the '+' symbols, a branch of the tree will open containing other modules.

To save your synth module, right click on the module and choose **Save as New Module**.

To learn how to make your synth module into a full-fledged **Device**, with its own Preset List, read the chapter *Device Design*.

Module Tree

Basics

This chapter will describe the fundamental properties of Modules and Pads, such as their numerical ranges, the types of signals available to them, and how they behave at different sampling frequencies. Before we go on, though, we'll discuss some of the terminology (known as ScopeSpeak).

As you have already seen in the tutorial sections, all the modules in a project are displayed in the **Module Tree.** This structure is very much like Windows Explorer.

To extend the analogy, the Project corresponds to a hard drive, and the modules at the highest level correspond to folders in the master directory of this drive.

Each module can contain additional modules, which can in turn contain modules, and so on.

Each module in the Project window consists of three things:

1. Functionality (it *does* something)

2. One or more connecting points (Pads)

3. Graphical objects

The **functionality** is acquired from DSP routines or through scripts.

The **Pads** provide connection points for audio or other signals. There are three basic kinds of pads - In pads, Out pads, and I/O pads. Graphic displays and pots and faders, for example, can both transmit and receive values.

The **Graphical Objects (GOs)** determine how the Module looks.

In addition to the 'default' Graphical Object used in the Project window you can also create sophisticated user interfaces using GOs. See the chapter "Surface Design" for more information.

Signal Types

Due to the variety of signals available Scope must enforce certain rules when Pads are connected. The following list provides a description of the different signals.

Audio signals, such as those created by an oscillator or modified by a filter, belong to the bipolar, synchronous category. These are indicated in Scope thus: (**bipolar, synchronous**).

Synchronous signals are updated with each wordclock (i.e. in sync with the word clock) whereas asynchronous signals are updated only when necessary (such as when a controller is changed).

MIDI signals carry data in the MIDI format.

Control signals are used for triggering and voice handling (Gate, Sync, Esyn).

Gate signals are used, for example, to start envelopes or other modules. (Asynchronous).

Sync signals are used when phase-accurate starts are required, as with FM operators. (Synchronous).

ESYNC signals carry MIDI voice control information required for MIDI voice handling. (Asynchronous)

Controller signals are those generated by pots, faders or other control elements. These signals are asynchronous, and can be either positive, or bi-polar.

Connections between the pads are represented using the following color coding:

Blue for **synchronous** signals

Red for **MIDI** signals

Green for **asynchronous** signals

It is possible to connect together synchronous and asynchronous Pads through the use of *converters*. However, it is not possible to convert MIDI signals to any other type.

Range and Resolution

Scope uses a 32-bit digital word for DSP processing. This word length provides a very high resolution for both audio and control signals.

For each Pad in a module there is an associated **Help** item that specifies both the input value range and the expected output. Use the Help facility to determine which values are meaningful or appropriate and which are not.

The following items describe the maximum ranges available and how they are referred to in the online Help.

Maximum Range: -2147483648 to 2147483647
In short form this is denoted as:

Max+/- or (bipolar)

Positive Range:
0 to (2^31) -1 = 0 to 2147483647
In short form this is denoted as:

0 to Max+ or (positive)

Negative Range:
-(2^31) to 0 = -2147483648 to 0
In short form this is denoted as:

-Max to 0 or (negative)

Although these are the maximum ranges, not all Pads can use the full range of possible values given by the 32-bit word. An example is the **Delay Time** parameter in the **Delay** module:

In this case the range of allowable values is 0 to 480,000. This DSP algorithm is not designed to handle delay times larger than 480,000 (ms) so values greater than this are not appropriate. This value is specific to this particular pad. Other pads may have similarly specified ranges.

Although a pad may be described as 'bipolar, synchronous', this does not necessarily mean that the full range of bipolar values is useful to it. A Pad's associated processing algorithm ultimately determines the appropriate range.

If a Pad is unconnected a default value is usually automatically assigned to it.

Help Window and Tool Tips

The **Tool Tips** feature, when enabled, displays a Pad's long name (among other things). For example:

Short name = Chan
Long name = MIDI Channel Number

The case of the first letter of a short name is significant in that it tells you something about the range of the pad. If the first character of a short name is capitalized, only positive values are allowed. If it is not capitalized, both positive and negative values are allowed.

More detailed information about a pad is always available in the Help window. The Help window is opened with the **<F1>** key (on PCs).

When the Help window is open, Help text for any particular module or pad is displayed when the mouse cursor is located over the object.

Help describes the most important properties of a module. The first property is always whether or not the module is **polyphonic** or **monophonic**.

When creating multi-voice synthesizers, polyphonic modules are required at *every* stage in the signal chain. Some modules, such as the parametric equalizer, are not polyphonic, and prevent polyphonic capability if used.

Pad properties are described in Help is follows:

1. In an area above the main Help text, the fundamental pad properties are displayed - Range (bi-polar, positive, negative), Signal Type (synchronous, asynchronous) and Data Type (Integer, Long...).

2. The first item described in the Help text itself is the actual permissable range of values. For example (as in the MIDI **Chan** pad):

Input Range = 0 . . 16

3. A description follows, such as how to use or interpret the above values:

Adjusts the MIDI Channel over the range 1 .. 16.
OMNI mode is set by selecting channel 16.

4. Additional pertinent information may also be included.

The range of **Out** pads is identical to the range identifier of the pad. For example:

Pad range is bipolar.
The output range is: -Max .. +Max

Please use the Help window for information on modules and pads as this will always be current and accurate.

Some Important Pads

The following are some important pads you will use frequently:

in pad (Audio Input), in1, in2 .. inL, inR
(bipolar, synchronous)
Audio inputs to connect to sources such as filter or amplifier outputs.

out pad (Audio Output), out1, out2 .. outL, outR
(bipolar, synchronous)
Audio outputs from modules such as filters, oscillators or amplifiers.

f pad (Frequency), f1, f2 ..
(bipolar, synchronous)
Required to adjust oscillators to the desired frequency. This pad is available in modules such as the MIDI Voice Control module so an oscillator frequency can be set to correspond to the MIDI note number.

f pad are scaled as follows:

A frequency range of 0 to 24kHz is generated from values of 0 to +Max.

The highest frequency (24kHz) is only available at sample rates of 48kHz and 96kHz. At 44.1kHz and 32kHz the highest frequencies are restricted to 22.05kHz and 16kHz respectively. These limits result from the Sampling Theorm.

f/Hz = input value * 24000Hz / (+Max)

Negative input values are permitted, and produce an output signal phase shifted by 180°. With filters, the effects of this phase shift are not really meaningful.

F pad (Frequency), F1, F2 ..
(positive, asynchronous)
These pads (denoted by a capital 'F') are scaled just like **f**-Pads, but accept only positive values.

Gain pad (Amplification), Gai1, Gai2, ..
(positive, asynchronous)
The scaling for this pad is linear, but you can implement logarithmic scaling by attaching a log-scaled potentiometer (the control ranger used for this is described later in this section). Most modules use the Gain pad to attenuate a signal only. The amplification factor depends on the respective module.

gain pad (Amplification), gai1, gai2, ..
(bipolar, asynchronous)
For positive values this pad behaves like the **Gain** pad. For negative values it produces a signal shifted 180°.

MIDI pad, M In, MIn1, Mout, MO1, Midi, Mid1 ..
(MIDI Data)
MIDI signals, such as those produced by a MIDI keyboard, can only be transmitted or received through MIDI pads.

Gate pad, gat1, gat2 ..
(Gate signals)
Gate signals are used, for example, to trigger an Envelope Generator from MIDI Note-On and Note-Off events. In this case the signal would appear at the Gate pad on the MIDI Voice Control module. Gate signals are also used by the FM Oscillators and LFO modules.

Val pad (Value)
(asynchronous, positive, controller signal)
All pots, faders, and range textfields have a Val pad. Controller values are transmitted or received at Val pads.

Procedures in Detail

The following sections provide some simple examples to help foster your understanding of the different groups of modules. You will also get some practice using the various dialogs. First, however, we'll go over the system's resolutions, ranges and limits.

Resolutions

For technical reasons sample ticks are used as a fundamental calculation element for all time and frequency values.

A **200 Delay DSP** module, for example, may have a maximum delay time of 200 sample ticks. There are exactly 48,000 sample ticks at a sample rate of 48 kHz, so the maximum delay possible is about t = 4.17 ms.

$$t=200/48000Hz=1/240s=4.166666ms$$

If now you adjusted the sample rate downwards, all delay times and frequencies would change. To avoid this, all values for time and frequency are converted by their pads to correspond to the current system sample rate.

The following conditions apply to circuit design calculations:

1. For any value, a sampling rate of 48 kHz is implied.

2. Any differences in in behaviour at different sampling rates only manifest themsleves at the upper and lower limits of the working range.

3. At a 96 kHz sample rate the maximum times of envelope segments or delay lines will be half of what they are at 48 kHz.

Examples:

1. As explained in the Basics section, only frequencies of up to 16 kHz can be produced or processed at a 32 kHz sampling rate.

2. Although oscillator frequencies of up to 48 kHz can theoretically be produced at a 96 kHz sampling rate, no values higher than 24 kHz are used internally.

Making Connections

Before diving into module interconnection techniques, a brief description of the operation of the Project Window, your workspace, is provided. The many tools available implies that there are many ways to create a circuit. In the following section we will not discuss all the ways these could be used in practice - we will focus here on functionality.

The **Wordclockpipe** module is always the first module in the module hierarchy when SCOPE starts. It notes the sampling frequency saved with the project. If you remove it from the module tree, and then save your project, the next time you load the project, its sampling rate is not remembered. This is diserable sometimes if you are working with several different digital devices, and you don't want the sample rate to change when the project loads (possibly creating havoc with attached devices).

There are three modes of operation in the Project Window - **Move** (**<Ctrl><M>**), **Use** (**<Ctrl><U>**) and **Edit** (**<Ctrl><E>**). Use **Move** mode when you want to arrange your modules and their connections in the window, **Use** mode when you want to operate a control (such as a fader), and **Edit** mode when you want to make changes to a module.

In Move mode, doubleclicking on a module takes you to the next lower level, and doublclicking on the Project window surface takes you up one level. You can drag modules with the mouse, or use the PC's arrow keys to make fine adjustments. The Toolbar's invisible, 'magnetic' grid feature is helpful to align modules in the Project window.

After you have dragged several modules from the File Browser into the Project Window, and connected them, you may want to 'clean up' the layout. In SCOPE you can 'fold' related modules together (e.g. a group of oscillators and their modulators) helping to keep the layout simple and informative.

To fold modules together first **select** them. Several modules can be selected by holding the **<Shift>** key while clicking on them with the left mouse button, or by drawing a wireframe around or through them (click and hold the left mouse button while dragging). When a group of modules is selected (highlighted) use the key combination **<Ctrl><F>** to fold them. A new, 'untitled' module will appear in the upper left hand corner of the Project Window.

In the new module, pads that were connected to other modules are automatically **exported** (see the *Sine Synth* tutorial). A pad can be further exported by selecting the option in its popup menu. If you delete an exported pad, the connection it provided to the original pad is lost. Exported pads are represented with an **arrow** are not shown in the Project Explorer's Routing drawer.

Since you must sometimes connect a pad to another pad at a different 'level', a **Store Pad** function is provided. This is found also in the pad's popup menu. Select **Store Pad** with the left mouse button. Now change levels and right click on the pad to which you want to connect the stored pad to call its popup menu. Choose the option **Connect to**.... At this moment, a red ball appears on each module to represent that the pads are connected at different levels. This is necessary because the connection will not be shown in the Project Window.

Controllers do not necessarily need visible pads, as **Store Val Pad** and **Connect to**... are available in the module's context-sensitive menu.

By putting modules in a **Module Group** you have the ability to copy modules at different levels at the same time, while maintaining their connections (see *Device Design*).

Grouping modules provides an important working feature, especially when working with display elements. You can use all the display elements of a module's complex surface as your surface without having to know anything more on how it is attached.

The use of grouping is best explained by example:

When you fold modules together into another (e.g. group them under the 'untitled' module) they are placed 'in' the module at the next lower level. Or, more technically, they become **child**ren of the module.

You can, at any time, move an object out of a **parent** object by selecting it and then using the keyboard combination **<Ctrl><Shift><I>** (for 'move Outside'). The real effect is that the module is moved up one level higher, and now resides at the same level as 'untitled' in the Project window. You can do this also with several selected child modules.

You can also move one or more modules into an existing group with **<Ctrl><I>** (for move **I**nside). When you strike this key combination the mouse cursor changes to the 'drop' cursor, and selected modules will be made children of the next module you click on.

This relationship is fundamental in SCOPE, and important to understand since some properties of a parent are transferred to child modules. More in depth information is found in the *Device Design* section.

The copy functions, **<Ctrl><X>**, **<Ctrl><D>**, **<Ctrl><C>** and **<Ctrl><V>** function with groups. Read the section on keyboard shortcuts for details.

Connections in the Project Explorer:

The Module Tree in the Project Explorer is, strictly speaking, just another representation of the Project Window offering only a few extra features. One thing you can only do in the Project Explorer is modify the name of a module. Click twice (slowly) on a module's name, or select it and press the **<F2>** key to enable editing. Type in the new name, and confirm with <Return>, or abort with **<Esc>**.

You can also move a module from the File Browser directly into the Module Tree at the desired level without having to go into the appropriate level in the Project window, and use the 'move In' function. Double click on a module name to open it up so you can see its children (just as it opens in the Project Window when you double click on it).

To drop a module into another all you have to do is drag it from the File Browser and drop it on the name of the module you want it to be a child of.

If, just before dropping the module, you press and hold the **<Shift>** key, the module will be dropped at an equal level as the selected module (as a *sibling*).

As sometimes it is desirable to hide a module's pads in the Project Window for clarity, the Project Explorer's Pad List is also useful. You can hide pads in the Pad List (press the **<H>** key while a pad is selected) and the pad disappears from the Project Window. It remains in the Pad List, however.

You can also modify a pad's short name and its values in the Pad List.

You open the **Routing** drawer by clicking on the icon in the upper left hand corner of the Pad List drawer. This is a list of a modules's pads, along with the pads they are connected to. You can delete connections from here. To delete, select the connection and press the **<Del>** key.

If you double click on a connection, the module and pad the connection is 'to' will be selected in the module tree and the routing list respectively.

If you hold the **<Ctrl>** key while double clicking on a routing, the module connected to is opened in the Project window. The view in the Project window will change to the appropriate level to display the module, if necessary.

To make effective use of the search functions, it is a good idea to name modules with some reference to their level - LFO 1, LFO 2, Filter LFO...

Exchanging Modules

Sometimes, while working on a design, you will want to exchange a module with another (perhaps a newer version) while retaining all the connections. If you delete a module, and then replace it with another, all connections are lost. Fortunately, SCOPE provides a **Module Exchange** feature.

First move to the appropriate level in the Project level so you can see the module you want to change. Then press the **<Alt>** key and drag the new module from the File Browser. When you drop the new module onto the old, the **File Exchanger** dialog will appear.

This dialog will let you reassign pads for both modules, but usually it is sufficient to check the connection assignments.

For more information on individual dialogs, see the *Dialogs* Chapter.

Device/Module Reference

Of the different types of module in SCOPE, one comes from the script class DEVICE (Empty Effect, Empty Synth). These modules are normally referred to as devices, and are well-known as SCOPE synthesizer, Big Mixer, etc.

Devices, by default, are loaded into a Project 'by reference'. This means that the Project data file contains only the name and location of the device - not the device itself. Any changes you make to a device will *not* be saved with the project. A referenced object, such as a device, is indicated with a 'diskette' symbol.

You can override the above default by pressing the **<Ctrl>** key while dropping the Device into the Project window. Now, when you save the project, the Device will be saved, with changes, as part of the Project data.

Modules, on the other hand, are not normally loaded by reference. To save a module by reference, select **Save as Reference** from its popup menu.

Saving modules by reference results in a smaller Project data file. Note that if you change a module, and then use **Save as Reference,** *any other projects that use the module will be affected.*

Envelopes

The **range** for envelope times (Attack, Decay, Release) in terms of sample ticks is Min = 12 and Max = 4194304. That means that for different sampling rates these limits will have different actual values:

Sample Rate	Minimum Value	Maximum Value
32.0kHz	3.75ms	131.0s
44.1kHz	2.70ms	95.1s
48.0kHz	2.50ms	87.4s
96.0kHz	1.25ms	43.7s

The real *practical* range is Min = 3.75ms and Max = 43.7s as these limits apply to all sampling frequencies.

In terms of sample ticks, then, this translates to:

Min = 180 and **Max** = 2097152

Calculating Envelope Times

Example: Attack = 20ms (desired value)

Attack = value/48kHz (value = required value)

value = 20ms * 48kHz = 20 * 48 = 960

For an attack time of 20ms, then, you would set the value to 960.

The following describes some of the other Envelope pads:

ESYN Pad: (voice handling)
(Esync signal)
For the correct internal processing of polyphony the ESYN (for *Envelope Sync*) pad of a MIDI Voice Control (MVC) must be connected to the pad of the same name of an envelope generator, even if you intend to operate monophonically. The ESync signal always carries the instantaneous value of the envelope.

As described in the 'Sine Synth Tutorial', the Esync and Gate signals are both necessary for the proper functioning of an MVC. Although a direct connection between the Gate and Esyn pads is possible on the MVC, voice handling will not be done as 'artistically' as it will if the MVC has the envelope level value available to it. Also, note that an envelope generator does not require an Esyn connection - it will operate with a gate signal only.

If you use several envelopes to control the output intensity of a voice, connect the Esync links of all envelopes with the inputs of an Esync Adder module, and then connect the output of this module with the Esyn pad of the MVC. The Esyn adder adds and normalizes the envelope information from the different envelopes to provide a single value for this voice.

Three pads are used for envelope time modulation:

TM pad (Time Modulation) TM1, TM2
(positive, asynchronous)
Time scaling (compression or expansion) of the envelope.

(logarithmic, 0.1 - 10)
(positive, asynchronous)

The two inputs are identical. The product TM1*TM2 controls the overall time compression or expansion.

0 corresponds to 0.1
+Max/2 corresponds to 1
Max corresponds to 10

LM pad (Level Modulation)
(positive, asynchronous)
The value at this pad controls the overall level of the envelope. This value could be set with a potentiometer, or by key velocity (Vel pad) for example.

Vol pad (Volume)
(positive, synchronous)
Usually connected to the pad with the same name on a 'Linear Volume Attenuator' module (without conversion). The FM oscillators also have a Vol pad.

When you want to use two envelope modules you must include the Esync Adder module. This adds the base values of two Esyn pads and passes the result on to the MIDI Voice Control module. Pitch envelopes do not require an Esyn check.

Connecting More than One Envelope Module

Envelope Modulation:

Here the overall envelope time and the sustain level are modulated. All modulation signals are asynchronous.

1. The note played modifies TM2. The Key Level Easy module extracts a weighting from the Note value.

2. The attack velocity determines the overall volume level. You could attach a potentiometer to the Amt pad of the Velocity Amt 1 module to modify the result.

3. The aftertouch (At) is also used to modulate the overall envelope time.

The illustration is only an example, although it produces a reasonable result. You might want to try some other arbitrary (or not) connections to see what happens.

Oscillators

The allowable range for some parameters, such as Fade-in Time, for example, is Min = 0 and Max = 20*48,000 sample ticks. This results in the following real limits at the different sampling frequencies:

Sampling Rate	Lowest Value	Greatest Value
32.0kHz	0.031ms	30.0s
44.1kHz	0.023ms	21.7s
48.0kHz	0.021ms	20.0s
96.0kHz	0.010ms	10.0s

The lowest value we can set is zero, so we will take this to be the lower limit. 10.0s is the highest limit that applies to all sampling frequencies, as shown above.

The useful working range then is Min = 0 and Max = 480,000. Only for the value '1' are there any differences at the various frequencies.

To Calculate a Fade or Delay Time

Example: Fade-in time = 2s (desired value)

Fade in = value/48kHz (value = required value)

value=2s*48KHz=2*48000=96000

The value to use to set a 2s Fade-in time is

value = 96,000

The following describes a few of an oscillator's specialized pads:

sp pad (start phase of an operator)
(bipolar, asynchronous)
The adjustable range is +/- 180°. More detailed information is found on the next page.

pw pad (pulse width of an oscillator)
(bipolar, synchronous)
A setting of 0 results in a symmetrical pulse. Positive values shorten the positive lobe, and negative values shorten the negative lobe.

FIn/**FOut** pads (fade in and fade out control for LFOs)
(positive, asynchronous)
Fade in or fade out durations of 0 to 10 seconds result from the working range of values, 0 to 480,000. Input values greater than 480000 are reset internally to 480000.

Positive lobe

Symmetrical pulse
pw = 0

Negative lobe

The following information is of particular importance when you are working with **LFOs** or **FM Operators**.

Oscillators can be either free running (free run) or started when a key is struck (Retrigger). Oscillators without an adjustable start phase (**sp** pad) always start at phase 0°. For the waveforms shown (rectangle, sine, and triangle) this results in an initial signal rise in the positive direction.

Some modulations, for example, volume modulation with an LFO, have a few implications. An LFO's output is a bipolar signal. However, the gain pads of an amplifier 'know' only positive values. The bipolar signal of an LFO must somehow be transformed into a positive signal. The lowest value (-max) becomes the minimum (0) while the highest value (+max) remains.

This implies that an **LFO** signal produces a gain of 50% of maximum volume at phase 0°.

Volume Modulation:

When using the ramp (sawtooth) waveforms in volume modulation undesireable audible effects can result from the characteristics of the wave shape.

Whereas with rectangle, sine and triangle waves the actual modulation cycle starts as expected at a phase of 0°, ramp signals start at the apparent 'center' of their period. For modulation to begin as expected you must start the wave with a phase offset of -180°.

Sawtooth down

Sawtooth up

The Random Number Generator outputs random values (+/-Max) at the adjusted frequency.

Random Number Generator

Modulators

Volume Modulation

Volume Modulator 1 has three inputs - two modulation inputs and an input for a constant offset.

The input signal is modulated by the addition of the offset to the product of the two modulation inputs. An offset may be required as this modulator disregards negative values (although it accepts them). Bipolar values from an LFO should be limited to half of +/- max and then given a positive offset (i.e. DC offset. See the following page).

m1, **m2** pads (Modulation inputs)
(bipolar, synchronous)
These would typically come from an LFO.

C1, C2 pads (Modulation inputs)
(positive, asynchronous)
Input from a control - a fader for example - to adjust the degree of modulation.

off pad (Offset)
(positive, asynchronous)
Used if a DC offset is required.

Signal Flow Diagram

The transfer function produces only positive values - negative values at the m1 or m2 inputs are disregarded. An offset is required to produce full modulation.

A full-amplitude, bipolar sine signal is received at an input, m1 for example (fig. 1).

Fig. 2 shows the resulting modulation signal (transfer function). Negative values of the input signal have been ignored.

Out = In * A (Transfer function 2)

A = ((m1*C1 + m2*C2) / Max + Offset) / Max

Fig. 3 shows a full amplitude modulation signal using an offset of +Max/2 with C1 = +Max.

Fig. 1:

+Max

-Max

Modulation input

Transfer function 1

Fig. 2:

+1

0

Modulation with
Offset=0
C=+Max

Transfer function 2

Fig. 3:

+1

Modulation with
Offset=+Max/2
C=+Max

0

Volume Modulator 2 behaves like Volume Modulator 1 with a constant offset of +Max/2.

Out = In * (M1*C1/(2Max^2) + 0.5)

Transfer function

1

0

Modulation with
C=+Max

Modulation by key position:

The **Key-Level Easy** module.

This module allows a MIDI keyboard to provide a modulation signal.

Modulation slope

Note Number

Val pad (controller input)
(positive, asynchronous)
You would normally connect this to the Note pad on the MVC. The MIDI keyboard serves then as a controller.

P pad Modulation in percent
(bipolar, asynchronous)
Range: -100% to +100%
This governs the direction and depth of the modulation.

Key pad (MIDI Note number)
Range: 0 bis127
(positive, asynchronous)
The MIDI Note number at which the modulation is 0 (crossing point).

Modulation at Note Number = 64

The **Velocity Sens 1** (Velocity Sensitivity) module controls the degree of volume modulation produced by MIDI key velocity information. The higher the value at the **Sens** pad, the greater the influence of the key velocity.

When Sens = 0, MIDI key velocity will have no influence and the notes will be played at maximum volume.

Maximum modulation is at Sens = +Max.

The **Velocity Sens 2** (Bipolar Velocity Sensitivity) module behaves like the **Velocity Sens 1** module, but permits inverse modulation - meaning high key velocities will produce weaker tones.

Maximum inverse modulation is at sens = -Max.

The **Aftertouch Amt** module has a maximum range of 127 semitones.

The **Aftertouch Amt Bipolar** module can be adjusted to raise or lower the pitch by up to +/-64 semitones.

The **Cutoff Key Follow** module responds to controller data. Normally the input pad is connected to the Note pad of a MIDI voice controller so that cutoff frequencies become a function of the MIDI note number. However, any positive, asynchronous pad can be connected to the input. The output should be connected to the asynchronous pad (C or CF) of a pitch modulator which is then connected to a filter's cutoff input.

The **Cutoff Key Follow 1** module produces an ascending values used to increase the cutoff frequency for a filter as the notes increase in frequency.

Val pad (for connecting to the controller)
(positive, asynchronous)

P pad
Range: -100% to +100%
(-100 to +100, asynchronous)

Key pad (Note number for where modulation = 0)
Range: 0 to 127
(positive, asynchronous)

The **Cutoff Key Follow 2** module behaves like the Cutoff Key Follow 1 module, but with in increased range at the P pad of -200 to +200.

Cutoff Key Follow 3 behaves like the Cutoff Key Follow 2 module, but produces no change after 100%.

Dialogs

The following paragraphs describe the **Project Explorer** dialog and the various context-sensitive menus available.

The Project Explorer Dialog

As mentioned earlier, the view of the module tree in the **Modules** drawer resembles Windows Explorer. The current Project includes all modules and their connections. In this view, however, no connections are displayed - only the modules themselves.

If a module contains other modules, it is marked with a '+' symbol: +

By clicking on the '+' symbol, or double clicking on a module's name, the child modules are made visible below it in a new branch. Within a branch you may be able to open additional branches - as many as required. If a module is protected ( ) you cannot open it up to view the children.

You can call up a context-sensitive popup menu for a module by right clicking on the module's name. Select the desired action (Cut, Copy, Load, etc.) with the left mouse button. Keyboard shortcuts are available and listed later in this section.

The **Properties** entry in the menu opens a window where you can view and edit a module's various parameters.

Save as reference ... stores a reference to a module. Now, when the project is saved, the module itself is not - only it's name and where it was located is saved with the project. Reference modules are indicated with a diskette symbol.

The **Optimizations** menu entry more properly belongs to the 'Device Design' section, and is described there.

Just as you can drag modules into the Project Window from the File Browser, so too can you drop them into the module tree at the desired level.

You can also drag modules already in the module tree to a new position or level. Select the module you want to move and drag it over another module. When you drop it, it will become a child to that module.

If you want to move a module to an equal level as another module, hold the **<Shift>** key while dropping it. Now when you drop it over another module it becomes a sibling to that module.
Once a module is selected and highlighted in the module tree you can use the arrow keys to move up and down through the tree. (This works in other trees and lists as well, such as the 'Pad list').

For details about the "**Protect Module**" option, refer to chapter "Device Design".

In the **Pads Drawer** you will find an overview of the pads belonging to the currently selected module. From here you can use the context-sensitive menu and keyboard keys to modify the values and appearance of the individual pads.

By clicking on the **Open Routings** icon you can open the **Routings** drawer where all the connections are visible. Clicking on the **Open Properties** icon opens the **Properties** drawer where you can view and modify additional properties.

When you select a new module, the first pad in the Pad list is also selected.

In complex circuits you will often want to organize the modules by 'folding' them. If you then want to connect a pad to a module at a different level you must first 'store' it and then connect it to the other pad. Use the **Store pad** icon (black plug) to store a pad, and the **Connect to...** icon (red plug) to connect it. The **Connect To...** icon will be enabled only when a pad has been stored. Each pad can thus be connected to another pad anywhere within a project (assuming they are compatible).

You can further organize your project for clarity by making some pads invisible, and arranging others so that connections are kept relatively short.

A pad's location on a module surface can be changed by clicking on its **Pad Orientation** icon. Repeated mouse clicks will cycle the possible pad locations - up, right, down, left.

To use a pad at the next higher level in the tree, select it and click on the **Export Pad** icon. A gray arrow appears, and a copy of the pad is created in the module's parent module. The exported pad is directly connected to the original.

By clicking again on the **Export Pad** icon, the copy is removed from the parent module.

Often it makes sense to change the graphical representation of the pad's type - In, Out or I/O. This would be the case, for example, if you export an In pad which is connected to an Out pad. You can change the appearance of a pad using the keyboard keys **<I>**, **<O>** and **<P>**, or by selecting **Type** from the popup menu.

The Pad List's Context-Sensitive Menu

Each pad has a long name and a short name. The short name is editable, and the first four letters of a short name appear on the module surface to identify it. To edit a short name, click and hold the left mouse button for a second while the mouse cursor is over the name, or select the pad and press the <F2> key. Long names are not editable and appear in the Tool Tips display.

Select **Default Layout** to restore the module's original representation. Modifications made to pad types will not be restored, however.

The **Restore Level** option selects whether the pad should be initialized with a value from a module's preset, or from the project's last 'save' state (its value at the time the project was last saved). A MIDI channel number should be restored from the project. The value of a level meter should not be restored at all (Don't restore).

To edit a value or change a name in a list click twice slowly with the left mouse button while the mouse cursor is over the name or value.

The interconnections of the modules are shown in the **Routings** drawer. In this example Routing List, the pad in1of the current module is connected to Out1 of the 'Inferno' synthesizer. If you double click on this entry, the connected module will be enabled in the module tree. If you hold the <Ctrl> key while double clicking you are brought to the appropriate level of the module as well. Although the connected module is now selected in the module tree, it does not appear selected in the project.

To remove a routing, select it and press the <Del> key.

A few remaining pad properties are found in the **Properties** drawer. The pad's range (e.g. bipolar), whether it is synchronous or not, and its Restore Level are displayed here.

Restore Level :

0: Pad value is restored from preset.
1: Pad value is restored from project's 'save state'.
2: Pad value is not restored

This value is not editable.

The **GO Tree** displays the graphical objects (GOs) for the selected module. Please see also the *Surface Design* section.

The illustration on the right shows the GO tree for the graphical elements of a simple on-off button. This is represented in a hierarchy as in the Module Tree. The entries are editable - delete using the **<Del>** key, or edit the names by clicking **<F2>** or by clicking twice slowly on an entry.

The on-off button is a script module. The response functionality is in the script and the graphical elements are GOs.

The designated names of the GOs (BitmapOn, BitMapOff...) are protected and cannot be changed, as they are used in the Script. They can, however, be replaced with other GOs. See the section on *Surface Design.*

To the left of the GO tree are the individual bitmaps for the button. The background is the 'Parent' GO and the others are 'children'. The arrangement in the tree determines the way the bitmaps are drawn - which is drawn over which.

The sequence of GOs in the tree, and whether they are parents or children, determine the way in which they are drawn.

1. Children are drawn over their parents.

2. GOs at the same level are drawn in the order in which they appear in the list.

In the **GO Properties** drawer you can adjust properties to further control the way in which objects are drawn.

Selectable, if enabled, means that the object can be selected by the user with the mouse. In Edit mode it will be selectable regardless of this setting.

Visible - the GO will only be displayed if this is enabled. In that case, it can be selected, moved etc.

Scrolling, if disabled, prevents the object from moving in response to scrollbar activity.

GO-Properties

Art des GOs

Selecting the **On Top** option means the GO will always be drawn on top of other objects. In the example, the text 'on/off' has this property set.

Likewise, the **On Bottom** property causes a GO do be drawn always underneath other objects.

Fixed Z-Order means it will alwys be drawn according to the rules.

Messaging must be set in order for the object to respond to user input (mouse clicks on the object, etc).

Clip Children increases drawing speed when set. It prevents the drawing of object areas that lie outside the dimensions of the parent object.

Custom Size applies only to the **Group** GO. If set, you can set a clipping rectangle as above, but the rectangle itself is invisible.

You can drag and drop GOs from the **Add Object** drawer into the GO tree or into other objects.

When you drag and drop a GO to an object in the GO tree it normally becomes a child to that object. If you press the shift key just before dropping, however, it will become a sibling ('brother' or 'sister') to the object. You can, for example, at additional text (standard or graphical text) or another bitmap to a graphicl object.

More information on GO types is found in the *Surface Design* section.

Context Sensitive Popup Menus in the Project Window

Some options not available in the Module Tree are found in the module popup menus.

Surfaces brings up a list of all surfaces a module uses. A single left mouse click opens the surfaces.

Add to Group includes the module in the list of modules in the 'Group Window'. This is a utility area in which to keep a collection of modules you want to duplicate frequently. Don't confuse this with the 'Group' module.

Properties opens a dialog in which you can make other adjustments (see *Device Design*).

A module's popup menu

Module Properties Dialog

A Pad's context sensitive popup menu contains the options for exporting, storing (to memory), connecting and deleting the pad.

Menus for Controllers are are basically identical to those for modules, except they have two extra options above the grey separator line - **Store Value Pad** and **Open Control Ranger**.

Store Value Pad is very useful alternative to using dialogs when making connections in a layout. Value Pads are often not visible, and without this option you would have to open the Project Explorer to make the connections.

The **Open Control Ranger** does as you would expect - it opens the Control Ranger.

Additional Controller Options

Pad Menu

Module Exchanger

Sometimes when working on a design you will decide that another module, a different filter module for example, would better serve a purpose. With the Module Exchanger you can replace the module with another one while maintaining all the connections.

The dialog opens automatically when wou drag and drop one module onto another in the Project window while pressing the <Alt> key.

In the example to the right a fader is potentiometer is being exchanged in favor of a fader. All the connections that could be assigned are indicated with a green check mark. Connections that could not be assigned are indicated in red.

It may be that you don't want to assign all the connections. In that case, double click on a pad name and it will disconnect and appear in red lettering on both sides of the list.

To re-connect, doubleclick on an entry again.

When you are satisfied, click on the **Exchange** icon.

Control Ranger

The Control Ranger is used to adjust the characteristics that determine the behavior of a control.

Often it is required to limit the range of a potentiometer or other automatic controller. You can do this in the Pad List, but you get a better overview in the Control Ranger. You can input the values directly, or use the sliders to adjust them.

Although a potentiometer's pad is identified as 'positive', these controllers *will* output negative values. All you have to do is set a negative **Min** value (as above). The **Max** value is the largest the controller will output.

Each controller has a 'taper' curve - linear, logarithmic, or exponential. Select the curve with the **Form** slider, and adjust the depth of the curve with the **Intensity** control. Any curve can be inverted by enabling the **Invert** option.

Settings for MIDI controllers are found in the **MIDI Controller Settings** drawer.

Miscellaneous

Error Messages

This section covers error messages you may encounter from time to time.

Cannot connect Converter pads can appear if you attemtpt to connect a pad to a pad that is already using a converter.

This error appears when a connection is not possible.

Keyboard Commands

Keyboard usage - General:

Press and hold the **<Ctrl>** or **<Alt>** key and then the required key in the list of key combinations below.

<Ctrl><O> Open Project

<Ctrl><S> Save Project

<Ctrl><Shift><S> Save Project As...

<Ctrl><N> New Project

<Alt><F4> Close application

<Ctrl><U> Change to 'Use' mode

<Ctrl><M> Change to 'Move' mode

<Ctrl><E> Change to 'Edit' mode

Use **<Esc>** to switch back and forth between 'Use' mode and 'Edit' mode.

<Del> Selected modules or routings will be removed from the Project Window.

SCOPE Window Keyboard Commands:

The following refer to your keyboard's function (**<F*x*>**) keys.

F9	Project Window on/off
F10	File Browser on/off
F11	Project Explorer on/off
F12	Toolbar on/off

Keyboard usage in the **Project Window**:

NOTE: The following copy functions do not work in Edit Mode. It's best to change to 'Move' mode. Connections to other modules are not taken into account when copying.

<Ctrl><D> Duplicate
Duplicates one or more selected modules. The mouse cursor changes to the 'Drop' form to place the module(s) anywhere you wish in the Project Window.

<Ctrl><C> Copy
Make a copy of the selected module in a copy buffer (clipboard).

<Ctrl><V> Paste
If you have copied a module as above, this will paste it into the project at the current level and at the current mouse position. Make sure you are at the desired level and position before using the paste function.

<Ctrl><F> Fold selected Modules
When a group of modules is selected use this key combination to fold them. A new, 'untitled' module will appear in the upper left hand corner of the Project Window. The new module contains all the selected modules, pads that were connected to other modules are automatically **exported** (see the *Sine Synth* tutorial).

<Ctrl><Shift><F> Unfold selected Module
The module vanishes and the circuit which was inside "pops out".

<Ctrl><I> Move Module Inside
Moves a module into another module as a child. The mouse cursor will change to a 'drop' form. When you now click on another module, the copied module will be removed from the current level and made a child of the new module.

<Ctrl><Shift><I> Move Module Outside
This moves the selected module to the next higher level in the module tree.

Arrow keys Move modules
Selected modules are moved left, right, up or down.

<Ctrl> Load as reference
If you press the control key while loading a module (i.e. dragging it into the project from the File Browser) it will be loaded as a 'Reference'. Devices are loaded as references by default - modules are not.

<Ctrl><G> Group
Selected modules are grouped into a single, fixed graphical unit. When you move one, the others move as well.

<Ctrl><Shift><G> Ungroup
Undoes the ‘Group’ function freeing the modules and removing the invisible grouping module from the module tree.

<Ctrl><R> Reroute
Recalculates the connections.

Keyboard usage in the **Module and GO Trees**:

Note: Connections to other modules are not taken into account when copying.

<Ctrl><C> Copy

Makes a copy of the selected module in a copy buffer (clipboard).

<Ctrl><V> Paste

Select the module into which you want to drop the copy (as a child) and press <Ctrl><V>.

<Ctrl><X> Copy a module *with connections*

The selected module is copied to a buffer, with its connection information. When you now select another module and press <Ctrl><V> the copy is moved to the new module with its connections preserved.

Arrow Keys

A selected module is moved up and down in the module tree with the up and down arrow keys.

<F5> Update List

Updates the module list to reflect values that might have changed.

Keyboard Usage in the **Pad List**:

First select the pad, and then strike a key to perform the following actions:

Arrow keys Move through the tree.

Use the up, down arrow keys to select pads, and the right, left arrows to select their parameters.

Pad locations on a module's default surface:

T Top

B Bottom

L Left

R Right

H Hide

Pad Type:

I In Pad

O Out Pad

P I/O Pad

A Reset layout to all default colors, sizes etc. Only the representation of the pads is preserved. Use with caution.

Pad Connections:

S Store Pad (save reference in memory)

C Connect to stored Pad (connect to saved pad)

E Export pad to next higher level