The SCOPE Manual

Preface

Return To Main Table Of Contents

Foreword

Welcome to the SCOPE. This is the beginning of your journey into the beautiful future of audio. You have surely already recognized that the SCOPE is not a conventional product designed for one specific task. It is a new technology platform for audio that is potentially suited to any audio task imaginable, going much further than any previous audio product. The potential extends far beyond the applications (devices) included in this first release of the platform – the supplied devices (like synthesizers, effects etc.) are just the beginning of a new universe of applications and possibilities in audio. But even the initial functionality is much more powerful than any product before now, and the longer you work with the system, the more possibilities you will discover. Never before has so much functionality, versatility and pontential been combined in one system. There are no limits on how to use the many items in the library – the SCOPE is a unique, powerful tool that lets you get creative again!

It's impossible to compare SCOPE with any other audio system because the nature of the SCOPE is so different from any convential approach. You cannot get an accurate overview of the full scope of the SCOPE with just a brief look at it. Maybe this sounds as if SCOPE is highly complicated and difficult to handle. The good news - it's not. In the audio universe of the SCOPE, there are just a few generic rules to follow – all the rest is pure freedom.

Once you grasp the intuitive concept behind the SCOPE, you can start to travel through a new universe of audio that has no known limits. We, the developers, have been SCOPE pilots for a while now, and it is still exciting to make new discoveries – and it's getting more exciting every day, especially now, as the development environment becomes available to the public. From now on we won't be travelling alone. It is great to welcome YOU onboard along with the many other innovative minds – we are the pioneers in a new audio world and it is up to us, the SCOPE community, to innovate and shape the future of audio engineering.

The first release of SCOPE emphasizes the development platform functionality as well as analog and FM synthesis, effects, and mixing. Again, this is just the starting point. Unlike other products, SCOPE is suited as a tool for many areas in the audio world. The pro audio recording and post production studio will find the SCOPE useful as a stong synthesis engine and for its versatile mixing and routing functionality. The unmatched flexibility makes the system a valuable tool for those involved in sound installations. SCOPE is equally well suited for many applications in the test and measurement field. Audio developers will find the SCOPE the ideal development environment, for rapid prototyping of their own algorithms or as a universal test environment for any audio development.

Foreword

As the functionality of the platform and environment grows, the SCOPE will embrace more and more applications. If you think you are missing a certain functionality, it may well be that, due to the system' flexibility, you will find another way to reach your goal. If you still find a limitation, all you need is a little patience. The more people that become involved with the platform, the faster the system will grow. Never before has a system provided such creative power to the user. The extensive development environment of the SCOPE enables audio developers to create new applications easily, and this, in turn, will expand the reach of the platform significantly. In fact, you should expect the system to grow exponentially. It is our goal to make the SCOPE the most broadly supported and most powerful audio DSP platform on the planet.

For now we hope (and expect) your work with SCOPE will be both fun and rewarding. We're proud to have you on board – have a pleasant flight.

Sincerely yours,

Frank Hund

President CreamWare GmbH

About The Manual

SCOPE wasn't built in a day. And no one will master it in a day – there's simply too much to know. However, you can and *should* expect to do interesting things with SCOPE during your first session as a SCOPE user. The SCOPE user documentation has been designed to promote this goal.

The main components of the documentation – in the manuals and in SCOPE itself – are described in overview below.

The Quick Tour For Designers

The best place for all users to begin, but especially budding designers. This chapter provides basic orientation regarding the SCOPE workspace – what the major features of the workspace are, what they're called, how to find them, and how to get around in SCOPE in general.

Using SCOPE In The Studio

SCOPE offers practically limitless design possibilities. However, it's also an outstanding platform for straight-ahead studio work, and includes a plethora of fully-developed, finished devices. For those who wish to simply "use" SCOPE, rather than jumping immediately into design, this chapter provides the information needed to get up and running (you can even skip the Quick Tour). This chapter also provides full information regarding general configuration of the SCOPE hardware and software. Additional chapters describe the individual synthesizer, mixer and effects devices in full detail.

If you've already used Pulsar, you'll be able to pick up and go with SCOPE right away. Using SCOPE as a studio tool is almost exactly the same as using Pulsar – but SCOPE puts significantly more power into your hands.

The tutorials

Here's where to begin learning the SCOPE development environment, after you've read the *Quick Tour* chapter. The tutorials walk you step-by-step through typical SCOPE circuit construction procedures based on simple circuit and device examples. They don't explain everything there is to know about the SCOPE features they demonstrate. The emphasis is on learning by doing. This puts you on the fastest track to becoming familiar and competent with SCOPE as a development tool.

Tutorial 1: Basic Techniques guides you along a direct path toward the creation of a simple SCOPE device, using the simplest and most fundamental methods.

Tutorial 2: Further Techniques extends the knowledge you gain from *Tutorial 1* with the introduction of more advanced circuit manipulation techniques, including an in-depth tour of the Module Properties tool, and the basics of device surface construction and modification.

The Circuit Design and Device Design chapters

These sections go into depth on a wide range of topics of interest to advancing SCOPE developers, including application techniques for specific modules, device design conventions and considerations, SCOPE signal characteristics, and advanced tool functions – with special emphasis on the "left side" of the Project Explorer. Numerous examples are included to illustrate the points explained. These chapters are the place to go when you find yourself needing to understand more of what's going on "behind the screens" than is explained in the tutorials. The tutorials prepare you to be able to make use of this material.

The Graphical Design chapter

This chapter provides thorough background information regarding the advanced graphics environment which is a central component of SCOPE, along with a detailed explanation of the individual graphical design tools, including the "right side" of the Project Explorer. The theory and practical techniques which are presented here will enable you to make full use of SCOPE's graphical design capabilities, not only for custom device surface design, but also for creation of custom graphical control modules. Examples are provided to make the application of this information clear.

SCOPE online help

The Help Window is a SCOPE tool which makes SCOPE a friendlier place to work. It provides instant plain-text descriptive information for individual SCOPE circuit modules or their individual connection points, simply by positioning the mouse cursor over them. These descriptions explain the function of a module or connection and provide helpful tips regarding applicatons and usage.

Tool Tips, the smaller cousin to the Help Window, provides instant pop-up information directly next to the cursor, including full-length names for module connection pads and, in some cases, their signal units and instantaneous values.

A few general suggestions

1. Expand the Acrobat Reader window to full size for most comfortable viewing of the manual.

2. As much as possible, run SCOPE *while* reading the manual. Get hands-on with SCOPE, right from the start.

3. Have fun!

About SCOPE

Congratulations! Welcome to SCOPE. Still wondering just a little bit what exactly it is that you've bought? The next few pages offer an overview of the product and the possbilities it provides.

SCOPE combines the SCOPE board, featuring state-of-the-art digital signal processing (DSP) hardware and extensive input/output (I/O) facilities, with cutting-edge software to produce an environment in which virtually anything having to do with digital audio is not only possible, but also remarkably easy. Furthermore, it is an open and open-ended environment which permits the user to "dig in" as deep as desired. Barriers to creative use of the system have been not merely avoided, but aggressively eliminated. Our goal is to encourage serious involvement on the part of the user, as well as to promote active third-party development.

SCOPE hardware

The SCOPE board provides a *very* generous amount of general-purpose DSP power and an expandable hardware platform which is independent of the host computer into which it is installed, along with the ability to connect directly to almost any type of external sound device via MIDI, analog audio, S/P-DIF and ADAT optical digital audio interfaces.

General-purpose DSP power means: full flexibility. The functions which the DSP section performs are defined entirely by the software which is loaded into it. There are no desgined-in functional constraints, no pre-defined fixed commitment of computing resources, and no up-front compromises in audio quality.

Host independence means that the audio performance of the system is essentially the same regardless of the specifications of the host computer (CPU speed, amount of installed memory, etc). It also means that the SCOPE board is inherently capable of operating in various types of host computer. Windows PCs are only the beginning.

Expandable hardware means that the SCOPE concept encompasses systems with multiple SCOPE boards operating in tandem, to multiply available DSP power and I/O capacity. Host independence is not compromised here. Communication among multiple SCOPE boards takes place via a proprietary hardware bus which connects the boards directly to one another and is fully independent of the host system bus.

SCOPE software

The SCOPE software comprises advanced DSP software and an immensely flexible and powerful graphics-based system running on the host, which makes the power of the DSP system readily accessible to the user and provides a previously unattained degree of power, ease of use and creative potential (not to mention outstanding aesthetics!).

The DSP software

The DSP operating system provides efficient, transparent and above all *flexible and dynamic* management – loading, interconnection and removal – of DSP software modules. These modules – SCOPE is delivered with a library of well over a hundred of them – perform building-block functions ranging from extremely simple (e.g., combining two audio signals) to quite complex (such as voice management for a polyphonic MIDI synthesizer). The library of DSP modules is continually being expanded by both Creamware and third-party developers.

The DSP operating system has been designed from the ground up to permit minor or drastic changes to the complement of loaded modules, or to the connections between them, *while the system is in use*, without excessive delays or unpleasant side effects, and to efficiently facilitate the distribution of these modules among the multiple DSP chips on a SCOPE board – or on multiple boards – without any need for the user to be concerned with or even aware of the details.

The host software

The host system software coordinates the activities of the DSP operating system and presents the user with a highly-intuitive graphical representation of the loaded DSP modules and the connections between them. Manipulation of DSP modules via this interface is slightly less complicated than connecting the components of a stereo system together.

The host software provides an additional library of building-block components – the graphical elements which implement onscreen display functions and mouse-operated controls. Via the same graphical interface used to manipulate DSP modules, these elements can be connected directly to DSP modules, thus providing the basis of the user interface to the DSP system and making it fully interactive.

The graphical object library further includes surface elements upon which the display and control elements can be arranged to form complete custom control panels for devices consisting of a handful or hundreds of DSP modules. The host software tool set includes a full suite of graphics tools for altering every conceivable aspect of the appearance of these surfaces and the other onscreen elements, permitting highly stylized "front panel" design.

But surface appearances are only part of the story. Sophisticated tools and methods are also provided for dealing with the structure of a design. As just one example: groups of library modules, along with the user-created connections between them, can be combined into custom modules featuring user-designated inputs and outputs. These can be saved as new components and used just like library modules, extending the library. Complex user-defined module circuits can thus be kept ready at hand, neatly encapsulated, with no need to recreate them from scratch for each new use or to confront their internal details. Naturally, user-defined modules can likewise be incorporated into further user-defined modules. This hierarchy can be extended to any number of levels, facilitating the construction of highly complex devices while retaining a clearly organized, manageable structure. Of course, the available design tools and methods go far beyond this one example.

SCOPE applications

The circuit/graphics design process described above culminates – *if* desired – with the incorporation of interconnected module groups and associated custom control surfaces into integrated devices – full-function software equipment, actually, such as the mixers, synthesizers and effects devices included with Pulsar. It should come as no surprise to hear that SCOPE was the development system used to create these devices. Properly packaged, the devices you create with SCOPE can likewise be used in Pulsar. This is, in fact, one of the specific intended applications for SCOPE.

However, creation of finished devices is by no means the only use for SCOPE. You can also use it to create custom setups of all kinds – to suit your signal processing or routing needs of the moment, for expermentation of all kinds, or to augment existing devices with just a little added functionality. There's no requirement that you make your SCOPE setups look nice or package them for mass consumption. "Quick and dirty" is also okay – and with SCOPE, *very* quick. The graphics tools and custom control surface design capabilities are available for use, but are not mandatory – likewise for the hierarchical circuit design options. If you want, you can drop knobs and displays, raw DSP modules and complete synthesizers onscreen randomly and hook them up – this also works.

On the other hand, there's no reason why the facilities of SCOPE can not also be applied to the development of devices which are ultimately to be realized as stand-alone hardware units. SCOPE lends itself quite well to rapid prototyping of both internal functional aspects and front panel layout – potentially even for equipment which has nothing to do with synthesizers, effects or even audio.

Speaking of synthesizers – the full set of finished synthesizers provided to Pulsar users is of course also provided with SCOPE. But the SCOPE user has the advantage of the SCOPE board, with almost four times as much DSP power as the Pulsar card. So, it's also entirely possible to use SCOPE simply as a super-Pulsar. The manual chapter *Using SCOPE In The Studio*, in conjunction with the chapters describing the individual synths and other devices, give you all the information you need for this. We highly recommend it – it doesn't exclude any of the other possibilities, and it's a helluva lot of fun!

Installation

Return To Main Table Of Contents

Quick Start Guide v1.2

DSP Powered Music Production Environment

Installation Guide

About this Guide

In this guide you will find important information with regard to the installation of your SCOPE hardware and software, and its integration into the PC environment.

Please read this guide carefully!

Along with the installation instructions you will also find some tips that should help get you startedwith some of the more popular Windows applications. These tips are intended to get you started only and do not replace the detailed information you will find in the next sections of this manual.

Welcome

Thank you for choosing the SCOPE system. With SCOPE, you've acquired not only a very high-quality, state-of-the-art I/O card, but also the most advanced DSP environment currently available. New SCOPE applications (effects, synths, etc.) are continually being created by hundreds of developers around the world ensuring the unfolding of a new audio world based on your new system. Our goal, with Pulsar and SCOPE, is to begin a new chapter in the development of audio technology - and we're quite pleased to have you already with us today!

Important Advice
Transients may occur on the SCOPE card outputs during startup of the computer or the SCOPE software. Equipment connected to the SCOPE outputs should be switched on only after starting the SCOPE program, or its volume controls should be turned down until the software has been started in order to avoid damage to this equipment. Likewise, the volume should be turned down, or the equipment switched off, before shutting down the computer.

The Scope card is hereby certified to conform to the requirements set forth in the guidelines for electromagnetic acceptability (89/336/EWG).

CreamWare Datentechnik GmbH, May 1999
Dr. Hans-Ulrich Hund

Before We Begin ...

System Requirements

Absolute Minimum System:

Processor:	Pentium 233MHz with MMX
RAM:	128MB main memory
Graphics card:	PCI or AGP, 4MB Resolution: 1024 x 768, 16-bit
Operating System:	Windows95 / Windows 98
Free Disk Space:	min. 100MB

Recommended Minimum System:

Processor:	Pentium II 300MHz or higher
RAM:	more than 128MB (256MB)
Graphics card:	AGP, 8MB or more, opt. 2nd graphics card Resolution of at least 1024 x 768, 16-bit
Operating System:	Windows95 / Windows 98
Free Disk Space:	min. 100MB

Important Notes

1. Some graphics display adapters can interfere with the data flow over your PC's PCI bus.

If you have a Matrox card and experience a problem with the PCI bus you can most likely resolve it by including the following entry in your **system.ini** file:

```
[mga.drv)
PCIChipset=1
```

2. If you are overclocking your PC note that the PCI bus speed will increase beyond the 33MHz PCI specification. Although SCOPE may work at bus speeds up to 37.5 MHz, this is in no way guaranteed.

Further information can be found in the **Support** PDF document on the CD-ROM.

The Hardware

SCOPE Board:

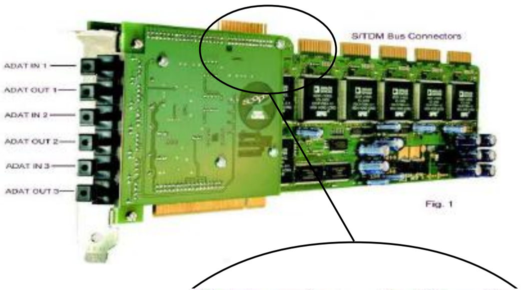


Fig. 1

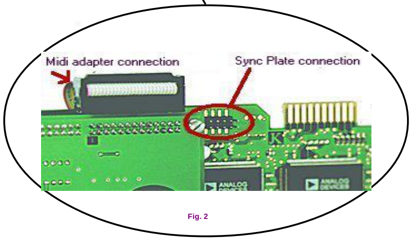


Fig. 2

The SCOPE board contains 5 S/TDM bus connections for cascading SCOPE, Pulsar or Pulsar SRB boards. You will also find a header to connect the Sync Plate just in front of the first S/TDM connection point. Please note the location of pin #1 (lower left) on this header. *This pin must be connected to the red colored line* of the Sync Plate ribbon cable (see below).

24 I/O Daughterboard:

The 24 I/O daughterboard is already affixed to the SCOPE board and provides ADAT ODI and MIDI I/O. The orientation of the ADAT inputs and outputs is shown in figure #1. The MIDI I/O cable whip attaches to a connector on a slot cover plate which is in turn connected to the 24 I/O board via a ribbon cable (figure #2). Note that the connector on the 24 I/O board is notched for the correct orientation of the ribbon cable.

Sync Plate:

The Sync Plate provides word clock snchronization through two BNC connectors and ADAT sync through two ADAT 9-pin connectors. Near the back of the card at the top is a header for connecting the sync ribbon cable. *When connecting this cable, ensure the red-colored line connects to pin #1 as marked on the printed circuit board.*

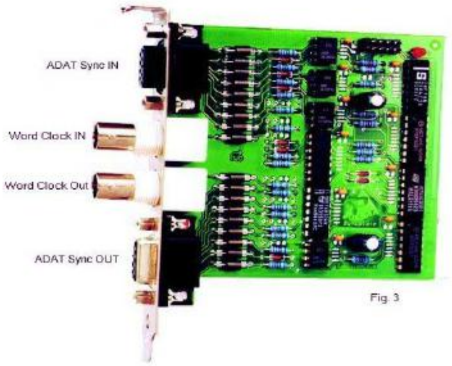


Fig. 3

With the Sync Plate it is easy to integrate SCOPE into an existing studio using either word clock or ADAT sync. As can be seen in figure #3, the Sync Plate provides both inputs and outputs for sync signals allowing SCOPE to switch between master and slave modes at any time.

With regard to sync we have assumed you are familiar with fundamental digital studio technology concepts. Please understand that, for the purposes of this installation guide, we cannot discuss the implementation of sync in depth. See the documentation for the devices you will be connecting to SCOPE, as well as the SCOPE manual on the CD-ROM for more information.

Installing Multiple S/TDM-Compatible Cards

Creamware developed the S/TDM bus so that additional compatible cards (for example, another SCOPE, Pulsar and/or the SRB) can be cascaded, thereby greatly extending the processing power of a Pulsar/SCOPE system. All that's requiredis the correct cable (supplied with the cards) to connect to the bus connection points provided on each CreamWare card of the Pulsar/SCOPE generation.

Fig. 4

2 x SCOPE, 1 x Pulsar

CreamWare has also designed the system such that the boards can share interrupts (IRQs) - thus greatly simplifying expansion. Only the appropriate Windows driver is required to enable a card once it is physically installed in the system. No further configuration is necessary (see page 14 for more information on driver installation).

Please, refer to the notes on "cascading".

Hardware Installation

1. Disconnect the power from your computer and remove the chassis cover or access panel.

2. Locate a free PCI card slot (usually lighter colored than the older ISA card slots, and smaller). It is generally a good idea to avoid using slot #1 if you can. See 'Notes' below.

3. Remove the rear panel coverplate for the selected card slot.

4. Discharge any buildup of static electricity from yourself by briefly touching a grounded metal object (such as a radiator or water pipe, etc.). **NOTE:** the ground contact of an electrical cord or electrical outlet should **NOT** be used for this purpose, as this introduces a shock hazard.

5. Remove the SCOPE card carefully from its packaging. **Do NOT,**
under ANY circumstances, touch the gold contacts (PCI/S/TDM bus) on the card!

6. Position the SCOPE card **carefully** in the selected PCI slot and then press straight downward on it from above until it is seated securely in the slot.

7. Use the retaining screw you removed in step 3 to secure the SCOPE card's metal mounting bracket to the computer chassis.

8. Locate two available spaces in the rear panlel of the chassis for the sync plate and the MIDI cablewhip connector. These spaces need not be associated with PCI or ISA slots. Use retaining screws to secure these metal brackets into the spaces, and make the connections to the SCOPE board as described earlier.

9. Replace the computer's housing or access panel and re-connect the power cable.

10. Connect the optical cables to the appropriate inputs and outputs on SCOPE and your other devices, connect the MIDI cablewhip to the MIDI jack, and connect your MIDI devices to the respective cable whip MIDI I/Os.

For information on making connections to the Sync Plate, see figures #2 and #3 on the previous pages, and consult the "Using SCOPE in the Studio" documentation for the devices you are connecting.

Software

Installing the Driver (Windows 98/95*)

1. Power up your computer and launch Windows 98.

2. Windows will report that it has found new hardware. This will be identified as a 'PCI Multimedia Device'.

3. Start the Windows **Add New Hardware Wizard** by clicking on **Next>**.

4. Now, when Winows asks, 'What do you want Windows to do?' select the option, 'Display a list of all the drivers in a specific location...' and click on **Next>**.

5. In the list of device types that appears, select **Sound, video and game controllers**. Click **Next>**.

6. Insert the SCOPE CD-ROM into your CD-ROM drive and click the **Have Disk...** button.

7. In the 'Install from disk' dialog click **Browse...** and select the CD-ROM drive from the drive selector drop-down list.

8. In the list of *.inf files that appears select **scope.inf** and click **OK**.

9. Back in the in the 'Install from disk' dialog click **OK**.

10. In the 'Select Device' dialog select 'ScopeBoard' and click **OK**.

11. The wizard now reports 'Windows searches for the device: Scope' Click **<Next>**.

12. Now the wizard copies the required files and updates the driver database.

13. When the message, 'Windows has finished installing the software' appears, click **Finish**.

14. When you are asked, 'Do you want to restart your computer now?', click **Yes**.

15.* When Windows has restarted, go to the Device Manager (**Start->Settings->Control Panel->System->Devices)** and locate 'ScopeBoard' under 'Audio, video and game controllers'. The driver is properly installed if there is no yellow warning symbol or red cross next to it in the device list.

* If you have cascaded more than one Pulsar, Pulsar SRB, or SCOPE boards in your system then Windows will again detect new hardware (as in step 2) when it restarts (step 15). Proceed for the additional card(s) exactly as described above for SCOPE, but use the appropriate corresponding driver(s) (in step 8) as indicated below .

SCOPE.INF	**= SCOPE Board**
PULSAR.INF	**= Pulsar Board**
PULSSRB.INF	**= Pulsar SRB**

* see Notes on following pages

Installing the SCOPE Software

The installation of the SCOPE program software is made quite easy by a custom Setup utility included on the CD. Just follow these simple steps to install SCOPE:

1. Insert the SCOPE software CD-ROM disc into your CD-ROM drive.

2. Close any programs that may be running.

3. In Windows Explorer, switch to the CD-ROM drive and doubleclick on **setup.exe** in the root directory.

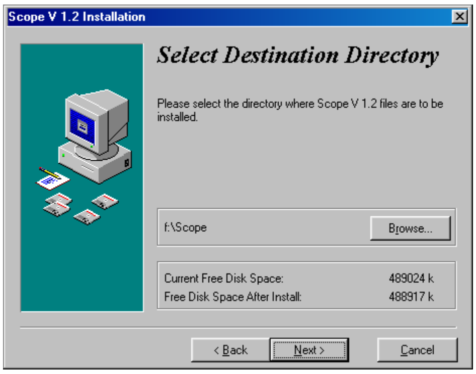
You can also start the setup program by typing <CD-ROM drive letter>:**\setup.exe** in the **Start** menu's **Run** utility (i.e. D:\setup.exe).

4. If, for some reason, Windows did not detect your SCOPE board, and therefore did not install the driver, you have the option now of running the hardware wizard to install the driver. Or, if you are upgrading to a newer version of the software, you should use this option to upgrade the SCOPE driver. If you choose to do so, follow the driver installation procedure described earlier.

5. In the 'Welcome' window, click on **Next>**.

6. The CreamWare License Agreement appears in the next window. Please read this and confirm that you agree by clicking on **Next>**.

7. (Note: at the time of writing of this guide the key is planned to be entered now. If this is not the case, proceed to step 8.)



Scope V 1.2 Installation

Select Destination Directory

Please select the directory where Scope V 1.2 files are to be installed.

f:\Scope

Current Free Disk Space: 489024 k

Free Disk Space After Install: 488917 k

If this is a first-time installation of SCOPE you will be asked to enter the **software key**. You'll find this number on the cover of this Installation Guide booklet. Enter the key carefully, then click on **Next** to continue.

8. Now you are asked to specify the directory into which you want to install the software. If you don't like the default directory, click on the **Browse** button to make your own target directory selection. Click on **Next**.

When installing a SCOPE update, you should always install the new version over the same directory in which the existing version is located. Each project which you have created with the older version includes a file path specification for each device included in the project. If your new SCOPE version is installed into a different directory than the previous version, and if the previous version is still present on your system, then loading of your projects under the new program version may nevertheless result in the loading of devices from the previous version.

9. In the 'Ready to Install!' Window click on **Next**.

10. All the required files will now be installed.

*Notes:

1. Installation under Windows 95 is essentially the same as it is under Windows 98, but some of the dialogs may vary slightly depending on which version of Windows 95 you are running.

2. If SCOPE was not found as new device on launching Windows (which is possible, depending on your computer and confirguration) then you must start the hardware wizard from the 'System' control panel applet. Then proceed as from step #3 from chapter 'Software'.

Note that the driver MUST be installed before it will be possible to launch the SCOPE software.

3. If, after installing the SCOPE driver, it does not appear in the control panel's **System** device list under **Audio, video and game controllers** then you should quit Windows, power down your computer, fire it back up and install the drivers again. We have noticed that Windows sometimes requires a few attempts before drivers are successfully installed (not just ours).

4. When you start the SCOPE software the first time, you may be asked to supply a software key in a 'Registration' dialog. This key is found on the cover of this guide. Enter the key, and click on **Save Key.**

SCOPE in the Windows Environment

The Windows Default Project (Winstartscope.pro)

Before going any further we should make one thing 'perfectly clear': SCOPE is not a 'sound card' in the conventional sense. When you install SCOPE, your computer is not only redesigned to provide new possibilities in the realm of digital audio, it is also made much more powerful.

In order to provide you with as reliable, stable and functional an environment as possible we have undertaken some rather unusual measures. But don't worry. Although we do things a little differently, you will grasp these concepts quickly as they are very intuitive. And soon you'll see what great flexibilitiy this approach adds to your system.

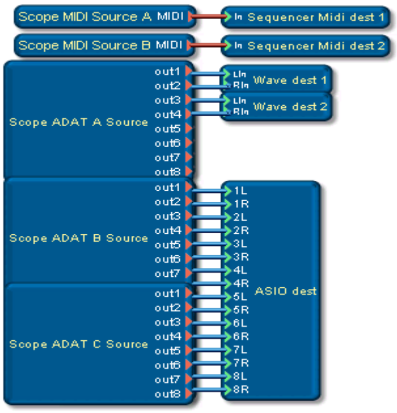
Immediately after Windows is started, SCOPE's physical inputs and outputs are available for your unrestricted use. They are enabled automatically by the **Winstartscope.pro** project, which is loaded in the background when Windows is launched.

Everything you need for seemless Windows multimedia and audio/ MIDI application software support is made available in this project. Drivers, in particular, are implemented as SCOPE **modules** (which you can view in the 'Project Window'). You can customize your Windows multimedia configuration by reconfiguring the **Winstartscope.pro** project. See the manual for details.

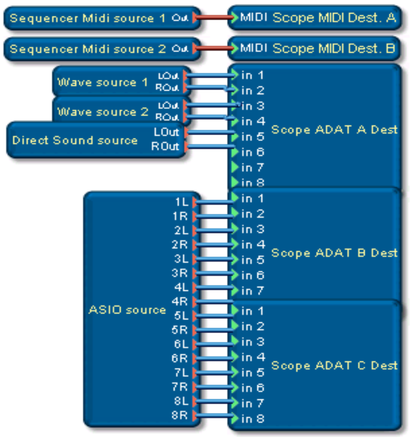
In general, any module labeled '**SCOPE** *xxx...*' represents one or a group of SCOPE physical inputs or outputs.

Facing next page: This is the default **Winstartscope.pro** interconnection diagram. Note that as this is a background application, you will not actually see the modules.

SCOPE Physical Input Destinations



Sources Routed to SCOPE Physical Outputs



SCOPE's two **physical** MIDI inputs and outputs (**SCOPE MIDI Source A** and **B**, and **SCOPE MIDI Dest. A** and **B**) are connected to **Sequencer Midi Dest 1** and **2** and **Sequnecer Midi Source 1** and **2** respectively. In your MIDI sequencer application the SCOPE MIDI I/Os will be identified as **SCOPE MIDI In/Out 1** and **2**.

The first 4 physical ADAT input and output channels (**SCOPE ADAT A Source**, 1-4 and **SCOPE ADAT A Dest.**, 1-4) are connected by default to the Windows **Wave dest** and **source 1** and **2** respectively. So, without having to run SCOPE, your Windows audio applications will have two stereo WAVE devices available (possibly connecting to our A8 or A16 A/D-D/A converters). In addition, the Windows **Direct Sound source** output is connected to channels 5 and 6 of the **SCOPE ADAT A Dest.** module.

Programs such as Cubase VST which use the ASIO driver specification to address the audio hardware more directly are supported by SCOPE via the **ASIO source** and **dest** modules (drivers). The advantage of using ASIO drivers over the normal WAVE drivers lies in the increased timing accuracy ASIO provides.

The 8 stereo (1L, 1R, 2L, 2R...) **ASIO dest** channels are connected to the **SCOPE ADAT B** and **C Source** modules, and the respective **ASIO source** channels are connected to the **SCOPE ADAT B** and **C dest** modules. Therefore 16 **ASIO** channels are available to any ASIO-compatible* program through **SCOPE ADAT B** and **C Source** and **dest** modules.

When Windows starts, **Winstartscope.pro** enables almost all of SCOPE's inputs and outputs. These are immediately available to your MIDI sequencing or audio programs - all you have to do is make the physical connections to your equipment and select the appropriate 'devices' in your application software.

In subsequent sections we will use three well-known software applications as examples to further illustrate the unique SCOPE multimedia implementation.

** At the time of writing only Steinberg's Cubase VST ASIO drivers are supported.*

Note: When you start the SCOPE software itself **Winstartscope.prov** routings are disabled and the SCOPE default project is loaded.

Click on the SCOPE icon in the 'System Tray' area (lower right corner of your screen) to call the Sample Rate Settings dialog. Here you can make adjustements to settings such as the sample rate and the sync mode. For more information check the later sections in this manual.

Creating a Customized Windows Default Project

The Windows default project makes SCOPE's inputs and outputs available to other programs without having to run the SCOPE software itself. All you need to do to create your own version is to set up the connections in a new project, and then click on the **Save as Windows Standard** button in the **SCOPE Settings** dialog (called from SCOPE's **Window** menu). The current project will then overwrite **Winstartscope.pro**.

If you save a complex setup involving several devices as a background project, be aware that Windows will take noticeably longer to load. In this case be careful to avoid launching the SCOPE software before the **Winstartscope.pro** devices have finished loading. You'll know the default project has fully loaded when the SCOPE icon appears in the task bar.

Some Tips On Using SCOPE With Other Windows Applications

You have now successfully installed the SCOPE hardware and software. To get you started using SCOPE with your favorite software we will provide notes and examples using three popular programs - Cubase VST, Cakewalk, and Logic Audio. These examples describe the basic settings for these programs and offer some other hints as well. Note that the tips we provide for Cakewalk users apply also to most other audio/MIDI applications.

Steinberg's Cubase VST

Using SCOPE with Steinberg's Cubase VST is easy but you must follow a few rules. There are two important items to observe to achieve maximum compatibility:

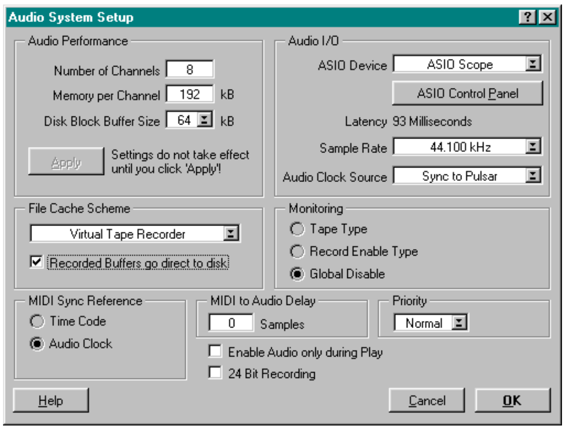
Ensure that the sampling frequencies are set identically in the two programs, and include the ASIO drivers in your project so VST can use them. (Also, make sure you are using at least version 3.55R3 of Cubase VST).

When Windows starts with the default **Winstartscope.pro** project, the **sample rate** is set to **44.1 kHz** (word clock master) and 8 stereo **ASIO** channels are supported through **ADAT** devices 2 and 3 (see previous page).

Before running VST, but after installing SCOPE, you should run Steinberg's 'Setup MME' program (Start->Programs->Steinberg Cubase VST...) to ensure that the MIDI inputs and outputs are enabled.

After starting VST check Audio-System-Setup to confirm that the SCOPE audio driver modules are available and properly integrated.

The following dialog appears:


Audio System Setup

Audio Performance
Number of Channels 8
Memory per Channel 192 kB
Disk Block Buffer Size 64 kB
Settings do not take effect until you click 'Apply'!

Audio I/O
ASIO Device ASIO Scope
Latency 93 Milliseconds
Sample Rate 44.100 kHz
Audio Clock Source Sync to Pulsar

File Cache Scheme
Virtual Tape Recorder
☑ Recorded Buffers go direct to disk

Monitoring
○ Tape Type
○ Record Enable Type
● Global Disable

MIDI Sync Reference
○ Time Code
● Audio Clock

MIDI to Audio Delay
0 Samples

Priority
Normal

☐ Enable Audio only during Play
☐ 24 Bit Recording

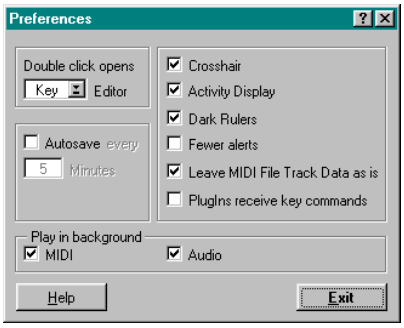
Under **ASIO Device**, select **ASIO Scope**. The **Sample Rate** and **Audio Clock Source** items cannot be changed since VST adapts to SCOPE's settings for optimal adjustment. You can change the sample rate settings at any time either by clicking on the SCOPE icon in the task-bar (if the SCOPE software is not running) or by selecting **Sample Rate Settings** from the **Window** menu in SCOPE. Cubase will inform you when you change the sample rate.

So you can always hear the audio and MIDI playback during the simultaneous use of SCOPE and VST, enable **Play in Background** in the **Preferences** dialog.

Which method you choose to use depends on your personal preferences and your sequencer's MIDI thru characteristics.

Consult the SCOPE and Cubase manuals to familiarize yourself with the various operational procedures of both programs before calling either CreamWare or Steinberg technical support. You will find that in a very short time you will begin to appreciate the vast possibilities available in this environment!

Have fun sequencing with SCOPE and VST!


Preferences

Double click opens
Key Editor

☐ Autosave every 5 Minutes

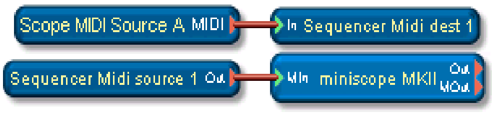
☑ Crosshair
☑ Activity Display
☑ Dark Rulers
☐ Fewer alerts
☑ Leave MIDI File Track Data as is
☐ PlugIns receive key commands

Play in background
☑ MIDI
☑ Audio

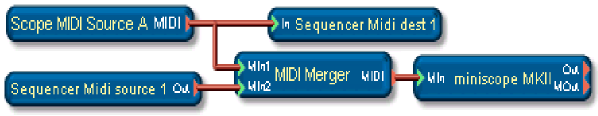
SCOPE MIDI source/dest A and **B** und **Sequencer MIDI dest/source 1/2** are already integrated in the default project.

The following are two typical examples of possible MIDI connections:

1. MIDI data are routed through to the **Miniscope** device with MIDI Thru in the sequencer = ON.



2. MIDI data are routed simultaneously to the device and the sequencer with MIDI Thru in the sequencer = OFF.


Scope MIDI Source A MIDI
In Sequencer Midi dest 1
MIn1 MIDI Merger MIDI MIn2
Sequencer Midi source 1 Out
MIn miniscope MKII Out MOut

Cakewalk

INFO The following example uses 16 audio tracks. Set Cakewalk and SCOPE for another number of tracks (2-32) in a similar manner.

SCOPE uses the Windows MME interface to deliver and receive audio to and from other programs. From the point of view of SCOPE, this interface appears as **WAVE source** and **dest** modules. Other programs will have **SCOPE Play** and **SCOPE Rec** 'devices' available to them.

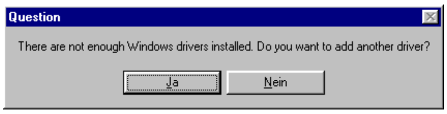
To use the MME interface follow these steps:

1. Make available the WAVE source and dest modules in SCOPE.

SCOPE will see as many audio tracks as you have made available through the number of **WAVE dest** modules you have included in the project. Each **WAVE dest** module allows for two tracks to be routed to a program. Therefore, if you want 16 channels of audio in Cakewalk, you must include 8 **WAVE dest** modules.

Similarly you must include as many **WAVE source** modules as you require to recevie audio playback from a given number of tracks.

Sooner or later, when adding modules, you'll probably encounter this dialog:

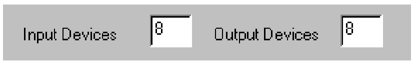

Question

There are not enough Windows drivers installed. Do you want to add another driver?

SCOPE is letting you know that there are not enough reserved Windows
drivers and asks you if you want to introduce another driver to Windows.
Click on **Yes**, and SCOPE will tell you that Windows must be restarted for the new driver to be activated. Before you do so, load the required number of modules and save your project. Then restart Windows.

2. Load the SCOPE MME Drivers

Now Windows supplies other programs with MME drivers. You can check
the number of drivers in the **SCOPE's driver setup** (Control Panel -> System -> Device manager -> Sound, video and game controllers -> ScopeBoard Properties -> Settings):



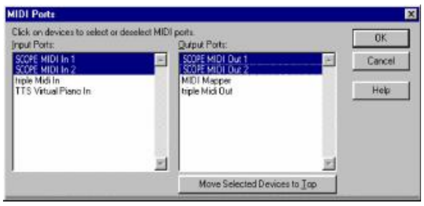
Input Devices 8 Output Devices 8

INFO As an alternative to the above method you can load the drivers directly in the 'ScopeBoard Properties' dialog. Close all applications and enter the desired number of input and output devices remembering that each one is a stereo driver. Then restart Windows.

To use these drivers you must restart SCOPE and load the previously created project - with it the drivers are loaded, too.

3. MIDI Settings

In Cakewalk, open the **MIDI Ports** dialog (Tools -> MIDI Devices).



MIDI Ports

Click on devices to select or deselect MIDI ports.

Input Ports: SCOPE MIDI In 1 (selected), SCOPE MIDI In 2 (selected), triple Midi In, TTS Virtual Piano In

Output Ports: SCOPE MIDI Out 1 (selected), SCOPE MIDI Out 2 (selected), MIDI Mapper, triple Midi Out

Enable the SCOPE MIDI Ports. The default is two MIDI Ins and two MIDI
Outs. You can specify the number of SCOPE MIDI ports in the SCOPE driver setup dialog (in the Windows Device Manager).

Note that the number of MIDI ports here is related not to SCOPE's physical MIDI I/O, but to the number of **Sequencer MIDI source** and **dest** modules loaded.

For connecting and using MIDI modules, see chapter "VST".

For detailed information on using SCOPE and Cakewalk consult the program manuals.

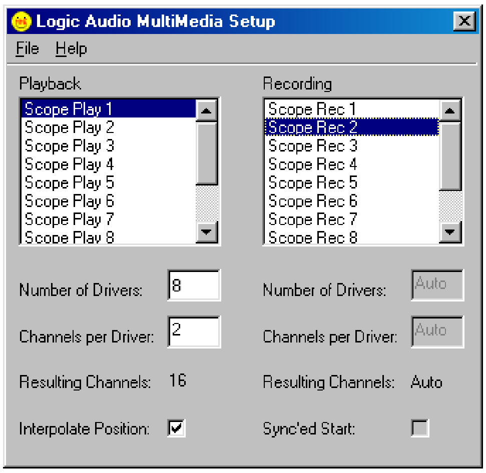
Emagic Logic Audio

INFO The following example uses 16 audio tracks. Set Logic and SCOPE to use another number of tracks (2-32) in a similar manner.

Follow procedures 1 and 2 in the preceeding CakeWalk section.

3. Tell Logic to use the MME drivers

In your Logic directory you will find the program MMSetup.exe. Start it, and the following dialog will appear:



Logic Audio MultiMedia Setup

Playback: Scope Play 1 (selected), Scope Play 2, Scope Play 3, Scope Play 4, Scope Play 5, Scope Play 6, Scope Play 7, Scope Play 8

Number of Drivers: 8

Channels per Driver: 2

Resulting Channels: 16

Interpolate Position: ☑

Recording: Scope Rec 1, Scope Rec 2 (selected), Scope Rec 3, Scope Rec 4, Scope Rec 5, Scope Rec 6, Scope Rec 7, Scope Rec 8

Number of Drivers: Auto

Channels per Driver: Auto

Resulting Channels: Auto

Sync'ed Start: □

Click on **Scope Play 1** and under **Number of Drivers** the number of Scope playback drivers you have used so far is displayed - probably 2. Enter the number of drivers you want to use to play audio tracks from Logic into Scope - in our example 8. This results in 8*2 = 16 channels.

Now follow the same procedure for **Scope Rec 1**.

If you click on a different driver (2 - 8, as shown in the Recording list), the
fields **Number of Drivers** and **Channels per Driver** should display the value **Auto**.

Note th at the number of drivers you specify here must correspond **exactly** to the number of Wave Source/Dest modules previously loaded in the SCOPE project. If the numbers don't match correctly, you will obtain the following error message upon starting Logic:

PC AV Error:5
Failed to recover device from playback/recording

INFO As an alternative to loading multiple stereo Wave modules, you can load the Multi-Interleaved Module, set it for the desired number of channels (e.g., 16) and connect it as appropriate. To use this module, you should enter the following settings for both playback and record in the Logic dialog described on the previous page:

Number of Drivers = 1
Channels per Driver = 16

For using and connecting MIDI modules, see VST section.

Now you're ready to mix 16 Logic channels in a SCOPE project, and bus 16 Big Mixer signals to Logic. For details see next sections of this manual and consult the Logic documentation.

Using multiple boards with S/TDM bus connector

Important notes on using multiple boards with S/TDM bus connector (Pulsar/ Pulsar SRB/SCOPE)

Order of hardware detection

If you add another Pulsar/Pulsar SRB or Scope board to your SCOPE system, different mainboards may detect the boards in different orders. Possibly this order is not what you prefer and you would like to change it - e.g. to maintain compatibility to previous projects. There are different ways to test and to change this order.

SCOPE + Pulsar

You already have an installed SCOPE board and you are adding a Pulsar board. After launching the application you find that the Pulsar hardware I/Os are loaded into your Default Project instead of the SCOPE I/Os which you want to use primarily.

This indicates that the mainboard's hardware detection detected the Pulsar board as first board so that your SCOPE board has been pushed up to the second position.

In this case you will also find that in the Device Manager (Start -> Settings -> Control Panels -> System -> Device Manager -> Sound, video and game controllers) the settings for the different driver options can be opened from the Pulsar board and not from the SCOPE board anymore.

To restore the logical order 1. SCOPE -> 2. Pulsar you can

- swap the physical slot positions of both boards on the mainboard,

or

- add two sections to the file **CSET.INI** in the directory **../Pulsar/App/Bin**. Open the file with a text editor (Editor or Wordpad) and add the following two sections:

```
[board0]
boardid=1

[board1]
boardid=0
```

Save the file, and restart Windows.

Now the 'correct' order of the boards should be restored. **[board0]** refers to that board which *should* be detected as first board - in our example your old first SCOPE board that has the **boardid=1** *(from the mainboard's point of view).* The second board *to be detected* **[board1]** is your new Pulsar board, which has the **boardid=0** *(from the mainboard's point of view).*

If you want to add a third board (Pulsar SRB, Pulsar or SCOPE), please note the position of the existing boards before installing the new board. If needed you can add another entry to the **CSET.INI** like

```
[board2]
boardid=2
```

to adapt the order to your requirements.

SCOPE + Pulsar SRB

You already have an installed SCOPE board and are adding a Pulsar SRB board. After launching the application you find that in the project Window the hardware I/Os are labelled with '2', e.g. SCOPE ADAT **source 2** or SCOPE Midi **source A 2 instead of** SCOPE ADAT **source 1** or SCOPE Midi **source A 1** etc.

This indicates that the mainboard's hardware detection detected the Pulsar SRB as first board so that your SCOPE board has been moved up to the second position.

In this case you will also find that in the Device Manager (Start -> Settings -> Control Panels -> System -> Device Manager -> Sound, video and game controllers) the settings for the different driver options can be opened from the Pulsar SRB and not from the SCOPE board anymore.

To restore the logical order 1. SCOPE -> 2. Pulsar SRB you can

- swap the physical slot positions of both boards on the mainboard,

or

- add two sections to the file **CSET.INI** in the directory **../Pulsar/App/Bin**. Open the file with a text editor (Editor or Wordpad) and add the following two sections:

```
[board0]
boardid=1

[board1]
boardid=0
```

Save the file, and restart Windows.

Now the 'correct' order of the boards should be restored. **[board0]** refers to that board which *should* be detected as first board - in our example your old first SCOPE board that has the **boardid=1** *(from the mainboard's point of view).* The second board *to be detected* **[board1]** is your new Pulsar SRB board, which has the **boardid=0** *(from the mainboard's point of view).*

If you want to add a third board (Pulsar SRB, Pulsar or SCOPE), please note the position of the existing boards before installing the new board. If needed you can add another entry to the CSET.INI like

```
[board2]
boardid=2
```

to adapt the order to your requirements.

These notes also apply to cascading SCOPE boards!

General

- If a **pulsar board is the first board detected**, the 'default project' **Winstart.pro** (in the ../Pulsar/App/Bin or ../Scope/App/Bin directory) will be loaded (see Installation Guide!)

- If a **SCOPE board with 24 I/O board is the first board detected**, the 'default project' **Winstartscope.pro** (in the ../Pulsar/App/Bin or ../Scope/App/Bin directory) will be loaded (see Installation Guide!)

Hardware I/Os of the board that has been detected as board #2 are always labelled with the number 2 at the end of the module name, e.g.

- if you add a Pulsar ADAT dest module to a SCOPE project with SCOPE I/Os (SCOPE board as board #1), the Pulsar ADAT dest module will be labelled with 'ADAT dest 2' (in a Pulsar project it will be labelled as Pulsar ADAT dest 2)

- if you add a SCOPE ADAT dest module to a SCOPE project with Pulsar I/Os (Pulsar as board #1), the SCOPE ADAT A dest module will be labelled with 'SCOPE ADAT dest 2' (in a Pulsar project it will be labelled the same)

Your CreamWare support!

Technical Specifications

SCOPE DSP Board Technical Data

Description

15 SHARC DSPs @ 60 MHz
PCI Busmaster with special support for delay lines
32 MByte SDRAM onboard for memory intensive signal processing
32 Bit Audio-Bus architecture

5 independent S/TDM extension busses for a total of 720 channels
Onboard connections for plug-in extension daughterboards

Sample frequencies from 24 kHz - 400 kHz

Digital Inputs and Outputs

S/TDM	SCOPE Bus connections	144 channels 32 Bit @ 44.1 kHz
SYNC	Sync Plate connectors	Wordclock In/Out 2 x ADAT 9-pin protocol
Direct I/O	Daughterboard connectors	Multifunction I/O-interface

General Data

PCI add-in card	Rev. 2.1 compliant 33 MHz (busmaster slot)
Vendor-ID	14B5 hex, **creamw@re**
Power consumption	15W
Operating temperature	0 - 55°C (Minimum, Maximum)
Dimensions	33.8 x 10.8 x 1.6 cm (not incl. metal bracket)

24 I/O Daughterboard Technical Data

Description

3 ADAT ODI Interfaces
2 MIDI Interfaces including cablewhip assembly
Mounts as a daughterboard to **SCOPE**

Sampling Frequency

Master	32, 44.1, 48 and 96 kHz
Varispeed	44,1 kHz +/- 200 cents 48 kHz +100/-300 cents (ADAT* compatible)
Slave	30 - 100 kHz (Wordclock)

Digital Inputs and Outputs

ADAT	Optical EIAJ connectors	8 Channels, 24-Bit, 3 In, 3 Out
MIDI	5-pin DIN	2 In, 2 Out, 2 Thru

General Data

Power consumption	0.4 W
Operating temerature	0 - 55°C (Minimum, Maximum)
Dimensions	10.9 x 10.5 x 1.1 cm

Sync Plate Technical Data

Description

Wordclock and ADAT* Timecode Synchronization (selectable)
Master and Slave
Mounts on PC chassis back panel (no slot used)

Sampling frequency	24 kHz - 400 kHz via BNC

Synchronization

Word clock	75 Ohm, BNC, input and output

Timecode

ADAT* 9-Pin connector	Input and output

General

Power consumption	0.1 W
Operating temperature	0 - 55°C (Minimum, Maximum)
Dimensions	12.3 x 9.8 x 1.5 cm

Package Contents

SCOPE-Board	with attached 24 I/O board Sync Plate MIDI adapter bracket MIDI cablewhip assembly Installation CD-ROM Quickstart Guide Akai Sample CD-ROM
CD Contents	Installation program Driver Software Manual (Adobe Acrobat)

**ADAT is a trademark of the Alesis corporation.*

Warranty and Disclaimer

CREAMWARE GmbH ("CREAMWARE") warrants this product to be free of defects in materials and workmanship for a period of one (1) year for parts and for a period of ninety (90) days for labor from the date of original retail purchase. This warranty is enforceable only by the original retail purchaser. To be protected by this warranty, the purchaser must complete and return the enclosed warranty card within fourteen (14) days of purchase.

During the warranty period CREAMWARE shall, at its sole and absolute option, either repair or replace free of charge any product that proves to be defective on inspection by CREAMWARE or an authorized service representative. In all cases disputes concerning the warranty shall be resolved as prescribed by law.

To obtain warranty service, the purchaser must first call or write CREAMWARE at the address and telephone number printed below to obtain a Return Authorization Number and instructions concerning where to return the unit for service. All inquiries must be accompanied by a description of the problem. All authorized returns must be sent to CREAMWARE or an authorized CREAMWARE repair facility postage prepaid, insured, and properly packaged. Proof of purchase must be provided in the form of a bill of sale, canceled cheque, or some other positive proof that the unit is within the warranty period. CREAMWARE reserves the right to update any unit returned for repair. CREAMWARE reserves the right to change or improve the design of the product at any time without prior notice. This warranty does not cover claims for damage due to abuse, neglect, alteration or attempted repair by unauthorized personnel, and is limited to failures arising during normal use that are due to defects in material or workmanship in the product.

ANY IMPLIED WARRANTIES INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE LIMITED IN DURATION TO THE LENGTH OF THIS LIMITED WARRANTY. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you.

IN NO EVENT WILL CREAMWARE BE LIABLE FOR INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES RESULTING FROM THE BREACH OF ANY EXPRESS OR IMPLIED WARRANTY, INCLUDING, AMONG OTHER THINGS, DAMAGE TO PROPERTY, DAMAGE BASED ON INCONVENIENCE OR ON LOSS OF USE OF THE PRODUCT, AND, TO THE EXTENT PERMITTED BY LAW, DAMAGES FOR PERSONAL INJURY. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights which vary from state to state. *(continued overleaf...)*

This warranty applies only to products sold in the United States of America or Canada. The terms of this warranty and any obligations of CREAMWARE under this warranty shall apply only within the country of sale. Without limiting the foregoing, repairs under this warranty shall be made only by a duly authorized CREAMWARE service representative. For warranty information in other countries please refer to your local distributor.

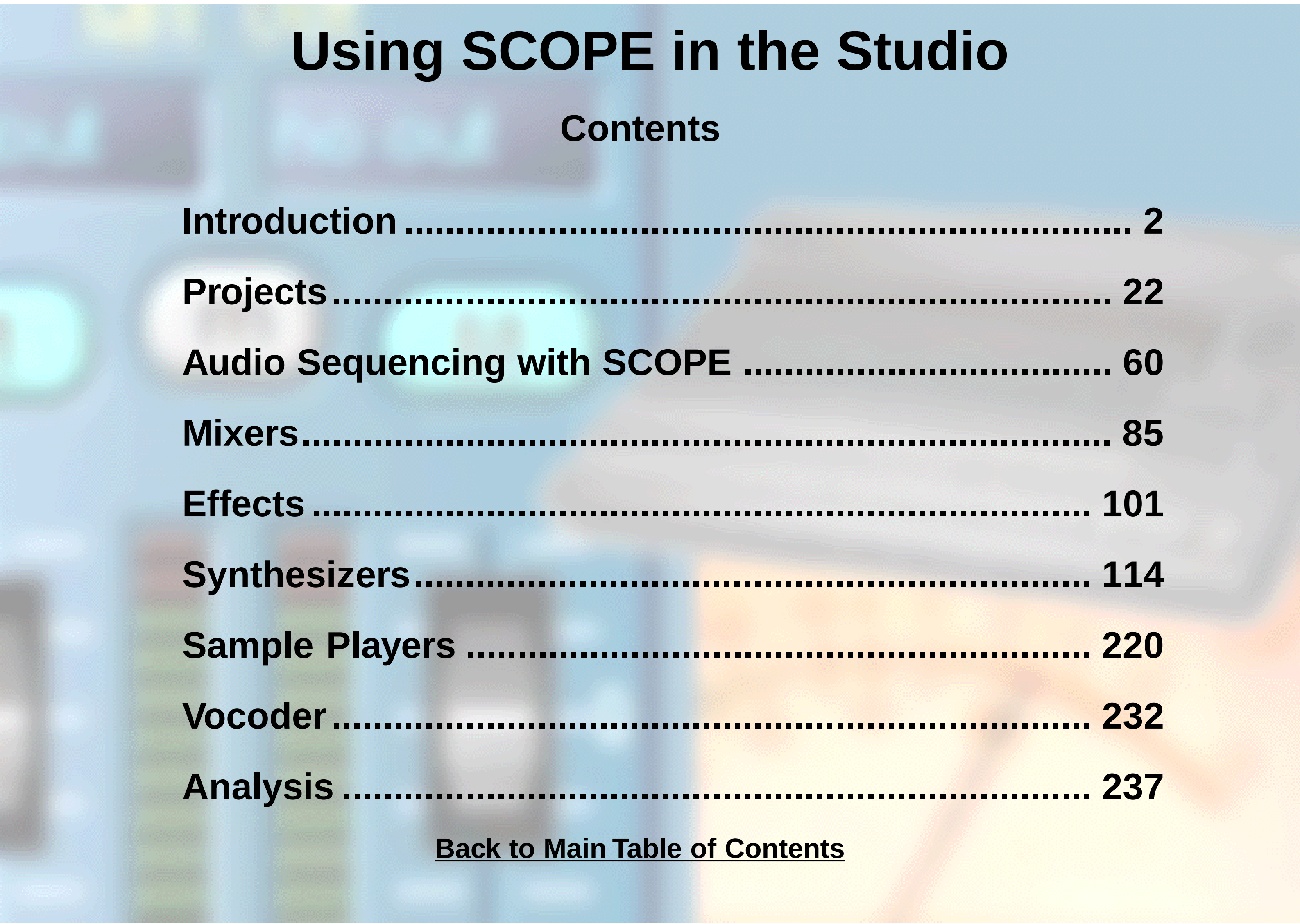
Shipping Address:

CreamWare US Inc.
855-C Conklin St.
Farmingdale, NY 11735
USA

Sales and Support Office:

CreamWare US Inc.
6915 Russell Ave.
Burnaby, B.C. V5J 4R8
Canada

Tel	604-435-0540
Fax	604-435-9937
Sales	800-899-1939
Support	604-435-5158



Using SCOPE in the Studio

Contents

Back to Main Table of Contents

Introduction

- Foreword

This chapter of the manual will help you learn the basics of SCOPE. The manual has been designed to be comfortable to use without the need to "zoom in" on things to be able to read them. It is intended to be used simultaneously with SCOPE, so you can immediately try out the things you read.

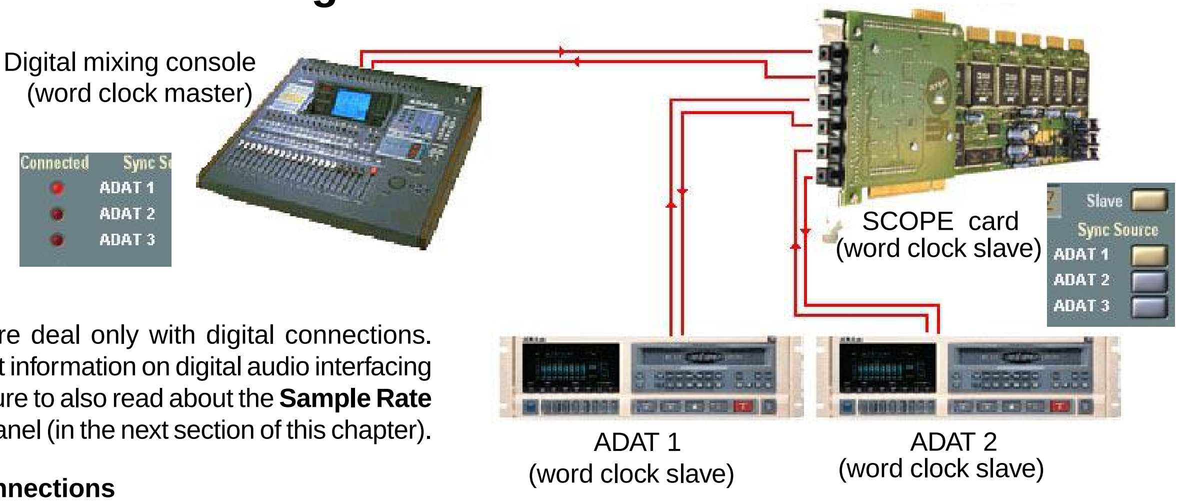
What makes SCOPE special is that it allows you to alter the functionality of the hardware *while* it is being used. The SCOPE card contains fifteen powerful digital signal processors (DSPs) which handle all signal processing. Their high combined processing capacity allows them to numerically simulate synthesizers, mixing consoles and audio effects in various combinations, in realtime.

In addition, the extensive interfacing capabilities of SCOPE – both hardware and software – not only allow the smooth integration of external devices such as ADAT recorders into the SCOPE environment, but also permit SCOPE to work directly with MIDI and hard disk audio recording software such as Cubase VST, Logic Audio and CakeWalk Audio *running on the same computer*. You can combine ADAT tracks with sequenced audio tracks and MIDI-sequenced SCOPE synths, mix everything down in a SCOPE mixer, and add SCOPE and external effects to the mix.

The number of synths, effects, etc. which SCOPE can simultaneously generate depends primarily upon the devices themselves. SCOPE synths, effects and mixers place varying demands upon the computing capacity of the DSPs. The SCOPE sample players additionally draw upon the computing and data transfer capacity of your PC. Thus, the limit also depends to some extent upon the configuration of your system.

But within these limits – there are scarcely any limits. If you want to sing live through a Modular synth filter – it's easy. If you can think of it, you can probably do it.

- Digital Connections



The examples here deal only with digital connections. For more important information on digital audio interfacing with SCOPE, be sure to also read about the **Sample Rate Settings** control panel (in the next section of this chapter).

Digital Audio Connections

The SCOPE **ADAT 1 In** and **ADAT 1 Out** connectors are shown connected to a digital mixing console. Two digital 8-track recorders (which just happens to be ADATs) are connected to **ADAT 2 In, ADAT 2 Out, ADAT 3 In** and **ADAT 3 Out**.

Digital Audio Synchronization – The Word Clock

When digital audio connections are made between devices, the devices must be **synchronized**. This is done with reference to a **word clock**. One device operates as the word clock **master** and produces the word clock signal. The connected devices operate as **slaves** and synchronize to this clock, which can often be derived directly from incoming ADAT audio signals.

In the example above, the digital mixer is the word clock master. SCOPE is set up as a word clock slave via the **Sample Rate Settings** control panel (see the next section) and synchronizes itself to the mixer's word clock via the ADAT 1 input signal. Both ADAT machines also operate as word clock slaves. These, however, synchronize themselves to the SCOPE card ADAT output signals, respectively.

- Sample Rate Settings

SCOPE can operate as either word clock master or slave for digital audio interfacing. Word clock settings are made via the **Sample Rate Settings** control panel, which is accessed via the Window menu.

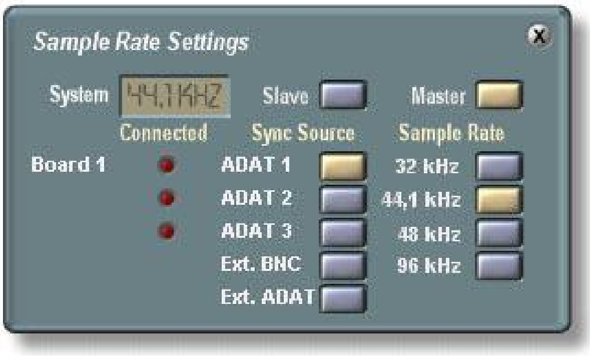
The preceding Digital Connections section provides some background on the role of the word clock in a system.

Settings

Under **System**, the current SCOPE word clock frequency or **sample rate** is shown (valid for both internal and external word clock sources). Below this, LEDs indicate the SCOPE inputs on which a digital audio signal is present.

Clicking on the **Master** button puts SCOPE into word clock master mode. In this mode, the buttons below allow selection of the system **Sample Rate.** Connected devices may limit your choices, since **all devices must operate at the same rate**, and not every device supports all rates. (44.1 kHz is almost universal, thus always a safe choice.)

Clicking on the **Slave** button puts SCOPE into word clock slave mode. Here it is necessary to click on one of the **Sync Source** buttons below to tell SCOPE which of its digital inputs to use as the word clock sync source. In slave mode, no sample rate selection is needed or possible, as this is determined by the master device.



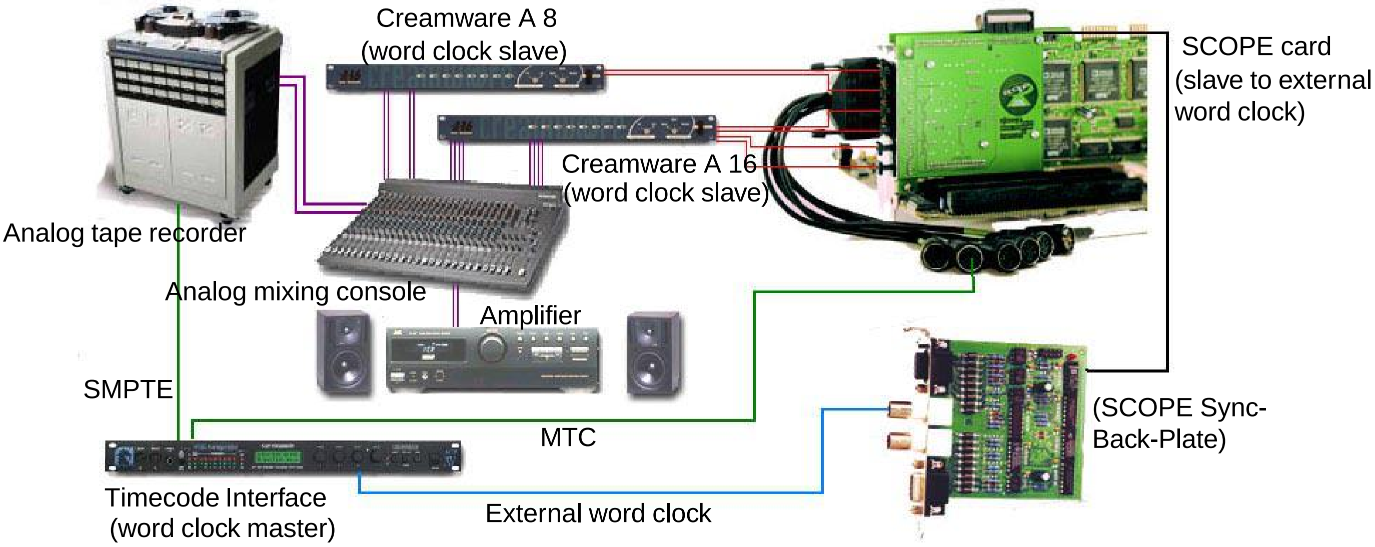
Note: The current sample rate setting is saved with your project. Before saving, you can delete the module WordClockPipe (this module can be found near the top of the module tree listing in the Modules drawer of the Project Explorer tool — refer to the Project Explorer chapter for more details). This permits this project to be reloaded without changing the sample rate currently in effect.

Word clock Settings And SCOPE Performance

It is important to be aware that the system word clock (internal *or* external) is also the SCOPE DSP sample clock. While higher sample rates do increase audio bandwidth and quality, they correspondingly reduce the available DSP computation time and thus the number of synths, synth voices, etc which SCOPE can maintain. Aside from specific "pro" uses, 44.1 kHz is generally optimal.

A lower sample rate *can* sometimes be preferable. If you're using *only* SCOPE synths (no audio or samples), or if audio fidelity is not highly critical, the reduced bandwidth at 32 kHz may not be objectionable. You might choose to trade it off against the increased functional capacity which you get in return. But when working with audio sequencers or external gear, this option may not always be available.

- Analog Connections



Extended possibilities for the use of SCOPE with external analog equipment (effects units, mixing consoles etc.) are afforded by the addition of an external **multi-channel A/D-D/A converter** such as the **Creamware A8** or **A16** into the setup. These devices connect to the SCOPE **ADAT interfaces** and provide an additional eight or sixteen analog inputs and outputs, respectively, for use as desired.

In the example, a timecode interface acts as the word clock master. It generates both an MTC stream and a digital word clock signal on the basis of the SMPTE signal coming from the analog tape recorder. SCOPE slaves to the external word clock, which is connected via the SCOPE SyncPlate. The A8 and A16 are word clock slaves to SCOPE's digital ADAT signal.

With this setup, an audio sequencer can be synchronized to an analog multitrack recorder.

- Getting Started

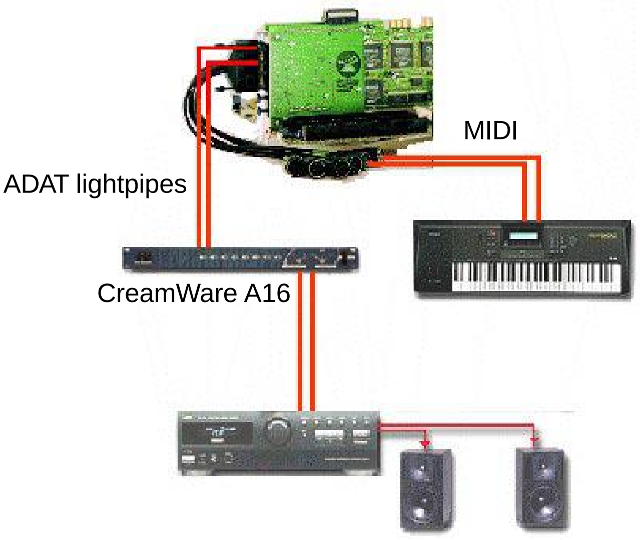
The previous chapters offered an overview of some of SCOPE's basic techniques and dialogs. In this chapter, the focus is on the use of SCOPE in your production environment.

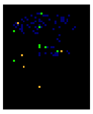
If you haven't already done so, connect the SCOPE MIDI A input to the output of a MIDI keyboard or controller of your choice, and connect the ADAT 1 output to an ADAT port on one of your external devices to permit monitoring.

When you're done, your setup should roughly resemble the picture at right. Now you've got everything you need to proceed through the following sections of the Introduction and be able to try things out as you read about them.

(If you absolutely can't wait and you want to poke around onscreen first, you can even do without the external connections. But – surely you'll want to *hear* something!)

To begin, *turn your amplifier down* (transients on the card outputs during program startup are possible). Double-click on the SCOPE icon on your desktop (see above right). In a few moments you'll be looking at the SCOPE screen.





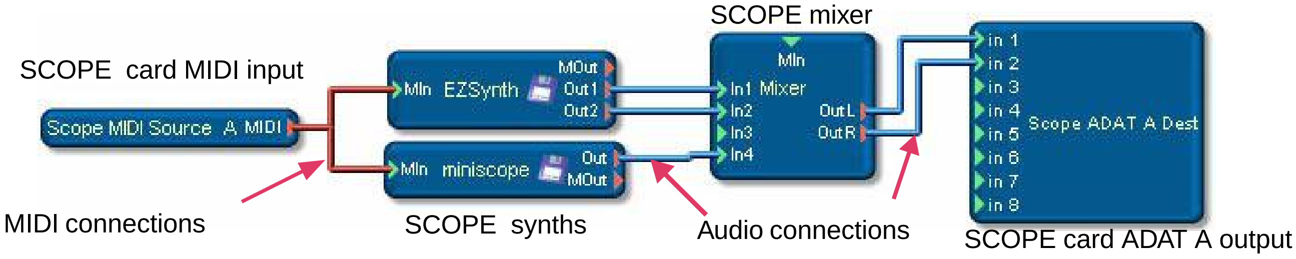
SCOPE Icon

Open the **File** menu at top left and click on **Load Project**. Navigate into the ..\Scope\Data\Examples folder and double-click on **Discover.pro**. When this is done loading, a small but complete SCOPE project will appear in the Project window. In this project, SCOPE is the word clock master. Make sure your external device is configured as a word clock slave. Go ahead – play a few notes!

The Discover project is discussed on the next page. Simple though it is, it illustrates many basic but important SCOPE features. After you've played around with it a bit, continue through the rest of the Introduction to get into the basic features in more depth. Don't be afraid to experiment or even to take the Discover project completely apart. That's the idea! You can always reload the project – directly from the CD, if all else fails.

Next, in the **Projects** chapter, you'll find information about more advanced features and applications, including the use of SCOPE in tandem with other PC audio and MIDI software and with external audio and MIDI devices.

- The "Discover" Project



This example project is installed with SCOPE. To load it, follow the instructions on the previous page.

The project consists of two SCOPE synths and a mixer. Both synths get their MIDI input from the Scope MIDI Source A *module*, which represents the SCOPE card MIDI input A. The synths are mixed via a small mixer, whose outputs are routed to the Scope ADAT 1 Dest *module*, which represents the SCOPE card ADAT Out 1.

Thus, the *modules* at both ends represent hardware. The ones in the middle – the synths and mixer – are strictly data. But they work, as you can easily demonstrate.

Play a few chords. Both synths are in Omni mode, so both will play at once. In this project, the **EZSynth** is 4-voice polyphonic, the **miniscope** monophonic. Thus, with each chord you play, the miniscope produces a bass tone on *one* of the chord notes.

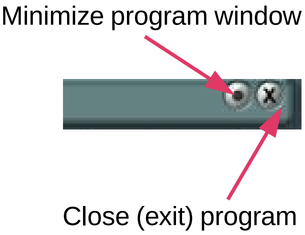
At the bottom of the screen you'll spot the *minimized* synth and mixer **icons**. Double-click on these to see the full-size **device surfaces**, where you can make adjustments such as setting MIDI channels or muting mixer channels.

On each device surface there is also a *Presets* button which opens the device's preset list. Simply double-click on a name in this list to load a new synth preset. Try this now with the EZSynth. To hear the changes more clearly, open the mixer surface (by double-clicking on the mixer icon) and mute mixer channel 4 to silence the miniscope.

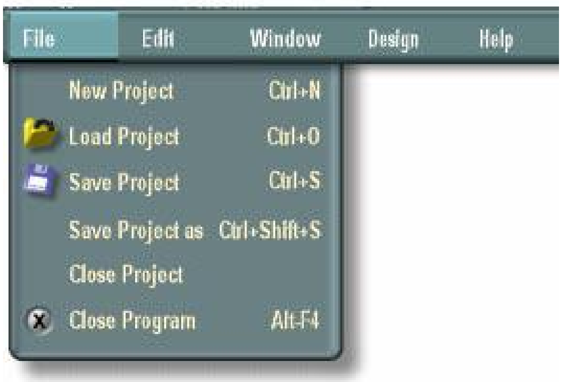
As you proceed through the rest of the Introduction, you can use this project to try out the things you read about. Be fearless! No matter how much you tear things apart, you can reload the original project at any time.

- Workspace Basics

This section and the ones which directly follow give an overview of the SCOPE workspace – the menus, windows and other onscreen widgets with which you carry out your activities in SCOPE. Much of this will be quickly recognizable to Windows users (even though the SCOPE version is substantially better-looking!)



Fixed at the top of the screen is the SCOPE **Menu** bar. At its far right end are two small buttons. The **Close** button ends the SCOPE program. The **Minimize** button leaves SCOPE running but sweeps it from the screen, letting whatever is behind it show through. Synths, mixers etc. remain active. This is handy for getting at the Windows desktop (program icons, etc.) without having to exit from SCOPE. (You can also switch directly to other *already-loaded* programs *without* minimizing SCOPE.) To bring SCOPE back onscreen, click on the SCOPE entry in the Windows Task bar or use the standard Windows **Alt-Tab** task switching function.menu

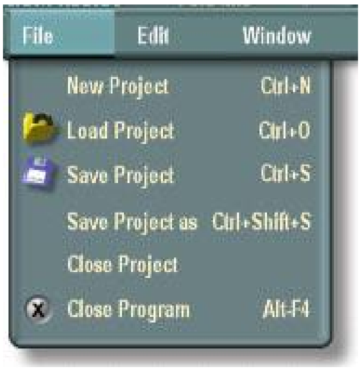


Also found on the Menu bar are the **File** , **Edit, Window, Design** and **Help** menus. The **File** menu is for the big stuff: loading, saving, and dumping projects and exiting from SCOPE. The Edit menu contains standard commands. The **Window** menu gives you control over your workspace, letting you open and close the various SCOPE windows and control panels. Any element of the SCOPE workspace can be hidden away when not needed.

- The File Menu

The **File Menu** is for opening and closing **projects**.

A **project** is a file which contains complete information about *everything* going on in SCOPE – which mixer and/ or synths are loaded, where they are onscreen, how they're connected, all of their settings, etc.



New Project sets the stage for new project creation by loading a default project (which you can customize to suit your own needs, by the way).

Load Project reloads a project exactly as it was saved.

Save Project stores a complete picture of your current SCOPE setup. Use it in the early AM before going off to bed, or if you leave the computer alone for a minute, or *anytime* you'd hate to lose what you've set up.

Save Project as lets you save a variation on a project under a new name or in a different place, when you don't want to write over the original project file.

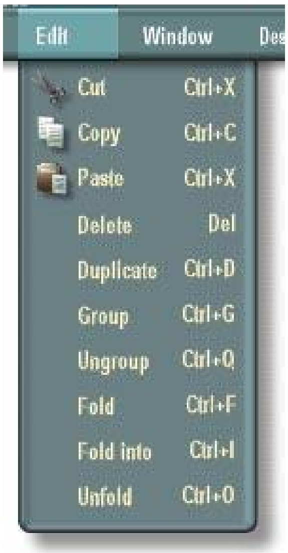
Close Project simply clears out the existing project. Your next step will be either New Project or Load Project. Or maybe **Close Program**, which exits SCOPE.

Load Project, New Project and Close Project/Program wipe out your currently loaded project. Thus, they all give you a chance to save it first, if it hasn't been saved yet.

- The Edit Menu

The Edit menu includes standard commands which will be familiar to you from other Windows applications, along with some commands which are frequently used when constructing custom devices.

Items In the Edit Menu



Cut removes an object from the work space and saves it into the Windows clipboard.

Copy saves an object into the Windows clipboard.

Paste copies an object from the Windows clipboard into the work space.

Delete removes an object from the work space.

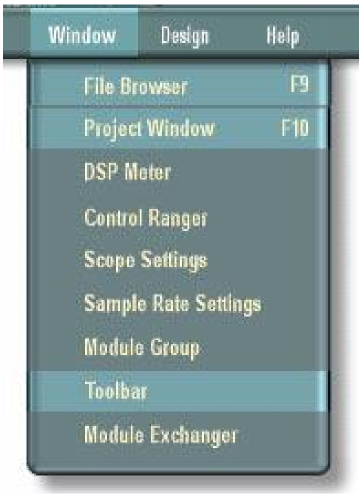
Duplicate creates and adds a new copy of an object.

Group combines multiple objects into a group. This operation can be reversed using **Ungroup**.

Fold, **Fold** Into and **Unfold** are explained in the circuit design chapter.

- The Window Menu

The **Window** menu lets you open and close SCOPE workspace windows and control panels.



Using The Window Menu

Any desired combination of SCOPE windows and control panels can be open at one time. If a window or control panel is already open, you switch to it simply by clicking on it. If it isn't, you use the Window menu to open it.

When a window or control panel is already open, selecting it in the Window menu will close it. Each window and control panel has a Close button (in its upper right corner) which can also be clicked on to close it directly.

Items In The Window Menu

Some of the items in this menu are covered later in the Introduction. A few require more in-depth description and are more appropriately dealt with elsewhere than in the Introduction:

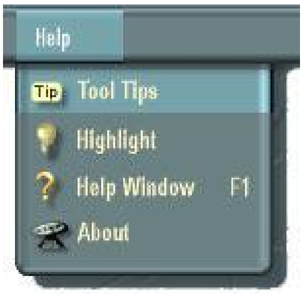
– Details of the **Project Window** and of the **SCOPE Settings, Control Ranger** and **DSP Load** control panels are presented in the *Projects* chapter.

– The **Module Group Window** and the **Module Exchanger** are explained in the **Circuit Design** chapter.

– The **Toolbar** and the items in the **Design Menu** are explained in the chapter **Surface Design**.

- The Help Menu

The Help Menu provides access to Tool Tips and the Help Window, the online help tools for SCOPE designers.

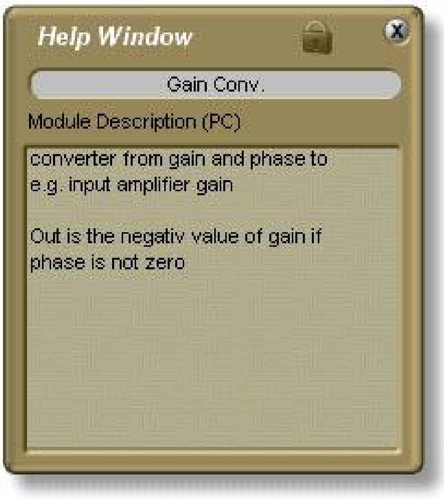


Items In The Help Menu

If **Tool Tips** is enabled, a bubble with a name or short description appears next to the mouse cursor whenever the cursor comes to rest over a pad or control. With some of these, the current value is also displayed.

If **Highlight** is enabled, all selected modules and connection will be highlighted.

The **Help Window** displays more detailed information. Simply position the mouse cursor over the module or pad you want information about. You can activate the lock function by hitting *F1*. This freezes the help text in the window so that it remains on display when you move the cursor away, allowing you to continue working and still have the text in front of you. Also, longer help texts which do not fit into the Help Window all at once can be viewed by simply dragging the frozen text upward and downward. Click on the lock button (or hit *F1* again) to unlock the Help Window.



Gain Conv.

Module Description (PC)

converter from gain and phase to
e.g. input amplifier gain

Out is the negativ value of gain if
phase is not zero

The **About** window displays the program version and the names of the programmers.

- Window And Control Panel Features

Window / Control Panel Common Features

– Each one can be freely **repositioned** onscreen via click-and-drag. With a window, you use the top side of the window frame for this. To move a control panel, click and drag on any blank part of the control panel surface. Windows and control panels become transparent while being moved so you can see what's behind them.

– Each one has a **Close** button which you can use to remove it from the screen. You can also use the Window menu for this. When a window or control panel has been closed, you must use the Window menu to open it again.

Features Which Apply Only To Windows

– Windows have **Minimize** buttons which shrink them to icons in the lower left-hand corner of the screen. Double-click on an icon to restore it to full size.

– Windows can be **resized** by dragging on side or bottom edges of the window frame or its lower right-hand corner.

– Windows *sometimes* have **scroll bars** on their right-hand or bottom edges. These appear and disappear automatically depending upon whether a window is large enough to display its entire contents at once.

- Device Surfaces

A Better Front Panel

Every SCOPE device has a "virtual" front panel via which you interact with it. In SCOPESpeak, this is called a **device surface**. Practically everything you can do with a SCOPE device is done via its surface. A SCOPE device surface contains an approachable (and of course, *very* attractive) layout of knobs, faders, switches and displays which invite you to *play*.

The powerful, flexible SCOPE graphics system permits the custom-crafting of device controls to fit the parameters they adjust. This extends to types of controls which have no counterpart in the "hard" world, but which help you visualize the thing they control – be it an envelope, an LFO, or whatever – or simply let you work more fluidly with it.

In turn, this flexibilty lets SCOPE devices offer features which otherwise would be technically possible but simply too difficult to use.

Moving Device Surfaces Around

Device surfaces can be moved wherever you want to put them. Just click and hold the left mouse button on an empty spot on the surface – it'll become transparent (if it doesn't, you need to pick a different spot – see below). Now you can drag it and drop it where you like – for example, way off to one side, leaving only a small corner visible to grab hold of later. (Note: for snappier screen response while moving device surfaces and windows around, switch off the *Display windows while moving* option in the **Scope Settings** control panel.)

Device surfaces can overlap one another just like windows on the desktop. Click on any visible part of a partially-obscured device surface to bring it all the way "up front" again. If it's completely hidden by another device surface, just move the other surface aside so you can click on the surface you want to use.

You do need to be a bit careful about where you "grab". You can't move a surface around by grabbing one of its controls – if you do, you'll just end up adjusting the control!

Grabbing a surface by one of its open drawers (see the following section) is also ineffective for moving it around.

The Works In A Drawer

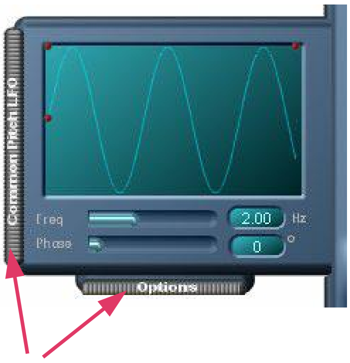
Some SCOPE device surfaces include "drawers" containing additional controls in logical groups. These stay tucked away out of sight when not needed, to keep the surface from becoming overly large or cluttered. Some drawers even contain drawers of their own! So, when you're first getting acquainted with a new device surface, it's a good idea to look for drawers and open them up to see what's inside.

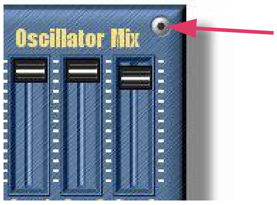
Drawer "handles" usually appear as ridged strips with labels indicating what's inside. Click on the handle to pop the drawer open or slide it shut again.

Screen Cleanup – Minimizing Device Surfaces

Every device surface has a Minimize button, generally somewhere near the upper right-hand corner of the surface. When you're using a device in your setup, but not planning to twiddle its knobs right away, you can click on this button to shrink the surface down to a tiny **icon** – a miniature version of itself. It appears along the bottom of the screen and can then be moved wherever you want.

Minimizing a device surface has no effect on the device – it stays connected and active in your setup. Only the controls are temporarily out of reach. When you want to get at them again, double-click on the icon to restore it to full size.

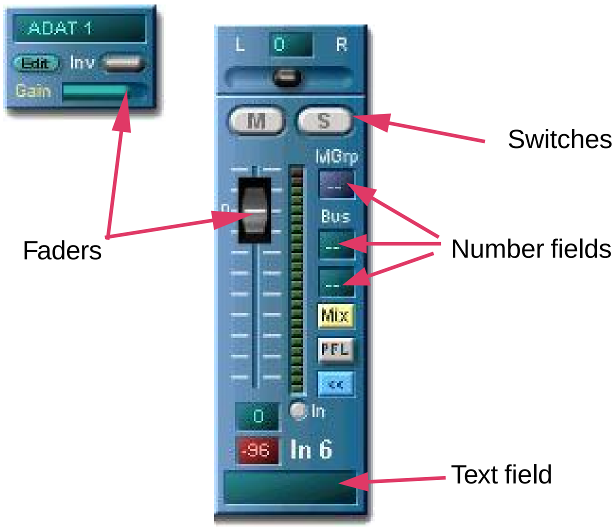




Minimized (icon) representation

- Device Controls

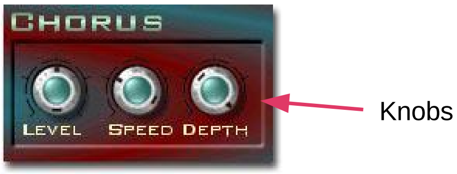
The control and display elements of SCOPE device surfaces take on a lot of different visual forms, thanks to the advanced capabilities of the SCOPE graphics system (and a bunch of over-the-top graphic designers!) But most of them fall into just a few categories which show up again and again in different devices. These are: **knobs and faders, switches, and number/text fields**.



Knobs And Faders

Knobs and faders are used for adjustment of **continuously-variable** parameters of all kinds – volume, filter and envelope settings, etc. To use them, you put the mouse cursor on the control – the center of a knob, or the handle of a fader – hold the left mouse button down, and move the mouse. When you let the mouse button up, the control is released and stays where you left it.

The various types of knobs and faders are basically alike except for the **mouse motion** they call for. You move the mouse cursor in circles around a knob, which rotates to "follow" the cursor. With faders, you use up-and-down or side-to-side motion, depending upon whether the fader is a vertical or horizontal one.



The mouse button must be held down while adjusting a control, but you *don't* have to keep the mouse cursor directly over a control after you click on it. In fact, a bit of distance often helps – the **adjustment becomes smoother and finer as the mouse cursor is moved further away from the control**. With faders this means moving off to one side of the fader (above or below it, if it's a horizontal fader).

Double-clicking on a knob or a fader handle will send it directly to its **center position**. Doing so again will restore it to the position it previously had.

Switches

Push buttons are simple – just click on them. They're either **momentary** or **push-on/push-off**.

Some **rocker** and **toggle** switches are also actually push-on/push-off – click anywhere on the switch and it'll switch to the opposite state each time. With others, you click on one end or the other to flip the switch in that direction.

Multi-position **slide switches**, such as are used for selecting oscillator or LFO waveforms, can be thought of as "stepped" faders, because you do have to "grab" the switch handle with the mouse and slide it around – just clicking on a different switch position won't make the switch go there.

Number/text fields

These are really another type of stepped fader control – but one which doesn't look like a fader at all. Device MIDI channel select controls are generally of this type. You click and hold with the mouse directly on the number or text value and move the mouse up and down to change it.

Knob/Number and Fader/Number Controls

Some knobs and faders have number fields attached. The control setting is displayed in the field and changes when you move the control.

You can also take it the other way around – you can click on the field and type in new values directly from the keyboard. The new value takes effect – and the knob or fader changes its position to match – when you do one of three things: hit the Enter key, click again on the field, or click somewhere else on the surface. With the first two methods, the field stays open and you can immediately type in another value.

Envelope And Curve Editing Controls

When dealing with complex functions such as multi-point envelopes, EQ curves and so on, a graph is often the most compact, effective and natural representation.

Some SCOPE devices include graphical "widgets" for editing such curves directly. Some of these widgets are small surfaces in their own right. Each one is tailored to the function it controls – therefore, each is described in detail in the manual section for its specific device. But knowing a few simple points which apply to most of them is enough to get you started quickly:

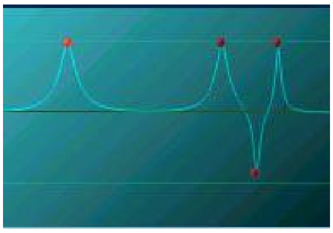
If you see colored dots on a curve, you can:

– Grab them with the mouse and move them around.
– Double-click on them to delete them.

If you *don't* see colored dots on a curve, you can:

– Double-click on the curve to create new dots.

The curves either fit themselves smoothly to the outline of the dots, or *points*, or are actually drawn point-to-point (that's right – connect the dots!)



EQ curve with dots

Projects

- Overview

This chapter picks up where the previous chapter left off. It assumes that you've been through the Introduction and have a reasonable understanding of the basic operating features of SCOPE. First, a brief overview of the SCOPE library is presented, along with a few additional program features not covered in the Introduction. This is followed by a discussion of working in the Project window. Next, the various hardware and software inputs and outputs are described. Finally, examples of the use of SCOPE with external software are given.

The rest of this overview provides some additional background relevant to working with projects in SCOPE.

Devices And Modules – The Project Perspective

In SCOPE, all *devices* and *modules* exist as files visible in the File Browser. You bring them into a project by dragging them from the File Browser into the **Project Window**.

Devices (**.dev** files) include **synths**, **mixers** and the like. They are relatively "large" and complex and have **device surfaces** with displays, knobs, switches and faders, etc.

Modules (**.mdl** files) are generally simpler items such as **audio and MIDI inputs and outputs** (hardware or software), most of which have **no controls**. These are also sometimes referred to using the computerese term "drivers", especially when they represent a piece of interface software between SCOPE and the "outside" world.

(The .mdl extension is also shared by Modular Synth module files, whose onscreen manifestations also share the use of the term "module". This is mentioned here merely to spare you some possible later confusion. These modules are strictly special accessories of the Modular Synth device which can be used *only within* that device.)

Projects And Project Files

The scope of a **project** includes the selection of hardware inputs and outputs to be used, whether a mixer is to be loaded, which synthesizers will be used and whether, for example, ASIO or Wave drivers are needed. Also included are the basic settings and screen positions of each of these elements, as well as all connections between them. The same goes for device preset lists, MIDI controller assignment lists, the audio synchronization (wordclock) configuration and various other SCOPE settings. A project is thus your complete SCOPE **setup** of the moment.

All of this information is stored in **project files** (identified with the **.pro** extension), using the Save Project (As) option of the File menu. You can also **load projects** via this menu, or by dragging them from the File Browser into the Project window (switch on the **Pro** button in the File Browser to make project files visible).

Just as important as what a project file includes is what it *doesn't* include – namely, the devices themselves. Instead, the project file includes **references** to the corresponding disk files. These references take up virtually no space in a project file. They prevent project files from becoming inconveniently large and avoid redundant storage of device files within every project.

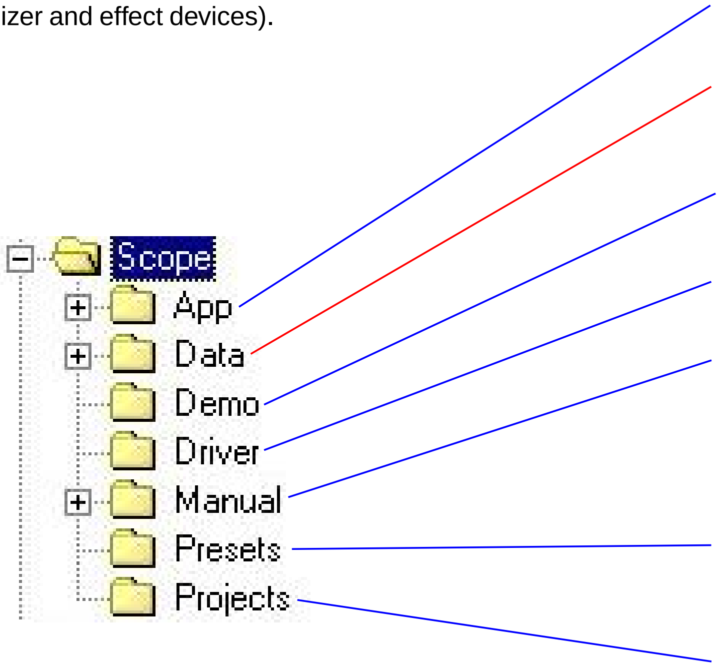
The use of referenced devices also keeps devices and projects separated from one another. This has implications for **device preset lists** and **MIDI controller assignment lists**, which are **stored within the device files** themselves. **Changes to these lists** for a particular device will **affect a device in every project** where it is used, since all of these projects refer to the same device file.

Since a device reference points to a file in a specific disk location, moving your device files to a different location can cause the references to go looking in vain. This might also occur with a project file created elsewhere than on your own system – for example, if SCOPE is installed on drive C: in one system and on drive D: in the other. When you attempt to load such a project, SCOPE will open a dialog asking you where it should look for the devices which it can't find on its own.

One **glaring exception to the referenced devices system** is worth pointing out: The Modular Synth is stored *directly* in the project file when a project is saved. The use of a reference is impossible with this synth, since (as a modular, user-modifiable device) its structure is assumed to be substantially different from one project to the next.

- SCOPE Library

The illustration below provides an overview of the complete SCOPE directory structure. On the next page, the structure of the Data subdirectory is described. This subdirectory contains all components, modules and devices (including synthesizer and effect devices).



Application files

Components, Modules and devices

Demo projects

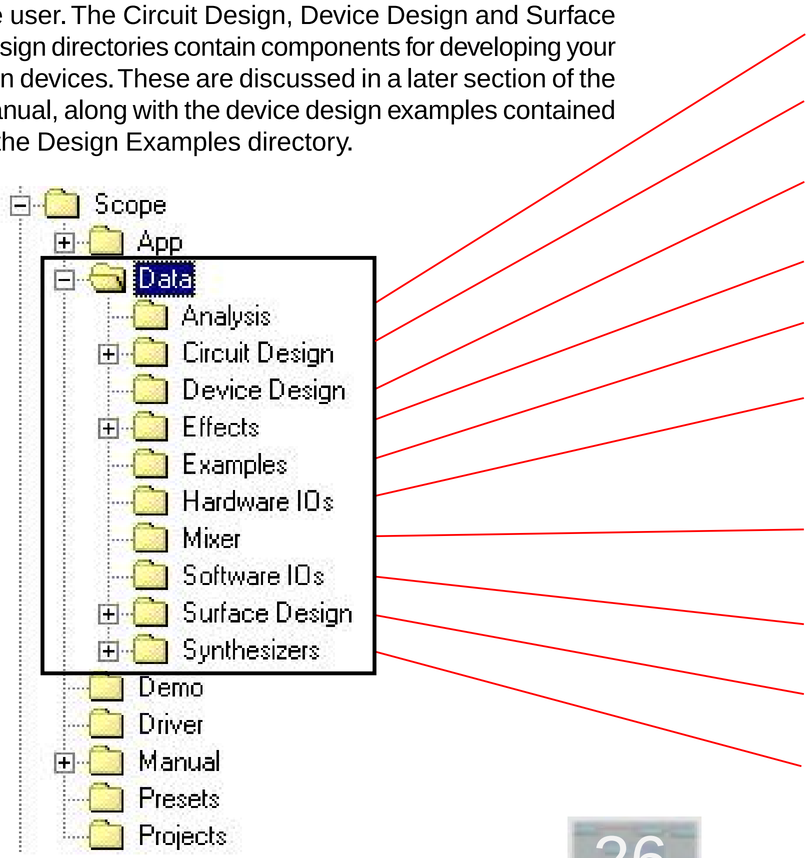
Copies of the current drivers

User manuals in Acrobat Reader (PDF) format (German and English versions)

Default directory for presets

Default directory for projects

Below is an overview of all directories containing modules and devices which are important from the standpoint of the user. The Circuit Design, Device Design and Surface Design directories contain components for developing your own devices. These are discussed in a later section of the manual, along with the device design examples contained in the Design Examples directory.



Analysis tools

Modules and graphical components for custom device development

Standard modules for custom device development.

Audio effects

Example devices and projects

Module representations of SCOPE's physical inputs and outputs

Mixers

Driver modules for communication with other applications

Modules for custom surface development

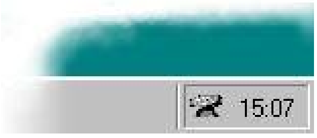
Synthesizers and samplers

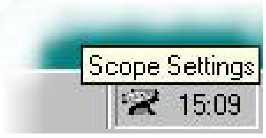
- The SCOPE Icon On The Windows Task Bar

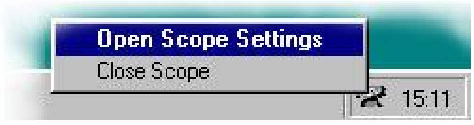
You'll probably have noticed this pretty quickly once you've gotten up and running with SCOPE, and may be wondering just what it means. In fact, it's your visible indication that a SCOPE **background project** is loaded and active in your system.

The purpose of this background project is to **set up the SCOPE card to serve as a default sound and MIDI device** in your system. Standard Windows sound playback (from multimedia programs, etc.) can thus occur via the SCOPE card as if it were a standard sound card, even when the SCOPE program itself is not running. Likewise, the SCOPE card's MIDI input and output become visible and accessible to Windows programs which use MIDI. Other default system connections are made as well. A complete picture of the devices and connections in the background project is presented in the Installation Guide.

Right-clicking on the icon opens a small context menu as shown at right. Via this menu, you can get direct access to the SCOPE **Sample Rate Settings** control panel (described in the *Introduction* section of this chapter) or **shut down the background project** (often advisable when installing a SCOPE update or other software).







When you start SCOPE , the background program is deactivated. The main program takes over full control of the SCOPE card and the icon vanishes. **When you exit the main program, the background program is automatically restarted** to make the SCOPE card's resources once again available to Windows as before.

You can replace the automatically installed background project with one of your own creation. Refer to the following section of this chapter (SCOPE Settings) to find out how this is done.

- SCOPE Settings

This control panel is opened via the Window Menu. It provides various options for configuring the SCOPE environment.

Appearance options

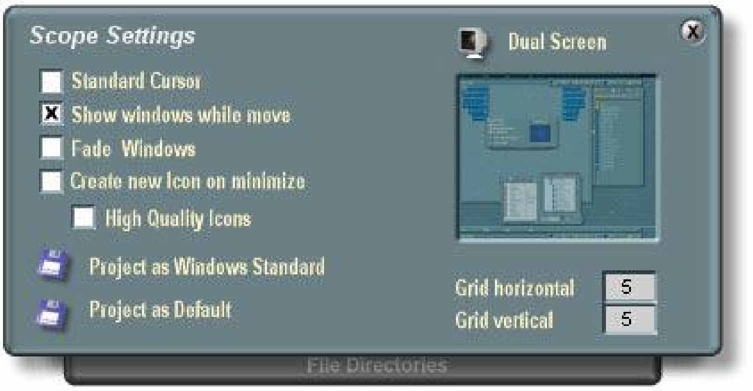
The ***Standard Cursor*** option is off by default. If switched on, SCOPE will use standard Windows cursors instead of its own animated ones. This can speed up screen response somewhat on slower computers.

The ***Show windows while move*** option is on by default. It causes SCOPE windows, tools, etc. to retain their normal appearance (apart from becoming transparent) while being dragged around onscreen. If this option is switched off, these objects change to simple outlines while being moved. This can noticeably speed up screen response while dragging objects around, even on faster computers, especially with larger objects such as the Project Window.

When **Fade Windows** is enabled, all windows, tools and surfaces are faded in or out gradually.

With **Create new Icons on minimize** enabled, the icon of a minimized device reflects the current state of the device surface, including positions of knobs, faders, and drawers.

High Quality Icons improves the resolution of the icon.



☐ Standard Cursor
☒ Show windows while move
☐ Fade Windows
☐ Create new Icon on minimize
☐ High Quality Icons

Grid horizontal 5
Grid vertical 5

The ***Screen*** box permits adjustment of the size and position of the SCOPE screen on your computer display. The box outline represents your entire computer screen. Activate the ***Dual Screen*** option if you're running dual monitors under Windows98. This divides the Screen box into two sections representing both of your screens, as shown at right.

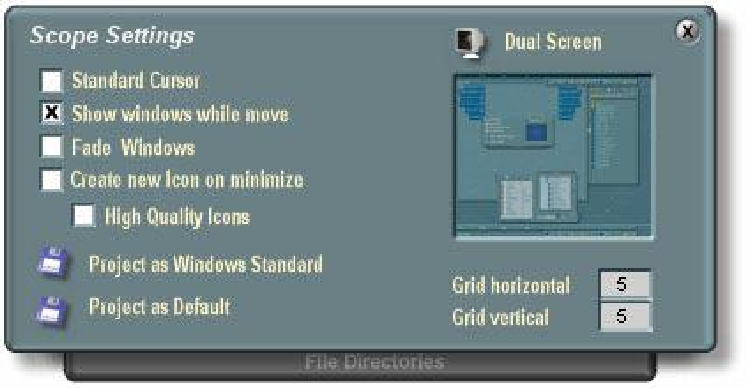
The rectangle *inside* the Screen box represents the SCOPE screen itself. It is full-screen by default but can be made smaller and then repositioned. Click and drag near any edge of this rectangle to move that edge. Once the blue rectangle is smaller than the Screen box, you can click and drag in its middle to move it around within the box. Note that the new size and position take effect only *after* you close the SCOPE Settings control panel.

Default project options

Clicking the ***Project as Default*** button sets the current project up as a default project to be loaded whenever SCOPE is started (or when you select *New Project* in the File menu). This is handy when you have a partial setup – e.g., a specific set of devices, modules and connections – which you often use as a starting point for new projects.

Clicking the ***Save Project as Windows Standard*** button causes the current project to be saved as the background SCOPE project. (Note: your new project overwrites the existing background project file Winstart.pro when you do this.)

The basic idea behind the background project is to automatically load a set of SCOPE device drivers and connections between them whenever Windows is started, or whenever you exit SCOPE. This permits use of the hardware resources of the SCOPE board in a logical way by other Windows programs, even when the SCOPE application itself is not running. For example, the SCOPE board MIDI and audio inputs and outputs can be used immediately by sequencer programs.



☐ Standard Cursor
☒ Show windows while move
☐ Fade Windows
☐ Create new Icon on minimize
☐ High Quality Icons

Grid horizontal 5
Grid vertical 5

However, you can save any project you like into Winstart.pro. Thus, you can have a complete SCOPE project with synths, mixer and effects – or maybe a sample player with a set of sounds to emulate a standard GM sound card – load automatically each time you start Windows (you'll be able to use it right away, but not interact with it or modify it). But take note: a large Winstart.pro project will also prolong your Windows startup.

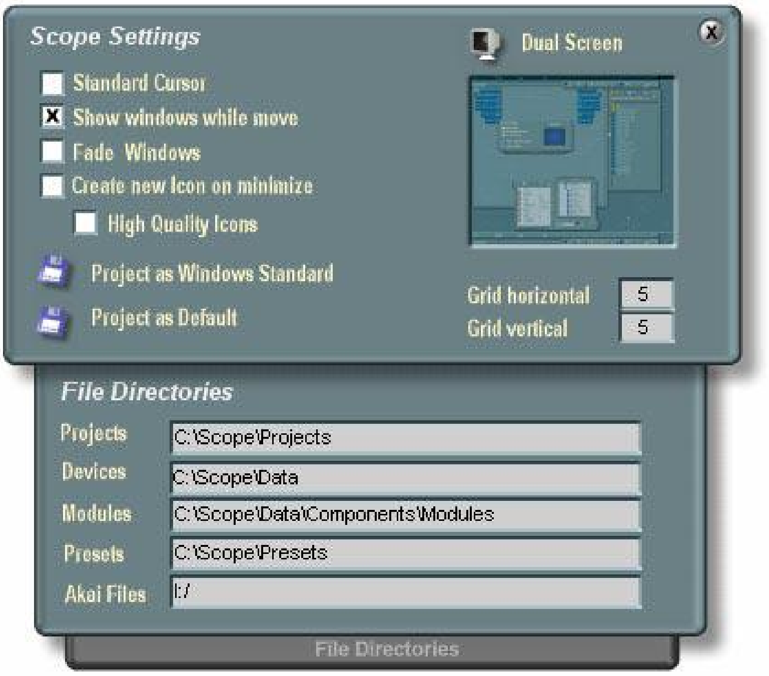
Grid options

Here you you can specify the horizontal and vertical width (in pixels) of the (invisible) grid in the Project Window. When you move a module or any of its parts using the mouse, it will snap to points in this grid.

Directory options

SCOPE maintains a list of directories in which it looks first for files of various types. A default list is created when SCOPE is installed.

The directory list can be modified in the drwaer of this control panel. You might want to do this, for example, to set up your own folders for storing your projects, rather than piling them all into the default SCOPE Projects folder. By changing the *Projects* entry to point to your new folder, you save some time and mouse-clicking in the File Menu when you want to load or save a project, because SCOPE will then automatically open this folder when you select *Load Project* or *Save Project* from the File Menu.



Scope Settings

☐ Standard Cursor
☒ Show windows while move
☐ Fade Windows
☐ Create new Icon on minimize
☐ High Quality Icons

Project as Windows Standard
Project as Default

Dual Screen

Grid horizontal 5
Grid vertical 5

File Directories

Projects	C:\Scope\Projects
Devices	C:\Scope\Data
Modules	C:\Scope\Data\Components\Modules
Presets	C:\Scope\Presets
Akai Files	I:/

- Presets / Number Of Voices

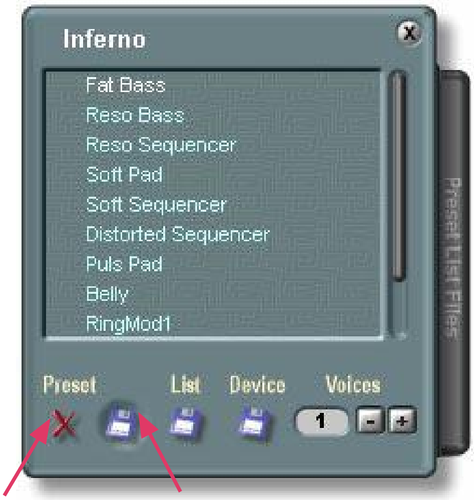
Presets can be easily created for any SCOPE device. These are complete device setting snapshots which can be recalled at any time. Storage, recall and management of device presets is done via the **Preset List** control panel, where you can also set the **number of voices** for a synth.

To access the preset list for a device, find the *Presets* button on its surface (it looks similar to the buttons on the control panel shown at right) and click on it. The **Preset List** control panel appears, displaying an alphabetically sorted list of the existing presets for the device. The list is automatically resorted whenever you make any changes.

To **recall (load) a preset**, double-click on its name.

To **create a new preset** from the current synth settings, click on **Save** (under Preset), answer "No" when asked if you want to overwrite the selected preset, type in a name (directly in the list) for the new "Untitled" preset and hit <Enter>. Alternatively, answer "Yes" to **update the selected preset** with the current synth settings.

To **rename a preset**, select it and hit F2, or just click and hold on it until the name-edit symbol ¶ appears next to it. Now type in the new name and hit <Enter>. Or, if you change your mind, hit <Esc> to leave the name unchanged.



Inferno

Fat Bass
Reso Bass
Reso Sequencer
Soft Pad
Soft Sequencer
Distorted Sequencer
Puls Pad
Belly
RingMod1

Delete Preset Save Preset

To **delete a preset**, click on the preset name in the list to select it, then click on the **Delete** button.

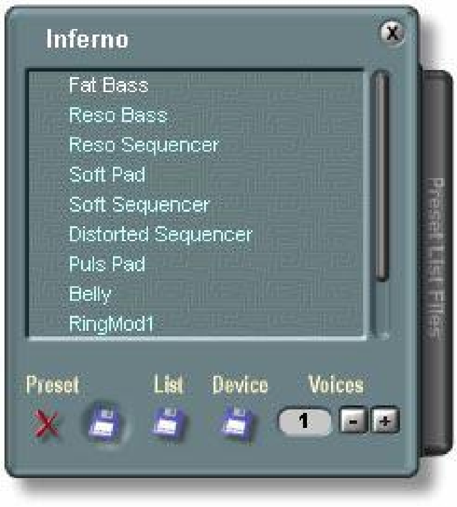
NOTE: Device presets created in this window are *not yet* permanent! The same goes for presets deleted or renamed here. The ▨ and ▨ symbols which appear next to the names of new and deleted presets serve to indicate this. In the following pages you'll find more information about device presets and how to work with them, including the use of device preset files and the two different ways of permanently saving device presets.

Number Of Voices

Via the Preset List control panel you can also set the **number of voices** a synth has. Click on the + and - buttons at the bottom of the panel or click and drag up/ down on the number display itself (it's a text fader).

Changes to this setting take effect immediately. The **upper limit is 16** . Remember – there's no free lunch here! **Each added voice uses extra DSP capacity**, so don't set the voice count for a synth higher than you actually need it to be in the current project.

Why is this setting located in the Preset List control panel? Because, like the device preset list, the **number of voices is a modifiable *property* of a device** which is stored within the device file itself. As with new, renamed and deleted presets, a new number-of-voice value dialed in here is not immediately permanent. However, the permanent stored setting can be updated via the **Save (Device)** button. Read further for more information about the Save (Device) function of the Preset List control panel.



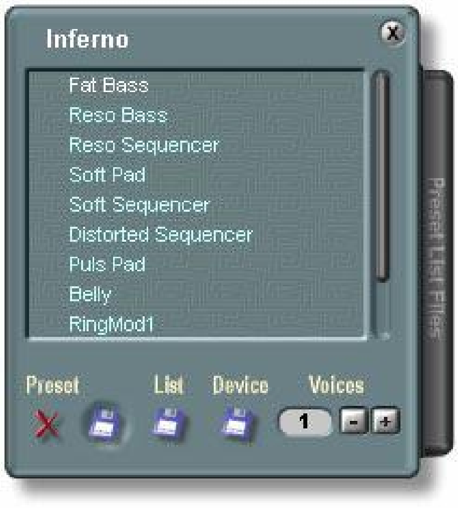
Saving The Preset List With The Device

Presets in the preset list are part of the device they belong to. However, the device itself is merely a temporary memory copy of a disk file. **Changes to the preset list – new ▨ or deleted ▨ presets – are thus likewise only temporary until they've been saved**.

Clicking on the **Device** button is the easiest way to do this. This causes the current preset list (and the current voice count setting) to be **saved *in the original device disk file.*** These thus become "permanently" part of the device (at least until the next Device Save) and come along whenever the device is loaded into a project.

Important to note here is that **the Save function affects the device in *every* project** where the device is used. You may not always want this. For example, you may want high polyphony for a synth in one project (to meet the needs of the music) and a low voice count for the same synth in another (to "shoehorn" it into a large project).

An easy way out of this conflict is to **use multiple copies of the device file**, with different settings saved in each, instead of using the same file in every project. Device files can be copied in Windows Explorer or directly in the SCOPE File Browser and given descriptive names such as "EZ Synth 16 voices" so you can tell them apart.



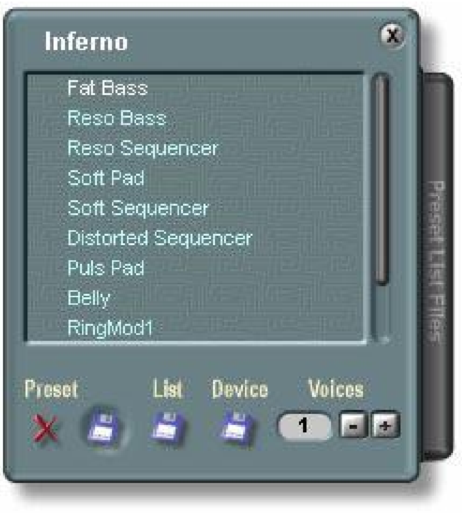
Continue reading to learn about the **other method of saving device presets** – namely, exporting the preset list as a file – and about using preset list files.

Exporting The Preset List As A File

The preset list for a device can also be stored separately from the device file by **exporting** it as a **preset list file**.

Click on the **List** button to open a standard Windows file save dialog. The preset list will be exported into a file under the name you give, with a .pre extension added automatically. SCOPE won't let you use preset files with devices which don't match them, but it's still a good idea to include some indication of the device name in the preset file name, to help you remember later what the file is for.

Preset list files let you **work more flexibly with presets** by separating them from the device (and from each other). You can organize them to suit your needs – for example, by breaking down a large preset library into categories – and avoid having to always work with your entire collection.

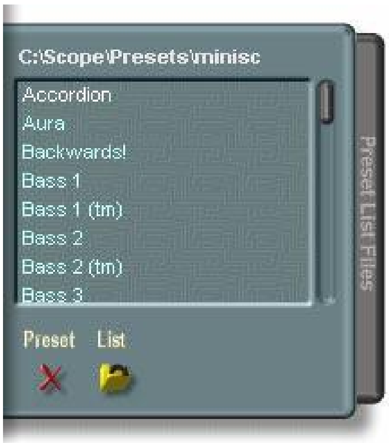


Exchange of presets with other users is also easy using preset files – as is downloading them via Internet – so your collections may very well grow quite large. If an updated version of a device becomes available, you can use preset files to transfer your presets to the new device. And if you use the trick (described on the previous page) of working with multiple instances of a device, preset files let you use your presets with all copies of the device, regardless of which copy the presets were created with.

Read further to learn about **working with preset list files**.

Working With Preset List Files

Preset list files are accessed via the **Preset List Files drawer** of the Preset List control panel. If a file was in use when the *device* was last saved, you'll see this preset list as soon as you open the drawer. Otherwise, you can open a file by clicking on **List**. In the file open dialog which appears, you can select an existing preset list file or type in a name to create a new file.



You can now **import individual presets from the file** by simply dragging them from the Preset List File window into the main Preset List window. You can **load a preset directly from the file** without importing it by double-clicking on the preset name in the Preset List File window.

Making changes to the preset list file is just as simple. Drag a preset from the main Preset List window into the Preset List File window to add it to the file. Click on a preset name in the Preset List File window, and then on the Delete button in this drawer, to delete the preset from the file.

Note that there is **no Save button** in the Preset List File drawer. When you add presets to a file or delete them from a file, the **changes are saved immediately to disk**.

Presets can be deleted with the button under Preset.

- MIDI Controller Assignment

Any knob, switch or fader on a SCOPE device surface can have a **MIDI Control Change** message assigned to it. This allows it to be adjusted not only onscreen via the mouse, but also under external (to SCOPE) MIDI control. In response to these messages, onscreen controls move just as if you were using the mouse to adjust them. But the parameter control response stays active even when the device surface (or SCOPE) is minimized.

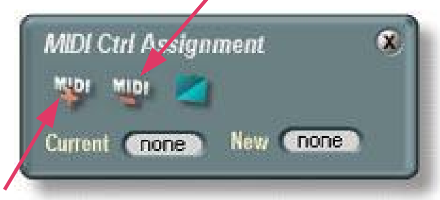
MIDI Control Change Message Sources

The **Mod Wheel** on a MIDI keyboard is the most obvious example of a MIDI Control Change message source. **MIDI fader boxes** are another. These are perfect, for example, as a quicker and more natural alternative to the mouse for controlling multiple faders on a SCOPE mixer. However, the controller message source can also be a **sequencer program** on the same computer, with its MIDI output routed into SCOPE – or an external sequencer.

It Works Both Ways

SCOPE device controls with MIDI controller message assignments don't only respond to *incoming* messages. **Corresponding messages are also sent out** (via the SCOPE MIDI Output) when you adjust controls with the mouse. These messages can likewise be sent to a sequencer for recording and subsequent playback (as a form of **MIDI-based control automation**) or used directly for live control of external MIDI devices.

Delete Controller Assignment



Add Controller Assignment

Making MIDI Controller Assignments

Right-click directly on any knob, fader or switch. The **MIDI Controller Assignment** panel pictured above appears. If the control already has a MIDI controller assignment, its message number is displayed in the **Current** field.

Now simply *move* the external MIDI controller (mod wheel, fader, data slider etc.) until its message number appears in the **New** field. Then click on MIDI. Finished!

Click on MIDI to clear an existing MIDI controller assignment.

The **Control Ranger** button ▨ opens the Controller Settings panel, which is covered in the following section.

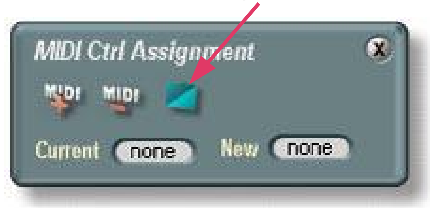
If nothing appears in the New field when you move the external controller:

– Make sure that the SCOPE device (synth, mixer, etc.) is receiving on the MIDI channel that the controller is sending on. (Note that if the control messages are coming into SCOPE by way of a sequencer, the sequencer could also be "bumping" them to a different MIDI channel.)

– Make sure that the MIDI input of the SCOPE device is cabled to the MIDI message source within SCOPE (SCOPE MIDI Input or Sequencer MIDI Out).

– Make sure that transmission of controller messages from the external controller hasn't been disabled.

Open Control Ranger

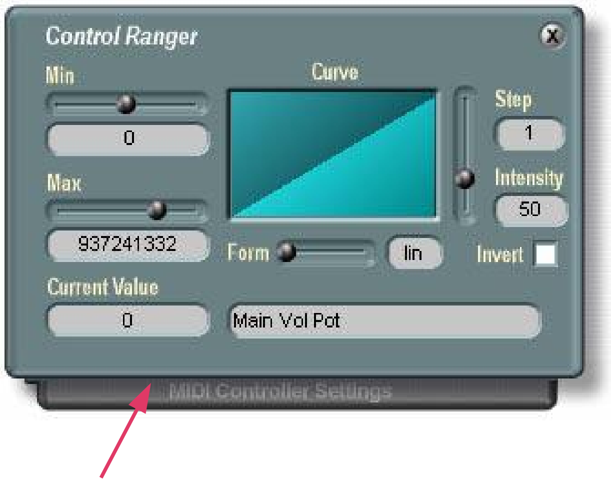


The Control Ranger panel can be opened directly from the Window menu, or by clicking on the *Ctrl List* button in the MIDI Controller Assignment control panel (see preceding section).

Via the Control Ranger, you can adjust the endpoint (minimum and maximum) values of the selected control object. In addition, you can select and fine-tune its characteristic curve (linear, logarithmic, exponential).

The use of the Control Ranger is described in detail in the Circuit Design chapter.

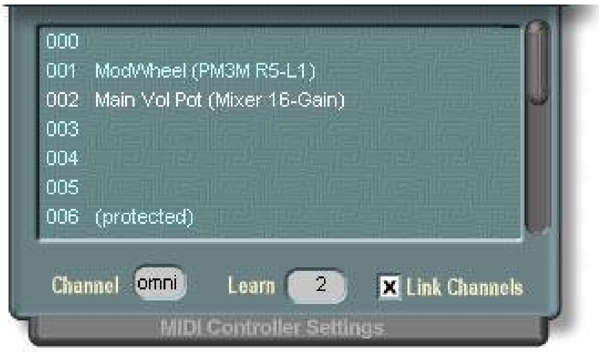
Accessible at the bottom of the Control Ranger is the MIDI Controller Settings drawer.



Min: 0
Max: 937241332
Step: 1
Intensity: 50
Form: lin
Invert: □
Current Value: 0
Main Vol Pot

Opens the MIDI Controller Settings

- MIDI Controller Settings


000
001 ModWheel (PM3M R5-L1)
002 Main Vol Pot (Mixer 16-Gain)
003
004
005
006 (protected)

Channel: omni
Learn: 2
☒ Link Channels

The MIDI Controller Settings panel can be opened by clicking on the drawer handle at the bottom edge of the Control Ranger (see preceding section).

Like the MIDI Controller Assignment panel, this one lets you manage MIDI Control Change message assignments to device controls – but with more of an overview.

In the window, a list of all available MIDI controller numbers is shown. Any device control which already has a MIDI Control Change assignment is listed by name next to the MIDI controller number which is assigned to it.

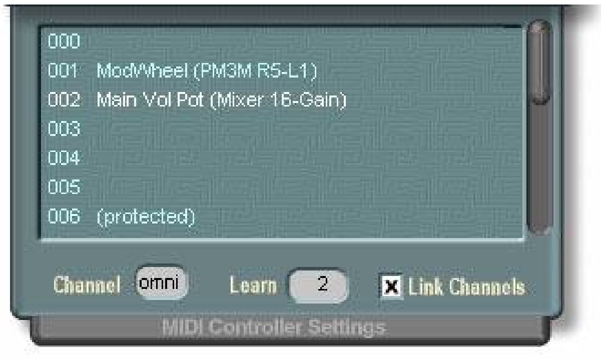
To **assign a MIDI control change number** to a device control (knob, fader, switch, etc.) via this control panel, first **click on the control itself** to select it. Then:

– Scroll to the desired controller number in the Controller Settings window and double-click on it. The name of the device control appears next to the number, indicating that the assignment has been made.

– **Or**, if the control change message is coming from an external MIDI controller (mod wheel, fader box, etc.), simply move the controller. The number of the message it sends will appear in the **Learn** box and highlighted in the list. Click on the highlighted number to confirm the assignment. If nothing happens when you move the controller, refer to the preceding *MIDI Controller Assignment* section under the subheading **Making MIDI Controller Assignments** ("If nothing appears ..") for possible causes.

To **delete an existing assignment**, double-click on it.

The **Link channels** option is on by default and causes the controller assignments for a device to be active on whatever MIDI channel the device itself is currently set to. (If this option is turned on, adjusting the **Channel** setting in this control panel will thus also change the device MIDI channel setting and vice versa.) If you turn this option off, the two settings are uncoupled. You can then **select a separate MIDI channel for the MIDI controller assignments**. For all other MIDI messages, the device will continue to use its own channel setting.



000
001 ModWheel (PM3M R5-L1)
002 Main Vol Pot (Mixer 16-Gain)
003
004
005
006 (protected)

Channel omni Learn 2 ☒ Link Channels

MIDI Controller Settings

Storage Of the MIDI Controller Assignment List

The list of MIDI controller assignments and associated settings for each device in a project is **saved with the project** when the project is saved. Additionally, the current assignments are **saved in the device file itself** when the device is saved via the Preset List control panel (refer to the appropriate section earlier in this chapter).

- DSP Load

The ***DSP Load*** control panel is like a tachometer for your SCOPE board. As with a tachometer, there's nothing you can actually *do* in this control panel, but the information it gives you can be useful.

The bar graph display shows you approximately what proportion of the DSP capacity of your SCOPE card is currently being used, and thus also how much unused capacity is left. This depends upon the current project and configuration. Leftover capacity means you can expand the project – you can add synth or mixer devices or effects, or increase the number of voices of an already-loaded synth, or make use of additional mixer channels or mixer channel EQs which are currently turned off.

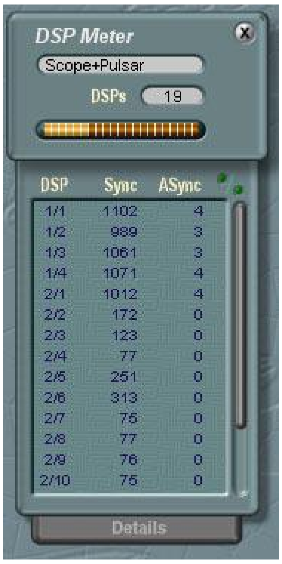
Unlike a tachometer, the DSP Load display *can* be safely ignored. There's no possibility of damaging the SCOPE board. In fact, SCOPE uses this info to prevent DSP overload and ensure that all loaded devices work properly. Running "in the red" is also okay. SCOPE devices are unaffected by DSP loading conditions, as long as there's enough DSP capacity for them.

The additional displays on this control panel identify your installed DSP card by model name and indicate the number of DSPs on the board (fifteen for SCOPE).

The slide-out drawer at right shows the capacity usage for each DSP individually, either as DSP cycle counts – synchronous and asynchronous cycle counts are tallied separately – or as percentages of the available maximum in synchronous and asynchronous categories (click on the % sign to switch display mode).

By watching how the displays change as you add and delete various devices or enable and disable various device features, you can get an idea of the associated "power consumption" of each. This can help you "trim the fat" from your projects and get the most out of your card.

Effective DSP capacity is directly related to available computing time, namely the duration of one sample clock. It follows that extra DSP capacity becomes instantly available when the sample rate is lowered – roughly 50 percent more, in fact, if it is taken down from 48 KHz to 32 KHz. However, this also shrinks the audio bandwidth. This may be noticeable with audio and samplers. Also, some audio sequencers and external devices can't work with a lower sample rate. Refer to the *Sample Rate Settings* section for more details.



DSP Meter

Scope+Pulsar

DSPs 19

DSP	Sync	ASync
1/1	1102	4
1/2	989	3
1/3	1061	3
1/4	1071	4
2/1	1012	4
2/2	172	0
2/3	123	0
2/4	77	0
2/5	251	0
2/6	313	0
2/7	75	0
2/8	77	0
2/9	76	0
2/10	75	0

Details

- Working In The Project Window

Whereas device surfaces are the front panels of SCOPE devices, the **Project window** can be thought of as the collective rear panel for an entire SCOPE project. More importantly, it's the place in SCOPE where you can see all and do all.

The Project window is opened via the Window menu, if it isn't already open. It can be maximized to fill the screen (unique to the Project window), minimized to an icon, or if not needed at the moment, closed completely.

This window gives you a detailed **schematic** or **"circuit diagram" view** of *all* devices and connections in a project. You can directly see and manipulate these connections, add devices to a project or remove them from a project.

In the Project window, both devices and modules (as defined in the Overview section of this chapter) are shown as simple "blue boxes". Each blue box is only as large as necessary to fit the device or module name in its middle and the requisite number of **pads** on its edges to represent the inputs and outputs of the specific device or module.

For example, the Project window view of a synth may include only one MIDI input pad (designated **MIn**) and one audio output (**Out**) pad. (A few other common pad designations: **In** for audio in, **MOut** for MIDI Out, **LIn**/**RIn** for left/right audio in, **LOut**/**ROut** for left/right audio out). In the Project window you add or remove point-to-point **cables** between these pads to create or modify a setup.

And now, a bit of bad news (well, not *that* bad) – namely, yet another definition for that overused word "module". Specifically: these "blue boxes" are referred to in the shorthand jargon of the Project window as – yes, you guessed it. (There – that wasn't *so* bad, was it?) Hopefully, the context should make this clear. If not – it doesn't really matter, there isn't much that can go wrong here.

Modules In The Project Window

When a project is loaded, the devices and modules it contains appear in the Project window as modules (yes, here it comes). Naturally, the connections between these modules, as defined in the project, appear here as well.

Devices and modules are added to a project via drag-and-drop from the File Browser into this window, as described in the *File Browser* section of the *Quick Tour* chapter. Make sure the **Dev** and **Mdl** buttons in the File Browser are pushed in, so that these file types will be visible in the File Browser.

(Incidentally, you can also **drag-and-drop an entire project** into the Project window, as an alternative method of loading a project.)

Modules in the Project window can be **moved around as desired** via simple click-and-drag, to clear up the project view or just for fun. All existing connections to a module hang on tight and go along with it when it is moved around.

Each module in the Project window has a **module menu** which is accessed by right-clicking on the module. The options in this menu are then selected via normal left-clicking. This menu will always contain the option **Save as new Device** or **Save as new Module** and **Delete module** or **Remove device** (thus providing us, finally, with a form of rescue from the confusion of "module" terminology – if you really want to know, once and for all, whether the module you're looking at in the Project window represents a device or a module, the module's module menu will reveal this to you. *Whew!*)

Where applicable, this menu also includes options for opening and closing control surfaces and preset lists – along with various other options useful for device development

A couple of other things you can do with modules in the Project window:

– You can **delete a module** by left-clicking on it once and then hitting the Delete key on your keyboard.

– You can select multiple modules by holding down the <Shift> key while left-clicking on them, or by using the mouse (via click-and-drag on the Project Window background) to draw a rectangle that encompasses all of the modules you wish to select.

When you delete a module, all connections to that module are also deleted. You've probably already noticed this. You may have also noticed that there is (unfortunately) **no** *Undo* function in the Project window. So – be careful, don't work too quickly, and save your project often.

SCOPE Cables

Devices are interconnected using *virtual cables*. In the Project window these are shown graphically as cables.

Virtual cables are nicer than the real kind in a number of ways. They weigh nothing and take up no space in your studio. You can't trip over them and your friend can't borrow them and "forget" to bring them back and your cat can't chew them up. They don't "go bad" and they won't pick up transmissions from the idiot next door with the 1000-watt CB rig. You always have "just one more" when you need one, it doesn't cost anything extra, it's just the right type, and – it's long enough.

Cables are used to connect device pads to one another. To **make a connection**, you click on a source pad (such as Audio Out), and then on a destination pad (e.g., Audio In). A new cable appears onscreen to represent the new connection, which instantly becomes active in the setup. You can **remove a connection** in the same way – click on two pads which have a cable connecting them, and the connection vanishes.

Cabling Tips

Basically, working with SCOPE cables is just as simple as described above. But, there are a few useful tips worth knowing. These are presented here, in no particular order:

– Rather than clicking on pads directly, you can **click on their text labels**. It works just the same and it provides a larger target which can be easier to hit, especially when there are already cables connected to a pad.

– When connecting or disconnecting two pads, you can **click on either one first** – the order doesn't matter.

– An easier way to get rid of a connection is to **click on the cable itself**. It becomes highlighted (so you can make sure you've hit the right one). Hit the Delete key on your keyboard to make it go away.

– SCOPE **allows only connections which "make sense"**. It won't let you connect outputs to outputs, nor (by consequence) two outputs to the same input, and it won't let you connect audio pads to MIDI pads.

– SCOPE ***will* let you connect inputs to one another**. This merely means that any signal source which is connected to one of these inputs will feed the other connected inputs as well.

– Cables appear in different **colors** to let you know what type of signal they carry (red for MIDI, blue for audio).

– When making multiple „parallel“ connections between two modules, you can use a keyboard shortcut to speed up the job. Make the first connection – e.g., between Out1 on one module and In1 on the other – „by hand“, using the mouse. Then hit <N> as many times as necessary to create the additional corresponding connections.

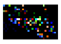
– For help in „cleaning up“ the appearance of your cabling, you can hold down the <Shift> key while moving a module. This defeats the normal „snap to grid“ behavior, allowing very fine movements. When the pads on both modules are precisely aligned with one another, the connections between them become perfectly straight.

– **The mouse cursor gives you clues** as to where you stand while cabling (here it is assumed that you're using the SCOPE graphical cursors and haven't switched them off, via the *SCOPE Settings* control panel, in favor of the standard Windows cursors):

Normally, the cursor looks like this:

When you move the cursor over a pad (or over its text label) it changes to a black plug to indicate that you can initiate a cabling action via a left-click:

When you actually click, the cursor briefly does a little twist. Now, when you move it away, it changes to a red plug:



This is your signal that the cursor is "armed" and ready to complete a connection.

As you continue to move the cursor around, you'll see the black plug again whenever you position it over a pad which can be connected to the first pad. Click when you see this and you'll make the connection.

On the other hand, if you position the cursor over a pad which is *not* an acceptable candidate for connection to the first pad (for any of the reasons listed on the previous page), you'll get this cursor:

You may have also spotted this shape when you clicked on the first pad, before you moved the cursor away from it. This is just SCOPE's way of pointing out that you can't connect a pad to itself.

If the cursor switches to this shape:

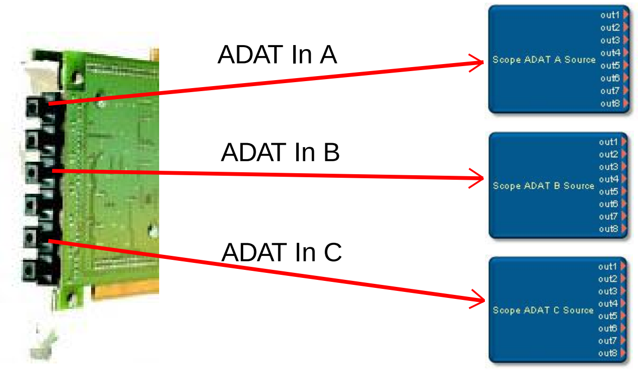
it means that the pad you're aimed at already *is* connected to the first pad. Clicking will disconnect it.

- Hardware Input/Output

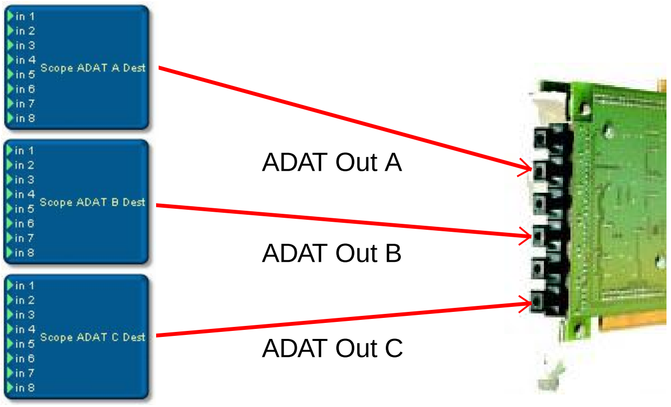
Hardware inputs and outputs include all of the actual physical inputs and outputs (I/Os) of your SCOPE system. The SCOPE 24 I/O board features three ADAT ports. You can additionally use a Pulsar board, which has one stereo analog I/O, one S/P-DIF stereo I/O and two ADAT ports (for a total of 20 audio I/O signals). In addition to the audio I/Os, the SCOPE I/O board features two MIDI ports, versus one on the Pulsar board.

All I/Os are „virtualized“ as modules which you can drag from the File Browser into the Project window. These are located in the Scope\Data\Hardware IOs\ directory. Only those modules which match your hardware setup can be loaded. By making onscreen connections to these modules, you're actually routing signals to and from these inputs and outputs.

These modules are designated as **Source** and **Dest** (destination). To keep this straight, remember that they're named *from the point of view of the Project window.* Thus, a Source module brings signals *into* SCOPE from outside (and so has *output* pads for hookup in the Project window), while a Dest module carries signals *out* of SCOPE (and has *input* pads to which SCOPE signals can be routed).

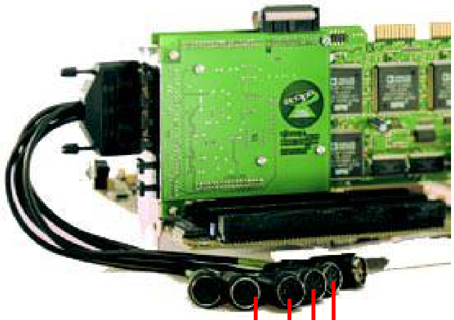


SCOPE ADAT A-C Source

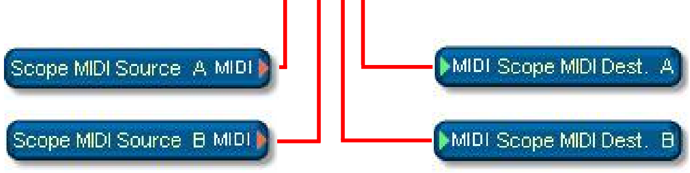


SCOPE ADAT A-C Dest

There are three sets of ADAT I/Os on the 24 I/O Board, each with 8 input channels and 8 output channels. These ADAT I/Os are represented in SCOPE by **Scope ADAT 1-3 Source** and **Scope ADAT 1-3 Dest.** An ADAT connected via ADAT I/O 1 appears on channels 1-8 of both modules. The same applies to channels 1-8 of both ADAT I/O 2 modules for an ADAT connected via ADAT I/O 2. However, there's no requirement that all of these signals be used. If you're using just one ADAT with SCOPE, you can load just the appropriate modules.



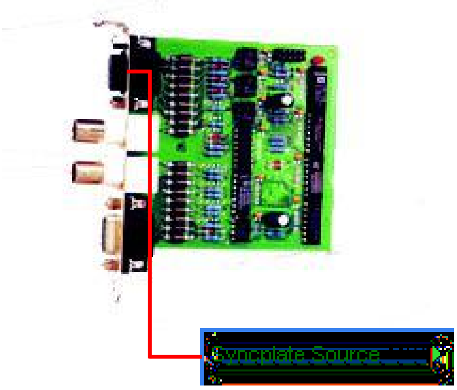
SCOPE card with 24 I/O Board and Cable Assembly



The MIDI signals, which are accessible via the **Cable Assembly,** are likewise represented in the software by modules.The modules for the **MIDI Input** and the **MIDI Output** are separate from one another. The **MIDI Thru** connector is not virtualized as a SCOPE module, since it receives its signals via a direct hardware connection to the MIDI Input.

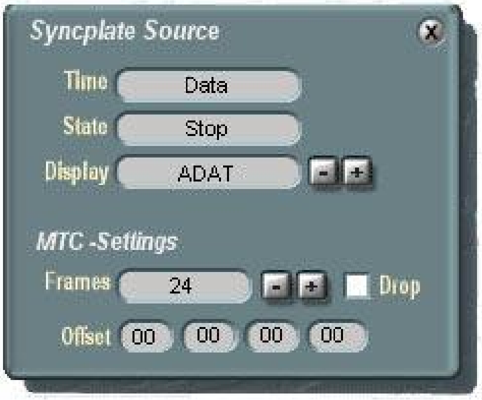
If you are also using a Pulsar card, refer to the Pulsar manual for a description of the software modules that represent the physical I/Os of this card.

Sample-accurate synchronization to an ADAT device is achieved with the Sync Backplate which provides the required hardware connections.



ADAT sync timecode is implemented by including the Syncplate Source module in your project. When this is present, ADAT sync is converted to MIDI Time Code (MTC) which can be directed to a Sequencer MIDI Destination module. You must then select the appropriate MIDI input in your sequencer to recognize the MTC.

Options in the Syncplate Source dialog



Syncplate Source

Time Data

State Stop

Display ADAT - +

MTC -Settings

Frames 24 - + □ Drop

Offset 00 00 00 00

With the **Display** toggle switch you can select whether the **Time** field displays MTC format or ADAT format.

The **State** field displays the current state of the ADAT device.

You can specify an **Offset** value (format hour:min:sec:frames) between source time and generated MTC time.

Under **Frames** the frame rate can be selected from 24, 25, 29.97 or 30 fps. The format 29.97 or 30 fps can be switched into drop-frame mode with the **Drop** switch.

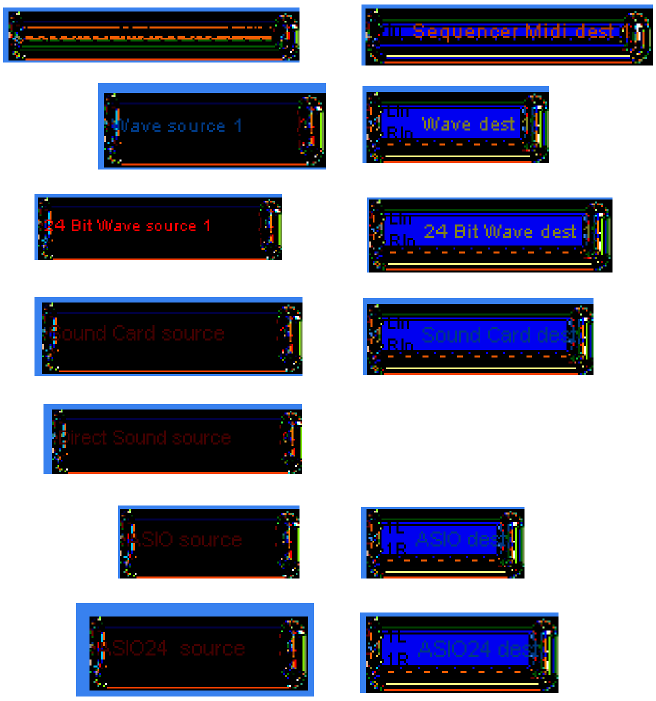
- Software Input/Output

Software inputs and outputs include **all audio and MIDI connections to other Windows programs running on the same computer** – MIDI and audio sequencers, wave/media players of various types, sound editors, etc. – as well as to non-SCOPE audio and MIDI hardware.

These connections permit MIDI and/or audio data streams to flow within the computer as if conducted via cables. This is what allows you, for example, to use a MIDI sequencing program to sequence your SCOPE synths, or to mix down audio tracks from a hard disk recording program using the SCOPE mixer, with MIDI mix control – or both at once – on *one* computer, without cables.

As with the SCOPE hardware I/Os discussed previously, the software I/Os show up in SCOPE as **modules** which can be dragged into the Project window (from ..\Scope\Data\Software IOs) and cabled. Each one is designated as **Source** or **Dest** (destination). **To keep this straight, remember that they're named *from the point of view of the Project window*.** Thus, a Source module brings data *into* SCOPE from outside (and so has *output* pads for hookup in the Project window), while a Dest module carries data *out* of SCOPE (and has *input* pads to which SCOPE signals can be routed).

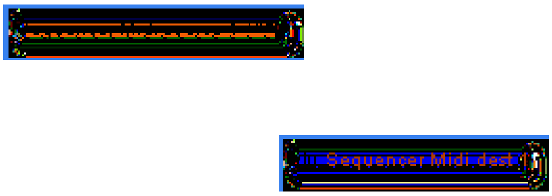
Outside of the Project window (or SCOPE) context, the software connections are referred to as *drivers*.



MIDI Connections To Other Programs

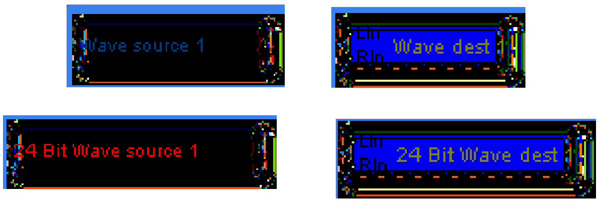
The **Sequencer Midi Source** and **Sequencer Midi Dest** modules are Windows MIDI drivers which can be accessed from outside of SCOPE. By default, SCOPE provides two such inputs and outputs for MIDI communication with other Windows programs. Your sequencer program will list them as "Scope MIDI In 1" and "Scope MIDI Out 1". If you add another Sequencer MIDI module to your project, it will be designated as "2".

In the default SCOPE configuration, two MIDI inputs and outputs are registered in Windows. If you add yet more Sequencer MIDI modules, you'll need to go into the Windows Device Manager and increase the number of SCOPE MIDI drivers registered there (and then restart the computer) before the additional Sequencer MIDI modules can be used. To get there, right-click on *My Computer*, select *Properties* in the menu which appears, then select *Device Manager*. Double-click on *Audio, Video and Game Controllers*, then on *Scope*, and finally select *Settings*, where you can change the number of SCOPE MIDI input and output "devices". Don't forget to restart Windows!



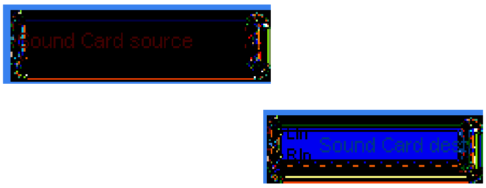
General Purpose Software Audio Connections

The **Wave Source** and **Wave Dest** modules for Windows audio communication are, like the Sequencer MIDI modules, assigned numbers sequentially as they are added to a project. If you go beyond four Wave inputs or outputs (i.e., two Source or Dest modules – they're stereo), SCOPE will make the Settings changes for you in the Device Manager (but you'll still need to restart Windows before you can use the additional I/Os).

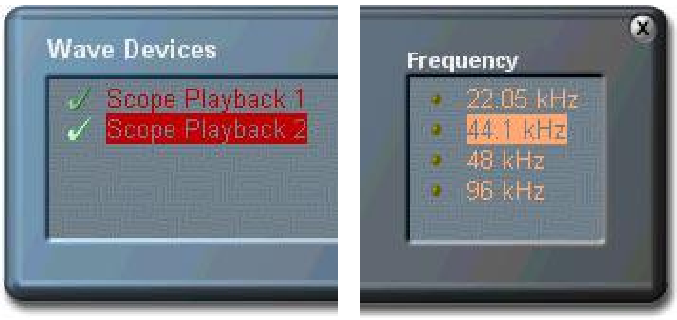


The Wave drivers are provided in 16-bit ("standard") and **24-bit** versions. The 24-bit versions preserve full fidelity and dynamic range when used to transfer multichannel mixes, etc., into or out of SCOPE. However, they should not be used unnecessarily, as each 24-bit Wave input or output consumes twice as much data transfer capacity on your computer's PCI bus as its 16-bit counterpart.

SCOPE also provides **Sound Card Source** and **Sound Card Dest** modules which permit direct audio communication between SCOPE and a standard PC sound card. Thus, you don't *have* to throw your existing sound card away when you install SCOPE. Instead, you can integrate it into the SCOPE system and continue to use it for audio input and/or output.



However, it should be noted that audio connections via **standard sound cards are prone to relatively high delays** (latency) compared to the inputs and outputs of the SCOPE hardware. This means that they are **suitable for general-purpose non-time critical uses** (talkback mike or other auxiliary audio source input, cue/monitor output) but not necessarily for use as an effects loop interface to external effect devices.



Via the **control surface** of the Sound Card Source and Sound Card Dest modules, you can **specify sample rates and signal sources** to be used with the card.

By means of the **Direct Sound Source** module, SCOPE can serve as a substitute sound card for programs (e.g., games) which deliver their audio output via the **DirectX** sound interface. (There is no corresponding dest module.)

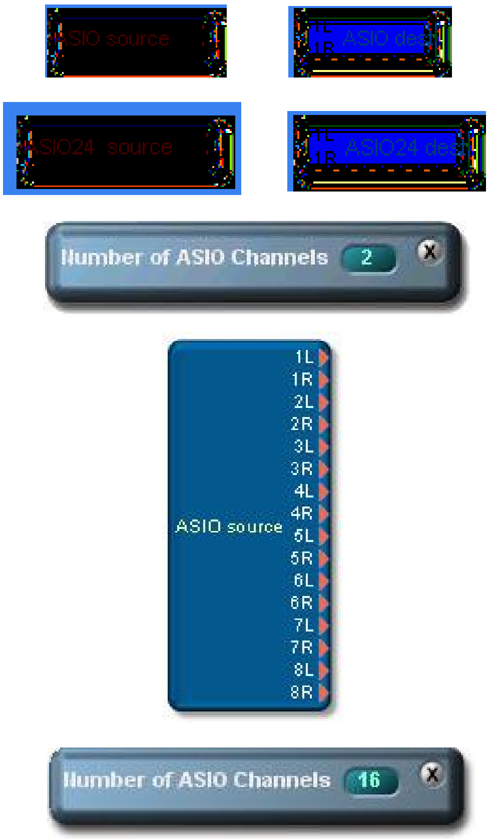
ASIO Software Audio Interfacing

These modules are provided for audio communication between SCOPE and programs which use the ASIO interface model (see the next section, *SCOPE And Cubase*, for more information on this topic).

The ASIO drivers are provided in 16-bit ("standard") and **24-bit** versions. The 24-bit versions preserve full fidelity and dynamic range when used to transfer multichannel mixes, etc., into or out of SCOPE. However, they should not be used unnecessarily, as each 24-bit ASIO input or output consumes twice as much data transfer capacity on your computer's PCI bus as its 16-bit counterpart.

Unlike most SCOPE I/O modules, the **ASIO Source** and **ASIO Dest** drivers have small control surfaces accessed via the right-click-on-module menu. These let you set the number of ASIO input or output channels (up to a maximum of 32 channels). Channels are added or removed dynamically – the module resizes itself onscreen accordingly as you change the setting, as shown in the illustration at right (here, the module has eight stereo channels – half the maximum possible number).

Note that changes to the number of ASIO channels are best done ***before*** the associated external audio program is started.



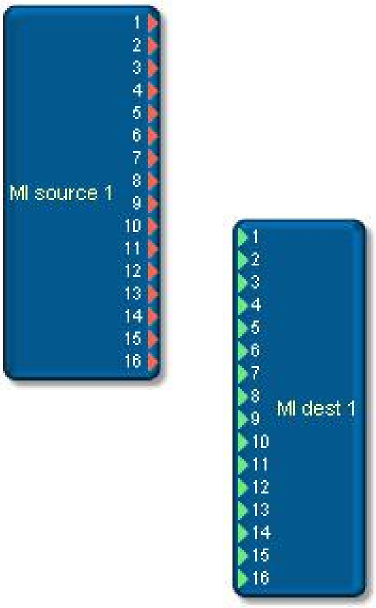
Number of ASIO Channels 2

Number of ASIO Channels 16

16 Wave Interleaved Source/Dest

These modules permit audio communication between SCOPE and Emagic's Logic Audio (see the next section, SCOPE And Logic, for more information on this topic).

Both the Source and Dest modules have 16 channels. We have found that Logic actually performs better with multiple stereo drivers (Wave Source/Dest modules) than it does with the interleaved modules. Nevertheless, the latter have been retained for the sake of compatibility.

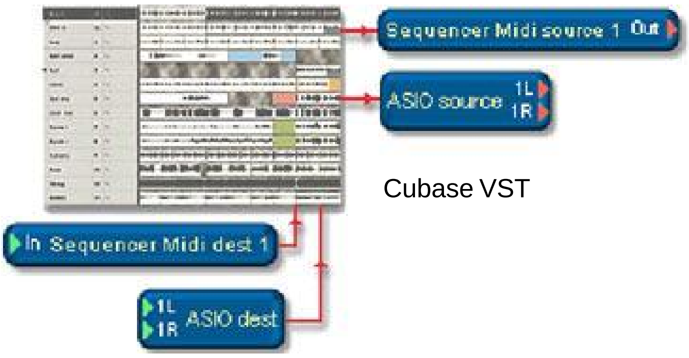


- SCOPE And Cubase - Introduction

Because it uses ASIO drivers instead of standard Windows Wave drivers, Cubase VST is an attractive option for sequencing work with SCOPE. ASIO drivers such as those provided by SCOPE permit more direct access to the system hardware used in the hard disk recording process and can therefore transfer audio data more efficiently to and from other programs. When using SCOPE with programs which are not ASIO-compatible, longer latency times (delays) in the audio tracks can be expected.

A brief overview of the use of Cubase with SCOPE is provided here. **For more details, you should refer to the example in the next chapter that describes how to setup SCOPE and Cubase for 16 audio channels**.

The signal flow when using Cubase VST together with SCOPE is diagrammed at right. Audio and MIDI originating in SCOPE, whether from SCOPE synths or the SCOPE hardware inputs, exit SCOPE via the ASIO Dest and Sequencer MIDI Dest modules and arrive in Cubase. Signals from Cubase enter SCOPE via the respective Source modules. The Sequencer MIDI Dest module appears in Cubase as a MIDI input (Scope MIDI In 1), ASIO Dest as a pair of VST inputs (Scope 16 bit 1L and 1R). The Dest modules appear correspondingly as outputs in Cubase.

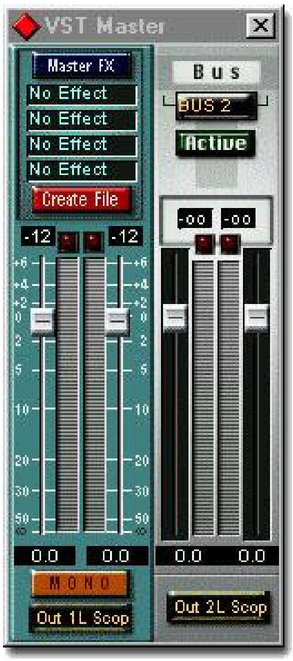


Cubase VST

Cubase VST must be started *after* starting SCOPE, so that the SCOPE ASIO and MIDI drivers can be in place when Cubase starts.

ASIO outputs are represented in Cubase as **busses**. The first bus is defined as the **master** (appears in green in the image at right). Any additional outputs/busses appear to the right of the master.

In this example, the ASIO Source module in SCOPE is set for a channel count of four. **If you change this setting in SCOPE, the change is immediately relayed to Cubase** and registered there. In the process, the **existing bus settings in Cubase are lost**. Therefore, for every song you work on in Cubase, you should make sure to save a corresponding SCOPE project.



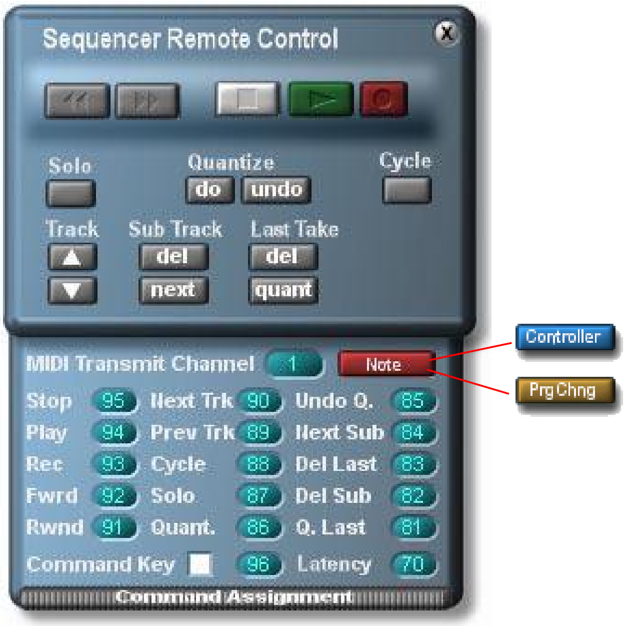
Bus 1
SCOPE 16 bit Out 1

Bus 2
SCOPE 16 bit Out 2

The **Sequencer Remote Control** allows the remote control of Cubase VST from within SCOPE. Since it achieves this through the use of standard MIDI messages (which are transmitted via the Sequencer MIDI Dest module to Cubase), and since the MIDI message for each of its buttons is redefinable, the Sequencer Remote can also be used for remote control of other programs. It comes preconfigured to match the default Cubase VST remote control setup.

To use the Remote, go into the **Remote Control** dialog in Cubase and turn on the **Remote active** option. To avoid inadvertent disabling of remote control, the **Remote key** option should be disabled. It shouldn't be necessary to modify any of the MIDI message settings if they are still in the program default configuration.

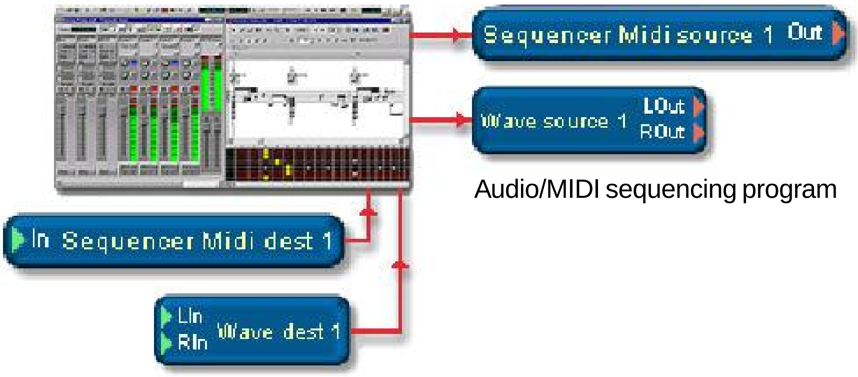
To reconfigure the Remote for other uses, open the **Command Assignment** slide-out drawer. There you can specify the type of message (Note, Controller or Program Change) as well as the specific message value to be sent for each button. The MIDI channel on which these messages are sent can also be adjusted (one setting applies to all messages).



MIDI Transmit Channel 1 Note

Stop	95	Next Trk	90	Undo Q.	85
Play	94	Prev Trk	89	Next Sub	84
Rec	93	Cycle	88	Del Last	83
Fwrd	92	Solo	87	Del Sub	82
Rwnd	91	Quant.	86	Q. Last	81
Command Key	□		96	Latency	70

- Hard Disk Recording



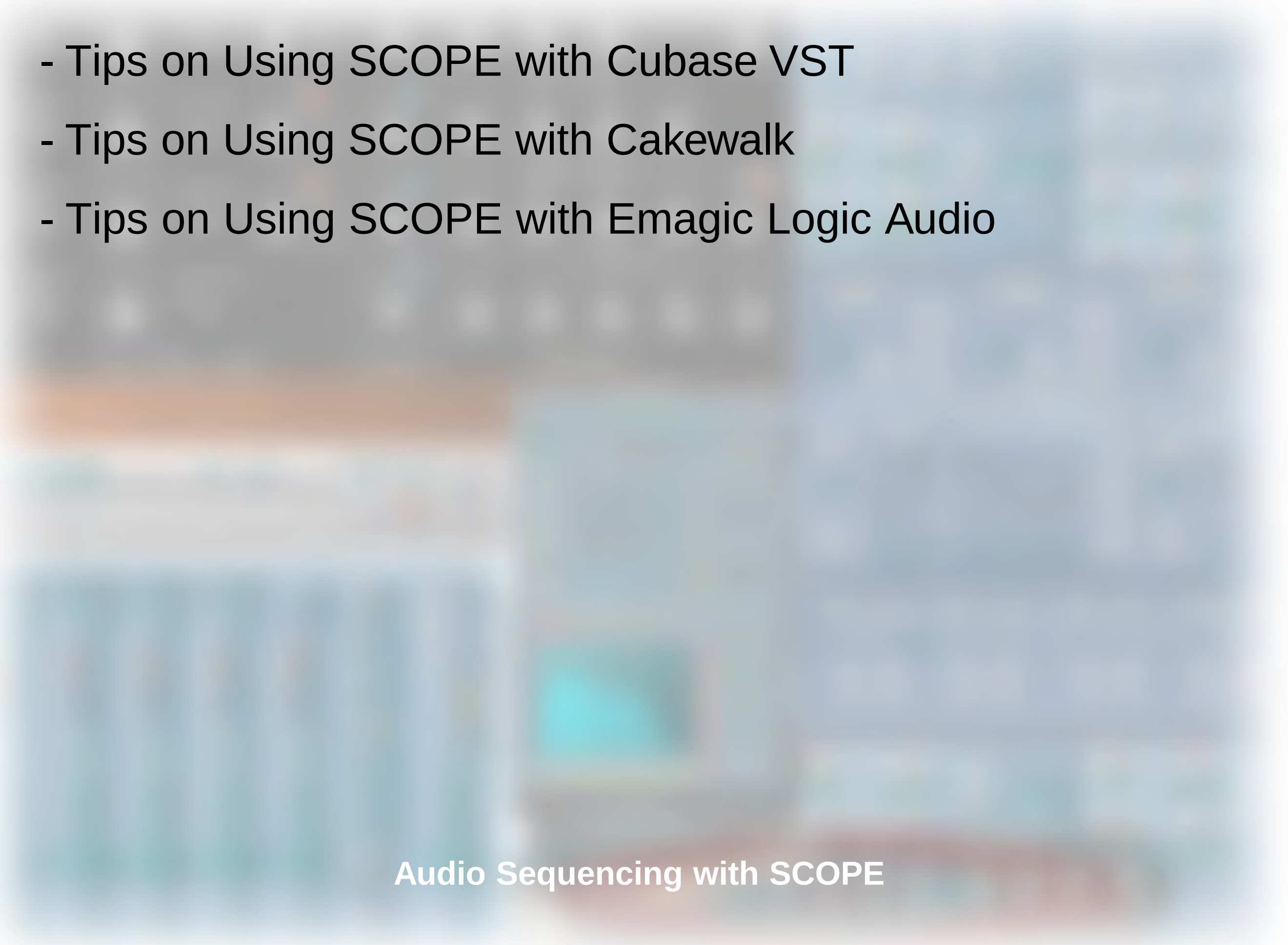
Hard disk recording using Wave drivers works fundamentally the same in SCOPE as otherwise. All that is required is an **appropriate setup project in SCOPE** and the corresponding settings in your recording software.

SCOPE should always be started first, followed by the recording program, to ensure that the SCOPE drivers required (and expected) by the recording program are already in place before it starts and maintain an uninterrupted presence.

Note that the requisite drivers can also be a part of the **background project** which manages the configuration of the SCOPE card when the SCOPE application is not running. This is workable if you just want the SCOPE resources in place to support recording and won't need to work directly in SCOPE (e.g., to operate the BigMixer) during your session. But **if you need to start SCOPE , close the recording program first and then start it again** when SCOPE is running.

Detailed information concerning the setup of SCOPE and a few of the leading audio sequencing programs for combined use is provided in the next chapter. Additional information on the background project can also be found there, as well as in the beginning sections of this chapter of the manual.

Audio Sequencing with SCOPE



- Tips on Using SCOPE with Cubase VST
- Tips on Using SCOPE with Cakewalk
- Tips on Using SCOPE with Emagic Logic Audio

Audio Sequencing with SCOPE

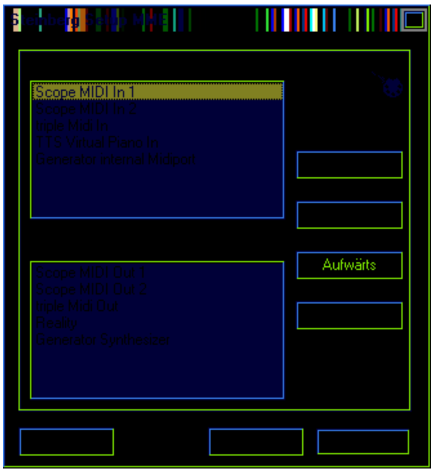
- Tips on Using SCOPE with Cubase VST

Simultaneous operation of SCOPE together with Steinberg's Cubase VST is easy to achieve and quite reliable, as long as you're aware of a couple of things.

Two important features make these programs compatible with one another: identical sample rate settings in both programs and the use by Cubase of ASIO drivers, which are provided by SCOPE for audio playback. (Also, you should use a version of Cubase no older than VST 3.55R3!)

When Windows starts, a SCOPE project is automatically loaded in the background. This project is set up to use a **sample rate** of **44.1kHz**. By default, SCOPE is configured as a **wordclock master**. Additionally, the project makes **ASIO drivers** available and connects these to the ADAT I/Os #2 and #3 of the SCOPE 24 I/O board.

After installing SCOPE, and before starting VST, you should run the Steinberg setup utility **Setup MME** (**Start .. Programs .. Steinberg Cubase VST ..**) and verify that the MIDI inputs and outputs are activated:

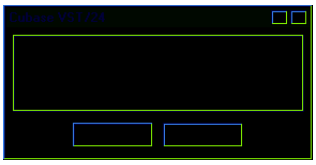


Steinberg Setup MME

Scope MIDI In 1
Scope MIDI In 2
triple Midi In
TTS Virtual Piano In
Generator internal Midiport

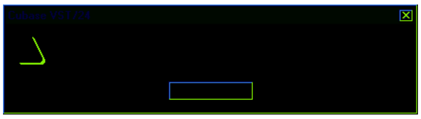
Scope MIDI Out 1
Scope MIDI Out 2
triple Midi Out
Reality
Generator Synthesizer

Now, upon starting VST (without having first started SCOPE) you may see the following message:



Click on **NO**, and VST will use SCOPE's sampling frequency anyway.

If you click on YES, the following message appears:

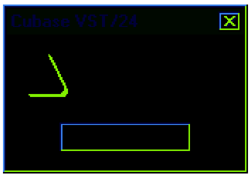


After confirming with **OK**, VST will again use SCOPE's sampling frequency.

This example makes clear that SCOPE determines the sampling frequency.

If you want to work in VST with a sample rate other than the one established in advance by SCOPE via the background project, you can open the appropriate dialog via the SCOPE icon which appears in the Windows Task Bar (at bottom right on your screen). If you've started SCOPE, you can find the same dialog within the program itself as „Samplerate Settings" under the Window menu.

If you at some later point modify your Windows start project so that it no longer includes ASIO modules, you will obtain the following error message upon starting VST:

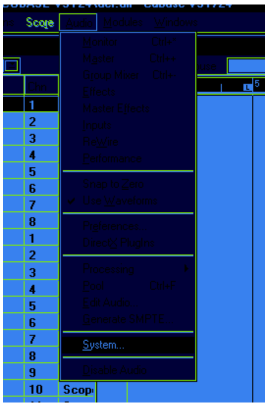


This message comes up when you start VST and there are no **ASIO modules** in the Windows start project (or the one currently loaded in SCOPE).

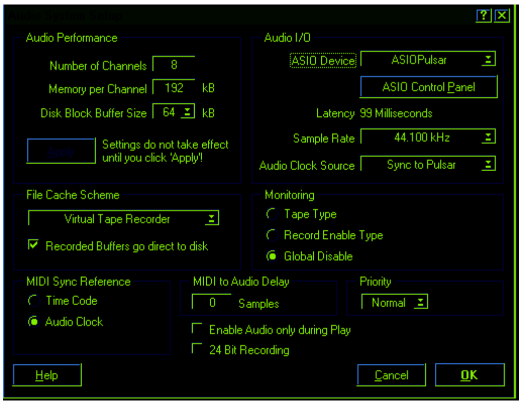
When you want to end your SCOPE/VST session, close Cubase VST **first**, **before** closing SCOPE. Otherwise you'll get this message and possibly others as well.

For MIDI integration, the **SCOPE MIDI Source/Dest** and **Sequencer MIDI Source/Dest** modules are already included in the "**default project**". Once you've selected the desired master sample rate in SCOPE (via the **Samplerate Settings control panel**), you can go ahead and start VST. Depending upon the rate you've selected, the '**Song Sample Rate does not match ...**' message may still appear. If this occurs, click on **NO**.

For seamless integration of the driver modules provided by SCOPE, your first step in VST should be to open the **Audio System Setup dialog**.



The following dialog box appears:



Audio Performance

Number of Channels 8

Memory per Channel 192 kB

Disk Block Buffer Size 64 kB

Settings do not take effect until you click 'Apply'!

Audio I/O

ASIO Device ASIOPulsar

ASIO Control Panel

Latency 99 Milliseconds

Sample Rate 44.100 kHz

Audio Clock Source Sync to Pulsar

File Cache Scheme

Virtual Tape Recorder

☑ Recorded Buffers go direct to disk

Monitoring

○ Tape Type

○ Record Enable Type

● Global Disable

MIDI Sync Reference

○ Time Code

● Audio Clock

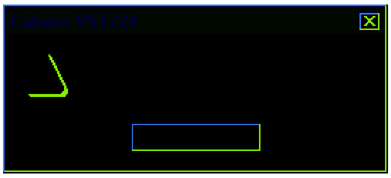
MIDI to Audio Delay 0 Samples

Priority Normal

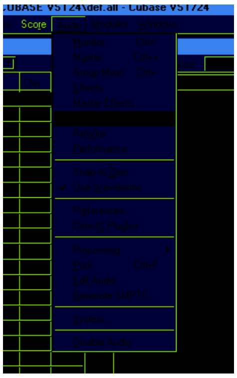
☐ Enable Audio only during Play

☐ 24 Bit Recording

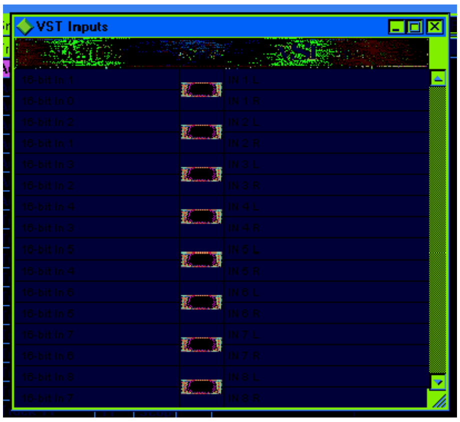
Select ASIO Scope as the **ASIO Device.** Directly below this are the **Sample Rate** and **Audio Clock Source** settings, which cannot be changed, since VST must use the same settings as SCOPE for proper synchronization. If you change the sample frequency in SCOPE, you will obtain the following message (in this case, upon a change from 48kHz to 44.1kHz). Simply confirm with **OK**.



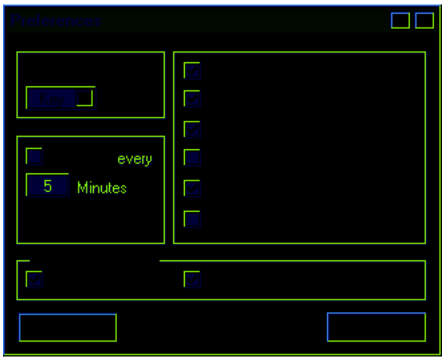
Next, the VST **audio inputs** should be selected. This selection is reached as follows:



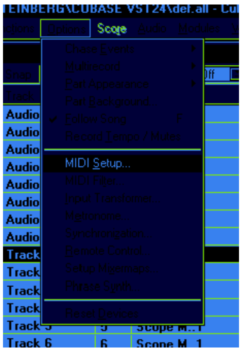
The VST Inputs window appears. It will show you eight stereo inputs, if you have selected 16 as the number of channels for the **SCOPE ASIO dest** module (via the the **Channel Settings** control panel of this module).

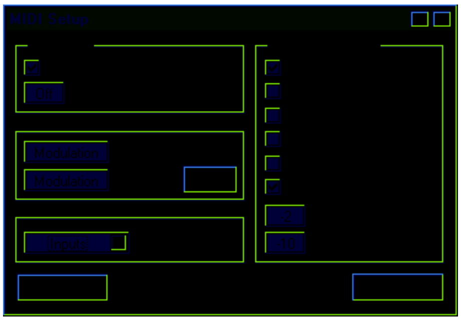


In order to enjoy uninterrupted MIDI and audio output during simultaneous operation of SCOPE and VST, you need to make sure that the **Play in background: Audio and MIDI** options in the **Preferences** dialog are activated:

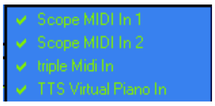


For **MIDI integration setup**, go into the **VST MIDI Setup** dialog:



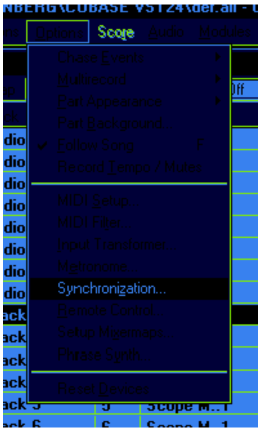


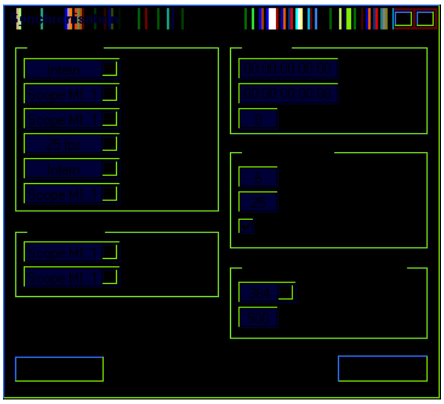
In the **Enable .. Inputs** pop-up list, activate the SCOPE **MIDI inputs**:

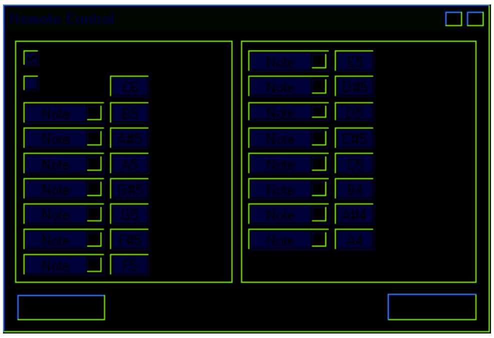


In order to have the MIDI messages which come in via the physical MIDI input of the SCOPE card be immediately available in both programs, the SCOPE MIDI Source module must be cabled to both the Sequencer MIDI Dest module and to any synthesizers you wish to hear during MIDI recording. Previously recorded MIDI events which are played back by VST appear in SCOPE via the Sequencer MIDI Source module, and can from there be routed to SCOPE devices as desired. If there is already a connection to the MIDI input of a particular device (e.g., from SCOPE MIDI Source), you can connect both MIDI signals to this input using the SCOPE MIDI Merge module (see illustration on next page). In addition, when you do this, you should make sure to disable the "MIDI Thru“ option in Cubase (Options .. MIDI Setup).

Specific **synchronization settings** can be made via the appropriate VST dialog:

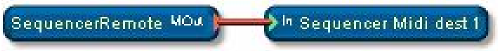




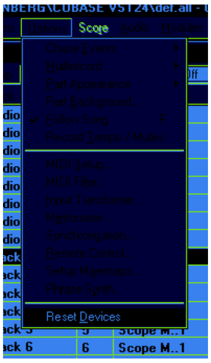


If you want to use the SCOPE **Sequencer Remote device**, which allows you to remote-control Cubase VST from within SCOPE, open the **VST Remote Control dialog** and turn on the **Remote active option**. The SCOPE Sequencer Remote device is pre-configured to work directly with the standard VST remote control values.

With this done, a simple connection in SCOPE between the **Sequencer Remote module** and the **Sequencer MIDI Dest module** is all that is needed to enable **remote-control operation** (refer to the previous SCOPE diagram).



In the course of normal operation it can sometimes become necessary to reset your MIDI devices. This can be done via an item at the bottom of the VST **Options** menu:



Record and Playback of multiple audio tracks with Cubase VST

Note: Our example involves 16 audio channels. Setting up SCOPE and Cubase for any other number of channels (2-32) is done in a similar fashion.

How does SCOPE communicate with Cubase VST?

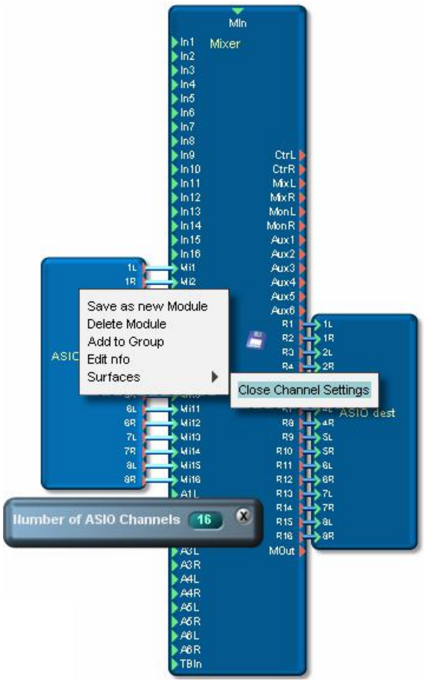
SCOPE supplies Cubase with audio channels via the ASIO interface. You can load ASIO source modules and/or ASIO dest modules.

Alternatively, you can use the Windows MME interface (Wave source and Wave dest modules). However, we do not recommend this approach, because it usually suffers from noticeable audio signal delays (latency).

The following example shows you how to set up SCOPE and VST to play 16 audio tracks from VST into SCOPE's BigMixer and to record the BigMixer's busses in Cubase.

1. Load the required ASIO modules in SCOPE.

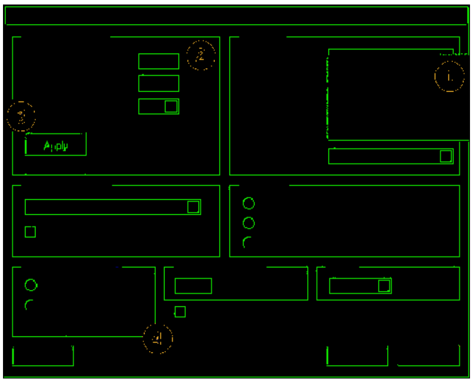
If you want to assign audio signals from your SCOPE project to Cubase, use an ASIO dest module. Use an ASIO source module to assign Cubase tracks to your SCOPE project. The following screen shot shows an example of how to connect the ASIO modules:



The dialog for adjusting the number of ASIO channels can be called up using the context menu or by double-clicking on the ASIO module. In our example, we've connected an ASIO source module with 16 channels to the Bigmixer's Mix inputs for mixdown of audio tracks from Cubase. You can additionally connect other SCOPE devices to the mixer inputs. The BigMixer's busses are connected to the ASIO dest module for routing back to Cubase, where they can be recorded.

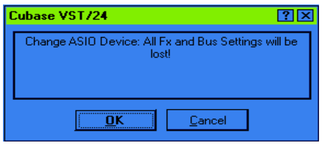
2. Tell Cubase to use SCOPE's ASIO interface!

Start Cubase and open the dialog "Audio Systems Setup" (menu Audio/System..):



2.1. Select SCOPE's ASIO interface:

Select the device "ASIO Scope" (1). You will obtain the following message:

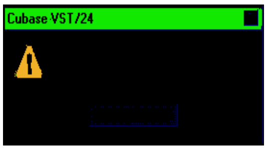


Cubase VST/24

Change ASIO Device: All Fx and Bus Settings will be lost!

Confirm with"OK" (or save your Cubase song to save the effect and bus settings of your current VST project).

If the sample rate of your Cubase song does not match that of SCOPE, you will obtain the following message (in this case, upon a change from 48kHz to 44.1kHz):



Simply confirm with OK. Now your Cubase song and your SCOPE project use the same sample rate.

2.2. Tell Cubase how many audio channels you want to use:

Select the desired number of audio channels (2).

In our example, 16 audio channels are being used. If you select more channels here than you have specified in your SCOPE project, you will not be able to use the addititional channels.

Confirm with the Apply button (3).

3. Configure Cubase to work with SCOPE

3.1. Enable "Play in background: Audio“

Refer to the last chapter.

3.2. Enable the inputs you want to use

Open the "VST Inputs“ window (menu Audio/Inputs, see illustration in the last chapter).

The inputs listed at left represent the SCOPE ASIO dest module channels. At right you can see which Cubase input each SCOPE channel is assigned to.

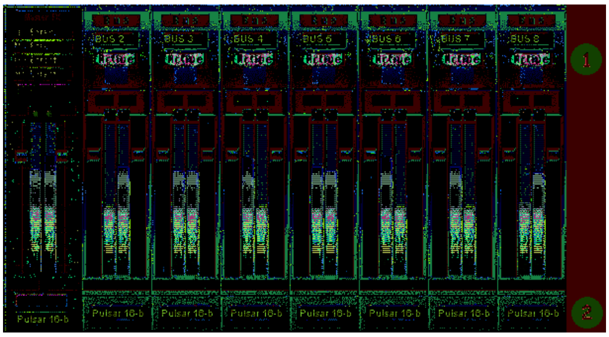
Enable the desired channels with the "Active“ button — in our example, all channels have been enabled, so that we can record on all 16 tracks simultaneously.

3.3. Assign the Cubase tracks

So you won’t need to keep jumping around from window to window, all of the settings for each window are listed together here.

Master Settings

Open the Master window:



You can assign two channels from Cubase to SCOPE using the Master Out. To send more than two channels to SCOPE, you must use Cubase busses. The busses are stereo channels, so you need one bus for every two additional channels. Click on the "Active“ button to enable as many additional busses as you need (1).

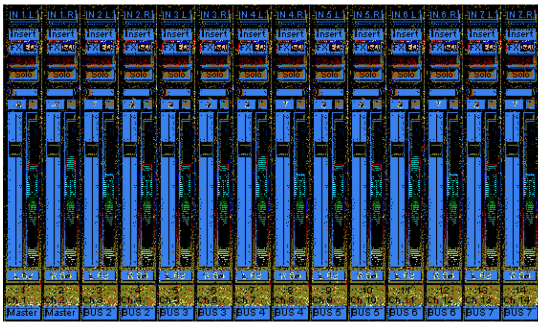
In our example, all 7 busses are enabled, so that we have 8 stereo channels (for 16 mono audio tracks)."

Select the outputs for the Master and all busses (2).

The Master has been assigned to "Out1L SCOPE 16-bit “ (the display cannot show the full label). In similar fashion, busses 2 - 8 have been assigned to outputs "Out2L SCOPE 16-bit “ through "Out8L SCOPE 16-bit “ respectively. The Master and all busses are now assigned to SCOPE ASIO source module channels in the correct order.

Mixer Settings

Open the Cubase Mixer:



First assign the inputs (SCOPE ASIO dest channels) to the audio channels (1). For each channel, select the input you wish to record from (Ctrl + left mouse button).

In the example, we want to record 16 bus signals from the BigMixer on 16 audio tracks in VST, so we've assigned the inputs to the corresponding channels. Later we'll assign audio tracks to these channels.

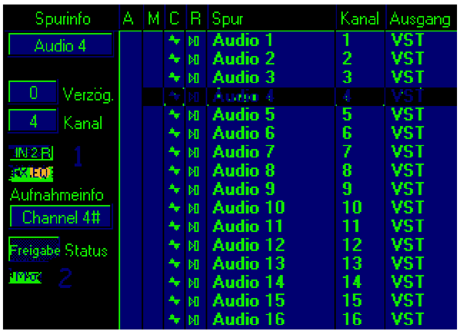
Next, assign the channels to Cubase outputs (2). We want to assign the 16 Cubase tracks to the SCOPE ASIO source module, so we select the Master out for channels 1 and 2, and busses 2 - 8 for the remaining ones. If the busses do not appear in the menu, you have probably not enabled them in the Master window (see "Master Settings")

Finally adjust the pan settings (3). Be aware that a channel receives full signal level only if its pan control is set full left or full right. As the Master and the busses are stereo channels, you must pan to the left to assign the signal to the left master or bus channel only. In our example, all odd channels have been panned to the left and all even channels have been panned to the right.

Enable Multi Record

Enable "Options/Multi Record/active" to record on several tracks simultaneously. In the arrangement window, a new column with the label "R" (Record) will appear. Here you can select the tracks you want to record on.

Arrangement Settings



A	M	C	R	Spur	Kanal	Ausgang
				Audio 1	1	VST
				Audio 2	2	VST
				Audio 3	3	VST
				Audio 4	4	VST
				Audio 5	5	VST
				Audio 6	6	VST
				Audio 7	7	VST
				Audio 8	8	VST
				Audio 9	9	VST
				Audio 10	10	VST
				Audio 11	11	VST
				Audio 12	12	VST
				Audio 13	13	VST
				Audio 14	14	VST
				Audio 15	15	VST
				Audio 16	16	VST

For each audio track, assign the desired playback and record channel. If you select a single track, its channel settings are displayed in the Info column.

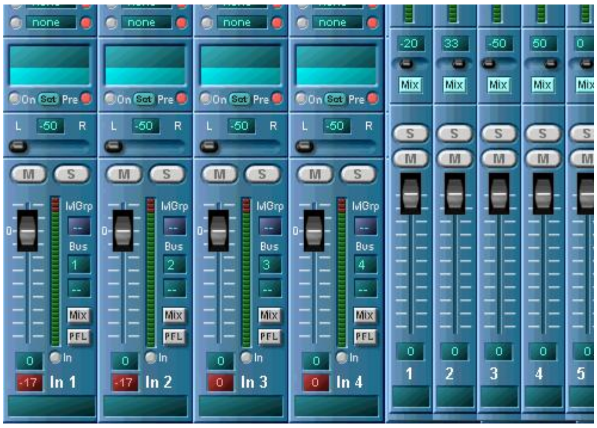
In the example, we've assigned the 16 channels one-for-one to the 16 audio tracks. The 4th track is highlighted, so its settings are displayed — for example, the assignment to input "IN 2 R" (1). As this channel is connected to the 4th channel of the ASIO dest module, this channel will be recorded to Cubase track 4.

To record on a different track, select the track in the "R" column, or click on the "Enable" (2) field of this track. In the example, we've activated all tracks, for a 16-track recording.

4. Assigning record busses in SCOPE's Bigmixer

Assign your input channels to the busses. We assigned channel 1 to bus 1, 2 to 2, and so on, using the upper Bus Select. Set the pan control of all input channels hard left to assign the signal to the upper Bus Select. Unmute the busses.

Press the Mix button in the bus channels and use the bus pan control to adjust the panorama as you want to hear it while recording. This does not effect the recorded signals.



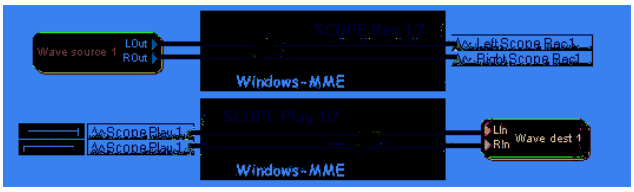
More extensive and detailed information can be found in the VST and SCOPE Users Manuals! Try to familiarize yourself with the step-by-step operation of both programs before contacting Creamware or Steinberg Customer Support for help. With a little time and patience, you'll soon know your way around in the program options.

Have fun sequencing with SCOPE and VST!

- Tips on Using SCOPE with Cakewalk

Our example is for 16 audio channels, setting up SCOPE and Cakewalk for any other number of channels (2-32) is similar.

SCOPE supplies other programs with audio channels using the Windows MME interface. You can load Wave modules in SCOPE that represent MME drivers. These modules appear in other programs as "Scope Rec“ and "SCOPE Play“ drivers. The following diagram shows the routing for the first two tracks:



To use this interface, proceed as follows:

1. Load the required Wave modules in SCOPE.

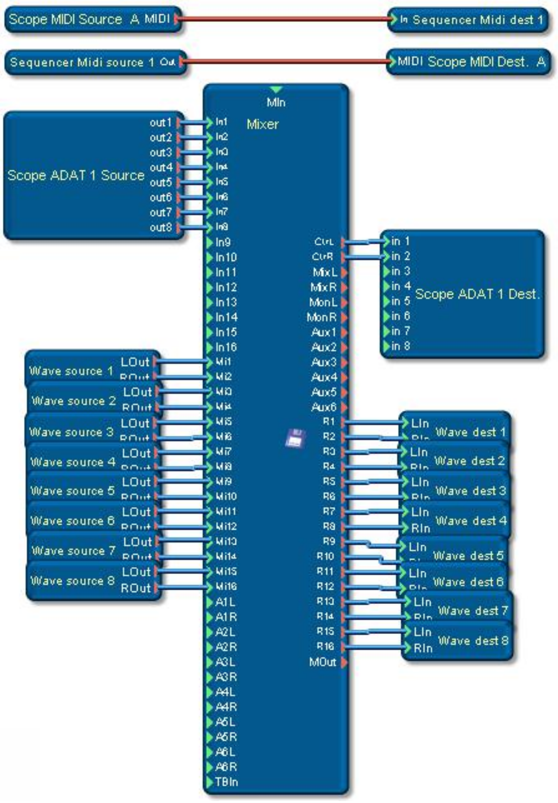
Generally you can record as many audio tracks in Cakewalk as you have loaded in SCOPE. Since a Wave dest module is a stereo device, you need 8 Wave dest modules for 16 audio tracks. Make sure your SCOPE project includes 8 Wave dest modules.

The same applies to the Wave source modules; load as many you need for the playback in Cakewalk.

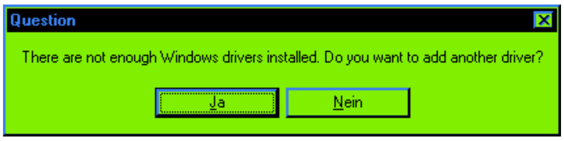
The screen shot shows how to connect the modules to record the BigMixer’s busses in CakeWalk.

In this configuration, SCOPE serves as a complex digital mixer for recording audio signals routed into In1 - In16 of the BigMixer. In the example, SCOPE’s analog input is connected to In1 and In2. The input channels must be assigned to the busses.

For controlling Cakewalk audio tracks during the mixdown process, 8 Wave source modules are connected to the BigMixer’s Mix Ins.



Sooner or later, when adding modules, you’ll probably encounter this dialog:



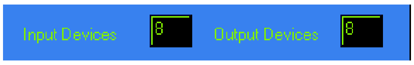
Question

There are not enough Windows drivers installed. Do you want to add another driver?

SCOPE is letting you know that there are not enough reserved Windows drivers and asks you if you want to introduce another driver to Windows. Click on "Yes“, and SCOPE will tell you that Windows must be restarted for the new driver to be activated. Before you do so, load the required number of modules and save your project. Then restart Windows.

2. Load the SCOPE MME drivers!

Now Windows supplies other programs with MME drivers. You can check the number of drivers in the Driver Setup of SCOPE (Control Panel -> System -> Device manager -> Sound, video and game controllers -> SCOPE -> Properties -> Settings):



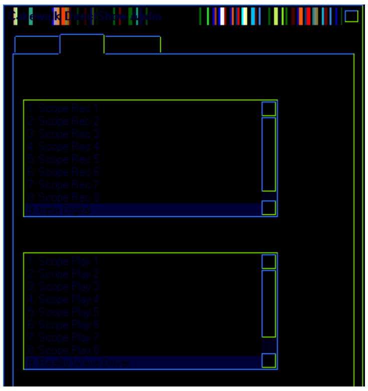
Input Devices 8 Output Devices 8

The "Input Devices“ are represented by Wave source modules, the "Output Devices“ are Wave dest modules.

If you want to make use of these drivers, you must start SCOPE and load the previously created project - with it, you load the MME drivers.

3. Tell Cakewalk to use the new MME drivers!

Start Cakewalk and open the "Cakewalk DirectShow Audio“ dialog (Tools -> Audio-Options ...). In the "Drivers“ panel, you will see the new SCOPE drivers. Highlight all SCOPE drivers to enable them.

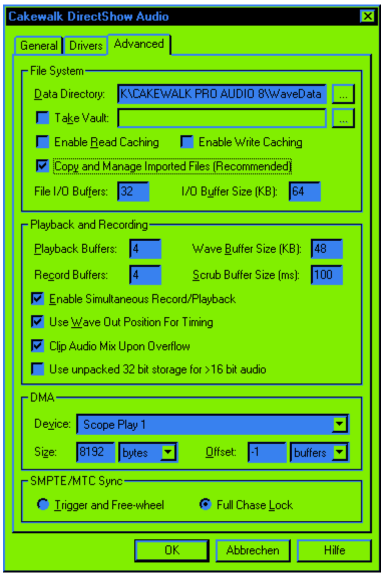


1: Scope Rec 1
2: Scope Rec 2
3: Scope Rec 3
4: Scope Rec 4
5: Scope Rec 5
6: Scope Rec 6
7: Scope Rec 7
8: Scope Rec 8
9: Triple Digital

1: Scope Play 1
2: Scope Play 2
3: Scope Play 3
4: Scope Play 4
5: Scope Play 5
6: Scope Play 6
7: Scope Play 7
8: Scope Play 8
9: Reality Wave Driver

In the "General“ panel make sure the selected sampling rate (1) is the same as in SCOPE (default 44.1kHz). Select SCOPE Play 1 and SCOPE Rec 1 as timing master for playback and record (2):

Next click on the ‚Advanced‘ tab. The only changes we made was to disable ‚Take Vault‘ and read and write caching.



File System
Data Directory: K:\CAKEWALK PRO AUDIO 8\WaveData
□ Take Vault:
□ Enable Read Caching
□ Enable Write Caching
☑ Copy and Manage Imported Files (Recommended)
File I/O Buffers: 32
I/O Buffer Size (KB): 64

Playback and Recording
Playback Buffers: 4
Wave Buffer Size (KB): 48
Record Buffers: 4
Scrub Buffer Size (ms): 100
☑ Enable Simultaneous Record/Playback
☑ Use Wave Out Position For Timing
☑ Clip Audio Mix Upon Overflow
□ Use unpacked 32 bit storage for >16 bit audio

DMA
Device: Scope Play 1
Size: 8192 bytes
Offset: -1 buffers

SMPTE/MTC Sync
○ Trigger and Free-wheel
● Full Chase Lock

Next, assign the Cakewalk tracks to the SCOPE ins and outs.

Record:

Select the driver to record from under "Source“. For example, if you want to assign the left channel of SCOPE’s first Wave dest module to the first track, select "left SCOPE Rec 1“ under "Source“ (1).

Playback:

In the "Port“ field of a track you can select the driver to which you want to send the track. For example, if you want to assign the first track to the left channel of SCOPE’s first Wave source module, select "SCOPE Play 1“ in the "Port“ field (2). In addition, you must set the track pan hard left (value 0) to make sure the signal appears only on the left channel of the Wave source module (3).

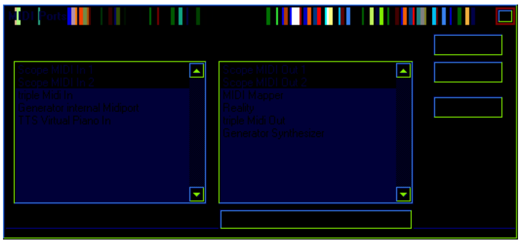
In our example, all odd tracks have pan set to 0 and all even ones have pan set to 127. All track signals will appear in their proper places in the BigMixer.

4. Assigning record busses in SCOPE’s BigMixer

This is done in exactly the same way as described in the preceding chapter on Cubase VST.

5. The MIDI setup

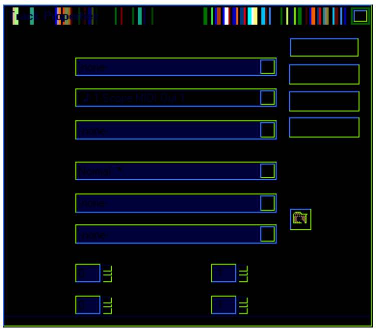
Open the MIDI Ports dialog of Cakewalk (Menu: Tools -> MIDI Devices). Enable the SCOPE MIDI Ports. The default is two MIDI ins and two MIDI outs. You can specify the number of SCOPE MIDI ports in the SCOPE driver setup dialog (in the Windows Device Manager).


MIDI Ports

Scope MIDI In 1
Scope MIDI In 2
triple Midi In
Generator internal Midiport
TTS Virtual Piano In

Scope MIDI Out 1
Scope MIDI Out 2
MIDI Mapper
Reality
triple Midi Out
Generator Synthesizer

Assign sources and ports for each MIDI track in the Cakewalk Track Properties dialog which appears when you double-click on the source or port field:


Track Properties

-none-
♪ 1-Scope MIDI Out 1
-none-
Normal *
-none-
-none-
0 | -1
0 | -1

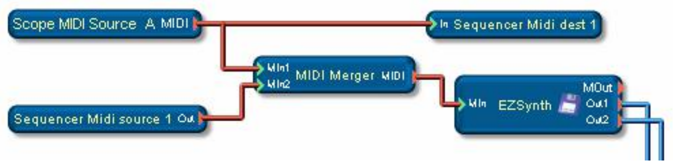
This example shows how to record MIDI from your keyboard into Cakewalk using SCOPE.

The keyboard is connected to the SCOPE MIDI input, represented by the ‚SCOPE MIDI Source‘ module. This module is connected to ‚Sequencer MIDI Dest 1‘ so that the MIDI signal will appear as an input in Cakewalk.

We want to record a solo using the EZ Synth. The MIDI signal from ‚SCOPE MIDI Source‘ is also connected to the input of the EZ Synth, so we can hear the synth while playing the keyboard,

In addition, ‚Sequencer MIDI Source 1‘ is connected to the input of the EZ Synth, so we can hear the recorded solo when it is played back from the sequencer.

The MIDI Merger is required in order to connect both ‚Sequencer MIDI Source 1‘ and ‚SCOPE MIDI Source‘ to the EZ Synth at the same time.



To avoid a doubled sound while recording, disable echo for the MIDI channel you are recording or for all MIDI channels (Echo Mode: None) in Cakewalk’s Project Options dialog (Tools -> Project Options -> Midi Input).

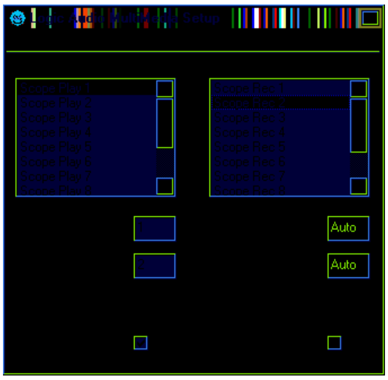
- Tips on Using SCOPE with Emagic Logic Audio

Our example involves 16 audio channels. Setting up SCOPE and Cubase for any other number of channels (2-32) is done in a similar fashion.

Proceed at first exactly as described in points 1 and 2 in the preceding chapter on Cakewalk.

3. Tell Logic to use the new MME drivers!

In your Logic directory you will find the program MMSetup.exe. Start it, and the following dialog will appear:



Click on "Scope Play 1"- (1) , and under (2) the number of SCOPE playback drivers you have used so far is displayed - probably 2. Enter the number of drivers you want to use to play audio tracks from Logic into SCOPE - in our example 8. This results in 8*2 = 16 channels (3).

Follow the same procedure for "Scope Rec 1"

If you click on a different driver (2 - 8, as shown in the Recording list), the fields "Number of Drivers" and "Channels per Driver" should display the value "Auto" (4).

Note that the number of drivers you specify here must correspond exactly to the number of Wave Source/Dest modules previously loaded in the SCOPE project. If the numbers don't match correctly, you will obtain the following error message upon starting Logic:

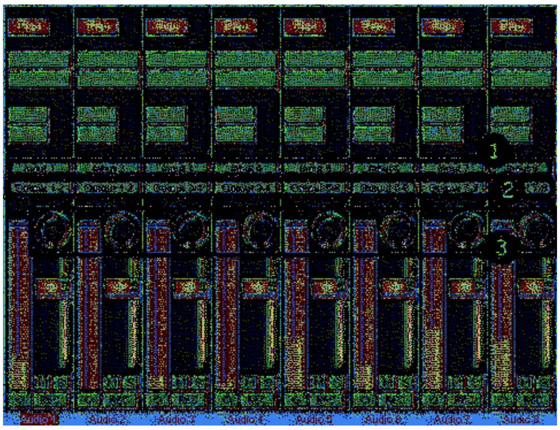
PC AV Error:5

Failed to recover device from playback/recording

Start Logic and open the "Audio-Extensions" dialog (Options -> Audio-Setup -> Audio-Extensions). You will see that the new playback and recording drivers are now available in Logic:

Next you can route the audio tracks to the desired SCOPE channels in the Track Mixer.

(The picture shows the channels 1-8 only; repeat the same procedure for channels 9-16)



Record:

To record an audio signal that is routed to a Wave dest module in SCOPE, select the respective Logic input for the track you want to record to. The first Wave dest module is assigned to Logic input 1 and 2, the second to 3 and 4, and so on. In the picture the first 8 busses of the BigMixer are recorded on the first 8 Logic tracks (1).

Playback:

To play a Logic track into a SCOPE project, set the output of the Logic track to the respective Logic output. Logic output 1 and 2 are assigned to the first Wave souce module, and so on. In the picture, the first 8 Logic tracks are routed to the Mix Ins 1 - 8 of the BigMixer (2).

As you probably want to assign tracks 1,3,5,..., 15 to the left channels and tracks 2,4,6,...,16 to the right ones, adjust the pan control for all odd tracks to left and for the even ones to right. This ensures that track one is routed to channel 1, track 2 to channel 2, and so on.

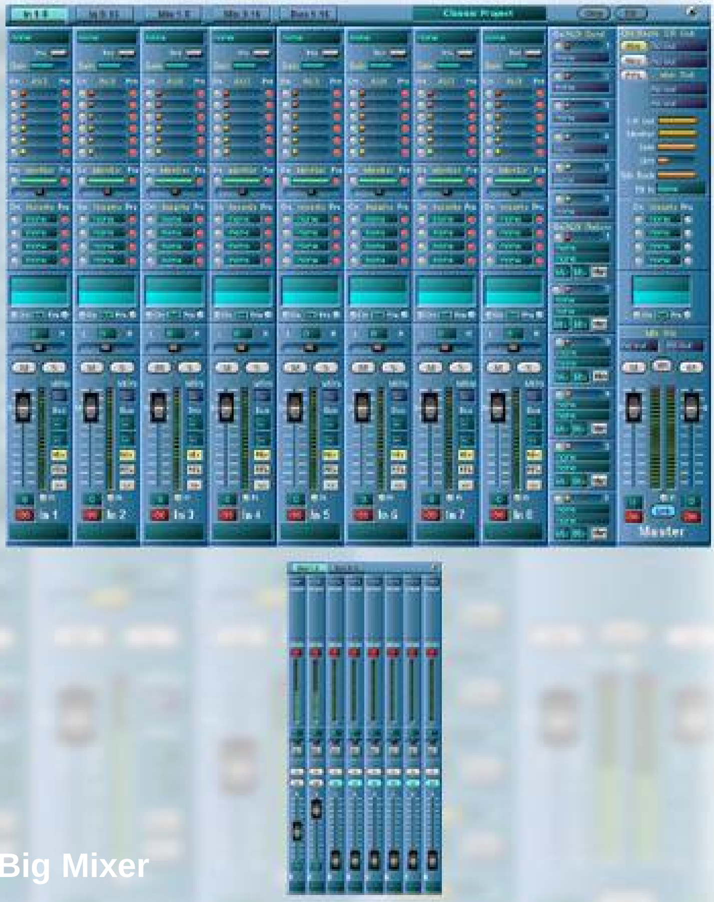
4. Assigning record busses in SCOPE’s Bigmixer

This is done in exactly the same way as described in the preceding chapter on Cubase VST.

Now you can mix 16 Logic tracks in your SCOPE project and record the signals from the BigMixer’s busses in Logic. Please see your Logic manual for further details.

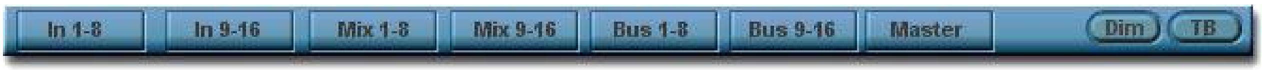
Big Mixer

- Overview
- Input/Mix Channels
- Bus Channels
- Aux Section
- Master Section
- Connections



Big Mixer

- Overview



The BigMixer is located in the Scope\Data\Mixer directory.

The BigMixer has **32 channel strips**. Sixteen of these are designated as mix channels and sixteen as input channels. There are also **sixteen busses**, **six auxiliary sends** with stereo returns and a **stereo monitor bus**.

The BigMixer is a full-featured recording mixer which **requires a large amount of computing power**. When you wish to use several synths at the same time, and don't need all of the recording-oriented features of the BigMixer the smaller **Dynamixer**, a relatively simple 8-channel line mixer, can be a better choice.

The button strip pictured above lets you **switch the display** among the various channel groups, the bus groups and the master section.

The **Dim switch** produces an adjustable level change in the master outputs.

The **Talkback** button is likewise activated from here. In the master section, the talkback signal can be routed to a particular hardware output.

Each input or mix channel as well as the master section) has **four effect insertion slots**. There are no restrictions on their use – you can use them all at once, on all channels. Nevertheless, care should be taken to **deactivate any unused insert slots, aux sends or EQs**, in order to keep the maximum possible amount of DSP power available for other synths, etc.

The distinction between input channels and mix channels in the BigMixer is somewhat artificial. It serves mainly to provide a measure of convenience and "comfort with the familiar".

Classically, input channels are used for recording. The channel output is sent to a recording device and is not routed through the master section or assigned to the main mix. When recording is complete, all 32 channels can be used as mixdown inputs.

Functionally, however, there is ***no* difference between mix channels and input channels in the BigMixer** – both types have the same features.

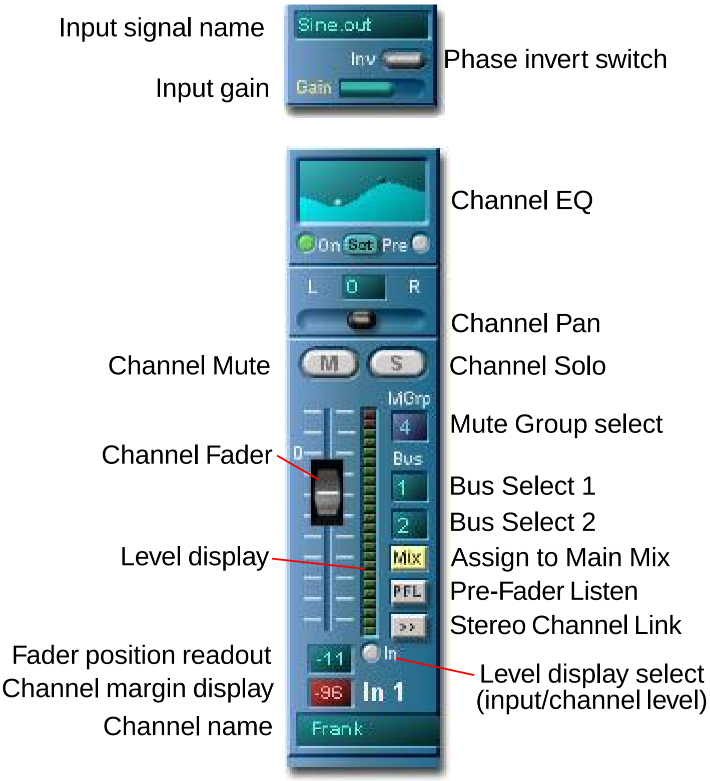
- Input/Mix Channels

A portion of a channel strip is shown at right. At the top is the **input unit**, where the channel input can be assigned. The **input gain** can also be adjusted here, and the **phase** of the input signal can be inverted if desired. **Gain** is adjusted by left-clicking and holding on the end of the Gain fader and moving the mouse right or left.

The **Pan** control has an associated numerical readout to allow pan settings to be precisely set. The control itself has a center-detent to allow channels to be centered easily in the mix using the mouse (this can also be done by double-clicking on the Pan fader).

Each channel can be **individually muted** via the Mute (**M**) button. It is also possible to assign several channels to a **Mute Group** (**MGrp**), so that all channels in the group are simultaneously muted or unmuted via the Mute button of any channel in the group. Mute groups, and likewise bus assignments, are adjusted by left-clicking and holding on the corresponding numeric display box next to a fader and moving the mouse up and down.

If a channel is routed only to one bus, then the **Pan** control must be set full left for an odd-numbered bus or full right for an even-numbered bus in order for the full channel signal level to be delivered to that bus. If two busses are selected for a channel, then the channel signal divides itself between them according to the Pan control setting.



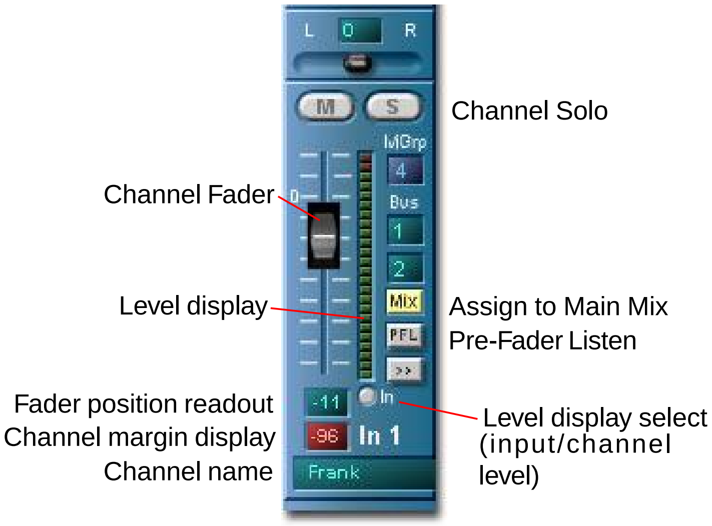
Any channel can be routed to the solo bus using the **Solo** (**S**) button. **The solo bus is a stereo bus** which receives channel signals **post-pan**, allowing a channel's pan positioning to be checked while soloing it (**solo in place**).

Pre-Fader Listen (**PFL**) allows monitoring of the channel signal before the channel fader but after the input gain control. The **level display** shows either the **post-fader channel output level**, or – if the **In** button directly below it is pressed – the level of the **unmodified channel input signal**. The **fader position readout** indicates the channel fader setting numerically and allows it to be set by entering numeric values directly.

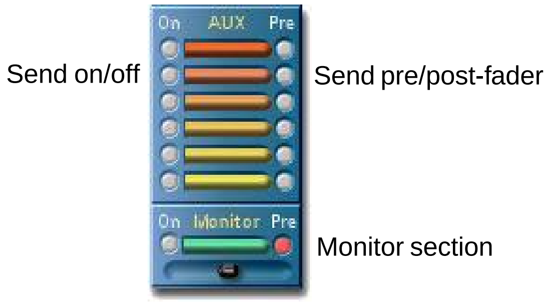
The **Margin** display captures instantaneous channel peak signal levels and shows them as dB headroom (0 = overload). It can be cleared via mouse-click.

Channel auxiliary send levels are individually adjustable in the **Aux section**. All Aux sends in the BigMixer can be individually switched for **pre-fader or post-fader send**.

The **Monitor section** is essentially an additional aux routing, with features similar to those of Aux section routings. However, **the monitor bus is a stereo bus** and the channel send to the monitor bus includes a separate pan control.



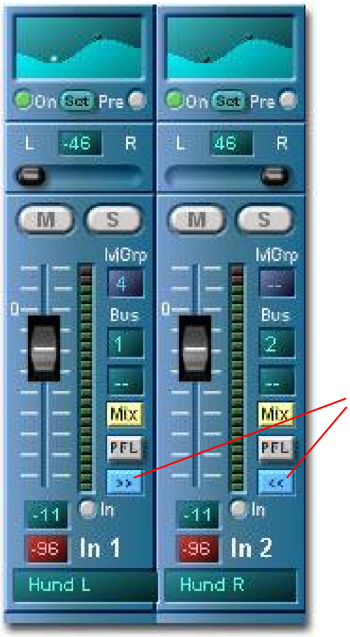
Aux section



The **Stereo Channel Link** buttons permit individual odd/even-numbered channel pairs (1-2, 3-4, etc.) to be linked together to function as stereo channel strips.

The following channel controls are then locked together and can be adjusted for both channels from either strip: Channel Fader, Assign To Main Mix button, Mute (M), Solo (S), Pan, Channel EQ, Input Gain and all Monitor Send controls. As can be seen in the illustration at right, the **Pan controls in linked channels automatically "mirror" one another**.

All other channel controls – including effect inserts – retain their separate functions, as do the level and margin displays.



Stereo Channel Link

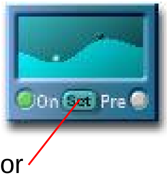
Below the aux sends and the effect insert slots is the **full-parametric EQ**. The EQ can be switched on or off here and switched to operate either **pre-fader or post-fader**.

Channel EQ is fully editable directly within a channel strip. **Double-clicking in the EQ field creates a new full-parametric filter** (up to a maximum of four per channel) which is represented by a point in the field. The **center frequency** of this filter can be adjusted by left-clicking on the point and dragging it left or right. Dragging it up or down adjusts the **passband gain** of the filter. Dragging it up or down with a right-click adjusts the **filter Q**.

Switching a channel's EQ off (by clicking on the On button) causes not only the channel signal but also the SCOPE card itself to bypass the EQ. This **reduces the DSP workload of the SCOPE card** and should be done for any channel in which the EQ is not being used.

The **Set** button opens a special **EQ editor** which allows more detailed and precise graphical EQ editing by virtue of its larger display area. It also has a set of controls, each with a numeric display, for **direct adjustment** of frequency, gain and Q values for a selected filter point (**select a specific point by left-clicking on it**).

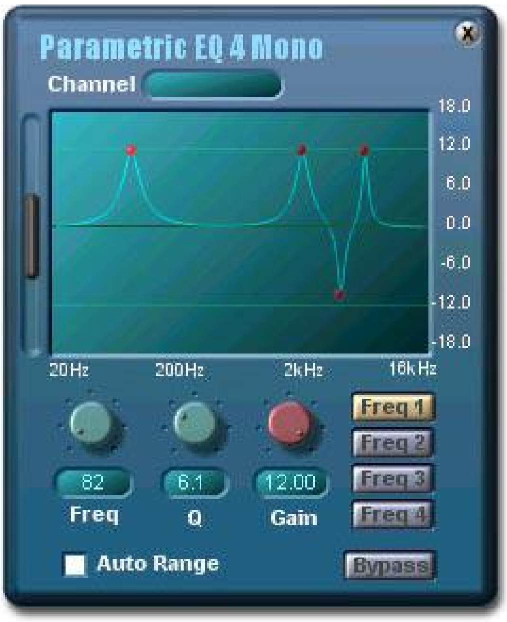
EQ section



EQ on/off

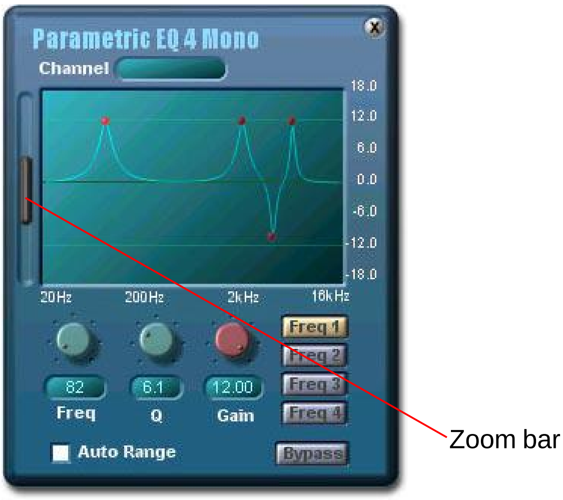
EQ pre/post-fader

Open EQ editor



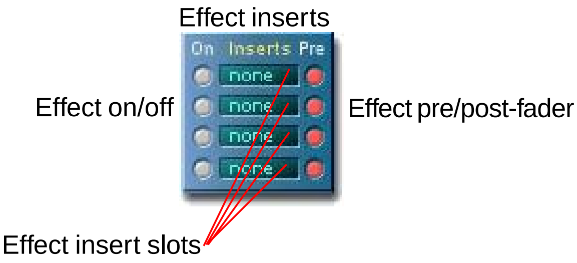
Freq 82
Q 6.1
Gain 12.00

The EQ editor's **zoom bar** allow vertical rescaling of the display for **optimal viewing or increased precision** during graphical editing. Clicking and dragging on the ends of the zoom bar resizes it and causes the display to rescale itself accordingly, letting you zoom the view in to **allow finer adjustments to be made** with the mouse. The resized zoom bar can also be moved by clicking and dragging on its middle, letting you scroll the display to find the specific point of interest. Double-clicking on the zoom bar results in an **instant full zoom-out**.



Each channel strip has four **effect insert slots**, each of which can operate **pre-fader or post-fader**. Each one can also be independently switched on or off. Effect insert slots which are not being used should be switched off to conserve DSP capacity.

Connect an insert effect by simply **dragging its name** from the File Browser and **dropping it on an insert slot**. You can use either existing software effects (from the folder **..\Inserts\Mono\**) or your own self-defined (external) effects modules as insert effects. **To remove an insert effect**, select it and hit the Delete key on your computer's keyboard, or delete via the menu which pops up when you right-click on the insert slot.



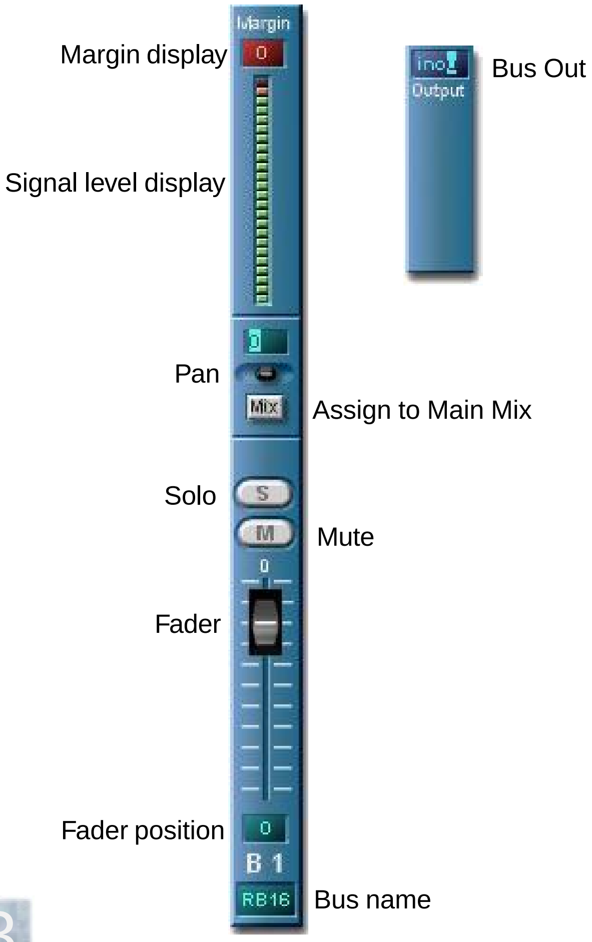
- Bus Channels

Bus channels are essentially **simplified input/mix channels** which are intended for use either **as recording busses or as subgroups**. They include a selected subset of the features found in input/mix channels.

For recording use, input/mix channels which are handling signals to be recorded are assigned to one or two bus channels. The outputs of each bus channel which is being used as a record bus is connected to one of the inputs of the recording device – for example, an ADAT. (Refer to the SCOPE Applications chapter for more information about using SCOPE with external devices).

Once all recording is complete and recording busses are no longer needed, **bus channels can be used for the mixdown as subgroups**. For this use, bus channel outputs are assigned to the main mix, while input/mix channels which are carrying signals for mixdown are assigned to bus channels according to the needs of the particular situation. For example, all background vocals or all drum tracks could be grouped together on one bus or bus pair to allow them to be easily controlled as a unit.

To speed up the process of switching between recording and mixdown setups and save yourself some repetitive button pushing, you can save each setup as a recallable mixer preset.

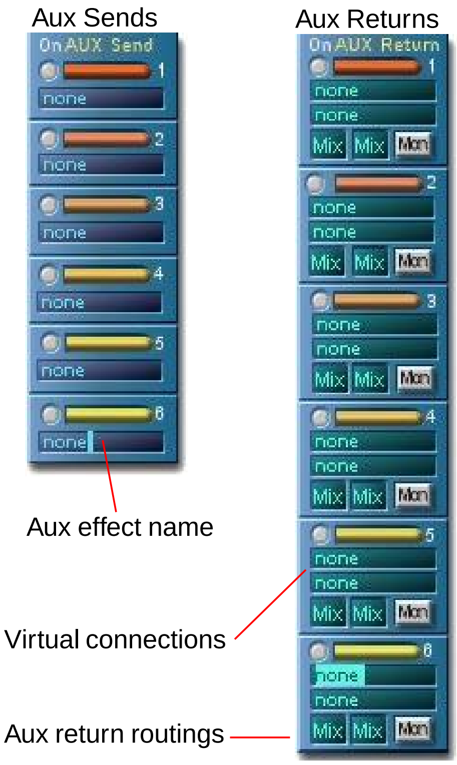


- Aux Section

The Aux section provides for **global control of aux send and aux return levels**. In addition, both the **send and the return** for a particular aux path can be **switched on or off** in this section.

Aux paths can be used both **with internal software effects** and with **external effects devices connected via the SCOPE card's hardware I/O**, since the mixer introduces practically no signal delay. Aux path routings can be set up most conveniently via the Rack Window. Aux path setup in the Project Window is also possible.

While the aux sends are mono sends, the aux returns are stereo. Aux returns can be routed to the main mix or to any one or two of the 32 mixer channels (allowing, for example, processing of an effect signal through a channel EQ) as well as to a virtual connection. In addition, any aux return can be routed to the monitor mix, allowing **effects to be added to the monitor mix**.



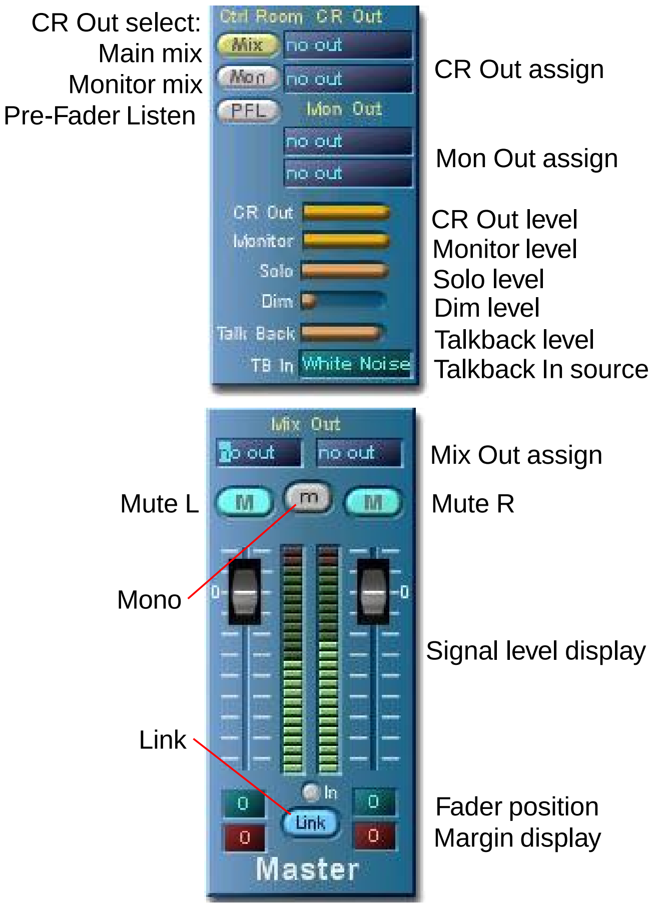
- Master Section

The Master section controls the **stereo main mix**. All signals which are routed to **Mix** show up in this mix, which has its own EQ as well as its own set of effect insert slots. The Master section also includes controls for the **Monitor mix**, **Pre-Fader Listen (PFL)** and the Mixer outputs.

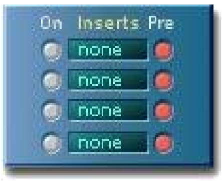
There are three freely-assignable stereo outputs: the **Control Room (CR)** output, the **Monitor (Mon)** output and the **Main Mix**. To use the **Talkback** function, a hardware input must be assigned to the Talkback Input.

Mon Out always carries the monitor mix and **Mix Out** the main stereo mix. **CR (Control Room) Out** can be switched among the main mix and the monitor mix. In addition, Solo and Pre-Fader Listen functions operate via CR Out. Activation of the **Solo function** on any signal delivers this signal **directly to CR Out**. All signals for which **PFL is active** are delivered simultaneously to **CR Out** when the PFL function is activated in the Master section.

Link causes the left and right master faders to be connected so that both move together when one is adjusted, allowing the stereo balance of the mix to be maintained at all times while making adjustments to the master mix output level using the mouse.

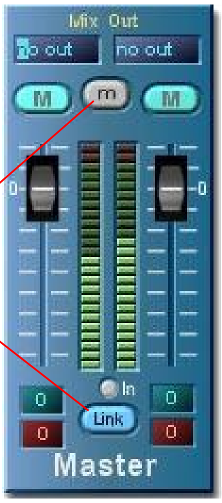


The Master section offers its own set of **four effect insert slots** for processing of the complete main mix. In contrast to the Mix channels, **effects used in the Master section must be full-stereo effects** (i.e., they must have stereo



Insert on/off

Insert pre/post-fader



Mix Out assign

Mute L

Mute R

Mono

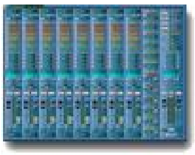
Signal level display

Link

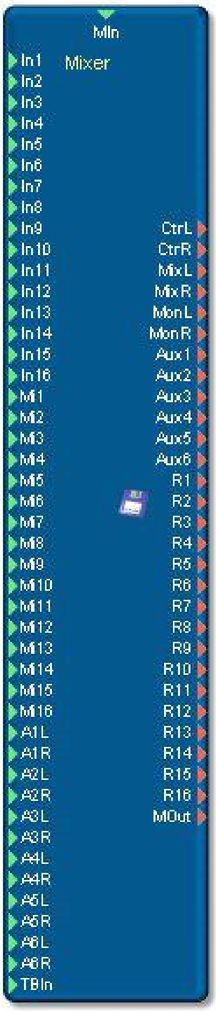
Fader position

Margin display

- Connections



Minimized (icon) representation



Midi In

In

1-16

Control Room Out

Mix Out

Monitor Out

Aux Send 1-6

Mix

1-16

Recording

Busses

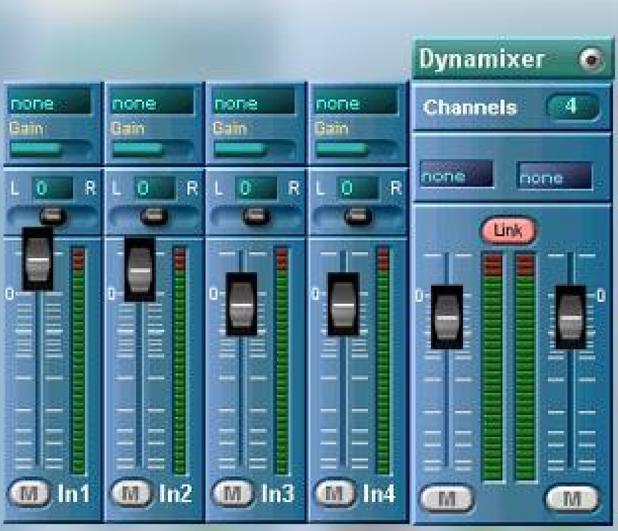
Aux Return 1-6

Midi Out

Talkback In

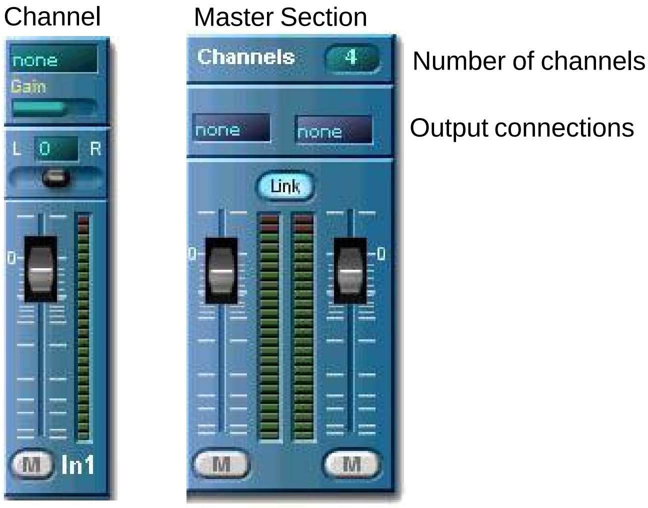
Dynamixer

- Overview
- Making Connections



- Overview

The **Dynamixer** is located in the Scope\Data\Mixer directory. It is a bare-bones, no-frills utility mixer. It uses the resources *very* sparingly. Not only is it constructed **without busses or aux paths**, but it also has **only the exact number of channels you need**, because *you* decide how many channels it should have – hence the name *Dyna*mixer. Whenever you just want to use two or more synths together and don't need the full functionality of the "big" SCOPE mixer, this is the mixer to use.

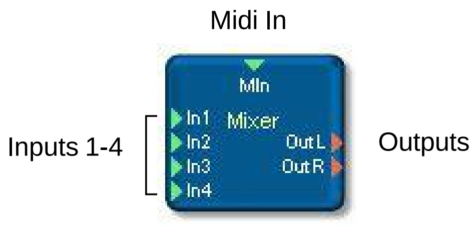


The text field at the top of each channel strip shows the **name of the connected input signal**. You can click in this field as if it were shown in the Rack window in order to make connections to devices which *are* represented in the Rack window.

The number of channels in the mixer is adjustable at any time via the **Channels** text fader in the upper right-hand corner. Channels are added or removed at the right (higher channel numbers) – the lower-numbered channels stay put. The module diagrams on the next page illustrate this for a 4-channel and 16-channel configuration.

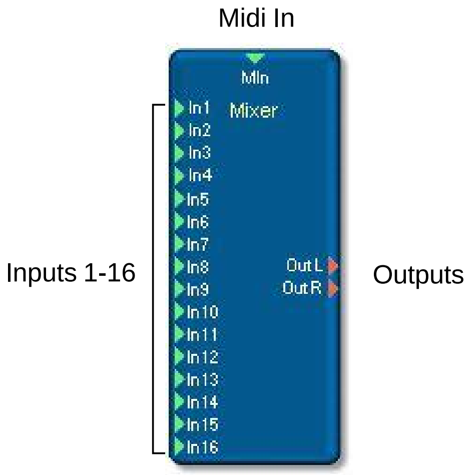
To move the Dynamixer surface around onscreen, click and drag on the "Dynamixer" tab at the top of the master section.

- Making Connections


Midi In

Inputs 1-4

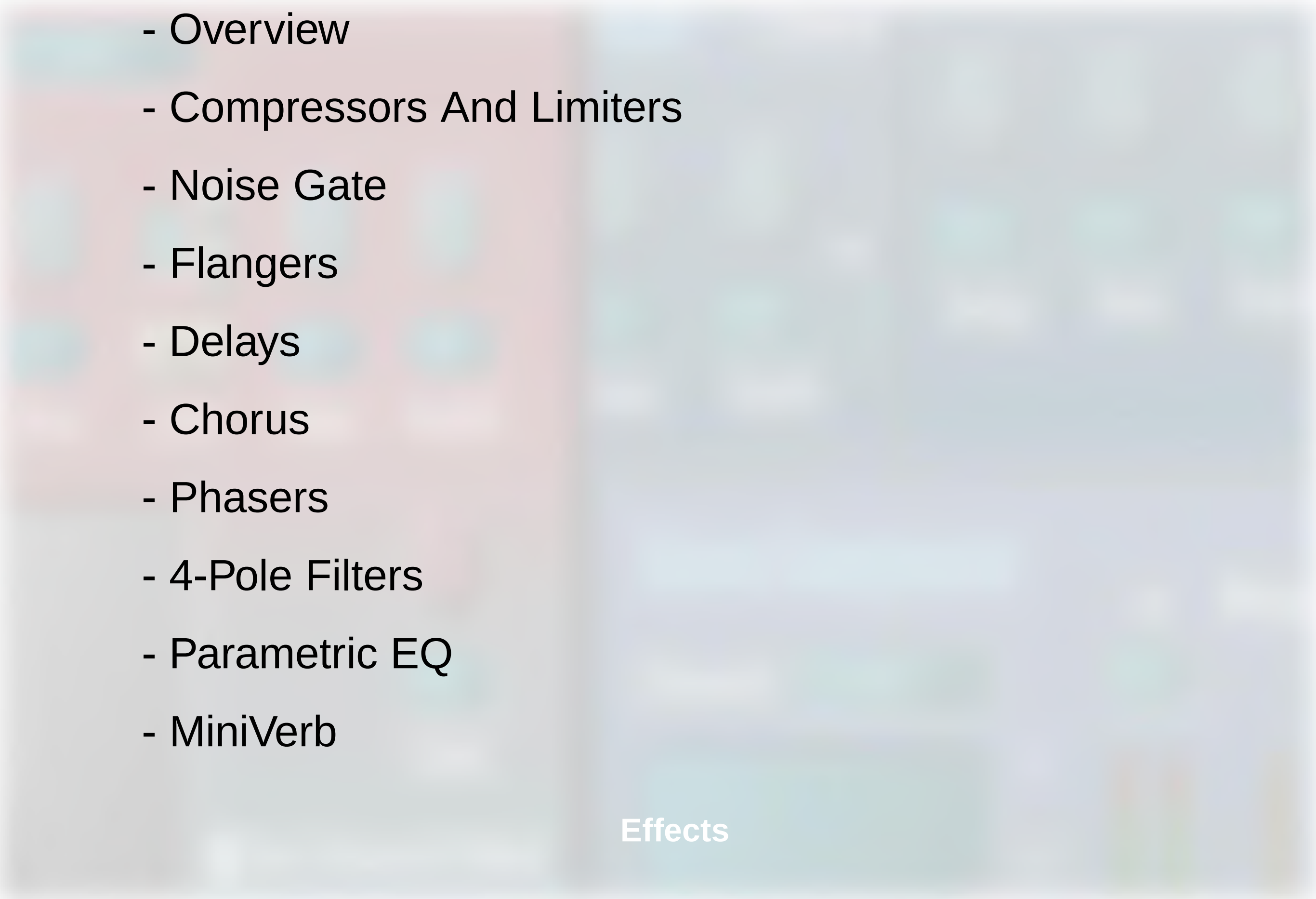
Outputs


Midi In

Inputs 1-16

Outputs

Effects

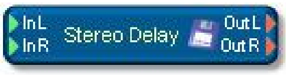


- Overview
- Compressors And Limiters
- Noise Gate
- Flangers
- Delays
- Chorus
- Phasers
- 4-Pole Filters
- Parametric EQ
- MiniVerb

Effects

This chapter describes SCOPE effects and shows you how to use them.

Effects devices can be **added to a project** via the Project window, using the usual method of drag-and-drop from the File Browser. An effects device is depicted in the Project window as a module with audio connections. By right-clicking on this module, you pop up a menu via which you can **open the control surface** of the effects device. Via the control surface, you can adjust all settings, **bypass** the effect or call up its **preset list**.

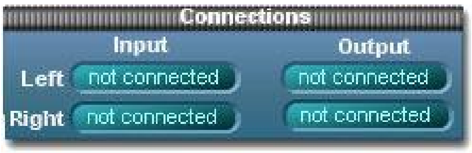


Insert And Aux Effects

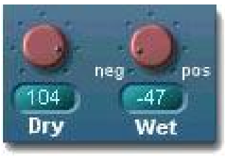
There are two main classes of SCOPE effects: **insert** effects and **aux** effects.

Insert effects are intended for direct insertion into the Big Mixer. Each one is provided in two varieties – mono for individual channel inserts, or stereo for main mix insertion. In addition to the usual Project window method of adding an effect to a project, hooking it up and opening its control surface, an insert effect can also be dragged from the File Browser and dropped directly into a mixer insert slot. Its control surface can then be opened by double-clicking on this insert slot. A display window on the surface indicates the mixer channel into which the associated effect is inserted.

Aux effects, as a group, feature relatively complex algorithms compared to those of the insert effects (and are therefore also more demanding of DSP capacity). They're not intended for insertion in several different mixer channels at once, but rather for singular use in a mixer aux path. Each aux effect has a **Connections** drawer which displays the connected signals by name. Cabling can also be done via this drawer – for example, click on the Input-Left window, and then on the Aux Send 1 window in the BigMixer aux section, to connect the mixer aux send to the effect input, etc.



With most effects, the amounts of input (**Dry**) and effect (**Wet**) signal sent to the output can be separately adjusted. The Wet control has its zero-output setting at the center of its rotation. Turning it further to the left again increases the Wet output amount, but with **inverted phase**.



Stereo effects can be applied to mono signal sources to produce a stereo output. To do this, activate the option **Use mono input (left)** and feed the source signal in via the left input.

External effects can also be used with SCOPE by connecting them to the card (via analog, S/P-DIF or ADAT I/Os as appropriate) and connecting them via the hardware I/O modules in the Project window.

- Compressors And Limiters

SCOPE includes a compressor and a limiter (each in mono and stereo versions). These are usually used as insert effects and thus can be found in ..\Data\Effects\Insert-Effects\

The compressor responds mainly to the average level of the input signal. The limiter focuses more on signal peaks. User controls are identical for both, as described below.

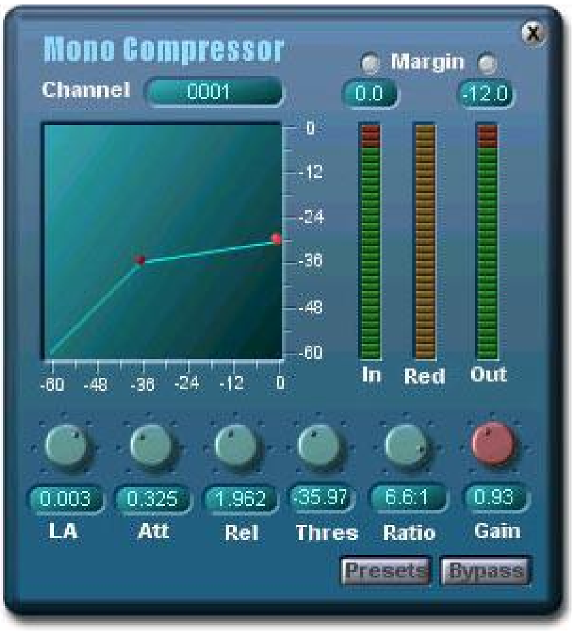
The individual parameters are:

LA (Look-Ahead): The compressor/limiter effectively looks into the future, evaluating the input signal ahead of the point where compression is actually taking place.

Att (Attack): Compressor/limiter reaction time. Allows the speed of response to input signal peaks (i.e., the rapidity of compression ratio increase) to be adjusted.

Rel (Release): Compressor/limiter "relaxation time". Controls the speed with which the compression ratio drops off after the input signal falls below the threshold level.

Thres (Threshold): Sets the input signal level above which compression (delayed by the Attack time) begins. When the input signal again falls below the threshold level, compression drops off (after Release time delay).



Ratio sets the actual compression ratio. **Gain** adjusts the output level.

- Noise Gate

The Noise Gate is full-featured but nevertheless requires relatively little DSP power. You can comfortably deploy multiple noise gates in several mixer channels at once. On the control surface, time values (LA, Att, Hold, Rel) are shown in msec, while level settings (Thres, Hyst, Floor, Gain) are in dB relative to the 0 dB maximum.

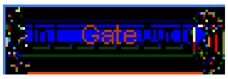
The individual parameters are:

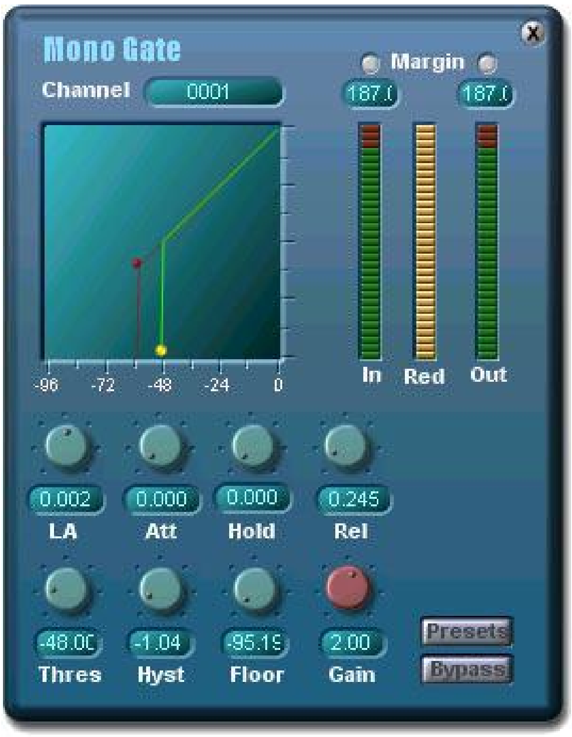
LA (Look-Ahead): The compressor/limiter effectively looks into the future, evaluating the input signal ahead of the point where compression is actually taking place. This setting is especially important with fast attack and release times.

Att (Attack): Amount of time for full opening of the gate once the input signal level has exceeded the turn-on (upper) threshold.

Hold: Minimum amount of time the gate will stay open once it is triggered (i.e., minimum delay between end of attack and start of release).

Rel (Release): Amount of time which gate takes to fully close once it begins to close (i.e., once the input signal level falls below the turn-off (lower) threshold, and after any remaining hold time has elapsed).





Thres (Threshold): Sets the input signal level in dB above which the gate will open. This is the turn-on (upper) threshold, shown in the display as a yellow point. The turn-off (lower) threshold is adjusted in tandem with the turn-on threshold, with an offset determined by the Hyst (hysteresis) control.

Hyst (Hysteresis): Adjusts the difference between the turn-on (upper) and turn-off (lower) thresholds. The turn-off threshold appears as a red point in the display. It can be set up to 10 dB below the turn-on threshold.

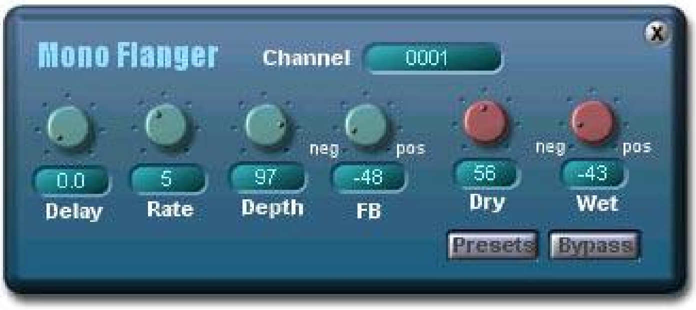
Floor: Sets the level to which gate gain drops when the gate closes. The gate can thus be set so that it does not close completely.

Gain adjusts the output signal level.

- Flangers

The SCOPE Flangers permit the use of extreme settings. The **Mono Flanger** is an insert effect, while the **Stereo Flanger** and **Cross Flanger** are aux effects.

Operationally, there are only minor differences between the various flangers. The controls include:



Channel 0001
Delay 0.0
Rate 5
Depth 97
FB -48
Dry 56
Wet -43

Delay: Sets the internal delay which produces the basic effect. (The effect itself is not delayed.) Lower delay values have the greatest effect upon low frequencies. With longer delays, high frequencies are more strongly affected.

Rate: Controls the rate of delay modulation.

Depth: Controls the amount of delay modulation.

FB (Feedback): Controls the amount of effect (i.e., delayed) signal which is fed back to the flanger input. For the **Stereo Flanger**, which has separate left and right delays, feedback is from the left delay output to the left input and from the right delay output to the right input – the Stereo Flanger is essentially two separate flangers in one. The **Cross Flanger** is basically a Stereo Flanger which uses cross-coupled feedback connections.

L/R Phase 57

L/R Phase: An additional control on the Stereo and Cross flangers which allows adjustment of the phase difference between the delay modulation of the left side and that of the right side.

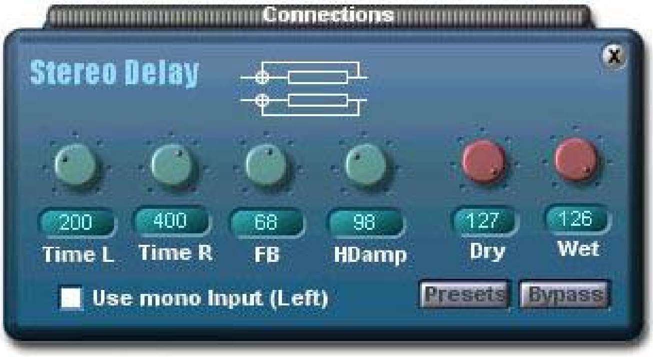
- Delays

SCOPE offers three Delay effects. The **Mono Delay** is an insert effect. The **Stereo Delay** and **Cross Delay** are stereo aux effects, with most controls affecting both left and right delay lines in common.

Operationally, there are no differences between the various delay effects. The controls include:

Time: Sets the delay time in milliseconds. The Stereo and Cross Delays offer separate left and right delay time controls.

FB (Feedback): Controls the amount of effect (i.e., delayed) signal which is fed back to the delay input, and thus, the rate at which the loudness of the echoes diminishes. For the **Stereo Delay**, which has separate left and right delays, feedback is from the left delay output to the left input and from the right delay output to the right input – the Stereo Delay is essentially two separate delay lines in one. The **Cross Delay** is basically a Stereo Delay which uses cross-coupled feedback connections.



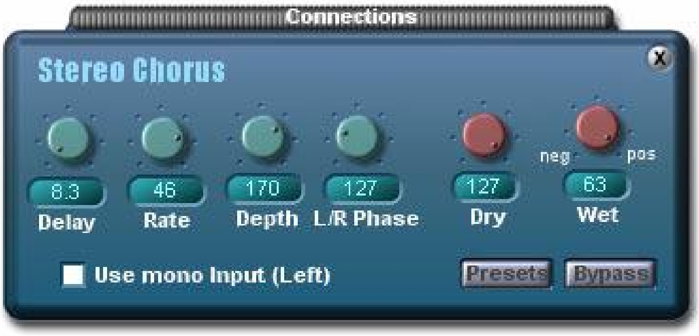
200 Time L
400 Time R
68 FB
98 HDamp
127 Dry
126 Wet
□ Use mono Input (Left)

HDamp (High-frequency Damping): attenuates high frequencies in the feedback paths, causing echoes to become progressively "softer".

- Chorus

SCOPE offers a **Stereo Chorus** as an aux effect. In essence, it is a Stereo Flanger with modified algorithms and longer delay times. The controls include:

Delay: Sets the internal delay which produces the basic effect. (The effect itself is not delayed.) Lower delay values have the greatest effect upon low frequencies. With longer delays, high frequencies are more strongly affected.



8.3 Delay
46 Rate
170 Depth
127 L/R Phase
127 Dry
63 Wet
□ Use mono Input (Left)

Rate: Controls the rate of delay modulation.

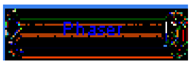
Depth: Controls the amount of delay modulation.

FB (Feedback): Controls the amount of effect (i.e., delayed) signal which is fed back to the chorus input.

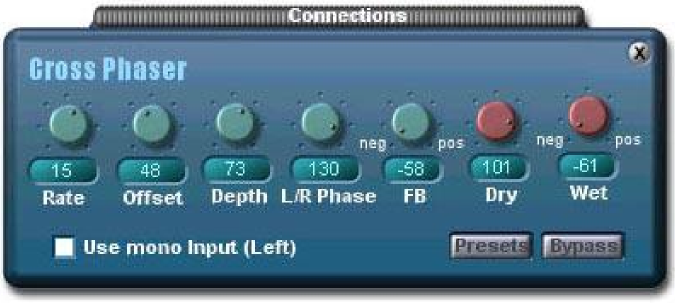
L/R Phase: Allows adjustment of the phase difference between the delay modulation of the left side and that of the right side.

- Phasers

The SCOPE Phasers are functionally similar to the flangers. The **Mono Phaser** is an insert effect, while the **Stereo Phaser** and **Cross Phaser** are aux effects.



Operationally, there are only minor differences between the various phasers. The controls include:



Rate 15
Offset 48
Depth 73
L/R Phase 130
FB -58
Dry 101
Wet -61
□ Use mono Input (Left)

Rate: Controls the rate of effect modulation.

Offset: Sets the internal delay which produces the basic effect. (The effect itself is not delayed.) Lower offset values have the greatest effect upon low frequencies. With higher settings, high frequencies are more strongly affected.

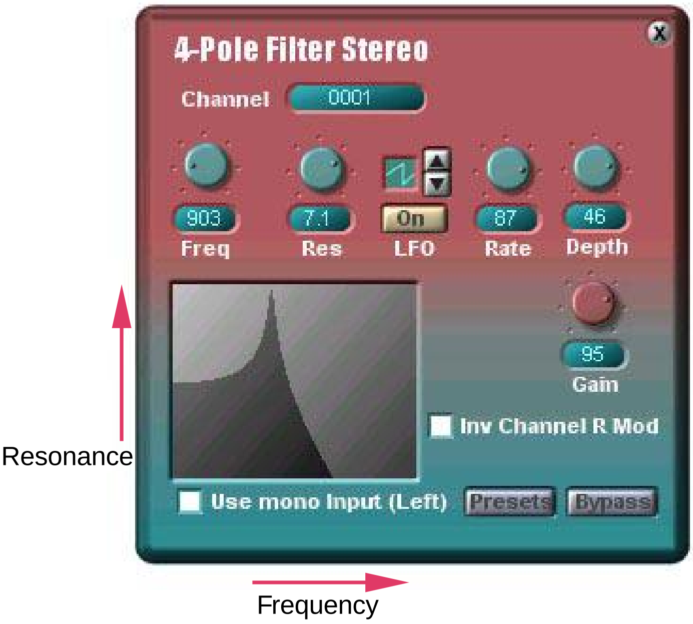
Depth: Controls the amount of effect modulation.

L/R Phase: An additional control on the Stereo and Cross phasers which allows adjustment of the phase difference between the effect modulation of the left side and that of the right side.

FB (Feedback): Controls the amount of effect output signal which is fed back to the phaser input. For the **Stereo Phaser**, which has separate left and right sections, feedback is from the left effect output to the left input and from the right effect output to the right input – the Stereo Phaser is essentially two separate phasers in one. The **Cross Phaser** is basically a Stereo Phaser which uses cross-coupled feedback connections.

- 4-Pole Filters

There are two versions of the **4-Pole Filter**: a mono-insert version and a stereo-insert version. Both are adjustable Q low-pass filters with identical functionality.



Channel 0001
Freq 903
Res 7.1
LFO On
Rate 87
Depth 46
Gain 95
□ Inv Channel R Mod
□ Use mono Input (Left)

The filter can be quickly and intuitively edited via the graphical display as indicated at right. Use the left mouse button with horizontal movements to adjust the low-pass cutoff frequency (**Freq**) and the right mouse button with vertical movements to adjust the resonance (**Res**). You can also adjust the filter via the Freq and Res controls, which additionally allow you to make finer adjustments.

An **LFO** is available for modulation of the cutoff frequency. It has **Rate** and **Depth** controls and can be switched in or out via the **LFO** switch. The **LFO waveform** can be selected via the arrow buttons or directly in the waveform display field next to the buttons.

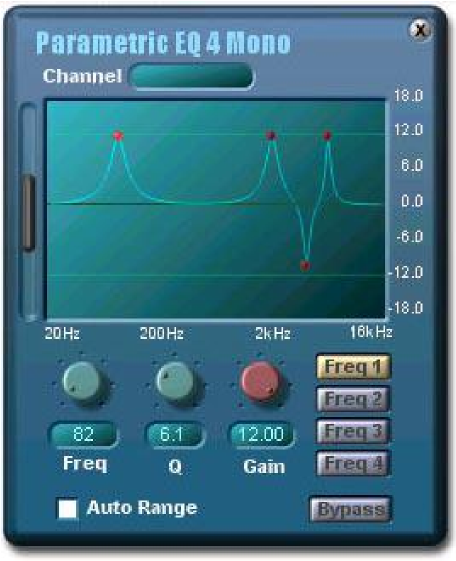
- Parametric EQ

The Parametric EQ is an insert effect. It is functionally identical to the EQ found in each channel of the Big Mixer.

The Parametric EQ permits up to four full-parametric filters (filter bands) at one time. These are represented by small red points in the EQ graphic window. New filters can be added (up to the limit of four) by double-clicking in this window.

The **center frequency** (**Freq**) of each filter can be changed by clicking and dragging with the left mouse button on the corresponding point in the graph and moving it sideways. By moving it up or down, you change the filter **gain**. Upward or downward dragging with a *right*-button mouse click adjusts the filter **Q**.

The same adjustments can be made via the corresponding knobs or via direct entry of numerical values in the associated number fields. The **Freq 1-4** buttons offer an alternative means of selecting individual bands.

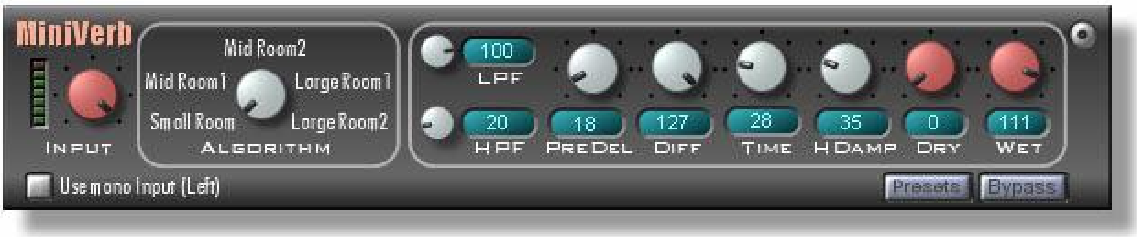


Freq 82
Q 6.1
Gain 12.00
□ Auto Range

The **zoom bar** can be made smaller by clicking and dragging on its ends, thus zooming the graphical display in for a closer view. You can then move the zoom bar by clicking and dragging on its middle to zero in on the desired display range. Double-clicking on the zoom bar toggles between the magnified and full-range views. With **Auto Range** enabled, the display range is selected automatically.

The **bypass** button defeats the EQ. This also takes the EQ out of the DSP circuit, reducing DSP loading accordingly. (Likewise, each deletion of an unused filter band reduces the DSP load slightly.)

- MiniVerb



LPF 100
HPF 20
PREDEL 18
DIFF 127
TIME 28
HDAMP 35
DRY 0
WET 111
□ Use mono Input (Left)

The MiniVerb is a reverberation module designed to be used as an Auxilliary effect. With the MiniVerb we have intended to reach a compromise whereby sound quality is maintained without excessive computational overhead (i.e. DSP usage).

The Input control adjusts the input sensitivity. The LED display provides a visual indicator of the input level.

The Algorithm rotary switch selects a program from among the various room simulation algorithms available.

The LPF and **HPF** settings adjust the low- and high-pass filtering of the signal to control the frequency spectrum, or tone color, of the result.

PREDEL (pre-delay) is used to adjust the time delay before the initial reverberation response.

DIFF controls the Diffusion of the reverberated signal.

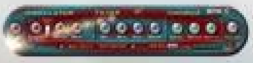
TIME sets the overall duration of the reverb response.

The HDAMP setting controls the rate at which high frequencies are dampened (die out) over time.

The DRY and **WET** setting adjusts the relative balance of the effect and the original signal. When using the MiniVerb as an Aux effect DRY should be set to 0, and WET to 127.

Synthesizer

BlueSynth



EZ Synth

INFERNO

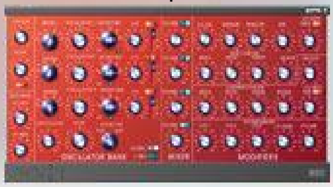
miniscope

Modular Synth

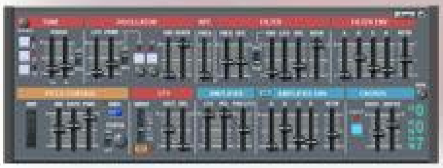
Synthesizer

FM One

miniscope MKII

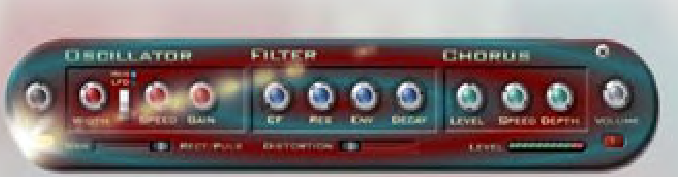


U KNOW 007

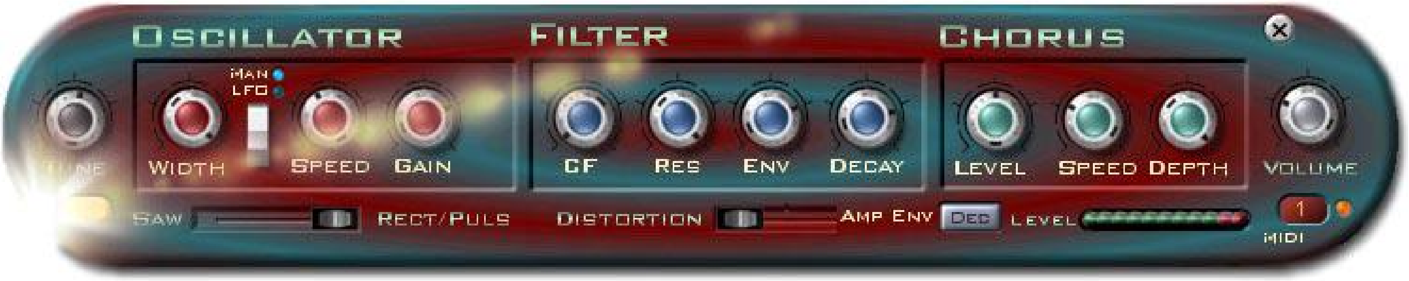


EZ Synth

- Introduction And Overview
- EZ Synth Controls
- Connections



- Introduction And Overview



Introduction

True to its name, the EZ Synth is easy to understand and use. Small and simple, it's a great choice for getting started with SCOPE synths.

Despite its simplicity, it's also easy to get good, useful sounds from the EZ Synth. Naturally, it can't match the Modular Synth or the FM Synth for range or complexity. But just for this reason, it's an "economical" unit which doesn't use up much of the SCOPE card's computing capacity. This means it'll continue to be useful to you even as an advanced SCOPE user.

Overview

The EZ Synth consists of three main sections: Oscillator, Filter, and Chorus. The front panel layout makes this clearly visible. The oscillator generates the sound. The filter emphasizes and/or removes specific frequency components of the sound. The chorus effect "fattens" the sound, making it more animated, and in the process, converts it from mono to stereo, adding a "motion" component. There's also a distortion effect to add "edge".

Two Important Things To Know

EZ Synth has a **power switch** -- you have to turn it on! It's the little button at far left, hidden in its own glare.

The **MIDI Channel** is selectable via the small window at far right. Just put the mouse cursor there, hold down the left mouse button and roll the mouse up and down to change channels. Default is Omni (all channels), so you don't need to set it unless you're also using other synths.

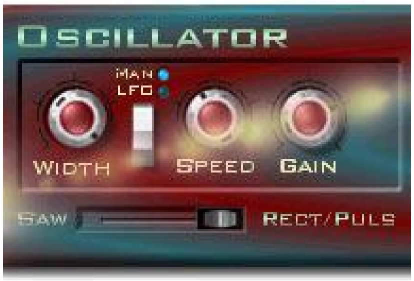
By now you should be hearing something! Read on ....

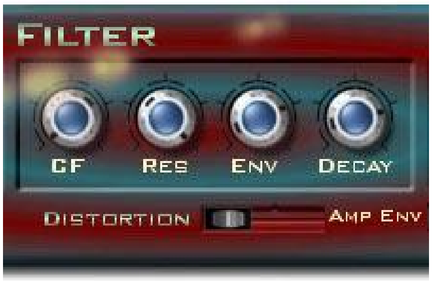
- EZ Synth Controls

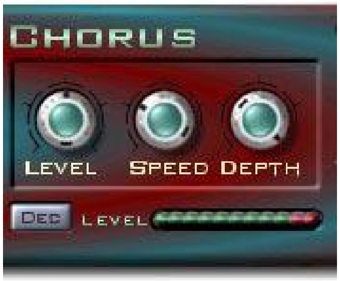
The **Oscillator** puts out either **Sawtooth** or **Pulse** waves, as selected via the slide switch at bottom. When the pulse waveform is selected, the **Man/LFO** switch selects whether the pulse width is set manually via the **Width** knob (switch up, “Man“ LED lit) or varies automatically with time under control of the oscillator **LFO** (down, “LFO“ LED lit). The oscillator LFO is in turn controlled via the **Speed** and **Gain** (LFO depth) knobs.

The **Filter** is a low-pass resonant type. Its response has a “peak“ around the cutoff frequency (**CF**) which becomes stronger as the resonance (**Res**) setting is increased. The filter envelope is triggered every time a note is played. The **Env** knob adjusts the amount of effect it has on the filter cutoff frequency. The **Decay** knob controls the envelope decay rate. The filter envelope is actually an ADSR envelope, but with preset attack, sustain and release settings -- to keep things EZ.

The amount of **Chorus** effect is adjusted by the **Level** control. The **Depth** control adjusts the amount of modulation of the effect, while the **Speed** knob adjusts the rate at which the modulation occurs.

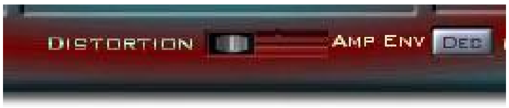






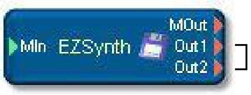
The **Tune** knob which is almost lost in the glare of the power switch at far left can shift the oscillator pitch up or down over a range of two octaves. The oscillator pitch can be easily restored to its normal non-detuned setting by double-clicking on this knob.

The **Distortion** fader (bottom center) controls the amount of distortion applied to the sound. Turning up the distortion will also raise the overall signal level. The actual synth output level can be monitored via the **Level** display (lower right) and adjusted using – naturally enough – the Volume knob.



The **Amp Env** button gives you the choice of two amplifier envelope modes. With **Dec**, the filter Decay control also sets the amp envelope decay time. The Sus setting sets the amp envelope into Attack-Sustain mode.

- Connections

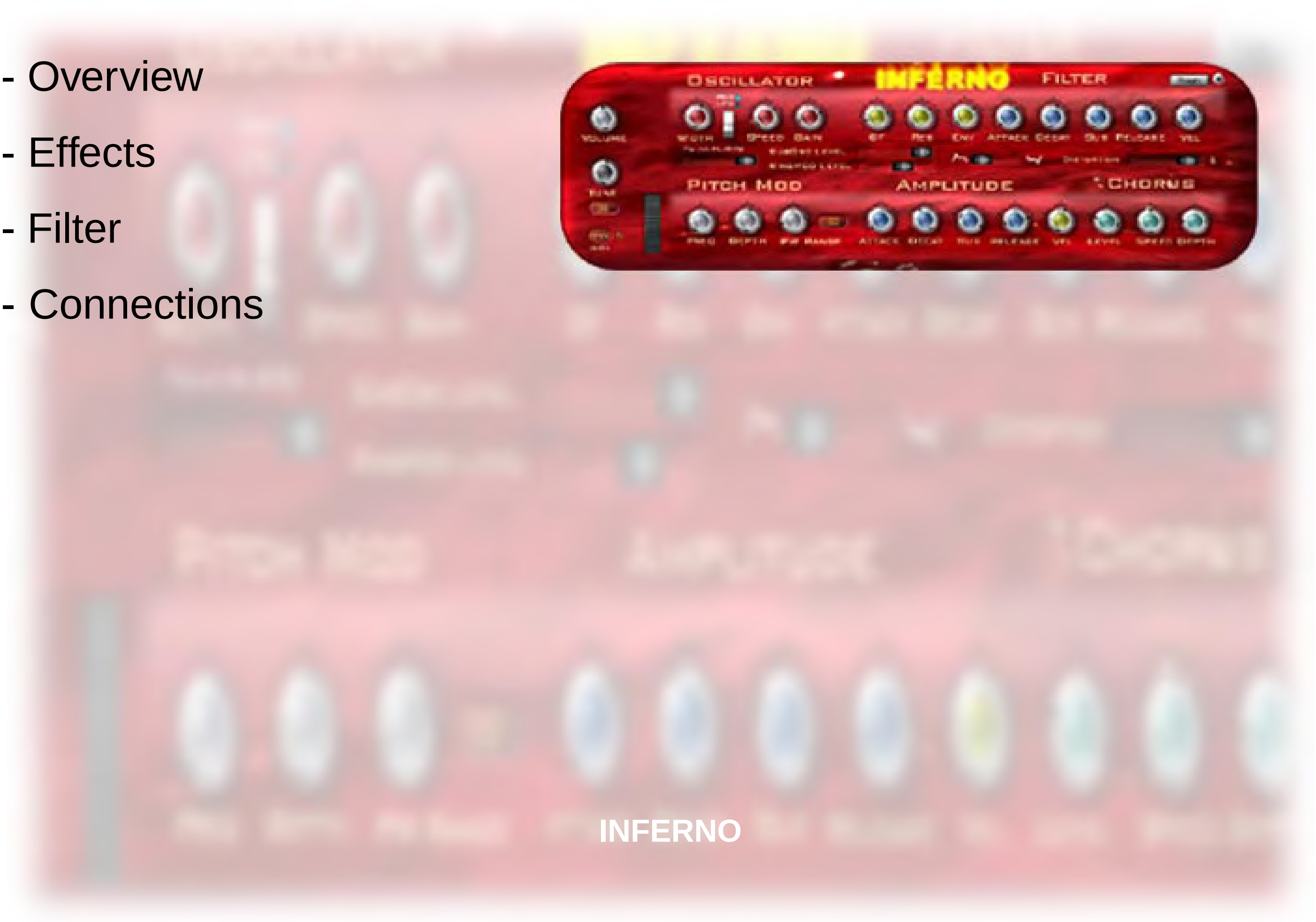


Project window (module) representation

Minimized (icon) representation

Inferno

- Overview
- Effects
- Filter
- Connections



- Overview

The **Inferno** has one oscillator with suboscillator, one filter, and effects (chorus, ring modulator, distortion). The suboscillator has a fixed square-wave waveform and is always tuned one octave below the main oscillator.

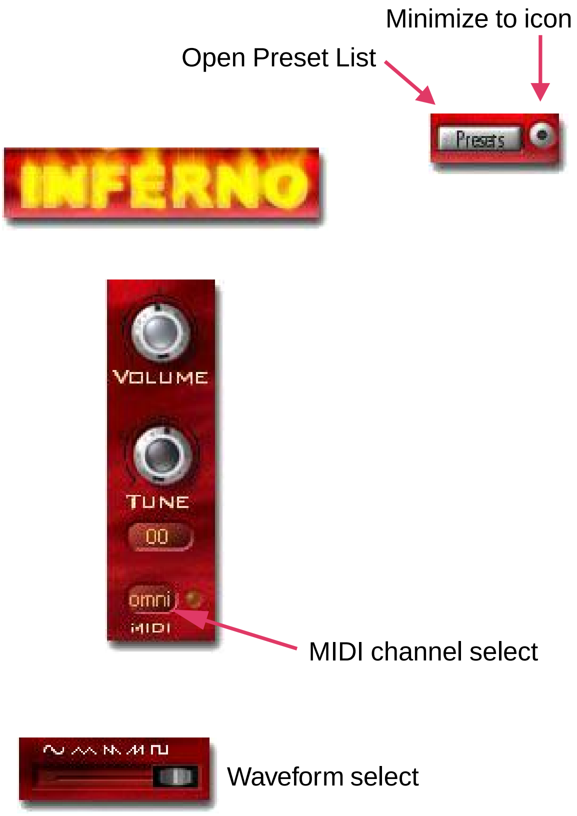
The animated **flaming INFERNO label** consumes a bit of computing time. By clicking on it, you can deactivate the animation.

The **Volume** knob controls the overall synth volume. When the Inferno is used polyphonically, and especially if the distortion effect is also used, the synthesizer output can overload. The Volume control can be used to correct this problem.

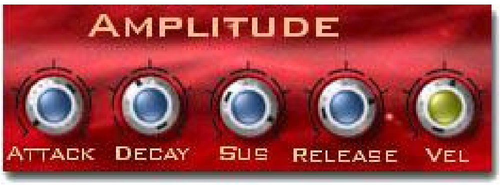
Tune adjusts oscillator pitch up to two octaves up or down. Double-clicking on this control sets it back to its center position (no detune).

The **MIDI channel** is selected via the small text fader window at bottom left on the surface.

The **Waveform** slide switch selects the main oscillator waveform (sine, triangle, sawtooth down, sawtooth up and square).

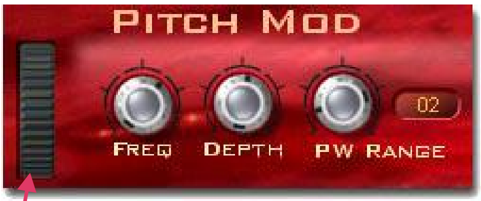


The **Amp Envelope** is a simple ADSR envelope which affects both oscillators in common. The yellow **Vel** (velocity) control adjusts the effect of note velocity on amplitude.



Amp envelope

The **Mod Wheel** is an onscreen version of the control found on practically every keyboard and/or controller. Use it when you can't (or don't want to) use an external one. Just like them, it's assigned to **MIDI controller 1** -- and correspondingly, **it will move on its own** to follow incoming MIDI Mod Wheel messages, if you *are* using an external one. The Mod Wheel **affects oscillator pitch** via an LFO whose speed is set by the **Freq** control. **Depth** adjusts the amount of modulation.



Pitch modulation

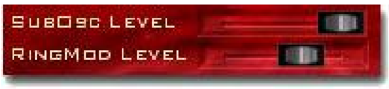
Mod wheel

The **PW Range** control sets the range of oscillator pitch response to incoming MIDI Pitch Bend messages. You can adjust the knob or enter values directly into the text field.

SubOsc Level controls the level of the suboscillator.

The **Ring Modulator** multiplies the oscillator and suboscillator signals. **RingMod Level** controls the amount of this signal product which is fed to the output of the Inferno.

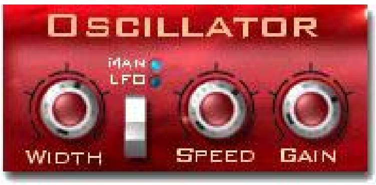
Suboscillator level



Ring modulator level

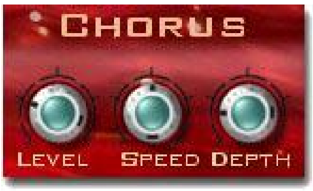
- Effects

When the pulse waveform is selected for the oscillator, the **Man/LFO** switch selects whether the pulse width is set manually via the **Width** knob (switch up, “Man“ LED lit) or varies automatically with time under control of the oscillator **LFO** (down, “LFO“ LED lit). The oscillator LFO is in turn controlled via the **Speed** and **Gain** (LFO depth) knobs.

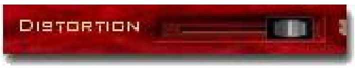


Pulse-width modulation controls

The **Chorus** is a stereo-out effect, which is why Inferno has two outputs. The amount of chorus effect is adjusted by the **Level** control. The **Depth** control adjusts the amount of modulation of the effect, while the **Speed** knob adjusts the rate at which the modulation occurs.



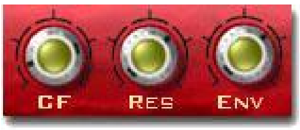
The **Distortion** fader controls the amount of distortion applied to the overall sound. Turning up the distortion will also raise the overall signal level. At extreme settings, you may need to turn the Inferno main volume control down to avoid output overloading.

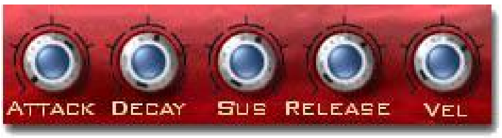


- Filter

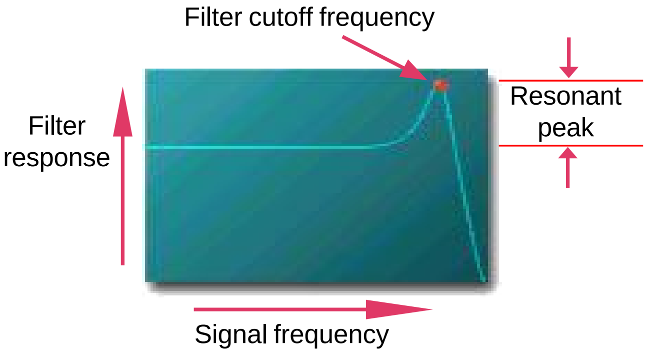
The **Filter** is a low-pass resonant type. Its response has a "peak" around the cutoff frequency (**CF**) which becomes stronger as the resonance (**Res**) setting is increased. The filter envelope is triggered every time a note is played. The **Env** knob adjusts the amount of effect it has on the filter cutoff frequency.

The filter envelope is an ADSR type with its own set of controls. The **Vel** (velocity) control adjusts the effect of note velocity on filter envelope depth.

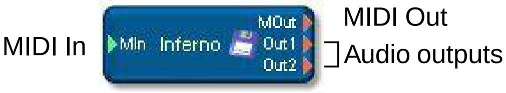




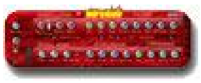
Filter envelope controls



- Connections



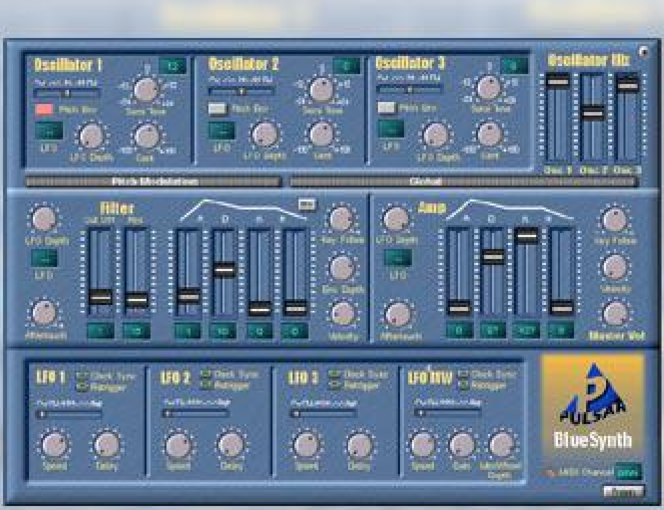
Project window (module) representation



Minimized (icon) representation

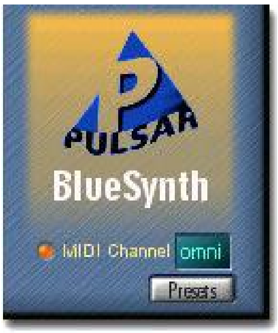
BlueSynth

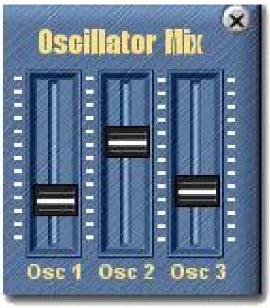
- Oscillator Mix / Amp
- Oscillators / LFOs
- Filter
- Pitch Envelope
- Global
- Connections

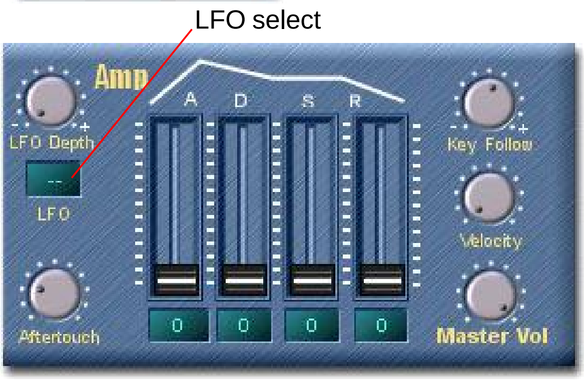


BlueSynth

- Oscillator Mix / Amp







The BlueSynth is an easy-to-use polyphonic synthesizer with three oscillators and four assignable LFOs plus a filter with its own ADSR envelope. The filter and the amp envelope can additionally be modulated by one of the four LFOs.

At bottom right is a small window where the **MIDI channel** can be dialed in. Next to this is a **MIDI activity indicator**.

In the **Oscillator Mix** section, the relative loudness of the three oscillators can be adjusted.

Master Vol is the main volume control for the BlueSynth.

The amplitude can optionally be additionally modulated by one of the LFOs, as selected in the small LFO window at right – just click and hold the left mouse button over this field and move the mouse up and down, then release the button when the number of the desired LFO appears. The **LFO Depth** knob controls the amount of LFO effect on amplitude.

The **Aftertouch** control adjusts the amount of effect which MIDI Aftertouch messages will have upon volume.

Key Follow causes the envelope times to vary according to the keyboard position from which a note is played. Positive settings cause the envelope to speed up as you go higher up the keyboard (similarly to the way that high notes on a piano decay more quickly than low notes). Negative settings produce the reverse effect. Via the **Velocity** control, the envelope depth can be affected to varying degrees by the velocity with which a note is played.

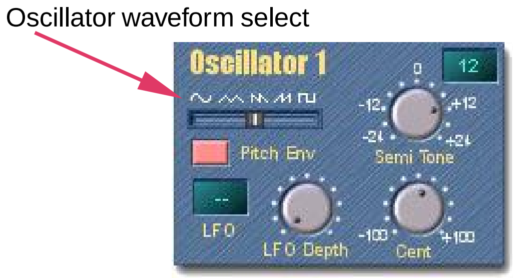
- Oscillators / LFOs

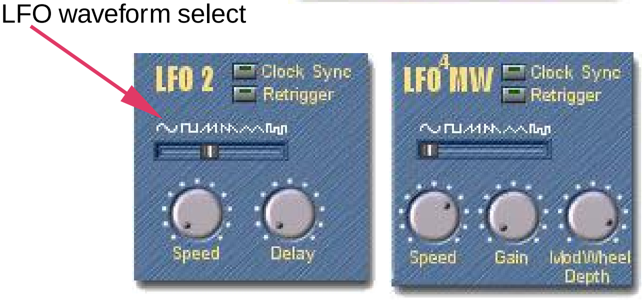
The three **oscillators** are identical. Available waveforms include sine, triangle, sawtooth down, sawtooth up and square. These can be selected independently for each oscillator. Likewise, each oscillator can be independently tuned via the **Semi Tone** control (“coarse“ control: 24 semitones up or down) and the **Cent** control (“fine“ control: 100 cents, or one semitone, up or down).

Each oscillator can be independently modulated by a pitch envelope (when the **Pitch Env** button is pressed) as well as by one of the four LFOs. Controls are provided for for **LFO selection** and **LFO Depth**.

Each of the four **LFO**s (Low Frequency Oscillators) has its own set of controls. Waveforms include sine, square wave, sawtooth up, sawtooth down, triangle and sample/hold (random). The **Speed** knob sets LFO frequency. LFOs can be set to kick in a short time after each new note, adjustable via the **Delay** knob. LFO 4, also known as **LFO MW**, lacks the delay but can be depth-modulated by MIDI Mod Wheel messages. **Mod Wheel Depth** adjusts this modulation, while **Gain** controls the LFO output.

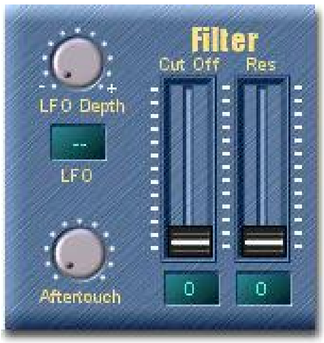
Normally the LFOs “free run“, whether notes are played or not. If **Retrigger** is turned on for an LFO, it is restarted with each new note, providing a consistent relationship between the LFO and the note start. This also lets you "free play" the LFOs in time with a rhythm. The corresponding LFOs in previous notes are not affected – a rhythmically useful side effect which lets you play the LFOs in multiple notes against one another.

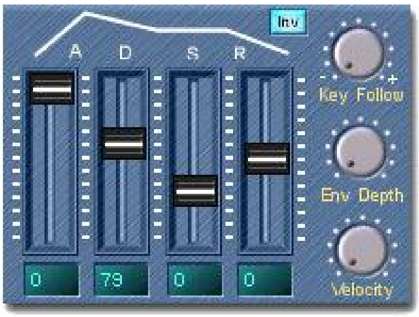




Activating **Clock Sync** causes the LFO waveform to synchronize to a MIDI clock (from a sequencer program or coming in on the SCOPE MIDI Input). The timing is determined by the **MIDI Clock Divider** setting. A separate setting for each LFO is provided in the Global drawer.

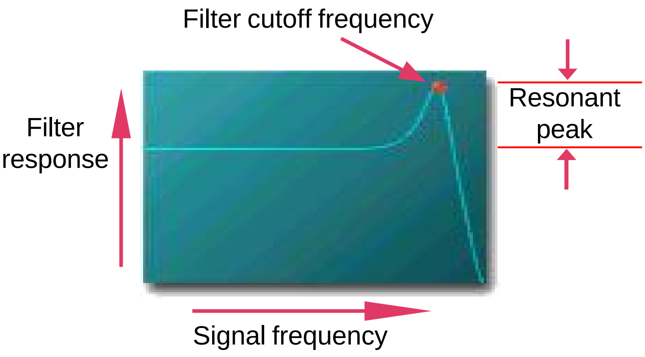
- Filter



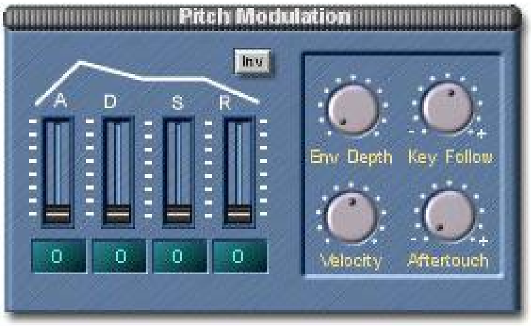


The low-pass **Filter** on the output of the BlueSynth is fully resonance-capable. The left fader controls the filter cutoff frequency, while the right fader controls filter resonance. Raising the Resonance setting increases the filter response in the vicinity of the cutoff frequency. At extreme resonance settings, the filter can itself begin oscillating.

The **cutoff frequency** can be modulated via an **LFO**, via MIDI **Aftertouch** messages and via the filter **envelope**. The **Env Depth** control sets the amount of effect the filter envelope has upon the filter. **Key Follow** and **Velocity** affect the filter envelope in the same ways as described earlier in the Amp section. The **Inv** button inverts the filter envelope.



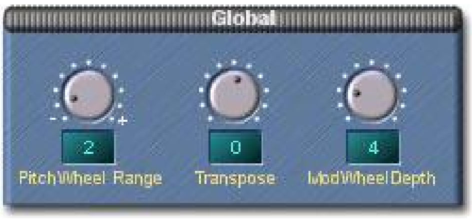
- Pitch Envelope



The controls for the Pitch Envelope, which modulates the pitches of the oscillators, are found in a slide-out drawer which opens up when you click on its visible edge.

Although the Pitch Envelope controls have a somewhat different layout, they operate in relation to oscillator pitch in exactly the same ways as the filter or amp envelope controls described previously.

The **Global** drawer contains controls which affect the basic tuning of the BlueSynth and its response to a couple of specific MIDI message types.



The **Pitch Wheel Range** control sets the maximum pitch bend amount (in semitones) up or down. This maximum bend is obtained when the (external) pitch wheel is deflected to its full limit in one direction or the other.

The **Transpose** control adjusts the overall pitch of the BlueSynth up or down in full semitones. As usual in SCOPE, you can double-click on this control to send it directly back to its default setting of zero.

The **Mod Wheel Depth** control regulates the depth of response to MIDI Mod Wheel input for BlueSynth controls assigned to this message.

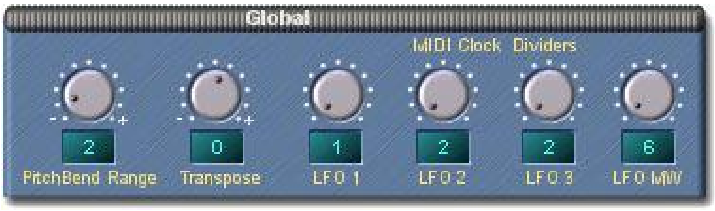
- Global

The **Global** drawer contains controls which affect the basic tuning of the BlueSynth and its response to a couple of specific MIDI message types.

The **Pitch Bend Range** control sets the maximum pitch bend amount (in semitones) up or down. This maximum bend is obtained when the (external/MIDI) pitch wheel is deflected to its full limit in one direction or the other.

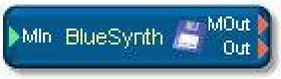
The **Transpose** control adjusts the overall pitch of the BlueSynth up or down in full semitones. As usual in SCOPE, you can double-click on this control to send it directly back to its default setting of zero.

The **MIDI Clock Dividers** determine the speeds of the individual LFOs when operated in Clock Sync mode. The 24 PPQN (Pulses Per Quarter Note) MIDI clock stream (from a sequencer program or external device) is divided by the selected value to obtain the rate at which the LFO waveform repeats.



- Connections

MIDI In



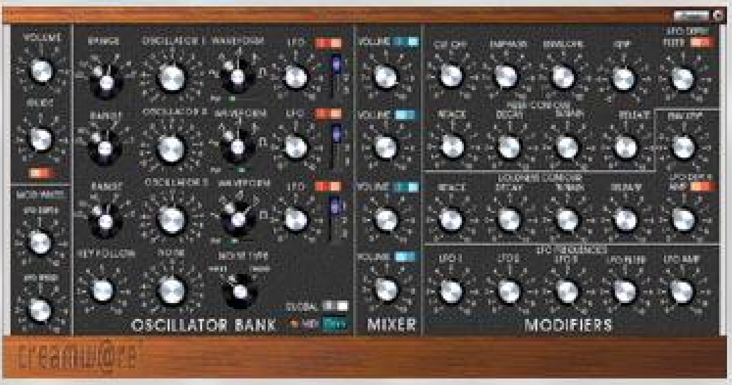
MIDI Out
Audio output

Project window (module) representation

Minimized (icon) representation

miniscope

- Overview
- Mixer / Oscillators / Noise
- Mod Wheel / Glide
- Modifiers
- Global
- Connections



- Overview

The **miniscope** is a reasonably faithful reproduction of an old-timey, monophonic, no-presets, 100 percent analog synth – just like the one your parents always had in their living room. We've hauled it down from the attic, dusted it off, given it a bit of a face-lift (but you'll still recognize it) and added a couple of things that it never had before. Step right up and start twiddling those knobs!

What It's Got

Three oscillators whose pitch and waveform are independently adjustable, a **glide** function, a **noise generator** producing full-spectrum or tuned noise, a **mixer** section for the oscillators and noise generator, an **amp envelope**, a **resonant low-pass filter** with its own envelope, and **six sinewave LFOs** – one for each oscillator, one for all three, one each for filter and amp.

What Else It's Got

MIDI control including – of course – note play, plus velocity sensitivty, pitch bend and mod wheel response, and the ability to **assign MIDI controllers to all controls** on the front panel. Adjustable **polyphony**. Nameable **presets** – however many you want – which **store and recall all front panel settings**. Always in tune (if you want it that way), no hum or noise (unless you *want* hum and noise), weighs nothing, requires no keyboard stand. And, if one miniscope happens not to be enough – just add another one into your project.

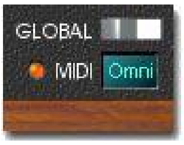
The Most Important Things To Know

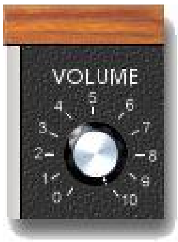
The **main volume control** is in the upper left corner, just where you *wouldn't* expect to find it.

The **MIDI channel** "text fader" control is front-row center. Click *and hold* with the left mouse button *directly* on the value, then move the mouse up and down to change it.

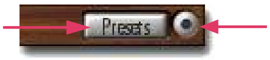
The **Preset List** is opened/closed via the button at top right (next to the **Minimize** button, which hides the synth).

The **Global** switch opens/closes a control panel which is discussed later in this chapter.



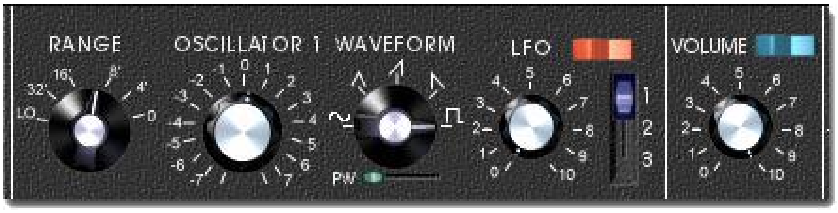


Show/hide Preset List



Minimize

- Mixer / Oscillators / Noise



Although visually separate, the Oscillator Bank and Mixer sections, together with the Mod Wheel and Glide controls, form a single functional unit which is described here in full.

Mixer

The first thing to note is that each oscillator has its own **Volume** knob in the Mixer section – nice for balancing the oscillators against one another – and **its own On/Off switch (blue) next to the volume knob**. Make sure the switch is on (flipped right) *and* the volume turned up before you start adjusting the other controls of an oscillator!

Oscillators

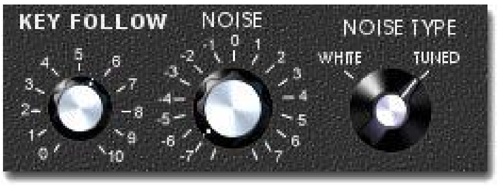
There are three oscillators. They are identical to one another and can be adjusted completely independently of one another. Oscillator **pitch** can be set in whole octaves via the **Range** selector at left and fine-tuned via the **Oscillator 1 - Oscillator 3** knobs.

The **Waveform** selector lets you dial in sine, triangle, sawtooth up, sawtooth down and pulse waveforms. The **PW** (Pulse Width) slide control directly below the waveform selector affects the pulse waveform. Full left produces a square wave (symmetrical pulse width). Moving the slider to the right causes the pulse width to become progressively narrower.

Oscillator LFOs

Oscillator pitch can be modified by any one of sinewave LFOs 1-3 as selected by the **LFO** slide switch. The **LFO** knob controls the amount of effect produced by the LFO. Note the **LFO On/Off switch**, which must be flipped to the right in order for the LFO to have any effect here. Assign a different LFO to each oscillator to get nice fat chorus-like sounds (with small LFO amounts) or extreme sound effects (with large LFO amounts). The speed controls for these LFOs are in the Modifiers section.

Noise

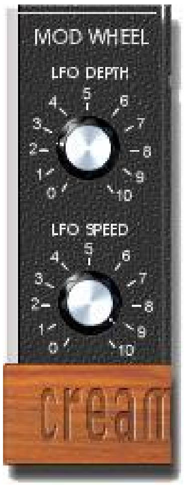


Included with the oscillators is a noise generator. Like the oscillators, it has its own **Volume** knob and **On/Off** switch in the Mixer section. It can produce plain white noise, or "tuned" noise which has been passed through a dedicated filter.

The **Noise** knob tunes the filter – it has no effect on white noise. The **Key Follow** control likewise affects only tuned noise, since it adjusts the tracking of the filter tuning in response to keyboard position. Mid-scale settings produce "normal" oscillator-like pitch tracking.

- Mod Wheel / Glide

Mod Wheel



Best for vibrato effects is the **Mod Wheel LFO**, which affects the pitches of all oscillators in common. Its controls are located in the lower left corner of the miniscope.

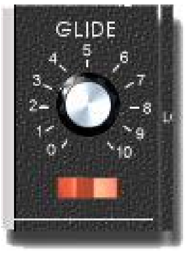
The amount of this LFO is controlled by **MIDI Mod Wheel messages**, so the effects of the settings described below will be seen only when you move your keyboard's Mod Wheel away from its zero position.

(For test purposes when editing presets, or in general if you find it more convenient, you can use the Mod Wheel control in the Global panel as an **onscreen substitute mod wheel** in place of an external physical mod wheel.)

The **LFO Depth** control here actually adjusts the sensitivity to mod wheel messages and **must be set higher than zero** for mod wheel motions to have any effect. The Mod Wheel **LFO Speed** control affects *only* this LFO.

The Mod Wheel LFO **affects only the oscillators**, but is completely independent from the individual oscillator LFOs. It can be used together with them or by itself, and is not affected by the individual oscillator LFO On/Off switches.

Glide



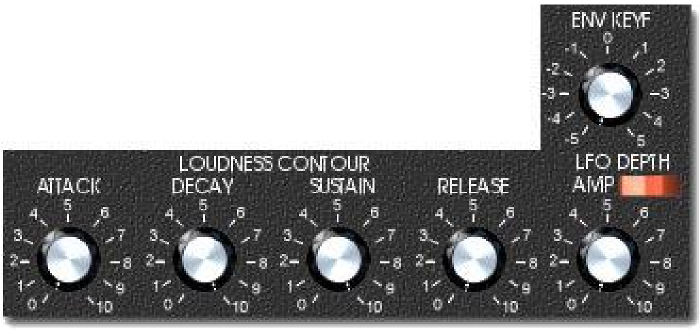
The Glide controls, tucked away off to the left below the main volume control, likewise **affect the pitch behavior of all oscillators**.

As always, the **On/Off** switch must be flipped to the right to activate Glide.

When Glide is turned on, oscillator pitches will glide smoothly from one note to the next instead of jumping instantaneously.

The **Glide** knob controls glide speed, with higher settings corresponding to *lower* speeds – in keeping with the old style. Note that glide speed doesn't depend upon how far the glide has to go. If you play notes which are close to one another on the keyboard, the glide takes relatively little time, and you'll need to turn the control up past 5 in order to even hear it. Lower settings are good for making big ELP-style pitch jumps which don't take forever to get there.

- Modifiers



The **complete oscillator/noise mix** emerging from the Mixer section goes through a **Modifiers section** consisting of a **low-pass filter and an amplifier** before finally making it to the miniscope output. The Modifiers section also includes the **speed controls for all LFOs** (except the Mod Wheel LFO).

Amplifier (Loudness Contour)

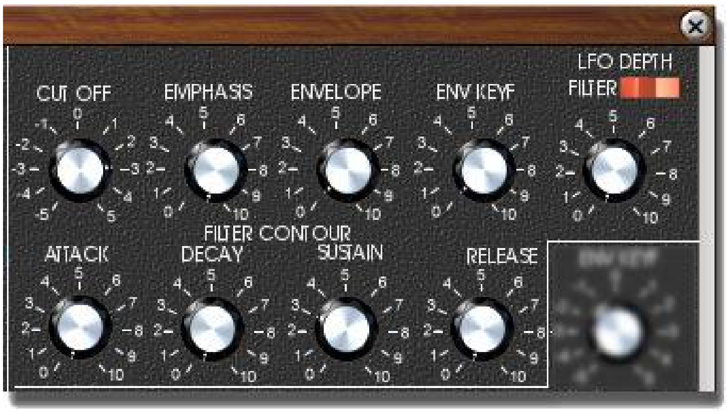
The Amplifier is controlled by a standard note-triggered ADSR (Attack-Decay-Sustain-Release) envelope, by note velocity, and by a dedicated amplifier LFO.

The **Sustain** control sets the level which the envelope ultimately gets to following the attack and decay phases, and where it stays for as long as a key is held down.

The **Attack**, **Decay** and **Release** controls are rate controls which determine how quickly the envelope level goes up or down in their respective envelope phases. Again in keeping with the old style, higher settings correspond to slower rates (or longer times).

The envelope starts at zero when a note is played and rises at the rate determined by the Attack control. The attack phase ends when the envelope reaches its "peak" level, which corresponds to the maximum Sustain setting of 10. Next comes decay phase, in which the envelope falls at the rate determined by the Decay control, until it reaches the level set by the Sustain control. Now comes the sustain phase, in which the envelope level stays put for as long as the key is held down. (If the Sustain control happens to be set to maximum, then the decay phase is basically skipped.) When the key is released, the envelope falls back to zero at the Release rate – even if it was still in attack or decay phase.

Adjusting the **Env Keyf** (Key Follow) control away from zero causes the envelope rates to change to varying degrees in response to keyboard position. The basic idea is to be able to mimic acoustic instruments such as the piano, in which higher notes decay more quickly, but by setting Env Keyf to negative values you can produce the opposite effect.



The **Amplifier level** can also be affected by an LFO. As with the oscillators, there is both an **on/off switch** for this effect and a control which adjusts the **intensity** of the effect when turned on. The Amplifier LFO speed control is in the LFO Frequencies group at the bottom of the Modifiers section.

Last but not least, Amplifier volume is affected by **note velocity**. This is a "default" feature for which there is no control.

Filter (Filter Contour)

Almost all of the features described for the Amplifier apply to the Filter as well. Only the features which are different or are unique to the filter are described here.

The Filter section includes controls for filter **cutoff frequency** and resonance (**Emphasis**) – the amount of boost near the cutoff frequency.

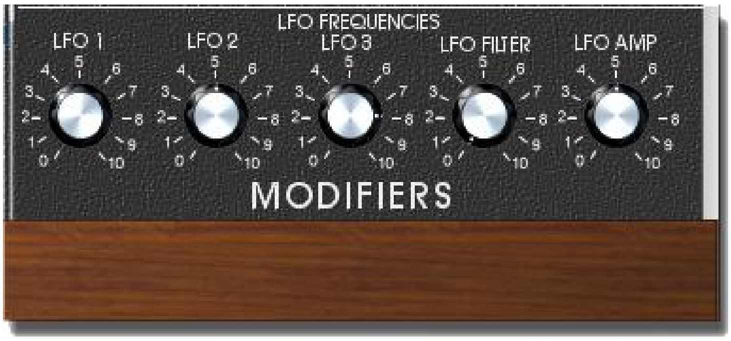
The Filter envelope affects the filter cutoff frequency. It has the same controls as the Amplifier envelope. However, unlike the Amplifier envelope, it is based not on a level of zero, but instead on the level set by the Filter Cutoff control. This means that the Filter envelope can "open up" the filter by raising the cutoff, but can't "close it down" further than the Cutoff setting itself.

Since the Filter envelope is not the only filter control, but rather an "extra" effect which works relative to the Filter Cutoff setting, there is also an **Envelope** control which lets you adjust the amount of effect the Filter envelope has on the filter. **Make sure this control isn't set to zero** before you start adjusting the Filter envelope controls!

As with the Amplifier, there is a **dedicated LFO for filter cutoff.** The Depth knob, on/off switch and speed control are all as per the Amplifier LFO.

Unlike the Amplifier, note velocity has no effect upon the Filter.

LFO Speed (LFO Frequencies)



The **speed controls for all LFOs**, except for the Mod Wheel LFO, are grouped together at the bottom of the Modifiers section.

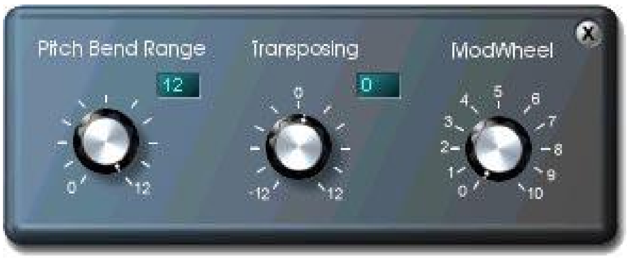
LFO 1, LFO 2 and LFO 3 are the **Oscillator LFOs**, which can be assigned to individual oscillators via the LFO select switches in the Oscillator section. **LFO Filter** and **LFO Amp** are the dedicated filter and amplifier LFOs. The depth controls for these LFOs appear in the control groups of their respective destinations.

Not appearing here is the speed control for the Mod Wheel LFO. This control is in the Mod Wheel Section.

As noted elsewhere, the miniscope has **sinewave LFOs only** – there are no LFO waveform select controls.

- Global

This control panel is opened and closed via the Global switch at bottom center (next to the MIDI channel select window).



Pitch Bend Range sets the range of pitch bend produced by incoming MIDI Pitch Bend messages. This adjustment is in semitones, up to a maximum of 12 (one octave).

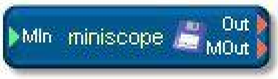
Transpose adjusts the pitch of the entire miniscope in semitone steps, up to a maximum of 12 up or down.

The **Mod Wheel** is an onscreen version of a standard mod wheel. Use it just as you would the real thing. It's convenient if you don't have a external controller keyboard handy, or if you don't want to bother with it while editing presets. Note that MIDI Mod Wheel messages are assigned to this control, which will therefore **move all by itself** when these messages are received.

Don't be fooled by the Global label here – these settings, like all others in the miniscope, are stored in presets and recalled by them.

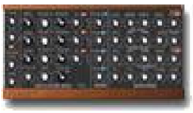
- Connections

MIDI In



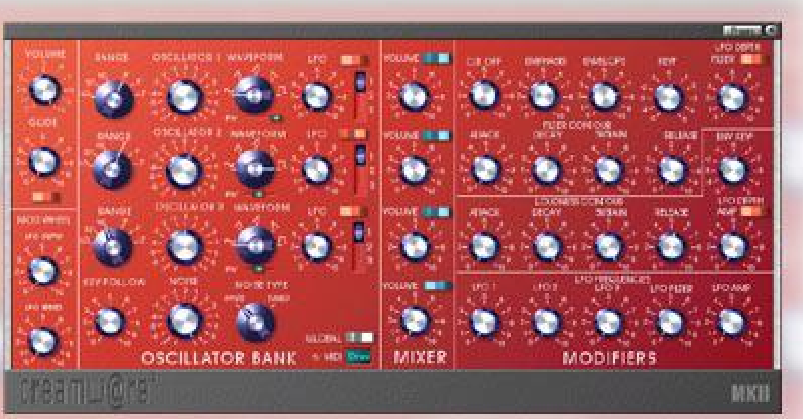
Audio output
MIDI Out

Project window (module) representation



Minimized (icon) representation

miniscope MKII



miniscope MKII

- Overview

The **miniscope MKII** is a reasonably faithful reproduction of an old-timey, monophonic, no-presets, 100 percent analog synth. In fact, it's virtually identical to the miniscope (see previous chapter). Apart from a new coat of paint, the miniscope MKII also has a better-sounding filter, which however does make it a bit more prone to overloading during polyphonic operation. If you run into this problem, simply turn down the individual oscillator levels.

What It's Got

Three oscillators whose pitch and waveform are independently adjustable, a **glide** function, a **noise generator** producing full-spectrum or tuned noise, a **mixer** section for the oscillators and noise generator, an **amp envelope**, a **resonant low-pass filter** with its own envelope, and **six sinewave LFOs** – one for each oscillator, one for all three, one each for filter and amp.

What Else It's Got

MIDI control including – of course – note play, plus velocity sensitivty, pitch bend and mod wheel response, and the ability to **assign MIDI controllers to all controls** on the front panel. Adjustable **polyphony**. Nameable **presets** – however many you want – which **store and recall all front panel settings**. Always in tune (if you want it that way), no hum or noise (unless you *want* hum and noise), weighs nothing, requires no keyboard stand. And, if one miniscope MKII happens not to be enough – just add another one into your project.

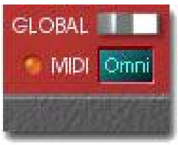
The Most Important Things To Know

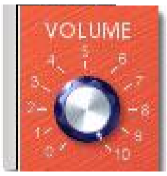
The **main volume control** is in the upper left corner, just where you *wouldn't* expect to find it.

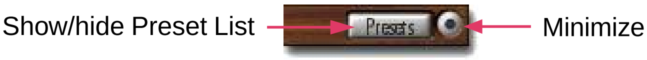
The **MIDI channel** "text fader" control is front-row center. Click *and hold* with the left mouse button *directly* on the value, then move the mouse up and down to change it.

The **Preset List** is opened/closed via the button at top right (next to the **Minimize** button, which hides the synth).

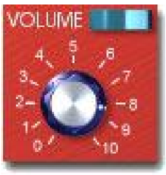
The **Global** switch opens/closes a control panel which is discussed later in this chapter.

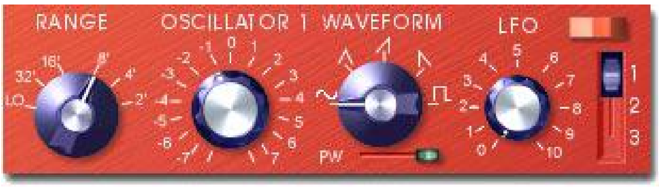






- Mixer / Oscillators / Noise





Although visually separate, the Oscillator Bank and Mixer sections, together with the Mod Wheel and Glide controls, form a single functional unit which is described here in full.

Mixer

The first thing to note is that each oscillator has its own **Volume** knob in the Mixer section – nice for balancing the oscillators against one another – and **its own On/Off switch (blue) next to the volume knob**. Make sure the switch is on (flipped right) *and* the volume turned up before you start adjusting the other controls of an oscillator!

Oscillators

There are three oscillators. They are identical to one another and can be adjusted completely independently of one another. Oscillator **pitch** can be set in whole octaves via the **Range** selector at left and fine-tuned via the **Oscillator 1 - Oscillator 3** knobs.

The **Waveform** selector lets you dial in sine, triangle, sawtooth up, sawtooth down and pulse waveforms. The **PW** (Pulse Width) slide control directly below the waveform selector affects the pulse waveform. Full left produces a square wave (symmetrical pulse width). Moving the slider to the right causes the pulse width to become progressively narrower.

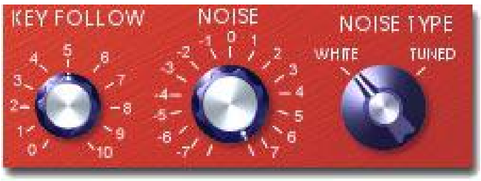
Oscillator LFOs

Oscillator pitch can be modified by any one of sinewave LFOs 1-3 as selected by the **LFO** slide switch. The **LFO** knob controls the amount of effect produced by the LFO. Note the **LFO On/Off switch**, which must be flipped to the right in order for the LFO to have any effect here. Assign a different LFO to each oscillator to get nice fat chorus-like sounds (with small LFO amounts) or extreme sound effects (with large LFO amounts). The speed controls for these LFOs are in the Modifiers section.

Noise

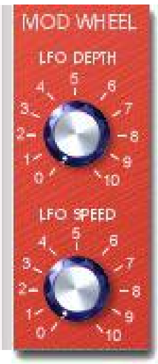
Included with the oscillators is a noise generator. Like the oscillators, it has its own **Volume** knob and **On/Off** switch in the Mixer section. It can produce plain white noise, or "tuned" noise which has been passed through a dedicated filter.

The **Noise** knob tunes the filter – it has no effect on white noise. The **Key Follow** control likewise affects only tuned noise, since it adjusts the tracking of the filter tuning in response to keyboard position. Mid-scale settings produce "normal" oscillator-like pitch tracking.



- Mod Wheel / Glide

Mod Wheel



Best for vibrato effects is the **Mod Wheel LFO**, which affects the pitches of all oscillators in common. Its controls are located in the lower left corner of the miniscope MKII.

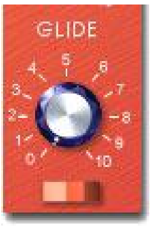
The amount of this LFO is controlled by **MIDI Mod Wheel messages**, so the effects of the settings described below will be seen only when you move your keyboard's Mod Wheel away from its zero position.

(For test purposes when editing presets, or in general if you find it more convenient, you can use the Mod Wheel control in the Global panel as an **onscreen substitute mod wheel** in place of an external physical mod wheel.)

The **LFO Depth** control here actually adjusts the sensitivity to mod wheel messages and **must be set higher than zero** for mod wheel motions to have any effect. The Mod Wheel **LFO Speed** control affects *only* this LFO.

The Mod Wheel LFO **affects only the oscillators**, but is completely independent from the individual oscillator LFOs. It can be used together with them or by itself, and is not affected by the individual oscillator LFO On/Off switches.

Glide



The Glide controls, tucked away off to the left below the main volume control, likewise **affect the pitch behavior of all oscillators**.

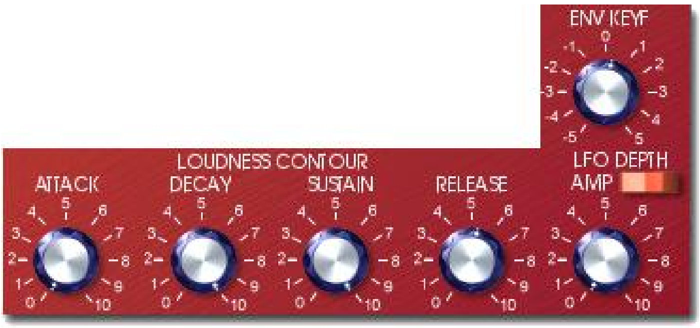
As always, the **On/Off** switch must be flipped to the right to activate Glide.

When Glide is turned on, oscillator pitches will glide smoothly from one note to the next instead of jumping instantaneously.

The **Glide** knob controls glide speed, with higher settings corresponding to *lower* speeds – in keeping with the old style. Note that glide speed doesn't depend upon how far the glide has to go. If you play notes which are close to one another on the keyboard, the glide takes relatively little time, and you'll need to turn the control up past 5 in order to even hear it. Lower settings are good for making big ELP-style pitch jumps which don't take forever to get there.

- Modifiers

The **complete oscillator/noise mix** emerging from the Mixer section goes through a **Modifiers section** consisting of a **low-pass filter and an amplifier** before finally making it to the output of the miniscope MKII. The Modifiers section also includes the **speed controls for all LFOs** (except the Mod Wheel LFO).



Amplifier (Loudness Contour)

The Amplifier is controlled by a standard note-triggered ADSR (Attack-Decay-Sustain-Release) envelope, by note velocity, and by a dedicated amplifier LFO.

The **Sustain** control sets the level which the envelope ultimately gets to following the attack and decay phases, and where it stays for as long as a key is held down.

The **Attack**, **Decay** and **Release** controls are rate controls which determine how quickly the envelope level goes up or down in their respective envelope phases. Again in keeping with the old style, higher settings correspond to slower rates (or longer times).

The envelope starts at zero when a note is played and rises at the rate determined by the Attack control. The attack phase ends when the envelope reaches its "peak" level, which corresponds to the maximum Sustain setting of 10. Next comes decay phase, in which the envelope falls at the rate determined by the Decay control, until it reaches the level set by the Sustain control. Now comes the sustain phase, in which the envelope level stays put for as long as the key is held down. (If the Sustain control happens to be set to maximum, then the decay phase is basically skipped.) When the key is released, the envelope falls back to zero at the Release rate – even if it was still in attack or decay phase.

Adjusting the **Env Keyf** (Key Follow) control away from zero causes the envelope rates to change to varying degrees in response to keyboard position. The basic idea is to be able to mimic acoustic instruments such as the piano, in which higher notes decay more quickly, but by setting Env Keyf to negative values you can produce the opposite effect.

The **Amplifier level** can also be affected by an LFO. As with the oscillators, there is both an **on/off switch** for this effect and a control which adjusts the **intensity** of the effect when turned on. The Amplifier LFO speed control is in the LFO Frequencies group at the bottom of the Modifiers section.

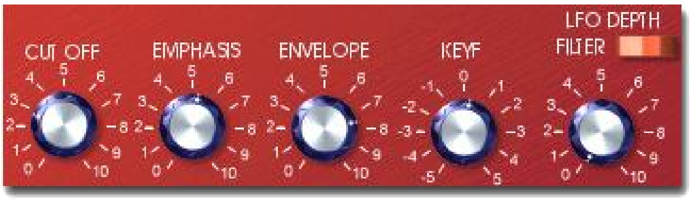
Last but not least, Amplifier volume is affected by **note velocity**. This is a "default" feature for which there is no control.

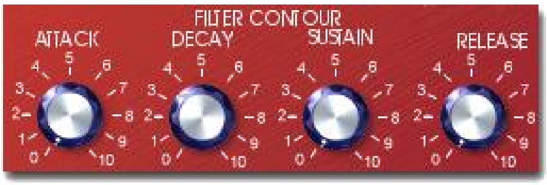
Filter (Filter Contour)

Almost all of the features described for the Amplifier apply to the Filter as well. Only the features which are different or are unique to the filter are described here.

The Filter section includes controls for filter **cutoff frequency** and resonance (**Emphasis**) – the amount of boost near the cutoff frequency.

The Filter envelope affects the filter cutoff frequency. It has the same controls as the Amplifier envelope. However, unlike the Amplifier envelope, it is based not on a level of zero, but instead on the level set by the Filter Cutoff control. This means that the Filter envelope can "open up" the filter by raising the cutoff, but can't "close it down" further than the Cutoff setting itself.





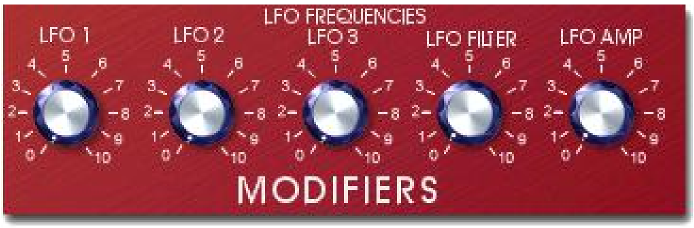
Since the Filter envelope is not the only filter control, but rather an "extra" effect which works relative to the Filter Cutoff setting, there is also an **Envelope** control which lets you adjust the amount of effect the Filter envelope has on the filter. **Make sure this control isn't set to zero** before you start adjusting the Filter envelope controls!

As with the Amplifier, there is a **dedicated LFO for filter cutoff**. The Depth knob, on/off switch and speed control are all as per the Amplifier LFO.

Unlike the Amplifier, note velocity has no effect upon the Filter.

LFO Speed (LFO Frequencies)

The **speed controls for all LFOs**, except for the Mod Wheel LFO, are grouped together at the bottom of the Modifiers section.



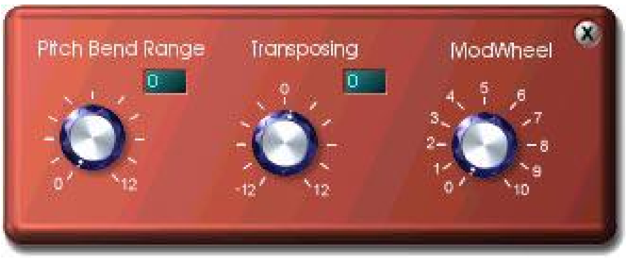
LFO 1, LFO 2 and LFO 3 are the **Oscillator LFOs**, which can be assigned to individual oscillators via the LFO select switches in the Oscillator section. **LFO Filter** and **LFO Amp** are the dedicated filter and amplifier LFOs. The depth controls for these LFOs appear in the control groups of their respective destinations.

Not appearing here is the speed control for the Mod Wheel LFO. This control is in the Mod Wheel Section.

As noted elsewhere, the miniscope MKII has **sinewave LFOs only** – there are no LFO waveform select controls.

- Global

This control panel is opened and closed via the Global switch at bottom center (next to the MIDI channel select window).



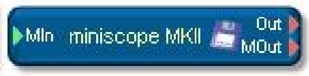
Pitch Bend Range sets the range of pitch bend produced by incoming MIDI Pitch Bend messages. This adjustment is in semitones, up to a maximum of 12 (one octave).

Transpose adjusts the pitch of the entire miniscope MKII in semitone steps, up to a maximum of 12 up or down.

The **Mod Wheel** is an onscreen version of a standard mod wheel. Use it just as you would the real thing. It's convenient if you don't have a external controller keyboard handy, or if you don't want to bother with it while editing presets. Note that MIDI Mod Wheel messages are assigned to this control, which will therefore **move all by itself** when these messages are received.

Don't be fooled by the Global label here – these settings, like all others in the miniscope MKII, are stored in presets and recalled by them.

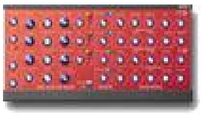
- Connections



MIDI In

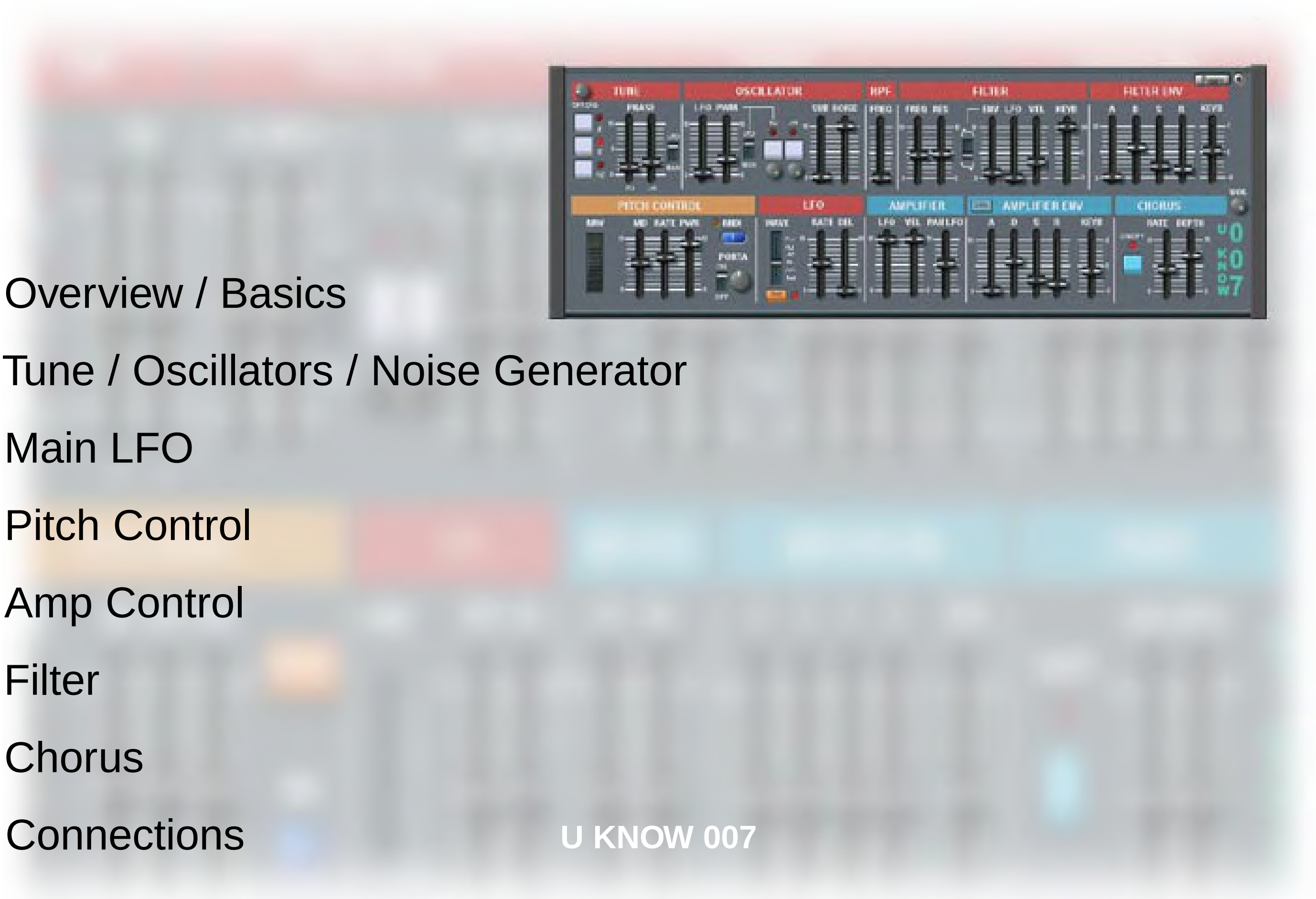
Audio output
MIDI Out

Project window (module) representation



Minimized (icon) representation

U KNOW 007



- Overview / Basics
- Tune / Oscillators / Noise Generator
- Main LFO
- Pitch Control
- Amp Control
- Filter
- Chorus
- Connections

- Overview / Basics

The **U KNOW 007** features a **pulse oscillator**,a **sawtooth oscillator**, a **square-wave suboscillator** and a **noise generator**. A **high-pass filter**, a **resonant low-pass filter**, a **stereo chorus** and an appropriate assortment of **envelopes** and **LFOs** round out the picture.

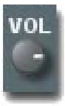
Do you think **U KNOW** where you've seen this one before?

Here are a few basic points to help with the quick start:

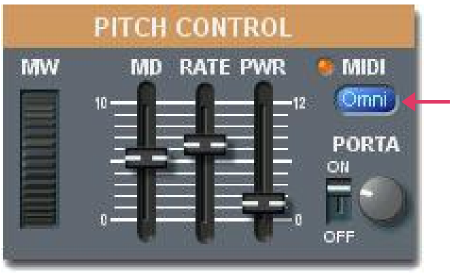
The **Main Volume** control (VOL) is squeezed in at the far right end of the synth, about halfway up the panel.

The **MIDI Channel** can be dialed in via a text fader window which is found in the upper right-hand corner of the Pitch Control section (or at any rate, that's where it was the last time we looked).

Buttons for opening the **Preset List** and for **minimizing** the synth control surface to an icon are located in the upper right-hand corner of the synth surface.

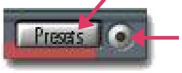


Main volume



MIDI channel

Open Preset List

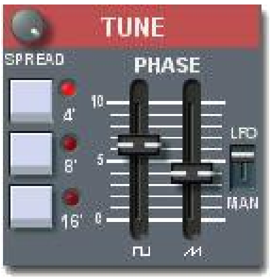


Minimize to icon

- Tune / Oscillators / Noise Generator

The oscillator section has a **variable-width pulse oscillator** and a **sawtooth oscillator**. There is also a **square-wave suboscillator** whose pitch is always one octave below that of the other two, and a **noise generator**.

Tune Section



The buttons **4'-8'-16'** select the octave range for all three oscillators. The **SPREAD** control "spreads" note pitches apart from each other just slightly to mimic that classic not-quite-perfect tuning (full left rotation = no spread).

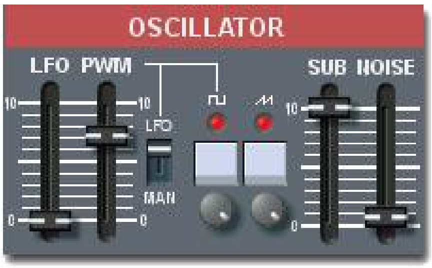
The pulse and sawtooth oscillators have separate **PHASE** controls with an associated **LFO/MAN** switch.

When the switch is in the **MAN** (manual) position, each Phase control **directly** adjusts the phase of its respective oscillator. In this mode, the phase is a static setting which does not change over time.

In **LFO** mode, the Phase controls adjust the **amount of (main) LFO modulation** applied to oscillator phase.

Although LFO-driven phase modulation of a single oscillator can produce an audible effect similar to vibrato, **the Phase features are intended for use with both oscillators switched on**, since the interesting phase effects are produced by the static or dynamic phase relationships *between* the two oscillators.

Oscillator Section



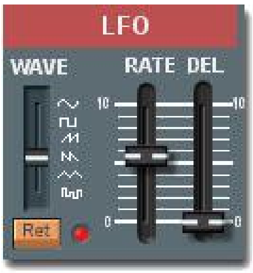
The white buttons at center **switch the pulse and sawtooth oscillators on and off** separately. The knobs directly below these buttons are the **oscillator level controls**. The faders further to the right are the **suboscillator and noise generator level controls**. With this set of controls, the levels of all four signal sources can be mixed with full flexibility.

At left are the **pulse-width controls** for the pulse oscillator.

When the switch is in the **LFO** position, the pulse width is modulated by the main LFO, with modulation depth set by the **LFO** fader. In this mode the PWM fader has no effect.

If **MAN** is selected, the **PWM** fader lets you manually set a fixed pulse width. In this mode the LFO fader has no effect.

- Main LFO



The main LFO can be used to modulate **amplitude**, **filter cutoff**, **pan position**, the **phase** of both oscillators and the **pulse width** of the pulse oscillator.

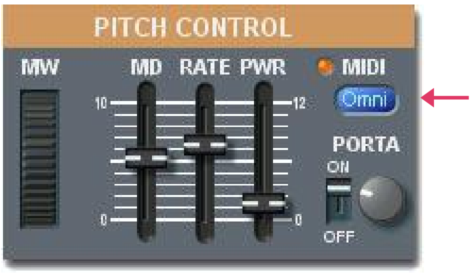
The **WAVE** slider offers a choice of six different LFO waveforms (sine, square, sawtooth up, sawtooth down, and random).

The **RATE** fader controls LFO speed.

The **Ret** (retrigger) button provides the option of resetting the LFO to the starting point in its waveform on each new note, instead of allowing it to continuously free-run.

The **DEL** (delay) fader permits adjustment of the delay and "ramp-up" (fade-in) time of the main LFO which begins when a note is played. At the minimum setting, the LFO comes on virtually instantly. At higher settings, both the delay and the ramp-up are gradually extended.

- Pitch Control



Set MIDI channel

The graphical **Mod Wheel** (MW) is an onscreen working model of the real thing, which indirectly affects oscillator pitch via a dedicated pitch-mod LFO. The Mod Wheel controls the output level of this LFO. The **MD** (mod depth) fader adjusts the range of this modulation, while the **RATE** fader adjusts the LFO speed.

MIDI Mod Wheel (controller #1) messages are assigned to the onscreen Mod Wheel and produce the same effect as the onscreen controller. In addition, **the onscreen controller moves by itself** in response to received MIDI Mod Wheel messages (provided, of course, that they are coming in on the selected MIDI channel).

The **PWR** (Pitch Wheel Range) slider sets the range of response to MIDI Pitch Bend messages in semitone steps, up to a maximum pitch-bend range of 12 semitones (one octave) up/down.

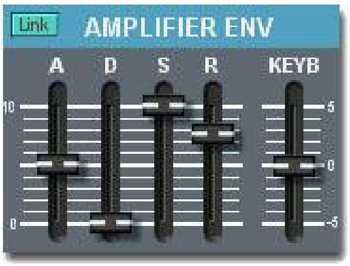
The **PORTA** (Portamento) section includes an on/off switch for enabling/disabling portamento and a knob which adjusts portamento speed.

Also located in this section is the familiar **MIDI channel select** text fader.

- Amp Control

Amplifier Envelope Section

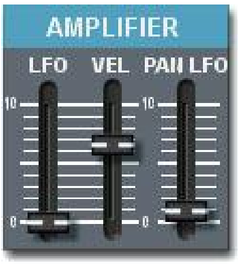
The Amp Envelope is a simple ADSR (with corresponding controls **A**/**D**/**S**/**R**) which affects the two oscillators and the suboscillator in common.



The **KEYB** fader adjusts the effect of keyboard position on envelope times (attack, decay, release). Positive settings produce longer times as you go up the keyboard, while negative settings produce the opposite result.

The **Link** switch at upper left in this section links the filter envelope settings to those of the amp envelope. When the switch is initially turned on, the current amp envelope settings are applied to the filter envelope. Thereafter, for as long as the switch remains on, the two envelopes are linked together, so that adjustments to the settings of either envelope are applied instantly to both envelopes.

Amplifier Section



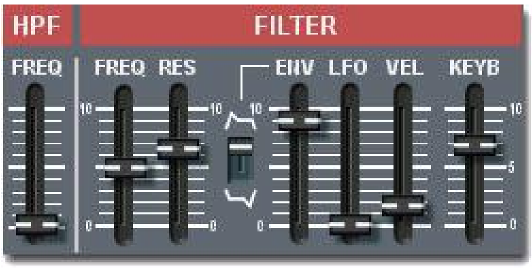
The **VEL** fader controls amplifier response to note velocity.

The **LFO** fader sets the amount of amp modulation produced by the main LFO.

The **PAN LFO** fader sets the amount of pan modulation produced by the main LFO.

- Filter

The U KNOW 007 filter section includes a resonant low-pass filter and a separate non-resonant high-pass filter **(HPF)**. Each of the filters has its own **FREQ** (frequency) control.



The **RES** fader controls filter resonance or sharpness for the low-pass filter.

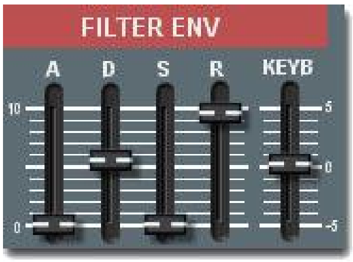
The RES control as well as the filter modulation sources described below **affect only the low-pass filter**.

The low-pass filter cutoff frequency set by the FREQ fader can be modified via four different **modulation sources**, each with its own fader to control modulation amount: **ENV** (filter envelope), **LFO** (main LFO), **VEL** (note-on velocity) and **KEYB** (keyboard position or note number).

The switch next to the ENV fader permits the **polarity of envelope modulation to be inverted**, so that the filter envelope causes the filter to close down during the attack phase and then reopen during decay and release phases.

The **KEYB** fader controls the tracking of filter cutoff against keyboard position (note number). At the maximum setting, filter cutoff tracks base oscillator (note) pitch more or less exactly. At the minimum setting, it is unaffected by keyboard position. Intermediate settings produce tracking amounts between these extremes.

The **Filter Envelope** is a simple ADSR (with corresponding controls **A**/**D**/**S**/**R**) which **affects *only* the low-pass filter**.

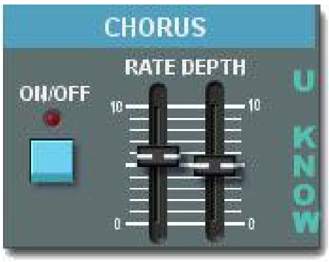


The **KEYB** fader adjusts the effect of keyboard position on envelope times (attack, decay, release). Positive settings produce longer times as you go up the keyboard, while negative settings produce the opposite result.

The **Link** switch at upper left in the Amp Envelope section links the filter envelope settings to those of the amp envelope. When the switch is initially turned on, the current amp envelope settings are applied to the filter envelope. Thereafter, for as long as the switch remains on, the two envelopes are linked together, so that adjustments to the settings of either envelope are applied instantly to both envelopes.

- Chorus

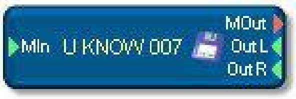
The chorus effect makes the sound of the U KNOW 007 more lush and "fat" and produces a stereo output from the otherwise monaural synth signal.



The **blue button** switches the chorus on and off. **RATE** controls the speed of the chorus modulation LFO, while **DEPTH** sets the amount of modulation (delay sweep) produced by the LFO.

- Connections

MIDI In



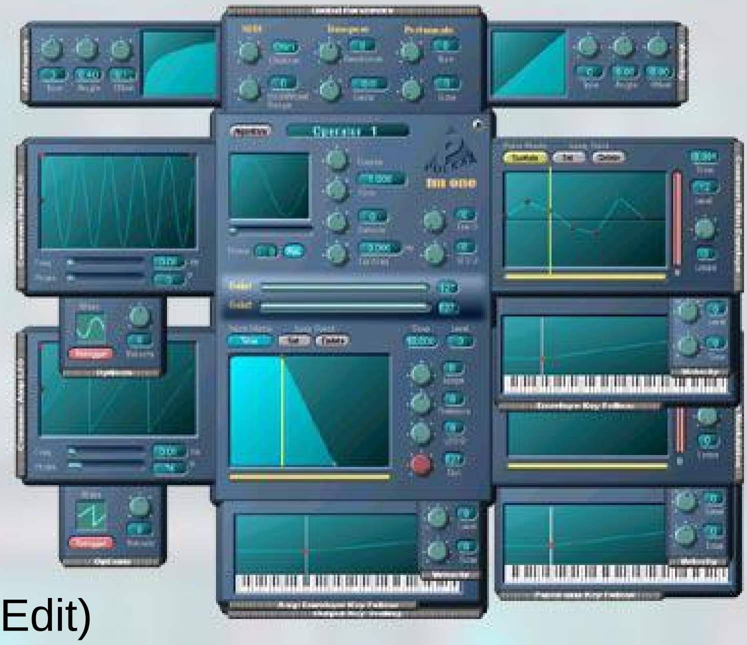
MIDI Out

Audio outputs

Project window (module) representation

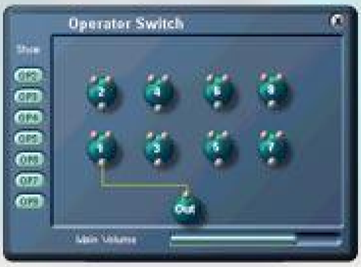
Minimized (icon) representation

FM One



- FM Synthesis
- Graphical Envelopes
- LFOs and Key Follow
- Global Parameters
- Operator Switch (Algorithm Edit)
- Individual Operator Parameters
- Common Operator Parameters
- Connections

FM One



- FM Synthesis

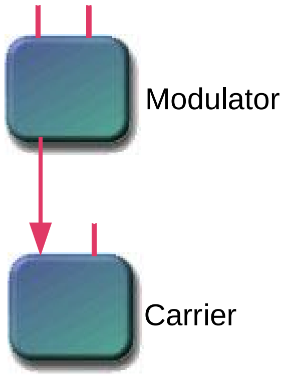
Prof. Dr. John M. Chowning discovered in 1967 that modulation of the frequency of a simple sine waveform by another sine wave results in complex sound spectra. FM means precisely Frequency Modulation – but what does *that* mean?

Quite simply, a sinewave oscillator produces a sinusoidal tone with a specific frequency. Changing the frequency by manually adjusting the oscillator's frequency control is a basic form of frequency modulation.

If this oscillator is outfitted with an amplitude envelope and one or more frequency modulation inputs, it becomes an FM operator. An operator can modulate other operators as well as be modulated.

A DX7 has six such operators which can be connected together in 32 different configurations or algorithms. This set of possibilities allows a large library of sounds to be created.

FM synthesis involves a large number of parameters, but the creation of an FM sound can be explained on a basic level according to a few simple principles.



An operator can be used either as a carrier or as a modulator. The simplest configuration – not FM at all – is a single carrier (with no modulator), producing a simple sine tone corresponding to the note played on a keyboard.

The simple FM circuit in the above diagram can create complex sounds via frequency modulation of the carrier.

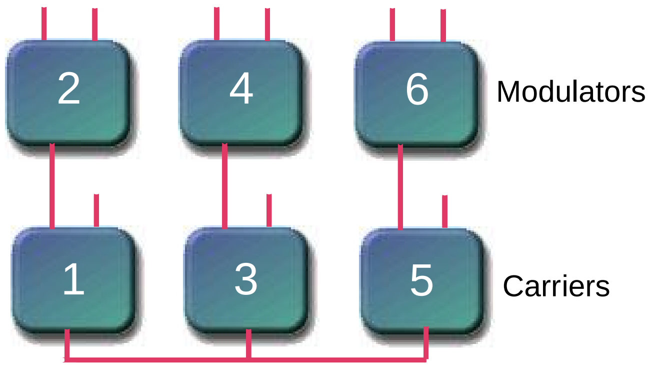
If an amplitude envelope is applied to the carrier, it affects the overall level of the generated sound. On the other hand, applying an amplitude envelope to the modulator causes the tone of the sound to vary with time. This is the basic essence of FM synthesis.

Actually, in FM synthesis there are few truly dependable statements which can be made regarding the effects of parameter changes on the sound. The results of changes in the amplitude or pitch of an operator depend upon the frequency relationship between the modulator and the carrier. Practically the only thing which can be said with any certainty is that the amplitude of a carrier directly affects the level of the composite sound.

In the six-operator circuit shown at right, each of the three carriers (operators 1, 3 and 5) determines the level of the sound generated by its branch of the algorithm.

This algorithm is actually three separate branches, each containing a single modulator/carrier pair, whose outputs are combined. Each branch can independently generate its own sound. For example, the first branch could generate a key-strike sound, the second a sustain-phase sound and the third a release-phase sound. The amplitude envelopes of each carrier are adjusted to create the transition from one phase to the next.

Another possibility is to generate similar sounds in each branch but slightly detune the branches from one another via the carrier frequency to result in a "fatter" sound.



Algorithm 42

- Graphical Envelopes

The graphical envelopes found in various SCOPE synths are all basically alike. The description here applies to all of them.

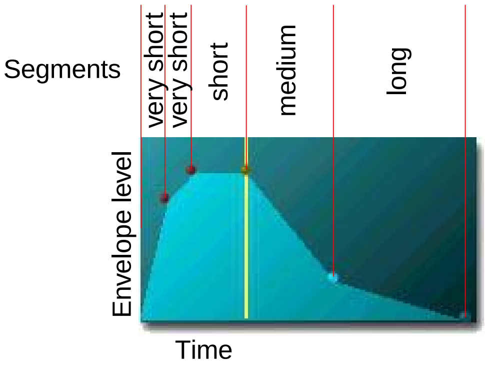
Segments And Points

A graphical envelope consists of a sequence of envelope **segments**. Envelope segments are defined using envelope **points**. Each point has a **value** (level) between 0 and 127 in a *unipolar* envelope. *Bipolar* envelopes also permit negative point values – the full range is -63 to 63.

A point's vertical position graphically indicates its value. In operation, the envelope begins at the left and rides up and down along the lines which are drawn between consecutive points. In the diagram, each envelope point sits at the right-hand end of its segment, since the point is where the segment ends.

Each segment has a **duration** (time). This is indicated graphically by the segment's width – the horizontal spacing between two points. Longer-duration segments appear wider than segments with shorter durations.

However, because **segment times can range anywhere from tiny fractions of a second up to ten seconds**, the horizontal time scale is adjusted individually for each segment, so that shorter segments are displayed proportionately wider (and longer segments proportionately narrower) than they would be if the same scale was used for all of them. This per-segment time scale "compression" yields a more usable overview with envelopes containing a wide range of segment times.



Edit both point value and segment time by dragging points with the mouse, or select a point and edit its level or time via knobs or direct keyboard entry of numerical values. The values/times of other point are not affected. **Add a new point** between two existing points by double-clicking in the space between them. You can have **up to 99** points. **Delete an existing point** by double-clicking on it. This deletes the associated segment and correspondingly cuts the segment's time out of the envelope as well.

Point Modes

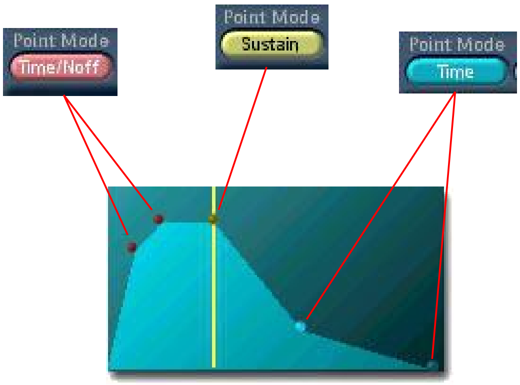
Clicking on the Point Mode button switches the selected envelope point (and its segment) between **three modes** which allow for versatile envelope construction:

Time mode: The simplest mode. The envelope always goes all the way to the segment endpoint. When the segment time is up – and only then – the envelope proceeds into the next segment.

Time/Note-Off mode: Same as Time mode, except that segments in this mode are cut short by a Note-Off (key release) event. Note-Offs also cause any Time/Note-Off segments which have not yet begun to be skipped.

Sustain mode: The envelope "pauses" at a Sustain point if no Note-Off event has yet occurred, and remains there indefinitely until the Note-Off occurs. Otherwise identical to Time/Note-Off mode.

In the display, the **mode** of each envelope point is **indicated by its color**. In addition, the Sustain point is visually highlighted by a vertical line running through it.



Using Point Modes

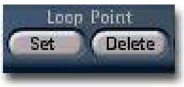
Note-Off events cause an envelope to skip "straight across" – without a level change – **to the first Time mode** segment (unless it is already in one) **and continue** from there.

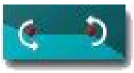
Time mode is thus useful in the release phase of an envelope (following a Sustain point) or for the last segment of any envelope (to ensure a non-zero release time and avoid note-off clicks). Envelopes consisting *entirely* of Time mode points are not affected by note-event duration – percussion envelopes are created in this way.

Time/Note-Off mode is normally used for attacks, decays and anything else coming *before* a Sustain point. It can also be used for envelopes with no Sustain point, such as piano/guitar-type envelopes, which decay steadily towards zero for as long as a note is held and cut off quickly if/when the note is released.

Envelope Loops

Envelope loops can be set up by clicking on the **Loop Point Set** button and then on two envelope points, which become the loop **start point** and **end point**. This is indicated graphically as shown above. An envelope can have only one loop. Creating a new loop clears any existing loop.





Any number of envelope points can be included in a loop. However, **a loop cannot include a Sustain point**. A loop must therefore be either completely before or completely after the Sustain point, if the envelope contains one.

The **Loops** control lets you dial in a specific repeat count 1-255 ("0" = infinite repeats). The loop repeats this number of times, or until note-off, if it contains Time/Note-Off segments. A loop containing *only* Time mode segments *always* repeats the specified number of times.

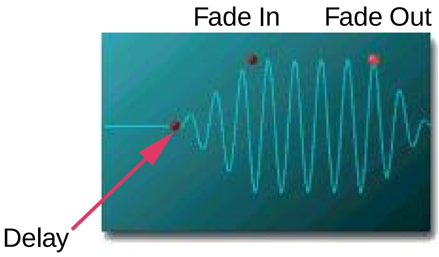
Loops are played by jumping from the end point back to the start point (forward-only looping). The time setting of the start point is applied to the end-to-start transition.

A loop can be removed by clicking on the **Loop Point Delete** button and then on one of the loop points. This merely clears the loop – the points which were included in the loop are not affected.

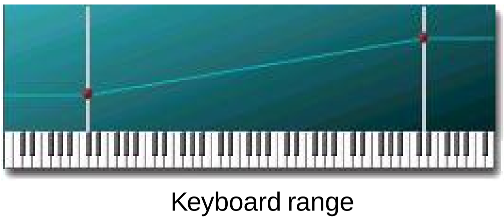
- LFOs and Key Follow

A **graphically-editable LFO** has four parameters which are adjusted via three points.

The first point sets the LFO **delay time**. The LFO waits this amount of time following a Note-On event before beginning its fade-in. The second point simultaneously sets both the **final LFO level** (after the fade-in) and the **fade-in time**. The third point sets the duration of the LFO **fade-out** which begins when a Note-Off event occurs. This point can also be used to set the final LFO level.



Key follow, or keyboard tracking, causes a parameter to vary depending upon which key on a keyboard is played. The key follow curve at right, applied to a low-pass filter cutoff, would cause the filter to close far down when low notes are played and be open fairly wide with high notes. Notes played on other keys would produce a gradual variation between these two extremes.



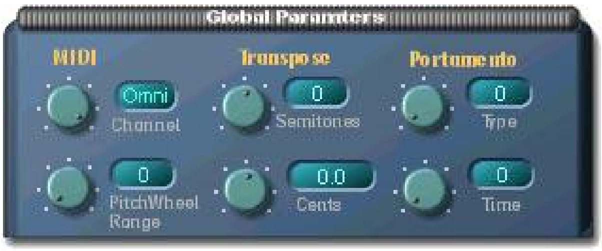
The vertical lines are curve *breakpoints* which can be moved side to side on the keyboard. As shown in the diagram, the curve slopes linearly between the red points which ride up and down on the breakpoints. Above or below the breakpoints, the curve is flat. The red points can be raised or lowered to set the *slope* of the curve between the breakpoints. The two points are coupled – raising one lowers the other by the same amount – making them both effectively part of a single slope setting.

- Global Parameters

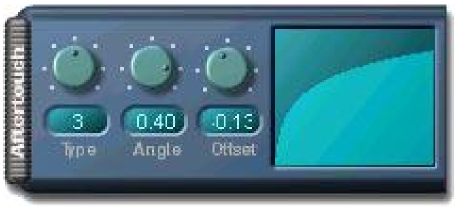
The **Global Parameters** (located in a drawer at the top of the **Operator surface**) include settings affecting the FM synth as a whole. Included here are **MIDI channel** and **Pitch Wheel Range** settings, coarse and fine tuning (**Transpose Semitones** and **Cents**), and **Portamento** controls.

The **Portamento Time** (Portamento/Glissando) control adjusts the speed of up/down pitch glide from one note to the next which is activated when the **Portamento Type** is set to a value other than 0 ("Off"). Both **Portamento** (smooth glide) and **Glissando** (glide in semitone steps) are available, each in two modes. In "plain" mode (settings 1-2), the pitch glide between two notes always occurs. In **Fingered** mode (settings 3-4), the glide occurs only when a new key is played *before* the previous key is released.

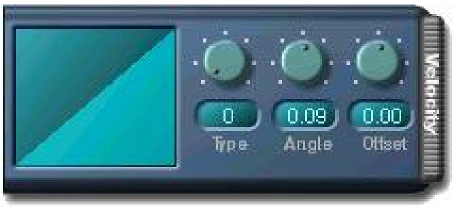
In the **Aftertouch** and **Velocity** drawers, you can tweak the curves which adjust the response of the FM synth to the corresponding MIDI values. Editing of these curves is fairly simple. The **Type** control at left selects one of six basic curve types (e.g. linear, logarithmic, etc.), the **Angle** control at center produces variations on the chosen curve type and the **Offset** control at right produces an overall shift the curve. (If that didn't completely make sense to you – never mind, just go ahead and try it. It's much easier to use than to explain.)



Channel: Omni
PitchWheel Range: 0
Semitones: 0
Cents: 0.0
Type: 0
Time: 0



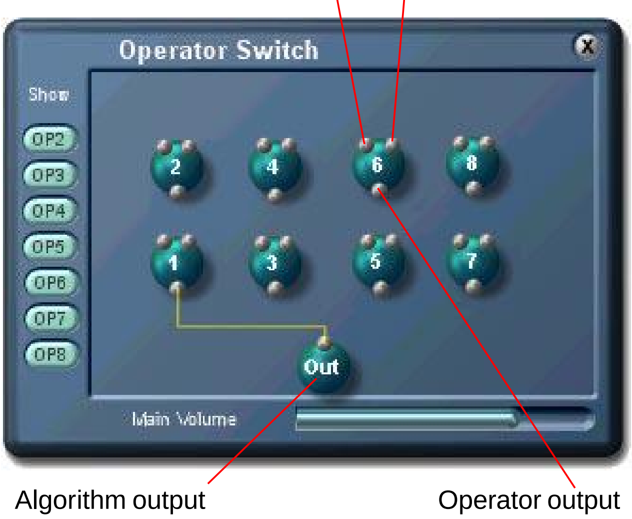
Type: 3
Angle: 0.40
Offset: -0.13



Type: 0
Angle: 0.09
Offset: 0.00

- Operator Switch (Algorithm Edit)

Operator modulation inputs 1 and 2



Algorithm output

Operator output

As discussed at the beginning of this chapter, the operators of an FM synth can be variously interconnected to construct FM *algorithms*. In the FM One, the eight operators can be interconnected more or less without restrictions, resulting in a very large number of possible algorithms.

The **Algorithm** button at upper left in the *Operator Surface* (a picture appears in the next section) opens the **Operator Switch** control panel, in which you can view and edit the algorithms of the FM One.

Each operator is depicted as a small sphere. In turn, each of these small spheres has three tiny spheres on its surface. The upper two represent the operator modulation inputs, while the one at bottom represents the operator output. These are connection points similar to module connection pads in the Project window and are used in the same way. By clicking on them, you can make or break point-to-point connections to custom-construct an algorithm. The connection from operator 1 to the **algorithm output** is a fixed connection which cannot be altered or removed.

The **Show** buttons let you switch individual operators on or off. Only *unconnected* operators can be switched off (hence no switch for operator 1, which is always connected). Switching off an unconnected operator is a good idea – it avoids unnecessary consumption of DSP capacity.

The **Main Volume** slider controls the output level of the algorithm (and thus the overall output level of the FM One, when using this algorithm).

To clear up the picture, the operator spheres can be dragged around as desired within the control panel.

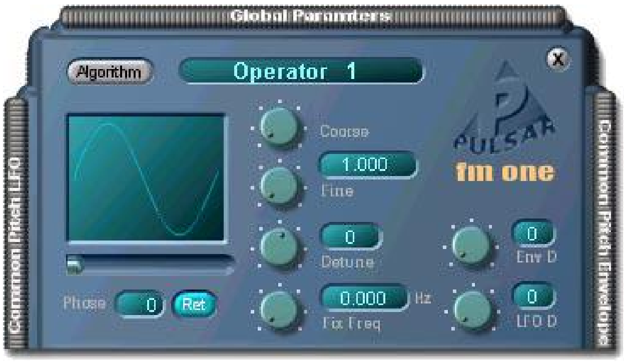
- Individual Operator Parameters

All sound settings for the FM synth are carried out via the **Operator** surface. On this surface, some parameters apply only to the selected operator, while other apply to all operators in common. In this section, the first group is described.

The FM synth has eight operators. Settings for a specific operator can be accessed by selecting the operator via the small window at the top of the Operator surface.

Coarse and **Fine** are used to adjust operator pitch. *Coarse* allows wide-ranging settings in integer steps between 0.5 and 32, where a factor of two change (from 1 to 2, or from 4 to 8) corresponds to an octave. *Fine* can dial in one more octave on top of the *Coarse* setting, in extremely fine steps. **Detune** allows a still-finer tuning adjustment of +/- 20 cents.

In conjunction with the key number of a note, these controls determine the pitch generated by an operator – as long as *fixed frequency* mode is not active. Setting the **Fix Freq** setting to a value other than zero activates this mode, in which operator pitch is set directly in Hz and is not affected by the key number of a note. In this mode, the *Coarse*, *Fine* and *Detune* settings have no effect.



Coarse 1.000
Fine
Detune 0
Fix Freq 0.000
Phase 0
Env D 0
LFO D 0

The starting **phase angle** of the operator is adjustable via the fader directly below the small waveform display window.

Ret (retrigger) controls whether an oscillator is restarted upon each new note (according to the *Phase* setting) or runs continuously. This option applies only to fixed frequency mode.

The **Env D** and **LFO D** (depth) controls allow the effect of the **Common Pitch Envelope** and **Common Pitch LFO** to be adjusted for each operator.

Each operator has two modulation inputs, whose sensitivity is adjusted via **Gain 1** and **Gain 2**. Below these are the operator **amplitude envelope** controls.

For detailed information about **editing of the envelope** itself, including the use of the **Point Mode** and **Loop Point** controls, refer to the *Graphical Envelopes* section earlier in this chapter.

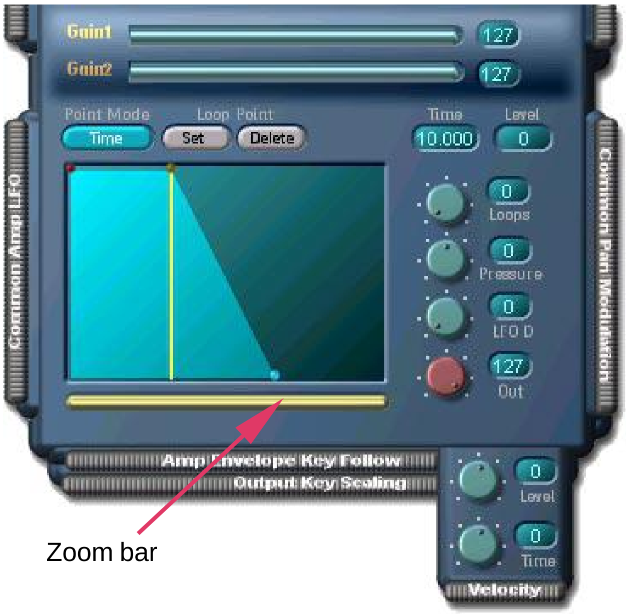
The **Time** and **Level** windows display the values for the currently selected envelope point. Values can be entered directly into these windows from the PC keyboard.

If there is a loop in the amp envelope, the **Loops** control can be used to specify how many times the loop should repeat.

The **Pressure** and **LFO D** (depth) knobs control the amount of effect which MIDI aftertouch and the *Common Amp LFO*, respectively, have upon operator **amplitude**.

The **Out** control sets the operator output level. The **output level** can also be affected by **key position**. This is adjustable in the **Output Key Scaling** drawer (next page).

The amp envelope can additionally be modified via *key follow* and *note velocity*.

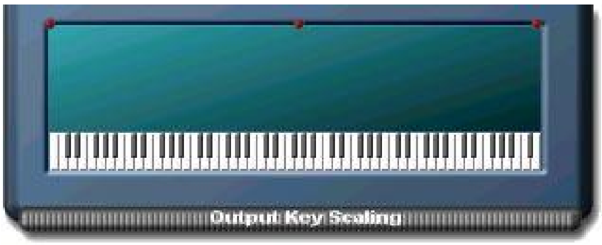


Gain1 127
Gain2 127
Point Mode Time
Time 10.000
Level 0
Loops 0
Pressure 0
LFO D 0
Out 127
Level 0
Time 0

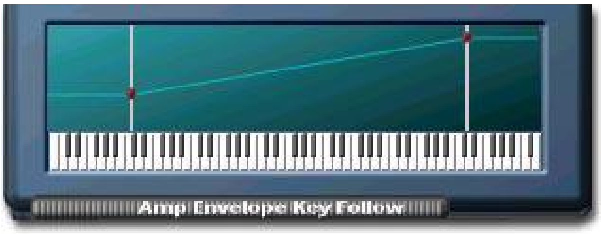
In the **Amp Envelope Key Follow** drawer (see next page), the influence of **keyboard position** upon amp envelope **times** can be adjusted.

The **Time** control in the **Velocity** drawer allows the effect of **note velocity** upon amp envelope **times** to be adjusted. The **Level** control sets the amount of effect which **note velocity** has upon the overall amp envelope **level**.

Output Key Scaling allows you to specify a curve to control the variation of an operator's output level based on keyboard position. Among other things, this is useful for creating keyboard split or crossfade patches in which two or more distinct sounds appear across the keyboard, or for controlling changes in the timbre of a sound across the keyboard.



Amp Envelope Key Follow lets you control how the amplitude envelope times of an operator will vary with keyboard position. An obvious use for this (of course there are other uses) is to mimic the typical characteristic of acoustic percussion and plucked-string instruments in which higher-pitched notes decay more quickly.



- Common Operator Parameters

This group of settings affects all operators in common.

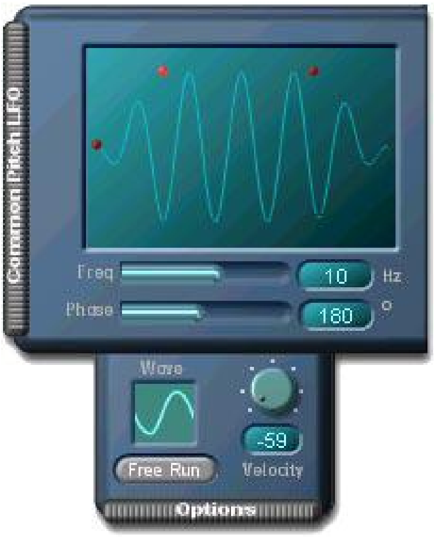
The **Common Pitch LFO** is graphically editable (refer to the *LFOs and Key Follow* section earlier in this chapter). In addition, LFO **frequency** and starting **phase** can be edited via the sliders below the LFO edit window.

The **Options** drawer contains a few more LFO controls:

The tiny **Wave** window allows LFO waveform selection. Left-click and hold here and move the mouse up and down to change the selection.

The **Velocity** control sets the amount of effect which note velocity has upon LFO **depth**.

The **Retrigger** button controls whether the LFO is retriggered (per the specified *starting phase*) with each new note or continuously free-runs.



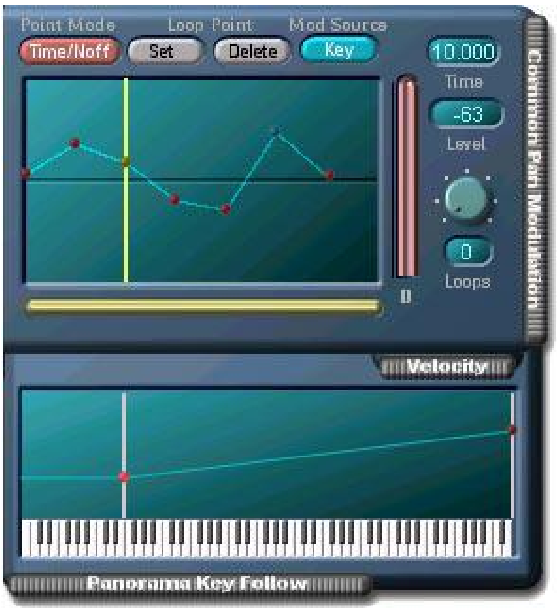
The **Common Amp LFO** permits modulation of voice output level. This LFO has controls identical to those of the *Common Pitch LFO* described above.

The **Common Pan Modulation** drawer contains options allowing modulation of voice pan position:

The **Common Pan Envelope** is is a *bipolar* envelope with controls largely similar to those of the operator amp envelope, including the options for velocity modulation of envelope levels and times.

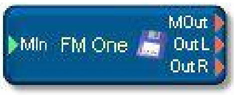
Alternatively, pan modulation is possible via **Pan Key Follow**, which allows a sound to be spread left-to-right based on key position.

The **Mod Source** switch determines which of these two pan modulation sources is active.



The options in the **Common Pitch Modulation** drawer affect the pitch of all operators in common. These options are similar to those of the *Common Pan Modulation* drawer described above, except for the absence of the Mod Source switch (both envelope and key follow modulation of pitch are available simulaneously).

- Connections



MIDI In

MIDI Out

Audio outputs

Project window (module) representation

Minimized (icon) representation

Modular Synth

Modular Synth

- Overview And Ground Rules

There's a lot to read here – don't get discouraged!

Loading The Modular Synth

The basic Modular Synth is located in the **..\SCOPE\Data\Synthesizers\Modular** folder. It appears in the File Browser under the name **Empty Modular.mdl** (make sure the **Mdl** button in the browser is pushed in, otherwise it *won't* appear!) and can be dragged from there into the Project window in the usual way.

The Empty Modular is a bare-bones framework suitable for creating new Modular setups from scratch. This makes it quite useful. But since it's "empty", it does need to be worked on a bit before it will even begin to make sounds.

If you're new to the Modular Synth, a highly recommended alternative is to drag in one of the **Modular Synth examples** found in **..\SCOPE\Data\Synthesizers\Modular\Modular Patches** (these are also .mdl files, by the way). With any of these (as with the Empty Modular as well), you'll still need to go into the Project window to make MIDI in and audio out connections to the Modular, but once you've taken care of that, you'll have a ready-to-play setup in front of you, complete with presets, which offers a concrete example of what can be done with the Modular and how to do it, and which can also serve as a starting point for your own forays into SCOPE Modular synthesis.

About The Modular Window

The Modular window is a special-purpose device surface into which Modular Synth *modules* can be dragged and then arranged and interconnected as desired (more about the modules later – *much* more). You get into this window via the usual methods for opening a device surface: open its module menu by right-clicking on the Modular device in the Project window, and then select *Open Modular Window* – or simply **double-click on this device or on the Modular Synth icon** which appears at the bottom of the screen when you load the Modular Synth into SCOPE.

The window itself is a standard movable, adjustable SCOPE window with only a few features. At the top are four buttons. Two of these are the usual ones found on every device surface. The **Presets** button displays or hides the Preset List for the current Modular setup. The small unlabeled **Minimize** button in the upper right corner shrinks the Modular window to an icon.

The **Routing** button toggles the display of cables in the Modular window on and off. Naturally you need to see the cables when modifying a Modular setup – but the cabling for a full-blown setup can get to be quite a rat's nest! When you're merely *using* a setup, you can hide the whole mess and make it much easier to see and do things. The **Save** button is discussed in the next section.

Saving Modular Synth Setups (Important!)

It's important to be aware that the Modular Synth requires a bit of special handling: **You have to save your Modular setups!** A brief explanation:

The structure of the typical SCOPE synth is stable – it's the same in every project. SCOPE projects therefore normally save only synth *references*. A given synth is loaded from the same device file whenever you load *any* project which uses it. This file can be shared by any number of projects. The synth itself doesn't need saving.

The Modular, on the other hand, has no single fixed architecture. It may contain any combination of modules, which can be cabled together in a huge variety of ways. It's likely to look quite different from one project to the next (or even within a single project, since you *can* have more than one Modular in a project). Think of having not just one but *many* unique and separate Modular synths.

This shape-shifting aspect of the Modular makes the technique of references to a single common device file unworkable. Therefore, **when you save a project which includes a Modular synth, *the complete Modular setup is saved directly within the project*.** This ensures that when you reload the project, the inlcuded Modular setup will be the same as it was when you saved the project.

Alternatively, the **Save** button at the top of the Modular window lets you **save a Modular setup as a *separate* module (.mdl) file**, in effect exporting it from its project. This Modular setup can then be used in other projects. Here you should be aware that: **1)** the setup thus saved retains ***no* connection** to the setup in the project – they're now entirely separate, and **2)** when you use the new copy in another project and save the project, the setup is saved again *in the new project*, and this setup is likewise now **independent of the copy in the .mdl file**. Remember the basic principle of many *unique and separate* Modulars.

In both cases, **the saved setup includes the Modular preset list**. By now, you can probably guess why: since Modular setups are so flexible, the **presets created for one particular setup are usable *only* with that setup**. You *can* export a preset list via the Preset List control panel, but this list will be usable only with the original setup or an *unmodified* copy. (Also note that the Preset List *Save Device* function does not apply to the Modular.)

The bottom line: to retain any changes you make to a Modular setup, *save the project* or *save the setup*. Otherwise, those changes are gone for good. We also heartily recommend, for all of the above reasons, that you give meaningful, descriptive or at least distinctive names to the Modular setups you save. Don't be lazy – use those **long filenames** that G— (Gates) gave you!

Adding, Removing and Copying Modules In A Setup

To **add new modules** to a setup, locate them using the File Browser in one of the module library folders (by default under ..\Scope\Data\Synthesizers\Modular – remember that you need to have the browser *Mdl* button pushed in). Click and drag on the name of the desired module, move the cursor over the Modular window and let go. The new module appears in the window after a brief loading delay.

To **delete a module**, select it by clicking on it, and then hit the Delete key on your keyboard. All connections to the module will likewise be deleted.

You can **copy modules** to save time and mouse-clicks when a module of the type you want to add is already present in a setup. Just click on it, then do the usual: hit Ctrl-C (copy) and then Ctrl-V (paste). Or, simply hit Ctrl-D (duplicate). The mouse cursor changes shape to show that it's ready to drop a new copy of the module wherever you next click. Existing connections to the original module are *not* duplicated – the new module shows up "naked".

Arranging Modules In The Modular Window

Modules occupy **positions on an invisible grid**, to which they automatically "**snap**" when dragged around in the window. Any connections you've already made to a module hang on tight and follow the module around when you move it. This makes the creation and maintainence of compact and manageable module arrangements both easy and quick.

Because of the grid, modules won't move until you drag them far enough. Don't be confused **if a module seems to be stuck in place**, especially when you try to drag it sideways – the grid spacings are wider in this direction. Keep dragging and it'll eventually snap. Also make sure you haven't grabbed the module by one of its active elements – a knob, switch or jack, etc. This will adjust the control or begin the making or breaking of a connection, but won't move the module.

The Modular won't let you arrange modules such that a module is completely hidden behind another.

With very large setups, screen response may become slow. To counter this, you can: **1)** Shrink the window and use the scroll bars to focus on the area of interest – for example, make the window tall and narrow when moving modules vertically, or short and wide for shifting them sideways. Where possible, dragging the window partway offscreen gets the same result. **2)** In the **SCOPE Settings** control panel, select *Standard cursor*. It doesn't look as cool as the SCOPE cursors, but it is updated much more quickly, even when the graphics workload is quite heavy.

Making Connections In The Modular Window

In the Modular, you **make a connection** by simply clicking on an input or output jack and then on the jack you want to connect it to. To **disconnect** two connected jacks, likewise click on one and then on the other – or, click on the cable itself and then hit the Delete key. (By the way, clicking on a cable highlights it and brings it to the front. This can help you to trace its path through the rat's nest.)

Note that there are a few different **types of jack** in the Modular which carry **different signal types**. A jack can only be connected to other jacks of the same type, since the different signal types are not compatible with one another. However, each jack type has its own color, so you don't have to strain your brain too much over this. Besides, SCOPE won't let you make an "inappropriate" connection. This means it also won't let you connect two outputs to the same input or to one another (just like in the "real" world), although you *can* connect multiple inputs to the same output. Here's a list of the different jack types you'll find:

Audio	- sound and control signals (LFOs etc)
MIDI	- MIDI messages into / out of the Modular
Gate	- envelope triggering (from note events etc.)
Freq	- oscillator frequency control
Esync	- polyphonic voice control feedback

This Stuff, You Gotta Have

There are a few **mandatory modules** which are present in every Modular setup and can't be deleted. A handful of specific connections to these modules must also be made before a modular setup can begin to function. The **Empty Modular** setup is a good basis for creating new setups, as it includes precisely these modules.

The **Audio Out** module lets you send audio out of the Modular Synth via any or all of its four outputs and into the SCOPE environment at large for connection to the Mixer, the Analog Outputs, etc. Likewise, audio fed to the Modular's two audio inputs is available via the **Audio In** module.

MIDI messages which are routed to the MIDI input of the Modular Synth from the SCOPE card MIDI input or from a sequencer program are "pipelined" into the Modular setup via the **MIDI In** module and appear at its output jack. From here they can be cabled as desired in the Modular setup.

Each setup must have at least one **MVC** (MIDI Voice Control) module, whose MIDI In jack must be connected to the MIDI In module. It's possible to have more than one MVC in a Modular setup. Among other things, this permits independent voicing control using separate MIDI channels within a single Modular setup.

The MVC module is the foundation of MIDI note response and voice control activity in the Modular Synth, as described in the next few paragraphs:

The **Freq** (frequency) output of the MVC module must be connected to the Freq input of the oscillator modules it is to control in order to have those oscillators – such as the Multi Oscillator – produce pitches which track, in the "usual" well-tempered way, the incoming MIDI notes.

The **Gate** output of the MVC must be connected to the Gate inputs of any envelope generator (ADSR) modules which are to be controlled by MIDI Note events. Typically this includes both an amp (volume) envelope and a filter cutoff envelope. Gate signals have other uses as well, such as retriggering of LFOs or synchronous starting of FM operators. Also note that gate signals can come from other sources, such as the MIDI Clock module.

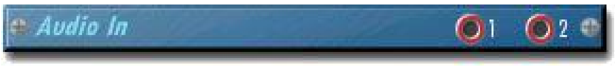
The **Esync** input of the MVC **facilitates proper polyphonic voice control**. It accepts feedback from the envelope generator (ADSR) module(s) used in a setup. If a setup has only one of these, its Esync output must be connected to this input. The Esync outputs of multiple envelope generators can be combined via one or more Esync Adder modules into a single signal which can be fed to this input. For more information about this, refer to the "*More About Esync*" section of this manual.

In "connection" with polyphony, the **Poly Out** module is important to be aware of. This module is not mandatory (and is not part of the Empty Modular setup), but it *is* **necessary for polyphonic output from the Modular Synth** (achieved by setting the *Voices* parameter in the Modular's Preset List to a value higher than 1). Simply connect it last in line before the Audio Out module when using the Modular in a polyphonic mode. Or, if you're using a mono-in/stereo-out effect such as the Chorus Mono To Stereo on the Modular output, connect the Poly Out module in line directly *ahead* of this effect.

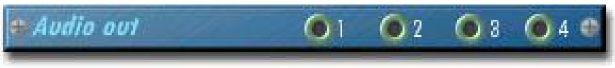
- Audio Input / Output

The Modular Synth has two inputs and four outputs for audio connections in a SCOPE setup. These are accessible inside the Modular window via the Audio In and Audio Out modules.

The **Audio In** module is included in the Empty Modular setup and comes pre-wired to the Modular Synth audio inputs. Audio signals sent to these inputs are available at the module output jacks for connection as desired in a Modular setup. The Audio In module can't be deleted, ensuring that its connections to the outside can't be lost.



The four jacks of the **Audio Out** module provide a direct line to the Modular Synth outputs. Signals generated in a Modular setup go out to the SCOPE system via these jacks. As with the Audio In module, this module is part of the Empty Modular setup and cannot be deleted, in order to ensure that these connections remain intact.



- Oscillators

The Oscillators category includes **single-waveform** and **multi-waveform conventional oscillator** modules, simple and advanced **noise generator** modules, **FM operator** modules and a **sample player** module.

This group contains a large *number* of modules. There are several different single-waveform oscillator modules. Each of these, as well as the multi-waveform oscillator, comes in multiple versions with various **external pitch modulation options**, allowing non-keyboard position related pitch control arrangements, and each variety of multi-waveform oscillator is furthermore available with or without a noise waveform. This systematic assortment lets you choose exactly the oscillator you want, without wasting DSP capacity or Modular window "real estate".

Pitch Modulation Input Options

Oscillator module name suffixes indicate the pitch mod options they are equipped with:

M1 (for example, Sine Oscillator M1) indicates a **single** pitch mod input with an **exponential** control characteristic.

M2 (e.g., Sine Oscillator M2) indicates **two** pitch mod inputs, one of each type: **exponential and linear**.

" " (the third option – i.e., nothing – e.g., Sine Oscillator) indicates **no** pitch modulation inputs.

Pitch Modulation Characteristics

The **exponential** characteristic is the more "musical" one. A specific mod input change always produces a corresponding **pitch** change. For example, if a given input change raises oscillator pitch by a whole tone, then a change which is twice as large will raise the pitch by two whole tones, etc. This is the best input for pitch-bend effects. The numerical setting on the control indicates directly in semitones the maximum pitch change (produced by full-scale *positive* mod input).

The **linear** characteristic, by contrast, relates mod input changes to oscillator **frequency**. For example, if a specific input change increases oscillator frequency by 100 Hz, then a 2x change will raise it by 200 Hz, etc. This input is good for vibrato, FM and other non-pitch related effects.

Pitch mod inputs accept **bipolar signals** (both positive and negative modulation is possible) and have controls which adjust the **modulation depth**. On exponential inputs, these controls have their zero point at center, permitting inverted modulation with left-of-center settings. LFOs, envelope generators, MVC MIDI control outputs or even other oscillators can serve as control sources.

The various oscillator modules are described in detail in the following pages.

Multi Oscillator – General

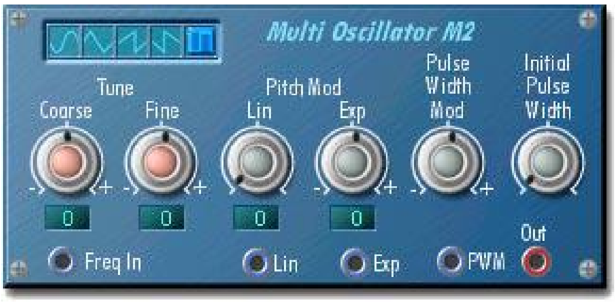
This is the workhorse oscillator. Select one of the basic unit's **five waveforms** (Sine, Triangle, Sawtooth Up, Sawtooth Down, and Pulse) by grabbing the blue highlight on the waveform selection bar and dragging it left or right.

Freq In *must* be connected to the Freq Out signal of an MVC (MIDI Voice Control) module for proper operation of an oscillator module. This causes oscillator pitch to track keyboard position in the usual way.

Coarse and **Fine** tune the oscillator, in semitones and cents, up to four octaves up/down from the center setting of zero which corresponds to "standard" keyboard pitch.

The **Initial Pulse Width** control sets the base pulse width of the Pulse waveform (50% at full counterclockwise, almost zero at full clockwise). This can be modulated via the **PWM** (Pulse Width Mod) input. PWM depth and polarity are adjustable via the control above this input.

The Multi Oscillator consumes more DSP capacity (and window area) than a single-waveform oscillator. Use it when you need it, or for trying various waveforms when working on a setup. If you end up using only one of the waveforms on a Multi Oscillator, you can "economize" by replacing it with the appropriate single-waveform module.



Multi Oscillator – Available Variations

The Multi Oscillator is also provided in versions which additionally include an **exponential** modulation input (**M1**) or both **exponential and linear** modulation inputs (**M2**), permitting non-keyboard related pitch control setups. Refer to the first page of the *Oscillators* section for details.

Each of the three varieties of Multi Oscillator described above comes in an alternate version which adds a **noise generator** as a sixth selectable waveform on the waveform selection bar. These versions consume slightly more DSP capacity than their noise-less counterparts.

The noise waveform is furthermore not affected by *any* of the module inputs or controls. This makes one of the simple **noise generator modules** a better choice in spots where the other oscillator waveforms will not be used.

Single-Waveform Oscillators

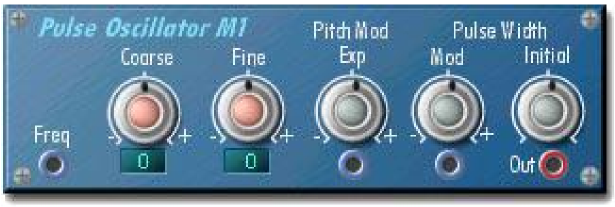
These modules are largely **functionally identical to the Multi Oscillator**, except that each one produces only a **single waveform**. These modules offer a simple advantage: they are more sparing in their use of both DSP capacity and window "real estate". They're the best choice if you know that you'll need only one specific waveform at a particular point in a setup – or as a replacement for a Multi Oscillator module, if you're actually using only one of its waveforms.

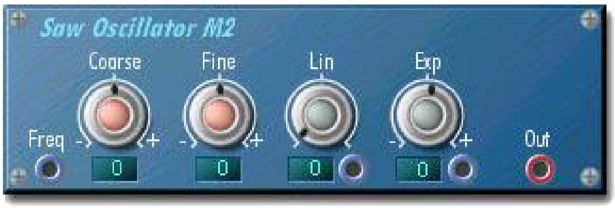
Each of the waveforms available on a Multi Oscillator (except for triangle wave) is also available in a single-waveform oscillator module. As with the Multi Oscillator, each member of the single-waveform series comes in **"regular", M1 and M2 versions**, each of which offers a different configuration of pitch modulation inputs (as described on the first page of the *Oscillators* section).

The **Pulse Oscillator** differs from the Multi Oscillator in the single detail that its Initial Pulse Width control produces 50% (square wave) pulse width at the *center* position (with almost zero or almost 100% at the extremes), rather than at full counterclockwise.

The **Square Oscillators** are equivalent to Pulse Oscillators with pulse width fixed at 50%, and no PWM inputs or controls. They take up less screen space than the corresponding Pulse Oscillators and consume less DSP capacity as well. They should be used instead of Pulse Oscillators when a simple square wave is desired.

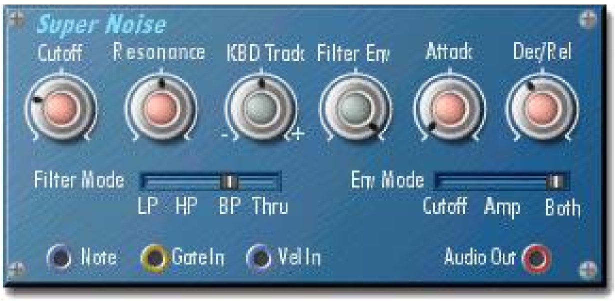
The simple **Noise Generators** are extremely space-efficient – they feature outputs only, and *no* controls. They provide pink as well as white noise via either separate modules or a single combination module.





Super Noise

The **Super Noise** module is a deluxe noise generator with multi-mode filter and envelope generator which offers generous mode and modulation possiblities.



The **Filter Mode** can be set for low/high/band-pass (or "by-pass" with the Thru setting). The **Cutoff** control sets the filter cutoff or peak frequency. The **Resonance** control is active in all modes.

The filter cutoff can be made to track keyboard position by connecting the **Note** input to the corresponding MVC (MIDI Voice Control) module output. The modulation depth and polarity can be trimmed via the **Kbd Track** control.

Use of the **Envelope** requires a **Gate In** connection from the MVC module or another gate signal source. The filter cutoff or the amplifier or both at once can be modulated by the envelope. **Decay** and **Release** times share a single control (there is no Sustain control). The **Filter Env** control adjusts the amount of envelope effect on filter cutoff.

Velocity modulation effects require a connection of the **Vel In** jack to the Vel jack on the MVC module. The velocity input operates directly on the envelope level. This means that velocity modulation of filter cutoff is possible only when envelope modulation of the filter is selected.

FM Operator / FM Operator FB

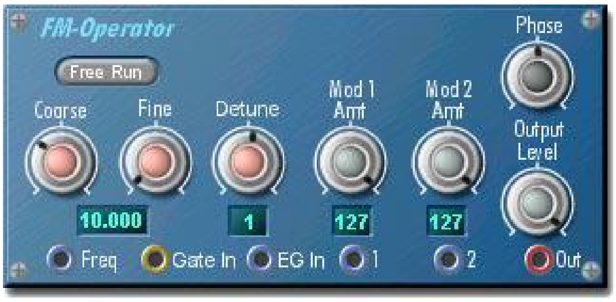
FM synthesis techniques can also be used in the Modular, extending all the way to construction of a multi-operator FM synth. These modules are essentially sine oscillator modules with *linear* pitch modulation inputs and other features to optimize them for FM synthesis.

The **Coarse** and **Fine** controls adjust a single *linear frequency* setting which is better suited for FM than the usual semitone/cents settings. The base setting of 1.000 corresponds to "standard" keyboard pitch (equivalent to a setting of 0 semitones on other oscillator modules).

The **Coarse** control sets the frequency in *harmonic* (rather than semitone) steps over whole-number values 1 through 32 (up five octaves) as well as 0.5 (one octave down). The **Fine** control gradually *scales* this setting over a range of 1.999:1 (*almost* one full octave up). The complete 7-octave span sits a good deal higher than that of most other oscillators, as you'll no doubt quickly notice.

The **Detune** control permits a still-finer shift of +/- 20 cents.

The **Freq** input *must* be connected to the Freq Out signal of an MVC (MIDI Voice Control) module for proper operation of the internal oscillator. This causes oscillator pitch to track keyboard position in the usual way.



EG In is the operator's gain control input. It's intended to accept the output of a DC (unipolar) **envelope generator**. Other control sources (e.g., LFOs) can be used with some success. In any case, you must connect *something* here in order to get any output from the operator.

The two **Mod** inputs are identical. Each has its own modulation amount control. FM happens here. Typical input signals are outputs from oscillators (classically sine), from other FM operators, or even from the *same* operator.

The **Output Level** control is useful for fine-tuning the interaction between operators in a multi-operator setup.

Gate In is used with the remaining controls, which are not essential for basic use of the FM Operator. They permit **precise phase control of the internal sine oscillator**, which can be quite useful for tweaking the timbre in multi-operator FM setups. A gate signal from the MVC or other gate signal source must be applied to the **Gate In** jack in order to activate these features.

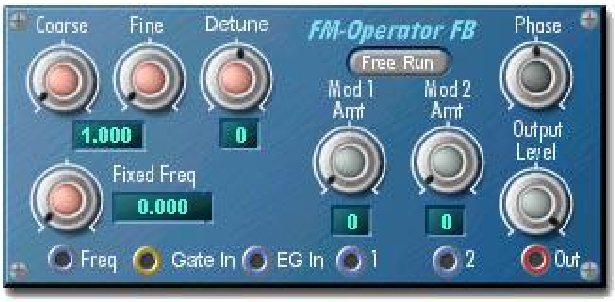
The **same gate signal must be applied in common** to all FM Operator modules which are to have their oscillator phases coordinated, in order for these features to produce their intended result.

The **Free Run / Retrigger** button controls whether the internal sine oscillator produces a continuous, uninterrupted waveform at all times or is reset to a specific phase point whenever a gate event arrives via the Gate In jack.

The **Phase** control adjusts the phase point to which the oscillator is reset in Retrigger mode.

FM Operator FB (FeedBack)

This module is largely identical to the "standard" FM Operator. The FM Operator description up to this point applies equally to both. Additional internal functions enable the FB version to complete an **operator modulation loop**. In a two-operator loop, for example, the output of each operator is routed to a modulation input of the other, producing a form of modulation feedback.



Such a configuration is a common component of FM algorithms. However, this is not possible using only the standard FM operator. To "close" a loop, whether it includes one, two, three or more FM operator modules, **at least one must be of the FB type**. *Which* one is not critical, since the loop has no beginning or end.

The FB module is also capable of fixed-frequency operation, another common FM component. **Fixed Freq** dials the frequency in directly (0.001 Hz - 9772.000 Hz). A non-zero setting here overrides the Coarse/Fine tuning setting and disables the Freq input (mod inputs still work).

A thorough discussion of FM synthesis is well beyond the scope of this manual. However, you can find some information on this topic in the *FM One* chapter, which describes the SCOPE FM Synth.

WAV Oscillator (Sample Player)

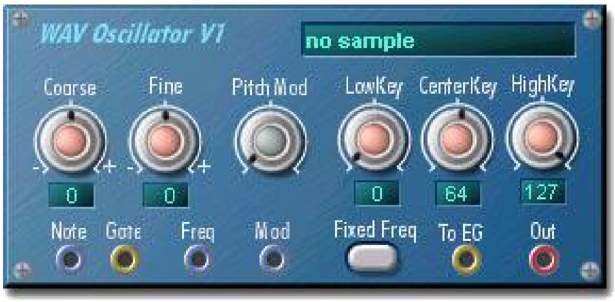
This module produces sounds by way of sample playback. As with the Sample Player devices, Akai S1000 format samples (but here, **single samples only**, not programs) are **dragged from the File Browser and dropped onto the window of this module** to load it. Otherwise, this module can be used just like any other oscillator module, with a few differences and restrictions as described below.

The WAV Oscillator ***must*** **be used with the MIDI Voice Control Sampler** module. The standard MVC won't work. (The MVC Sampler *can* control conventional oscillators.)

The **Freq** input of the WAV Oscillator must be connected to the **Freq Off** (offset) output of the MVC Sampler, and ***not*** to its Freq Out jack. (If you're layering conventional oscillators on top of the WAV Oscillator, *these* can be connected to the Freq Out jack of the MVC Sampler.)

Since the WAV Oscillator is sample-based and not a simple oscillator, it features **Note** and **Gate** inputs which must be connected to the corresponding MVC Sampler outputs.

The **Note** input provides MIDI note numbers which facilitate the WAV Oscillator's key-mapping functions (these will be familiar to any experienced sampler user).



The **Gate** input lets the WAV Oscillator know when a new note comes in, so it can trigger sample playback. For the same reason, **envelope generators** used in conjunction with the WAV Oscillator should obtain their Gate signal from the **To EG** output of the WAV Oscillator, instead of directly from the MVC module, as is normally the case. This allows the WAV Oscillator to ensure that the triggering of envelope generators is properly synchronized with the triggering of samples. (Esync connections between the envelope generators and the MVC Sampler are made in the usual way.)

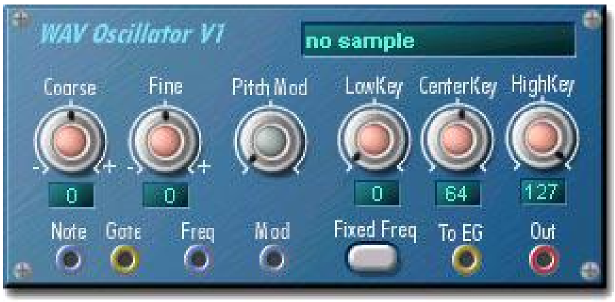
Once you've made these connections – plus of course an audio Out connection and optionally a Pitch Mod connection – *and* loaded a sample (don't forget!), you can use the key-mapping controls of the MVC to put the sample where you want it on the keyboard, and the tuning controls to further adjust its tuning.

CenterKey specifies the MIDI note number on which the sample will be played at its original or "native" pitch – that is, assuming that the tuning controls on the WAV Oscillator *and* on the MVC Sampler are all set to nominal. **LowKey** and **HighKey** set the lower and upper limits of the keyboard range over which the sample plays. Notes outside of this range produce no response.

The CenterKey value is *not* restricted to the LowKey-HighKey range. This may not seem entirely logical at first. However, all it means is that on every key in the key range in which the module responds, the sample is played back transposed from its original pitch.

Alternatively, if the **Fixed Freq** mode is activated, the sample will play back at its original pitch on all keys in the LowKey-HighKey range (again, disregarding tuning adjustments).

Fixed Freq mode is useful for creating percussion maps – as it allows the keyboard layout to be arranged without affecting sample tuning – for layering a fixed-pitch sample over a different sound (whose pitch may or may not be fixed), and for mapping a sample to multiple keyboard notes to make it easier to play or to conform to preexisting MIDI layouts (GM drum maps, sound effects, etc.).



As to the tuning controls – note that **Coarse** (semitone) and **Fine** (cents) controls appear both on the WAV Oscillator and on the MVC Sampler module. If you're using multiple WAV Oscillators connected to a single MVC Sampler, the MVC Sampler tuning controls affect all of the oscillators in common – thus serving as master tuning controls – while the tuning controls on each oscillator of course affect only that oscillator, permitting individual sample tuning corrections as well as the creation of detune effects or intervals.

- Filters

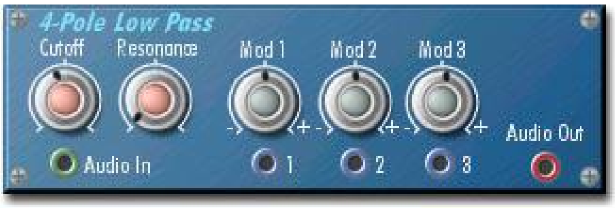
Operation of the filter modules is very similar from one type to the next.

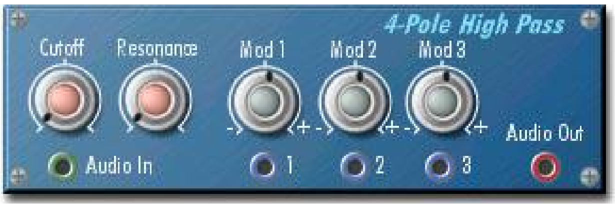
Each one has an audio input and output, a **cutoff** control which adjusts the filter's cutoff or passband-peak frequency, and three **modulation inputs**, each with its own depth control. Most also have a **resonance** control which adjusts the amount of signal boost or "ringiness" around the filter cutoff / peak frequency, and which can bring the filter itself right to the edge of oscillation.

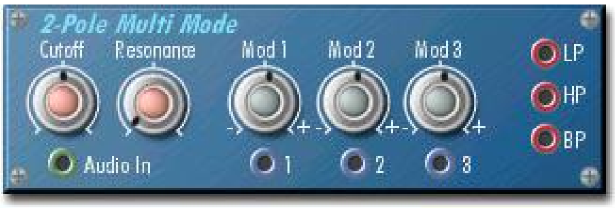
The **4-Pole Low Pass** filter lets low frequencies pass and reduces higher-frequency signal content by 24 dB per octave above the cutoff frequency.

The **4-Pole High Pass** filter does the converse, letting high frequencies pass and reducing lower-frequency signal content by 24 dB per octave below the cutoff frequency.

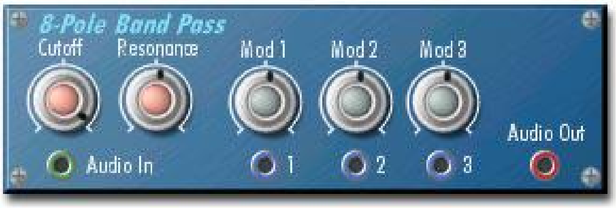
The **2-Pole Multimode** filter provides simultaneous low-pass, high-pass and band-pass filtering via separate outputs. The low-pass and high-pass functions cut off at 12 dB per octave beyond the cutoff frequency, while the band-pass function slopes off at 6 dB per octave on either side of the peak frequency.



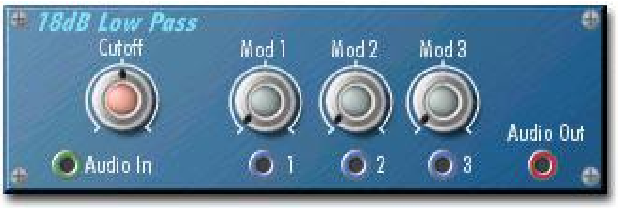




The **8-Pole Band Pass** filter allows a narrow band of frequencies around the passband peak frequency (adjusted via the Cutoff control) to go through, rather like an extremely exaggerated wah-wah pedal.



The **18dB Low Pass** filter has a more gentle cutoff slope than the 4-Pole Low Pass. In addition, this filter is not a resonant filter and thus has no resonance control. It's useful for more subtle high-frequency rolloff applications where the "synth-filter" sound is not desired.



- Envelopes

Overview

Envelope Generator (**EG**) modules are triggerable, "one-shot" generators of control signals which follow a defined course. They're usually used to control amplitude and filter cutoff but can be connected to virtually any control input to produce pitch envelopes, pan envelopes, etc.

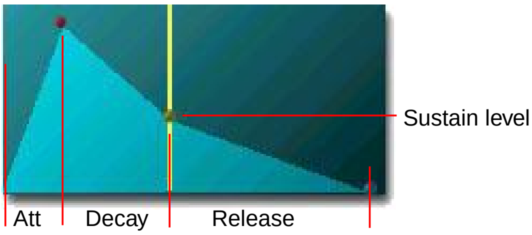
EGs are triggered via their **Gate** inputs by **gate events** (gate-on and gate-off). Gate events are usually derived from **MIDI note-on and note-off events** via the **MVC (MIDI Voice Control) module**. The Gate input of an EG is normally connected to the Gate Out of this module.

Gate events can also come from other sources such as the **MIDI Clock module**, which continuously generates **repetitive gate events** on its own (either in free-run mode or synchronized to an incoming MIDI clock) for automatic rhythmic envelope triggering.

The **Esync output** is a feature of most SCOPE EGs. EGs which have this output will function normally without any connection to it. However, for **proper polyphonic voice control**, it *must* be connected to the **Esync input of the MVC module** (either directly or via an Esync Adder module). This is discussed further below. (For more details, refer also to the *"More About Esync"* section of this chapter, or to the description of the MVC module.)

Envelope Generator Types

SCOPE Modular Synth EGs have two main distinguishing points: Each one is either an **ADSR or Multisegment** EG, and each one produces either **unipolar or bipolar** outputs. These points are explained below.



ADSR (Attack-Decay-Sustain-Release) Envelopes

The **Attack** phase begins when the envelope is triggered by a gate-on event. In this phase the envelope level rises from zero up to its (fixed) **peak level** at a rate determined by the Attack control.

When the peak level is reached, the **Decay** phase begins. The Decay knob sets the speed at which the envelope drops from its peak level down to the level set by the **Sustain** control. The amount of time this takes depends on the Sustain level setting. If the Sustain level is set fairly high, it is reached relatively quickly. If Sustain is set to maximum, this occurs virtually instantaneously and the Decay phase is essentially skipped.

When the Sustain level is reached, the **Sustain** phase begins. The envelope holds at this level until a gate-off event occurs. Naturally, this phase has no specific duration.

The **Release** phase begins when a gate-off event occurs. The envelope drops from its current level back towards zero at the rate set by the Release control. Release phase can also begin during Attack or Decay phase, if a gate-off occurs during one of these phases.

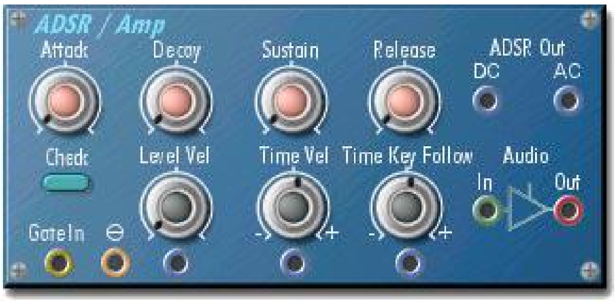
ADSR Module Common Features

The **Gate** input and **Esync** output are discussed in the Overview section on the preceding page.

Each of these modules features the basic **Attack**, **Decay**, **Sustain** and **Release** controls described just above.

The **TKF** (Time Key Follow) input is intended for connection to the Note output of the MVC module. This allows all envelope times to be lengthened or shortened in tandem in response to keyboard position. An associated control adjusts the depth and direction of this modulation.

The **Level Vel** input is intended for connection to the Vel output of the MVC module. This connection causes the envelope peak and sustain levels to vary in response to note velocity.



ADSR with Amp

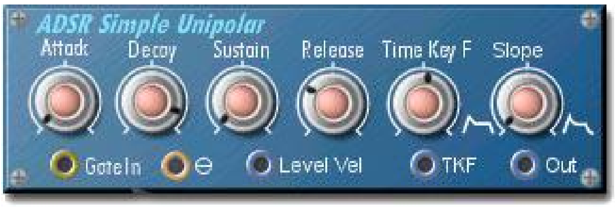
The **ASDR / Amp** combines ADSR and control amplifier (two separate control amps in the **ASDR / Stereo Amp**) within a single module – very handy, since amplitude envelopes find use in almost every Modular setup.

The envelope generator output is available directly via the **DC** and **AC** control outputs (the single output on the stereo module is DC). The DC output is for use with modulation inputs (such as on the VCA module) which accept only positive control values.

The **Check** button triggers the envelope directly on the module itself. This is convenient for testing envelopes without playing notes, but also opens up the performance option of (re)triggering individual envelopes independently of note events.

The Time Key Follow input and control discussed above are joined by a **Time Vel** input and control for adjustable modulation of envelope timings in response to note velocity info (from the Vel output of the MVC module).

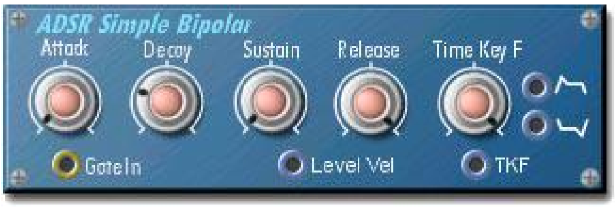
The modulation assortment is rounded out by the addition of a **mod depth control for the Level Vel** input.



ADSR Simple Unipolar

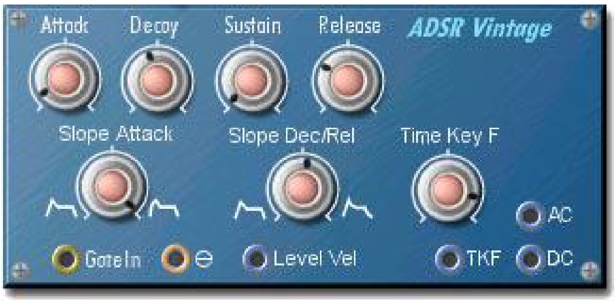
The special feature of this module is its **Slope** control, which affects the the shape of the decay and release portions of the envelope. It provides a continuous adjustment between **linear** (full counterclockwise) and **exponential** (full clockwise) curve shapes. Linear curves are the norm for Modular envelope generators. The exponential curves tend to have a smoother and more natural sound when **used to control amplitude**, especially during slow, extended decays and releases.

In fact, this module is mainly intended for amplitude control (via the VCA module). Therefore, its single output produces a DC signal (hence the name Unipolar), and it also includes an **Esync** output for connection to the MVC module (which coordinates polyphonic voice assignment on the basis of amplitude envelopes).



ADSR Simple Bipolar

This module is for **control of parameters other than amplitude**. It outputs simultaneous positive- and negative-going envelopes via two outputs. These envelopes are inverted versions of one another – both of them start and end at zero – and have linear curve shapes. In keeping with its intended usage, the module has **no Esync output**.

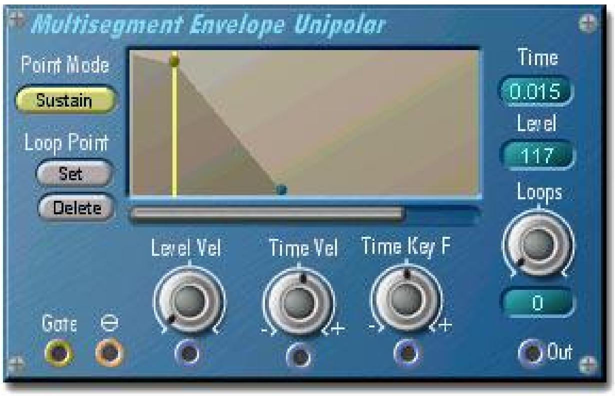


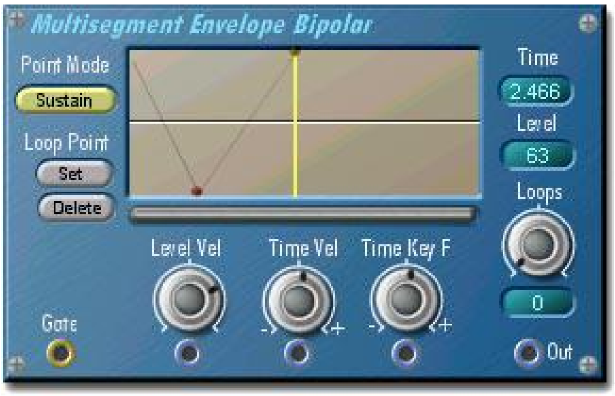
ADSR Vintage

This module expands upon the ADSR Simple Unipolar module by adding an **attack slope control** which varies the shape of the attack curve smoothly between linear and exponential. This is thus somewhat of an **all-around ADSR** – it offers linear curve shapes and an AC envelope output for control of pitch, filter frequency, etc., along with exponential curve shapes, a DC envelope output and an Esync output for amplitude control use.

Multisegment Envelope Unipolar And Bipolar

These are complex envelope generators permitting point-by-point envelope design. They are described in detail over the next few pages. The Unipolar, for amp control, has points with control values in the range 0-127. The Bipolar, for non-amp uses, has points with values -63 to 63, and correspondingly omits the Esync output.





- Graphical Envelopes

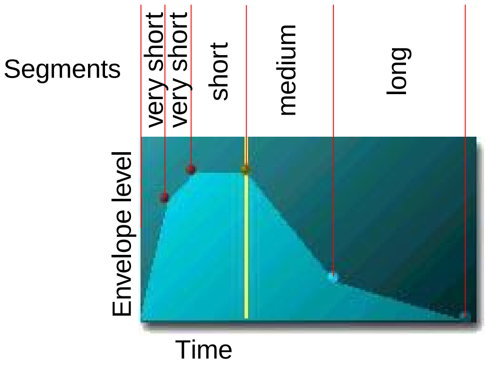
The graphical envelopes found in various SCOPE synths are all basically alike. The description here applies to all of them.

Segments And Points

A graphical envelope consists of a sequence of envelope **segments**. Envelope segments are defined using envelope **points**. Each point has a **value** (level) between 0 and 127 in a *unipolar* envelope. *Bipolar* envelopes also permit negative point values – the full range is -63 to 63.

A point's vertical position graphically indicates its value. In operation, the envelope begins at the left and rides up and down along the lines which are drawn between consecutive points. In the diagram, each envelope point sits at the right-hand end of its segment, since the point is where the segment ends.

Each segment has a **duration** (time). This is indicated graphically by the segment's width – the horizontal spacing between two points. Longer-duration segments appear wider than segments with shorter durations.



However, because **segment times can range anywhere from tiny fractions of a second up to ten seconds**, the horizontal time scale is adjusted individually for each segment, so that shorter segments are displayed proportionately wider (and longer segments proportionately narrower) than they would be if the same scale was used for all of them. This per-segment time scale "compression" yields a more usable overview with envelopes containing a wide range of segment times.

Edit both point value and segment time by dragging points with the mouse, or select a point and edit its level or time via knobs or direct keyboard entry of numerical values. The values/times of other point are not affected. **Add a new point** between two existing points by double-clicking in the space between them. You can have **up to 99** points. **Delete an existing point** by double-clicking on it. This deletes the associated segment and correspondingly cuts the segment's time out of the envelope as well.

Point Modes

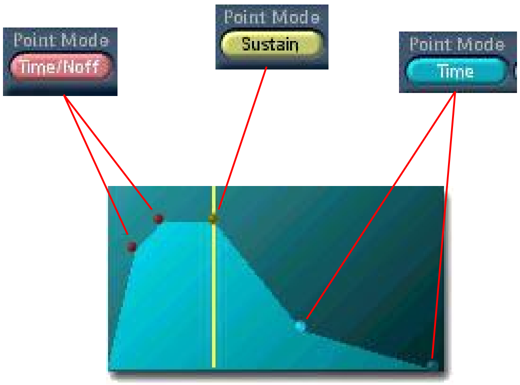
Clicking on the Point Mode button switches the selected envelope point (and its segment) between **three modes** which allow for versatile envelope construction:

Time mode: The simplest mode. The envelope always goes all the way to the segment endpoint. When the segment time is up – and only then – the envelope proceeds into the next segment.

Time/Note-Off mode: Same as Time mode, except that segments in this mode are cut short by a Note-Off (key release) event. Note-Offs also cause any Time/Note-Off segments which have not yet begun to be skipped.

Sustain mode: The envelope "pauses" at a Sustain point if no Note-Off event has yet occurred, and remains there indefinitely until the Note-Off occurs. Otherwise identical to Time/Note-Off mode.

In the display, the **mode** of each envelope point is **indicated by its color**. In addition, the Sustain point is visually highlighted by a vertical line running through it.



Using Point Modes

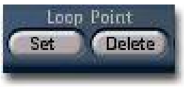
Note-Off events cause an envelope to skip "straight across" – without a level change – **to the first Time mode** segment (unless it is already in one) **and continue** from there.

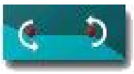
Time mode is thus useful in the release phase of an envelope (following a Sustain point) or for the last segment of any envelope (to ensure a non-zero release time and avoid note-off clicks). Envelopes consisting *entirely* of Time mode points are not affected by note-event duration – percussion envelopes are created in this way.

Time/Note-Off mode is normally used for attacks, decays and anything else coming *before* a Sustain point. It can also be used for envelopes with no Sustain point, such as piano/guitar-type envelopes, which decay steadily towards zero for as long as a note is held and cut off quickly if/when the note is released.

Envelope Loops

Envelope loops can be set up by clicking on the **Loop Point Set** button and then on two envelope points, which become the loop **start point** and **end point**. This is





indicated graphically as shown above. An envelope can have only one loop. Creating a new loop clears any existing loop.

Any number of envelope points can be included in a loop. However, **a loop cannot include a Sustain point**. A loop must therefore be either completely before or completely after the Sustain point, if the envelope contains one.

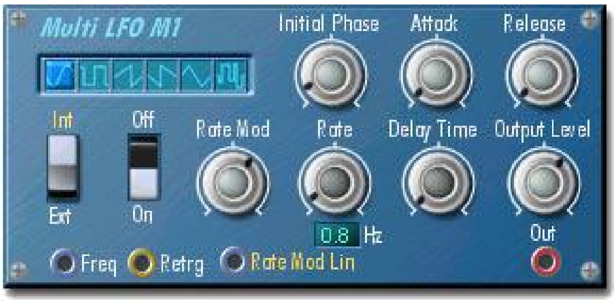
The **Loops** control lets you dial in a specific repeat count 1-255 ("0" = infinite repeats). The loop repeats this number of times, or until note-off, if it contains Time/Note-Off segments. A loop containing *only* Time mode segments *always* repeats the specified number of times.

Loops are played by jumping from the end point back to the start point (forward-only looping). The time setting of the start point is applied to the end-to-start transition.

A loop can be removed by clicking on the **Loop Point Delete** button and then on one of the loop points. This merely clears the loop – the points which were included in the loop are not affected.

- LFOs

Multi LFO / Multi LFO (Fast) M1 / Multi LFO b



These deluxe **LFOs** have six different waveforms (Sine, Square, Sawtooth Up, Sawtooth Down, Triangle and Random). These are selected by grabbing the blue slide switch and dragging it left or right.

The **Rate** control sets the base LFO frequency or speed (0.00-400.00 Hz for the Fast M1, 0.00-50.29 Hz for the others). The **Rate Mod Lin** input (M1 and Fast M1 only) permits external modulation of the LFO rate with a linear control characteristic. The associated **Rate Mod** control adjusts the modulation amount and polarity.

The normally free-running **LFO waveform can be reset** by gate-on events appearing at the **Retrg** (retrigger) input. The switch above this jack must also be set to On – set it to Off to use the LFO envelope (see below) with LFO retriggering disabled. The **Initial Phase** control sets the phase point to which the LFO is reset. Connect Gate Out of the MVC (MIDI Voice Control) module to this input to cause the LFO to be reset at the start of every note.

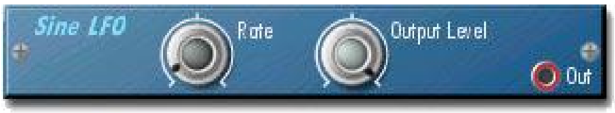
The Multi LFOs (excluding Multi LFO b) include a simple **envelope** which is likewise triggered via the Retrg input and operates only when there is a connection to this input. The **Delay Time** control adjusts the delay between the trigger (gate-on) event and the start of the LFO attack.

The **Attack** control sets the fade-in time of the LFO following this delay. The **Release** control sets the fade-out time following gate-off.

The LFO has two **MIDI clock sync** options which produce a one-cycle per beat (24 MIDI clocks) LFO speed with the **Freq** input connected to the Freq output of the MIDI clock module. For sync to an external MIDI Clock, the switch above this jack must be set to **Ext**. Setting it to **Int** allows sync to the Modular's internal MIDI clock.

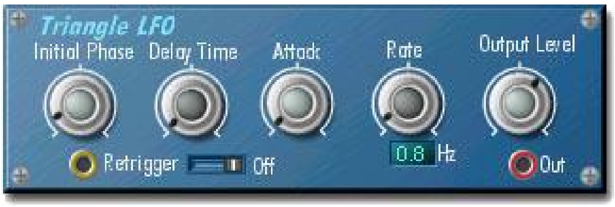
Finally, the overall output level of the LFO is adjustable via the **Output Level** control.

Sine LFO



This module is a good choice when the utmost simplest, continuous-running, unsynchronized, unmodulated sinewave LFO is all that is needed. It takes up less space onscreen, requires only a single connection and uses up less DSP power as well. Its controls include **Rate** for the LFO speed and **Output Level**.

Triangle LFO



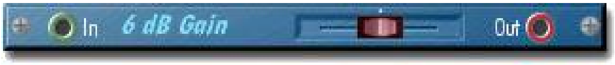
This is another fairly compact single-waveform LFO module with a mix of selected features from the Multi LFOs. It has an LFO envelope with **Attack** time and pre-attack **Delay Time** adjustments (the envelope has no release phase), plus waveform retriggering capability with **Initial Phase** control. A **Retrigger** input is provided to actuate both envelope and retriggering, along with a switch which disables retriggering only, allowing use of the envelope alone.

- Mix And Level

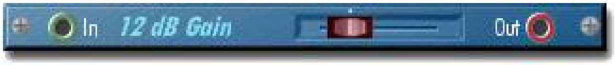
This is a set of simple modules for modifying signal levels. A small mixer is included in this group.

The **6dB Gain** and **12dB Gain** modules contain amplifiers providing a signal boost of x2 (+6dB) and x4 (+12dB) respectively. Each one has a slider which allows the output to be adjusted continuously all the way down to zero. These modules are often useful for boosting the output of a resonant filter module, as these filters reduce their gain internally when their resonance controls are turned up to avoid internal overloading.

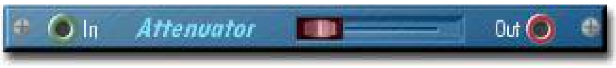
0 dB



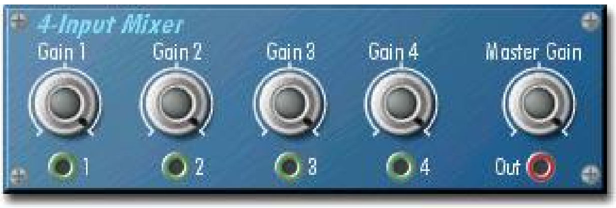
0 dB



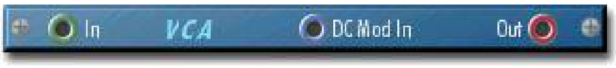
The **Attenuator**, like the above gain modules, includes a slider for adjustment of output level, but contains no amplifier. Thus, its output at the maximum slider setting is exactly equal to the input.



The **4-Input Mixer** provides a simple mixing function for up to four signals, such as multiple oscillators. It has a **Gain** (level) control for each input and a **Master Gain** control for adjusting the output level of the mix.



The **VCA** module is an attenuator whose gain is controlled by an external signal instead of a slider. It's intended mainly for amp envelope control. Its **DC Mod In** jack accepts the output of a **unipolar control signal** source such as the ADSR Simple Unipolar or the Multisegment Envelope Unipolar envelope generators.



- Switches

Switch modules allow the **selection of various destinations or sources for a signal.** This lets you build various configurations into a Modular setup which can be accessed quickly and repeatably without the need for recabling or for trying to remember "where that cable used to go?" or "what was connected here before?".

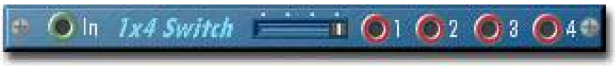
In addition to this basic convenience factor, switch modules also permit **signal routing variations** within an otherwise fixed setup to be **stored as presets**, since a Modular preset includes the switch positions of any switch module in a setup.

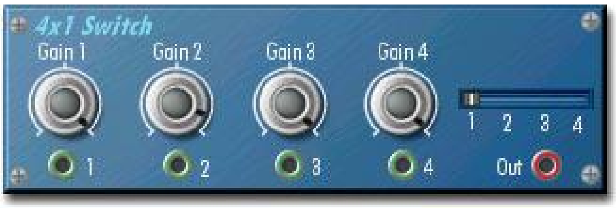
As a rule, both **normal audio signals and control signals** (LFOs etc.) can be routed using switch modules. These modules do not handle MIDI and Gate signals.

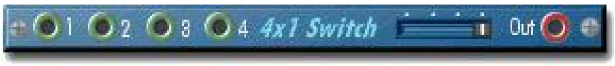
The **1x4 Switch** routes a single signal to one of four different destinations. For example, it could be used to apply an audio signal from the Audio In module to one of four completely different audio processing chains.

The **4x1 Switch with Gain** selects a signal from one of its four inputs. This module also has individual **Gain** controls on each of its inputs, letting you balance possibly unmatched levels on the different inputs to one another. This module could be used, for example, to select one of the three outputs of a Multi-Mode filter or the filter input signal instead (i.e., bypassing the filter) – or to select one output from a set of different Multisegment envelope generators, etc.

The **4x1 Switch** module is the same as the 4x1 Switch with Gain, but without the gain controls.





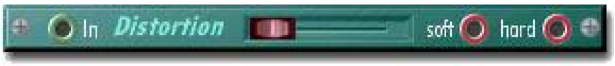


- Effects

SCOPE Modular effects fall into two main gategories.

Polyphonic effects are able to process the signals of multiple voices separately and preserve their "separateness" when used in a polyphonic setup. Behind the faceplate, polyphonic effects are actually **multiple copies – one per voice**. Therefore, they can be used anywhere in a polyphonic setup, but can be relatively "expensive" (in terms of DSP capacity consumed).

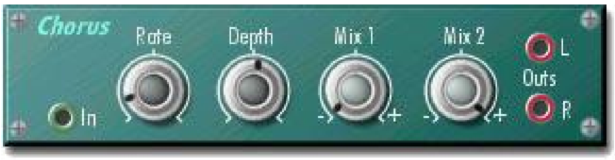
Monophonic effects (green faceplates) load only a single copy of an effect, but **defeat polyphonic voice control** beyond the point where they are used. In a polyphonic setup, they are best used following the Poly Out module. Most SCOPE Modular effects are of this type.



The **Distortion** module is available in both varieties and illustrates some of their differences. It produces soft- and hard-distorted versions of its input, with distortion amount adjustable via the slider.

The **green monophonic Distortion** module has an effect like that of an amp on a guitar – the *mix* of notes is distorted. They get mashed together into a single thick-sounding grunge whose texture is affected by each note. The **blue polyphonic Distortion** module distorts each voice separately, more like a group of lead guitarists each playing through their own amplifiers – the notes don't interact with each other at all and remain sonically distinct.

The remaining effects described here are **all of the monophonic type** unless otherwise indicated and should be used at the end of the line, *after* the Poly Out module.



The **Chorus** produces stereo output from mono sources. It's thus best used as the very last module before the outputs (i.e., *after* the Poly Out module!). The **Rate** control adjusts the speed of the internal sweep LFO. The **Depth** control adjusts the sweep amount. Each of two separately swept delay lines is mixed to one of the outputs. The amount and phase of delayed signal going to left and right outputs is adjusted by **Mix 1** and **Mix 2**, respectively.

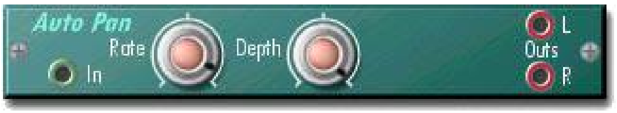
The **Auto Pan** module pans a mono source signal between its two outputs under control of an internal LFO. **Rate** and **Depth** controls for the LFO are provided. As with the Chorus, it yields **stereo output from a mono source** and **should be used after the Poly Out module**.

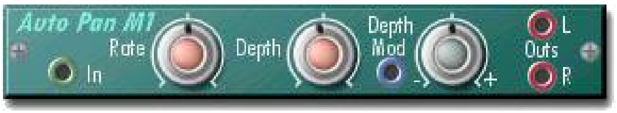
The **Auto Pan M1** module is like the Auto Pan module, but adds an input for **external control of pan depth**, along with an associated amount/polarity control.

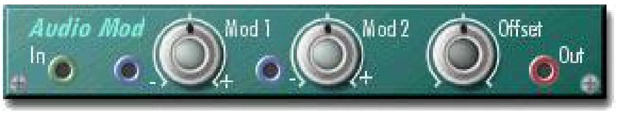
The **Pan** module has no LFO. **Initial Pan** can be set manually for static pan control. An input for **external pan modulation** is provided, with an associated depth/polarity adjustment.

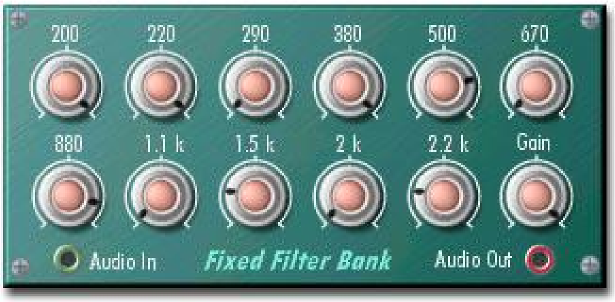
The **Audio Modulator** is similar to a VCA. The **Offset** control sets the initial gain, so that an adjustable amount of the input signal between zero and 100% is always fed to the output. The **Mod 1** and **Mod 2** inputs accept *bipolar* control signals which can then **increase or decrease the base gain** set by the Offset control. Each Mod input has an associated depth/polarity control. This module appears in both monophonic and polyphonic categories.

The **Fixed Filter Bank** is a set of boost/cut filters. Each can produce a gentle boost or cut within its own fixed frequency band, without strongly affecting other frequencies. **Gain** adjusts the overall output level.









MVC (MIDI Voice Control)

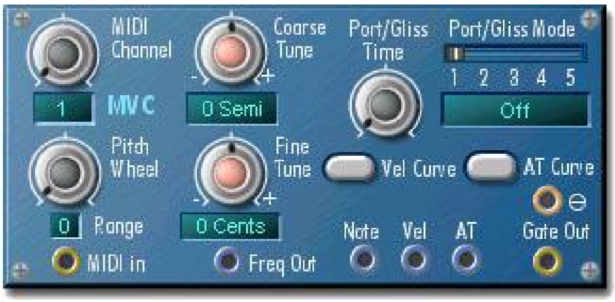
The **MVC** (MIDI Voice Control) module is the first line of processing for incoming MIDI messages. It manages MIDI-controlled voicing in the Modular Synth. Its **MIDI In** jack must be connected to the output of the MIDI In module.

NOTE: The MVC module is for use with *standard* oscillator modules. **When using the WAV Oscillator** (sample playback) module, **you must use the MIDI Voice Control Sampler** module (described following) in place of the MVC. Use of both types of MVC in one setup is possible.

The **MIDI Channel** control can be left on "Omni" (MVC responds on all MIDI channels) or set to confine MIDI response to a specific channel between 1 and 16.

Two or more MVC modules can be used within a single Modular setup to control separate voicing circuits. These will respond independently of one another if each MVC is set to its own MIDI channel. The MIDI channel setting can also be used to isolate the Modular from other SCOPE synths running in the same SCOPE setup.

Coarse Tune adjusts the pitch of controlled voices 4 octaves up or down in semitone steps. **Fine Tune** adjusts pitch in cents over a range of one semitone up or down.



The **Pitch Wheel Range** control sets the maximum pitch change of the controlled voices in response to incoming MIDI Pitch Bend messages.

The **Port/Gliss Time** (Portamento/Glissando) control adjusts the speed of up/down pitch glide from one note to the next which is activated when the **Port/Gliss Mode** switch is moved away from position 1 ("Off"). Both **Portamento** (smooth glide) and **Glissando** (glide in semitone steps) are available, each in two modes. In "plain" mode, the pitch glide between two notes always occurs. In **Fingered** mode, the glide occurs only when a new key is played before the previous key is released.

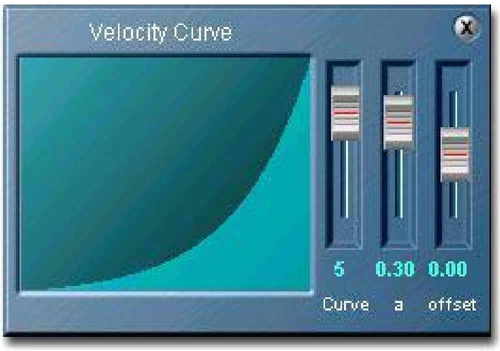
The **Gate Out** jack must be connected to Gate In of all envelope generators and LFOs which are to be triggered by MIDI note events.

Esync is a functional necessity for polyphonic voice handling (see "*More About Esync*" later in this chapter for more info). It must be connected to the Esync output of an envelope generator, typically the one which generates the primary voice amplitude envelope.

Freq Out is a control signal specifically intended for oscillator pitch control. It is derived from the combination of MIDI note number and pitch bend values, interpreted with respect to the current MVC settings for Coarse/Fine Tuning and Pitch Wheel Range, with possible MVC Portamento/Glissando effects thrown in on top. It must be connected to the Freq input of an oscillator in order to cause its pitch to respond to all of these factors in the usual way.

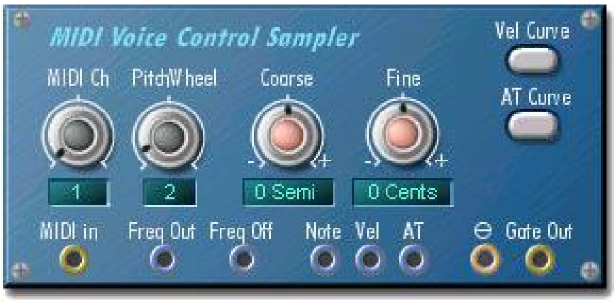
The **Note**, **Vel** and **AT** jacks send out control signals derived from MIDI note event Note Number and Velocity values and MIDI Aftertouch messages, respectively. These are in general intended for connection to correspondingly-named modulation inputs on other modules for keyboard tracking, velocity modulation etc.

The **Vel Curve** button opens a window for editing of a velocity curve which allows contouring of the response characteristic of the **Vel** output. The **AT Curve** button likewise allows editing of an aftertouch response curve for the **AT** output.



Editing of these curves is fairly simple. The **Curve** control at left selects one of six basic curve types, the **a** control at center produces variations on the chosen curve and the **Offset** control at right dials in a minimum value for the curve.

MIDI Voice Control Sampler



This module is similar to the "standard" MVC module, but has special features to support the **WAV Oscillator** sample playback module. This module *must* be used to control the WAV Oscillator – the standard MVC won't work.

Most aspects of the MVC Sampler are identical to those of the standard MVC. Only the differences between the MVC Sampler and the standard MVC are described here.

Naturally, the **MIDI In** jack must be connected to the output of the MIDI In module. This might seem too obvious to mention, but it's easily overlooked, since the MIDI In connection to the standard MVC is already made for you in the Empty Modular.

The **Freq Off** (frequency offset) output handles pitch control of the WAV Oscillator. The **Freq** input of the WAV Oscillator must be connected to *this* output, rather than to the accustomed Freq Out signal.

(The MVC Sampler has both **Freq Off *and* Freq Out** outputs, enabling it to **also control standard oscillators**. In fact, when using the WAV Oscillator, even if you wish to use standard oscillators along with it, you can delete the standard MVC altogether, if the other oscillators are to be layered onto the same MIDI channel as the WAV Oscillator, *and* if you don't need the Portamento/Glissando functions, which the WAV Oscillator does not support.)

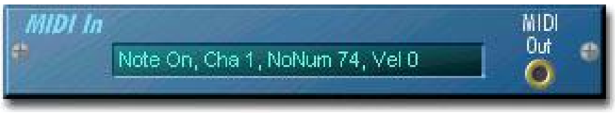
The **Note** connection from the MVC Sampler to the WAV Oscillator is required for correct keyboard pitch response (unlike with the standard MVC, where a Note connection is optional). **Gate Out** must go to the Gate input of the WAV Oscillator (which uses it to trigger sample playback), and *not* to an envelope generator.

Finally, as alluded to just above, the **MVC Sampler does not provide the Portamento/Glissando functions** found on the standard MVC.

For information regarding the WAV Oscillator, refer to the *Oscillators* section of this chapter of the manual.

MIDI In

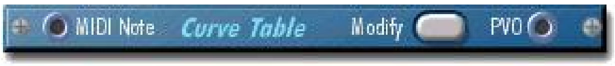
The **MIDI In** module provides MIDI input to the Modular. MIDI messages which are routed to the MIDI input of the Modular Synth from the SCOPE card MIDI input or from a sequencer program are "pipelined" into the Modular setup via this module and appear at its output jack, in addition to being displayed in its text window. (This can come in very handy as a diagnostic tool for MIDI problems.) This module is mandatory in a Modular setup and cannot be deleted.



Note On, Cha 1, NoNum 74, Vel 0

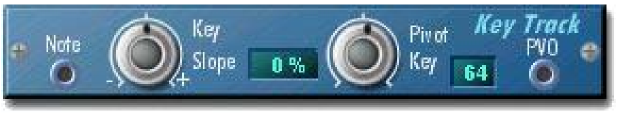
Curve Table

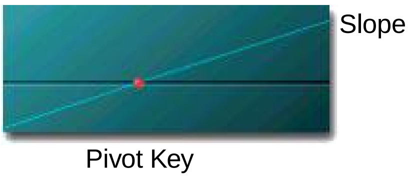
This module generates a control signal from **MIDI note numbers** (or other MIDI numbers such as controllers) by way of an editable transform curve similar to those offered by the MVC module for Vel and AT.



Key Track

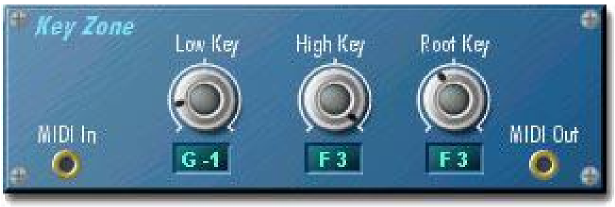
A simplified version of the Curve Table with a linear-only curve. **Pivot Key** sets the note number at which the module output is zero. The output rises steadily with increasing key number and drops steadily with decreasing key number (or vice versa), which is useful for such things as panning based on keyboard position. Rate and direction of the change are set by the **Key Slope** control.





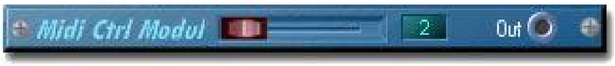
Key Zone

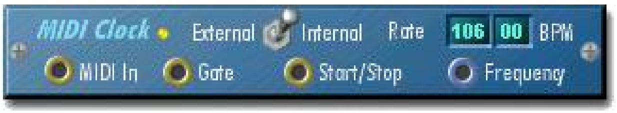
The **Key Zone** module filters MIDI note messages fed into its **MIDI In** jack. Only those messages whose note numbers are within the range set by the **Low Key** and **High Key** controls are passed through to the **MIDI Out** jack. The **Root Key** control allows for simultaneous transposition of the messages which get sent through, effectively shifting them up or down on the keyboard.



MIDI Controller

This module allows the use of **MIDI control change** messages as modulation sources in a Modular setup. Simply **assign a controller number** via the text fader, and then connect the output to modulation inputs as desired. The output can also be passed first through the Curve Table module to shape the control characteristic.



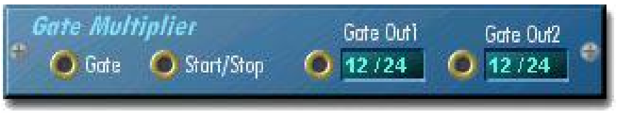


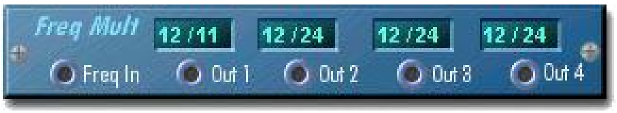
MIDI Clock

This module **translates external MIDI clock messages** received by the Modular into **tempo and gate messages** usable within the Modular setup. This requires that the output of the MIDI In module be connected to the MIDI In jack on this module).

Flipping the switch from **External** to **Internal** converts the module to a standalone MIDI clock source with tempo (**Rate**) adjustable via the text windows in whole and hundredth **BPM** (Beats Per Minute) according to the MIDI standard of 24 PPQN (clocks or pulses per quarter note).

In both modes, the **Gate** output produces one **gate on/off event per clock** for use with the Gate Multiplier module (see below). The **Frequency** output produces a control signal proportional to the tempo of the generated or detected MIDI clock stream. This is intended for connection to the Freq input of an LFO, in order to control its speed and thus **synchronize the LFO waveform to the MIDI clock** tempo. The **Start/Stop** output controls the Gate Multiplier module in response to MIDI Start and Stop commands from an external clock source.



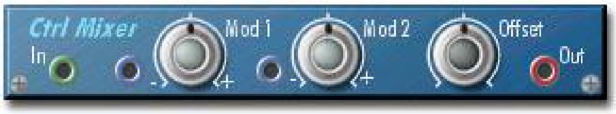


Gate Multiplier / Freq Multiplier

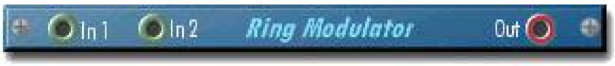
The **Gate Multiplier** module accepts a gate event stream from the MIDI Clock module and **divides it into slower streams** for synchronized envelope triggering, LFO retriggering, etc. It has two outputs with independent frequency settings. The **value to the right of the slash is adjustable** and sets the **number of input gate pulses** which are counted **for each output gate pulse**. With a 24 PPQN input from the MIDI Clock module, **setting this value to 24 yields one gate event per beat**. This module can also be controlled by the MIDI Clock module via the **Start/Stop** input in response to external MIDI Start and Stop commands.

The **Frequency Multiplier** module performs a similar function with respect to the Frequency output of the MIDI Clock module, permitting **four different simultaneous LFO speeds with waveforms synchronized to the same clock source**.

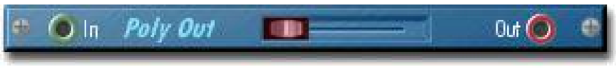
- Special



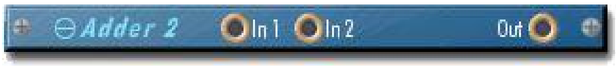
The **Control Mixer** allows amplitude modulation of control signals. For example, use it to apply an envelope to an LFO. The **Offset** control sets the base gain, thus controlling the amount of input signal fed to the output in the absence of modulation. The **Mod 1** and **Mod 2** inputs accept *bipolar* control signals which can then **increase or decrease the base gain** set by the Offset control. Each Mod input has an associated depth/polarity control.

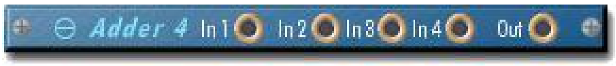


The **Ring Modulator** is a no-frills amplitude modulator effect which produces its output by simply multiplying its two (audio) input signals by one another – rather like a VCA whose control input is also a bipolar audio signal instead of an envelope or a steady DC level. Since there's no built-in gain offset, the Ring Modulator produces output only when there are signals present on *both* inputs.



The **Poly Out** module is a special-purpose module which is **required for polyphonic output from the Modular**. Its function is to properly deliver the signals stemming from multiple voices to the Modular output. Connect it inline **directly before the Audio Out module**. The **fader** can reduce the level of the combined multi-voice signal to **avoid digital overload**. This is often necessary, since each of the voices is otherwise individually capable of using up the full headroom of the Modular audio output. The more voices you use, the more you'll need to decrease the level.





Esync Adder 2 and **Esync Adder 4** permit **merging of Esync signals from multiple envelope generators** for return to the Esync input of the MVC module. Use these in particular when your setup includes **multiple amp envelopes**, to ensure that polyphonic voicing is correctly managed by the MVC. (For details about Esync, refer to the "*More About Esync*" section of this chapter.)

- More About Esync

Here's an explanation of the Esync (Envelope Sync) connection, for those who wish to know more about why it's there. If you're content simply to use it, feel free to skip over this part (but be sure to read the instructions regarding Esync in the "Overview And Ground Rules" section of this manual and elsewhere).

The Basic Issue

An important function of the MVC module is the coordination of voice reassignment or "stealing". Every synth has a specific number of voices as determined by its design. This limits the number of sounds it can make at one time. (This is actually not an issue with *real* modular synths, which are inherently monophonic anyway.) SCOPE synths let you specify how many voices they have, but are likewise limited by this setting. When all voices are in use and a new note is played, one voice must be "stolen" away from what it is doing to play the new note. Which one to steal? Ideally, the one least likely to be missed.

How SCOPE Synths Handle This

SCOPE synths resolve this question by scanning the amp envelopes of all voices to find the momentarily quietest voice. However, this often turns out to be one which has just been started and not yet heard (since notes often arrive in tight groups – e.g., in chords, on downbeats, etc). A preferable choice for stealing is a voice whose envelopes are already well along in the release phase, even if this voice is not absolutely the quietest one.

Why The Modular Synth Needs Help From You

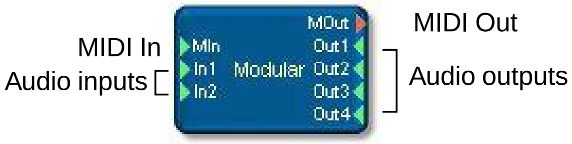
Most SCOPE synths are "hard-wired" – their structure is fixed. They use internal Esync connections to route the necessary amp envelope status info back to the internal equivalent of an MVC module.

The Modular has no fixed structure. A Modular setup may contain any number of envelopes, not all of which affect voice volume. SCOPE is not clever enough (not *yet*, anyway) to figure out which or how many ADSR modules should have Esync connections, so this is left up to the user to decide.

Can You Foul This Up Somehow?

What if you pick the "wrong" ADSR(s) for Esync hookup? Not to worry. You can't do any damage and you *may* not even notice a difference. Given the flexibility of modular setups, there could in some cases not even be a truly "right" choice. Most important is that *something* is connected to the MVC Esync input (even if only the MVC's own Gate output!) – otherwise, *nothing* at all will happen.

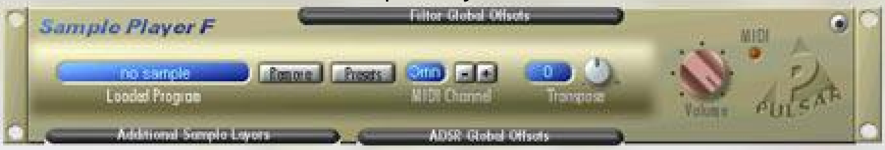
- Connections



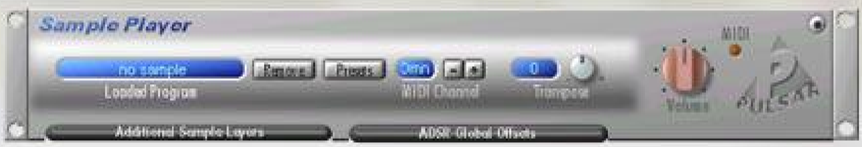
Project window (module) representation

Minimized (icon) representation

Sample Player F



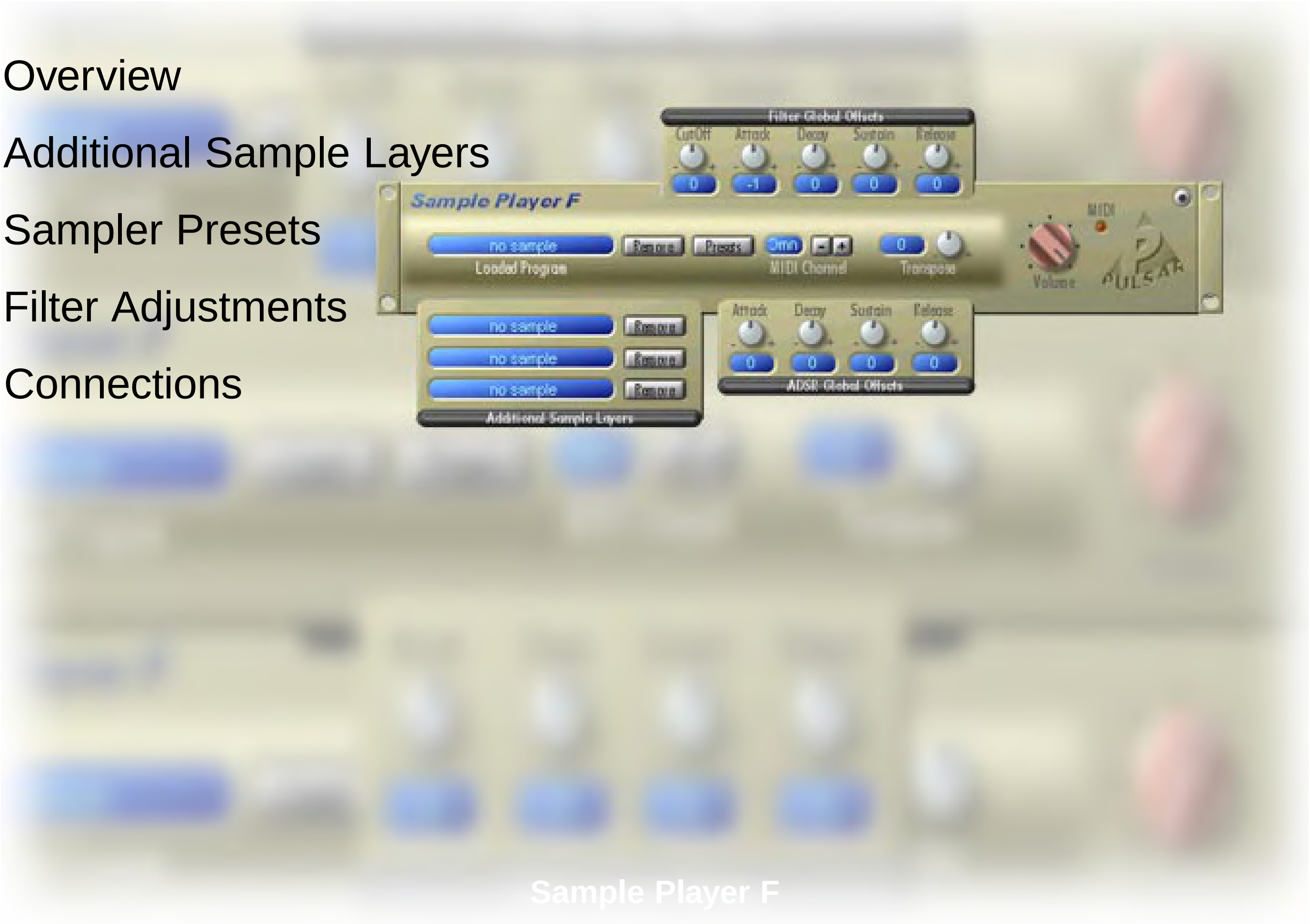
Sample Player



Sample Player

Sample Players

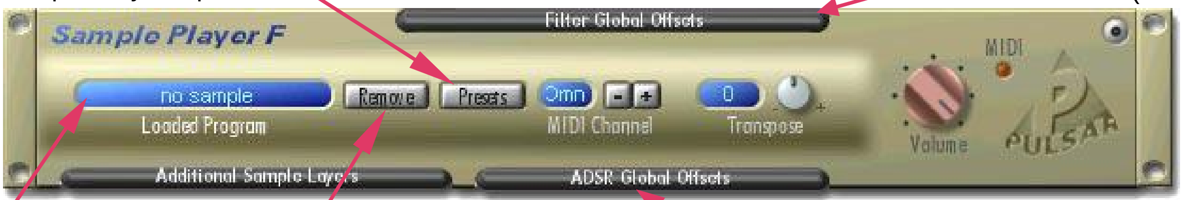
Sample Player F



- Overview
- Additional Sample Layers
- Sampler Presets
- Filter Adjustments
- Connections

Sample Player F

- Overview



Opens Sample Player F preset list

Filter offset controls (slide-out drawer)

Loaded Program window

Deletes loaded program

Amp envelope offset controls (slide-out drawer)

The **Sample Player F** can load and play complete **Akai S1000 programs** from S1000-format CD-ROMs. Via global offsets, it also allows you to modify amp envelope and filter parameters contained within these programs.

(The "regular" **Sample Player** is equivalent to Sample Player F, but allows no modification of program filter parameters – and places correspondingly smaller demands upon DSP capacity.)

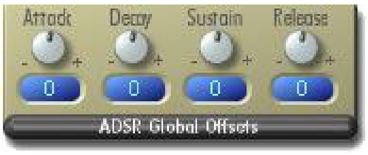
Programs are loaded from an Akai format CD-ROM via drag-and-drop of program (**P**) files from the **File Browser** onto the **Loaded Program** window of the Sample Player. Switch off the **S** button in the File Browser to hide sample files, which cannot be loaded individually into the sample player. This will make the program files easier to find.

The **Remove** button unloads the displayed program from the sampler.

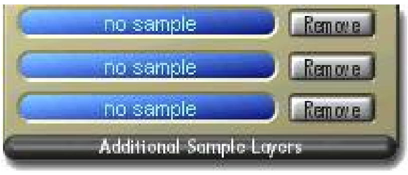
The Sample Players do not provide editing access to individual keygroup or sample parameters. The **Transpose** control affects everything in a program equally.

Existing amplitude envelope parameters are extracted from loaded programs and used by the Sample Players. These parameters can additionally be modified via global offsets which shift the corresponding settings in all amp envelopes up or down by the specified amount.

Naturally, if envelope *depth* settings in a program are set to minimum, or if envelope values in the program are set to one extreme or the other (or are not set at all), then modification of envelope settings via global offsets may produce little or no change.



- Additional Sample Layers



Opening this drawer gives you access to three additional windows into which programs can be loaded via drag-and-drop. Thus, up to four programs can be loaded at one time. Each window has its own **Remove** button.

All loaded programs play on the common MIDI channel in simple layered fashion. However, the envelope parameters from the additional programs are *not* used. Instead, the envelope parameters from the primary loaded program are applied to these layers as well (which is why they're called Additional *Sample Layers* rather than Additional Programs). The reason for this is as follows:

Akai programmers sometimes make use of of the S1000's ability to have multiple programs active on the same MIDI channel by creating sounds whose various components are split into a set of separate programs (some of the sounds in the samples CD which accompanies SCOPE are of this type). These programs are intended to be simultaneously loaded and active in the sampler in order to reconstruct the complete sound. Often, each of these programs incorporates velocity zone switching or some similar method to cause these programs to mesh smoothly with one another into a single coherent sound, instead of simply playing over one another.

Normally, these programs would all be loaded as a single volume. The purpose of the Additional Sample Layers is to compensate for the current inability of the SCOPE Sample Players to load complete Akai volumes. While the envelope parameters of the programs in the additional layers are substituted by those of the primary layer program, the keyboard mapping and velocity zone switching of the other layers *is* used in these layers. This allows such multiple-program sounds to be played properly by the Sample Players, as long as the envelope settings in all of the programs were more or less similar.

The results of this with randomly-combined programs depend completely upon the existing program keyboard and velocity maps, which are not editable in the SCOPE players. Only if the programs are preconstructed so as *not* to overlap one another on the keyboard will they be individually playable in separate MIDI note number ranges. Also, overloading and distortion is likely with programs combinations in which multiple layers produce sound at the same time. However, you can work around this situation by using more than one Sample Player at a time.

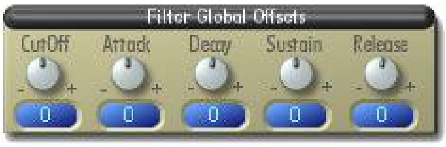
- Sampler Presets

Like other SCOPE devices, the Sample Players offer preset lists accessed via the **Preset** button. A sampler preset includes all front panel settings *and all loaded programs* as well. By using sampler presets, you can save yourself some File Browser work in the future, since the programs are reloaded automatically when you reload a preset. Keep in mind that sampler presets do not *contain* program files – you must have the CD in the drive for this to work.

Refer to the *Projects* chapter for info on using preset lists.

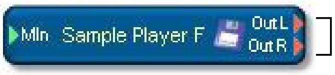
- Filter Adjustments

Sample Player F additionally permits **modification of the existing program filter parameters via global offsets**. As with amp envelope parameters, global filter envelope parameter offsets are effective *only* if the existing program parameters do not prevent this. However, offset adjustment of the filter cutoff frequency is always possible.



- Connections

MIDI In

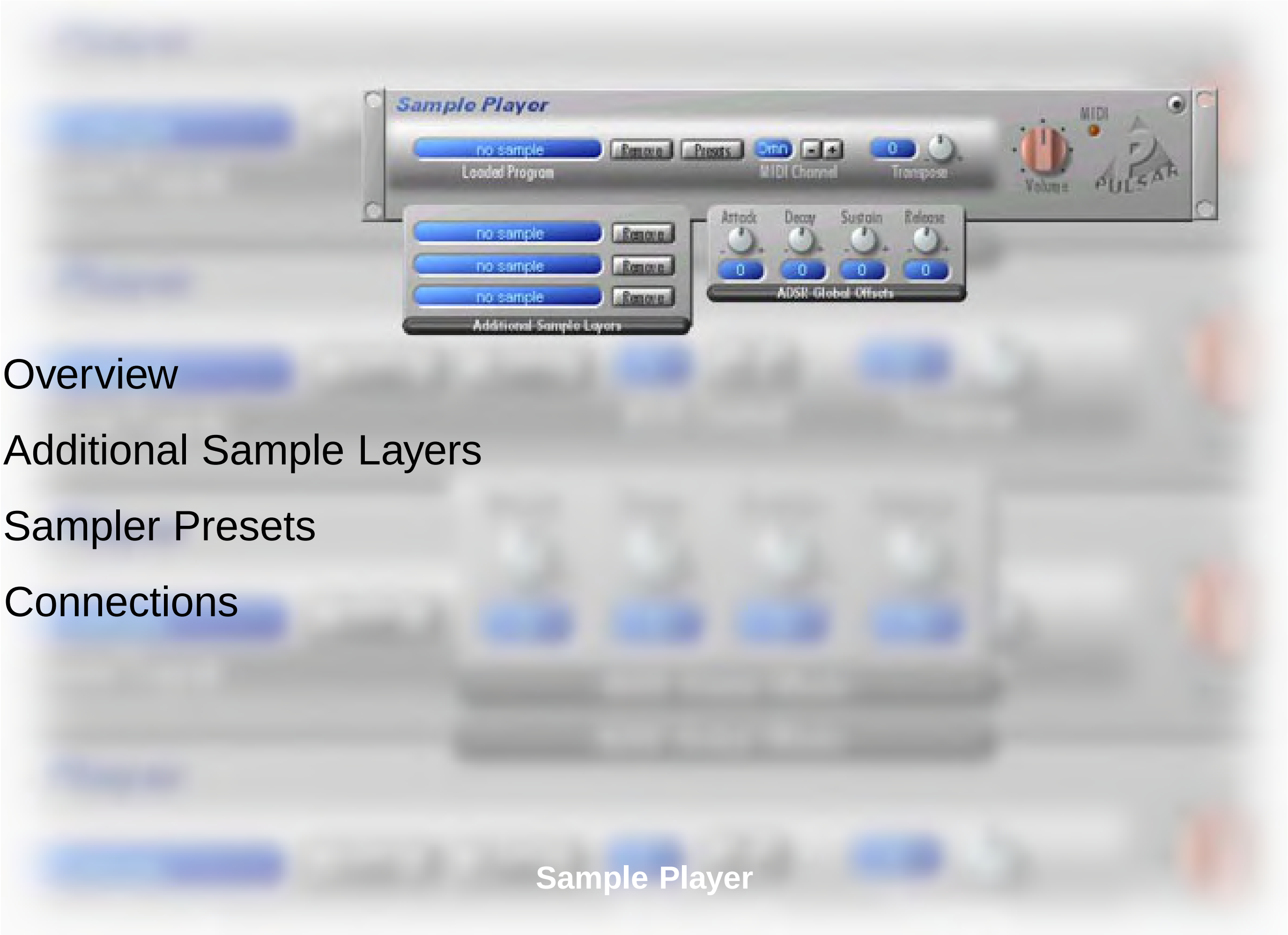


Audio outputs

Project window (module) representation

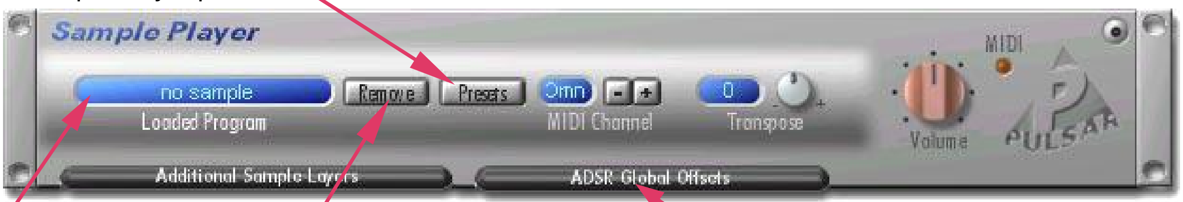
Minimized (icon) representation

Sample Player



- Overview
- Additional Sample Layers
- Sampler Presets
- Connections

- Overview



Opens Sample Player preset list

Loaded Program window

Deletes loaded program

Amp envelope offset controls (slide-out drawer)

The **Sample Player** can load and play complete **Akai S1000 programs** from S1000-format CD-ROMs. Via global offsets, it also allows you to modify amp envelope parameters contained within these programs.

(**Sample Player F** is equivalent to the "regular" Sample Player, but additionally allows modification of program filter parameters – and places correspondingly larger demands upon DSP capacity.)

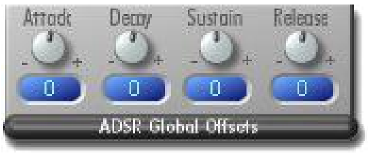
Programs are loaded from an Akai format CD-ROM via drag-and-drop of program (**P**) files from the **File Browser** onto the **Loaded Program** window of the Sample Player. Switch off the **S** button in the File Browser to hide sample files, which cannot be loaded individually into the sample player. This will make the program files easier to find.

The **Remove** button unloads the displayed program from the sampler.

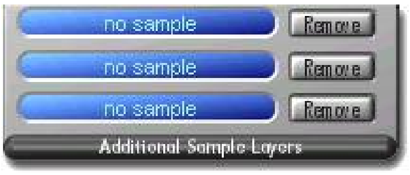
The Sample Players do not provide editing access to individual keygroup or sample parameters. The **Transpose** control affects everything in a program equally.

Existing amplitude envelope parameters are extracted from loaded programs and used by the Sample Players. These parameters can additionally be modified via global offsets which shift the corresponding settings in all amp envelopes up or down by the specified amount.

Naturally, if envelope *depth* settings in a program are set to minimum, or if envelope values in the program are set to one extreme or the other (or are not set at all), then modification of envelope settings via global offsets may produce little or no change.



- Additional Sample Layers



Opening this drawer gives you access to three additional windows into which programs can be loaded via drag-and-drop. Thus, up to four programs can be loaded at one time. Each window has its own **Remove** button.

All loaded programs play on the common MIDI channel in simple layered fashion. However, the envelope parameters from the additional programs are *not* used. Instead, the envelope parameters from the primary loaded program are applied to these layers as well (which is why they're called Additional *Sample Layers* rather than Additional Programs). The reason for this is as follows:

Akai programmers sometimes make use of of the S1000's ability to have multiple programs active on the same MIDI channel by creating sounds whose various components are split into a set of separate programs (some of the sounds in the samples CD which accompanies SCOPE are of this type). These programs are intended to be simultaneously loaded and active in the sampler in order to reconstruct the complete sound. Often, each of these programs incorporates velocity zone switching or some similar method to cause these programs to mesh smoothly with one another into a single coherent sound, instead of simply playing over one another.

Normally, these programs would all be loaded as a single volume. The purpose of the Additional Sample Layers is to compensate for the current inability of the SCOPE Sample Players to load complete Akai volumes. While the envelope parameters of the programs in the additional layers are substituted by those of the primary layer program, the keyboard mapping and velocity zone switching of the other layers *is* used in these layers. This allows such multiple-program sounds to be played properly by the Sample Players, as long as the envelope settings in all of the programs were more or less similar.

The results of this with randomly-combined programs depend completely upon the existing program keyboard and velocity maps, which are not editable in the SCOPE players. Only if the programs are preconstructed so as *not* to overlap one another on the keyboard will they be individually playable in separate MIDI note number ranges. Also, overloading and distortion is likely with programs combinations in which multiple layers produce sound at the same time. However, you can work around this situation by using more than one Sample Player at a time.

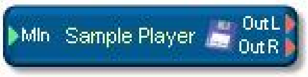
- Sampler Presets

Like other SCOPE devices, the Sample Players offer preset lists accessed via the **Preset** button. A sampler preset includes all front panel settings *and all loaded programs* as well. By using sampler presets, you can save yourself some File Browser work in the future, since the programs are reloaded automatically when you reload a preset. Keep in mind that sampler presets do not *contain* program files – you must have the CD in the drive for this to work.

Refer to the *Projects* chapter for info on using preset lists.

- Connections

MIDI In



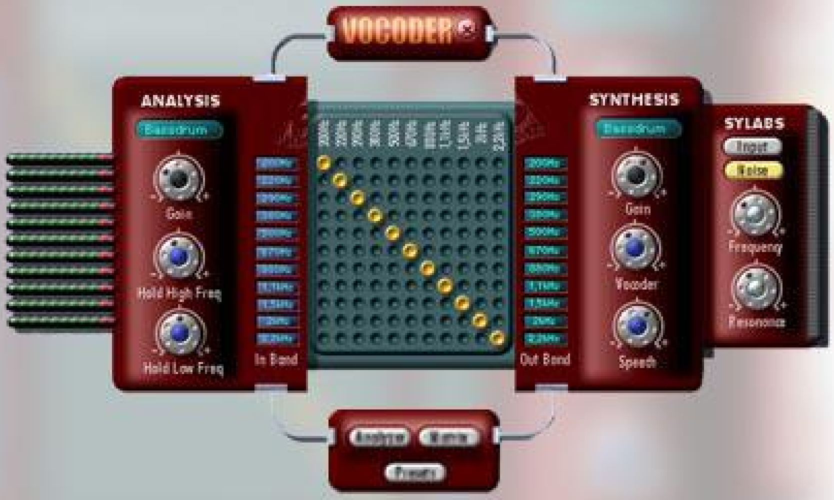
Audio outputs

Project window (module) representation

Minimized (icon) representation

Vocoder

- Overview
- Analysis
- Synthesis
- Connections



Vocoder

- Overview

The vocoder is located in the Scope\Date\Effects directory. It is especially useful for the processing of drum loops.

The **Analysis** controls at left apply to the signal to be analyzed:

Use **Gain** to adjust the level of the incoming signal. The Hold controls adjust the analysis properties of the vocoder.

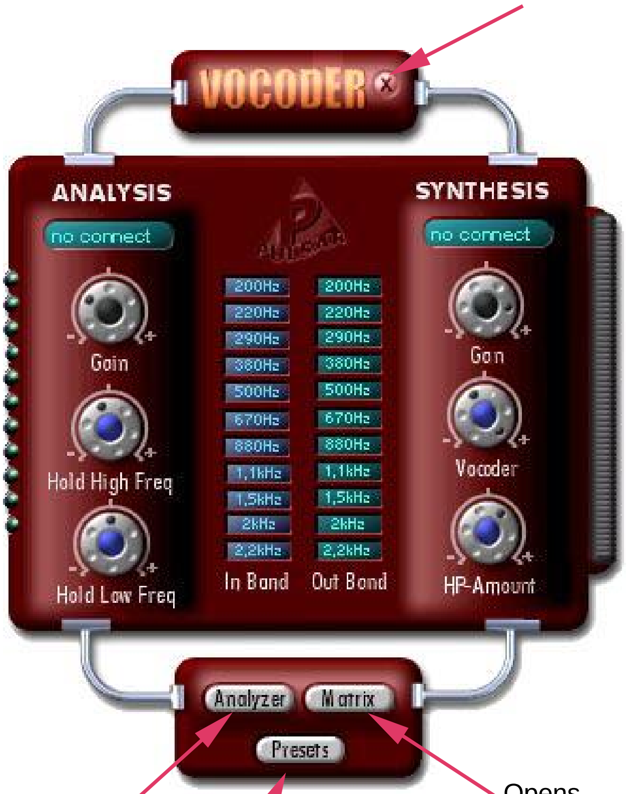
The assignment of frequency bands is displayed in the middle of the surface. The Matrix button at bottom center opens the editor for modifying the assignment matrix.

At right are the **Synthesis** controls:

The level of the signal to be processed by the vocoder is adjusted via the **Gain** control.

The **Vocoder** control adjusts the relative amount of vocoder effect in the vocoder output.

The **HP Amount** control adjusts the level of the output from the **High Pass** section.



Close button

Opens analyzer level display

Opens Preset List

Opens frequency band assignment matrix

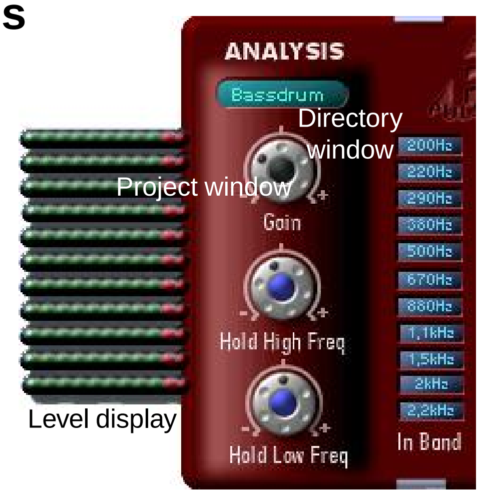
- Analysis

The input signal is divided into eleven frequency bands. The signal level within each of the bands is continuously analyzed. The **Level Display** shows the current level within each band.

The **Gain** knob controls the overall level of the signal coming into the analysis input of the vocoder.

The **Hold Low Freq** and **Hold High Freq** controls apply to the six lower-frequency bands (1- 6 at right) and five higher-frequency bands (7 - 11 at right) respectively. These are like speed controls for the level analysis within each band. Adjusting them towards the "-" end of the scale causes the vocoder to track level fluctuations in the associated frequency bands more rapidly and precisely. Moving them towards "+" slows down and smoothes out the tracking.

The frequency bands of the analyzer are assigned to the frequency bands of the synthesis section via a **matrix** with virtual jumper pins. By default, the matrix is set up as shown at right – each analyzer band is assigned to the synthesis band of the same frequency range. This arrangement can be "jumbled up" as desired by dragging the jumper pins sideways using the mouse. For example, if the pin in the 1.1kHz row is moved three steps to the right, then the frequency content of the input signal in the 1.1kHz band will affect the 2.2kHz frequency band of the synthesis section output.



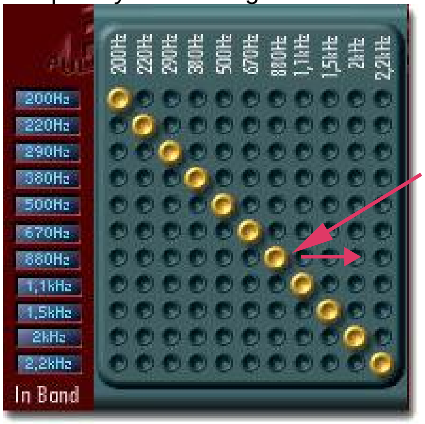
Directory window

Project window

Level display

Freq. band 1
Freq. band 2
Freq. band 3
Freq. band 4
Freq. band 5
Freq. band 6
Freq. band 7
Freq. band 8
Freq. band 9
Freq. band 10
Freq. band 11

Frequency band assignment matrix



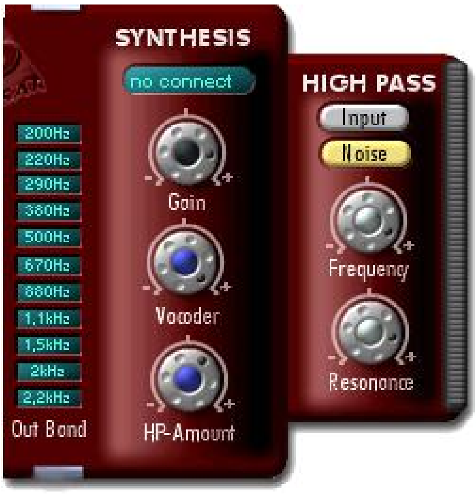
"Jumper" pins can be dragged sideways

- Synthesis

As with the analyzed signal, the synthesis input signal is divided into eleven frequency bands. The vocoder continuously controls the signal level within each synthesis band according to the signal level detected in the analyzer frequency band which is assigned to it.

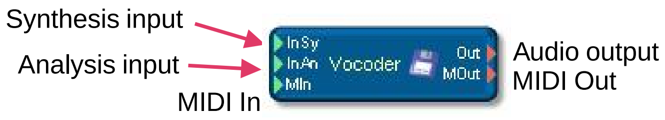
The **Gain** control on the Synthesis side adjusts the incoming level of the signal being processed by the vocoder. The **Vocoder** control adjusts the relative amount of vocoder-processed signal in the vocoder output.

The **HP Amount** control adjusts the intensity of an additional effect whose further controls are found in the **High Pass** drawer:

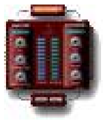


In addition to the “standard“ vocoder processing, the topmost analyzer frequency band (2.2kHz) – the “sybillance“ band – can control a separate signal path. As the input signal for this path, either the topmost synthesis band (2.2kHz) or a noise generator can be chosen. A resonant low-pass filter is included here to permit sonic tailoring of the “sybillance effect“.

- Connections

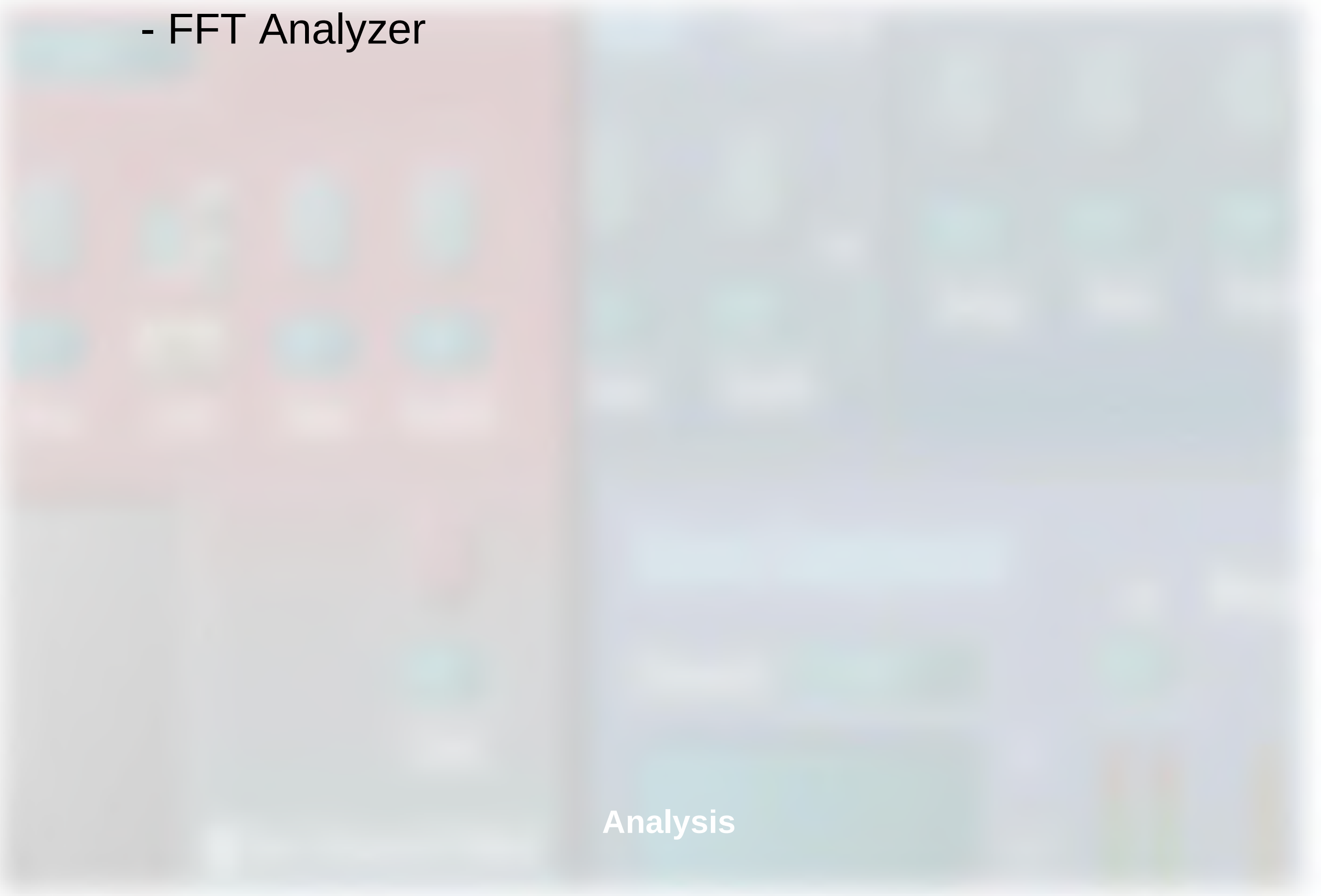


Project window (module) representation



Minimized (icon) representation

Analysis



- FFT Analyzer

The FFT Analyzer is located in the Scope\Data\Analysis directory.

The FFT Analyzer is a valuable instrument in the Test Instrument category. The FFT can monitor continuous audio signals and display the frequency spectrum accurately in real time. The FFT Analyzer does this with a greater degree of accuracy then traditional Filter bank analyzers. The FFT Analyzer reveals the finest details of an audio signal for you to explore.

FFT is an abbreviation for Fast Fourier Transformation, a mathematical construct based on the work of Joseph Fourier, the French mathematician. His theory is predicated on the assertion that any arbitrary signal can be reduced to a collection of pure sine wave components, each with its own frequency, amplitude and phase displacement. Fourier developed a transformation with which the representation of periodic signals can be transformed from the time domain to the frequency domain, and back. This theory would later be used also for analysis of aperiodic and digitally sampled signals. The Fast Fourier Transformation, developed in 1969 by Cooley and Tukey, is a more efficient refinement of the classic Fourier transformation in which the number of calculations required to complete a transform is greatly reduced.

Basic Fourier Transform theory assumes that the signal section subject to analysis is a complete cycle of a periodic waveform. For real signals, this is rarely the case. In practice, fixed length sections, or frames of the signal are analyzed in turn until the entire signal has been processed. The length of the section is determined by the desired spectral resolution - the greater the frequency range to be revealed, the longer the analyzed signal frame must be. In the continuous FFT, large sections of the signal are taken one after the other, transformed, and displayed on the screen. But since the sections probably do not match the frequency of the source signal, it is easy to see that the signal at the section boundaries will be discontinuous. The FFT interprets these discontinuities as impulses and superimposes them on the real spectrum, falsifying the expectedmeasurement result. This phenomenon is known by experts as ‘leakage’. To minimiz degree of distortion, each signal section is multiplied by special ‘window’ functions so that it fades to zero at each end of the frame. This way the length of the signal section is made to conform to the period of the FFT to allow continuous analysis while minimizing the leakage effects of a rectangular window. There are many different window types included in the FFT Analyzer. These should be chosen dependent on the nature of the signal, the quality to be emphasized, and by the requirements for leakage and selectivity.

Choosing the appropriate window type and interpreting the measurement results requires experience and background knowledge. If you are interested, we recommend that you read some of the technical literature on the subject. But even without a knowledge of the theory, after you've 'played' with it a while, the FFT Analyzer should be useful for tracking interference frequencies or helping to get the most out of the other available modules.

The FFT Analyzer Parameters

FFT Size: determines the resolution of the FFT, and is always a power of 2. Adjustable from 256 to 8192. Larger values result in a finer frequency resolution, but at the cost of computational overhead. If you have sampled at a 48 KHz sampling frequency an FFT size of 2048 will divide the frequency range (0 - 24 KHz) into 1024 spectral lines, each 23.44 Hz apart. This resolution is very good for high frequencies, but much too crude for the lower frequencies. A good frequency resolution for low frequencies requires adjusting the number of points to a higher value, which requires a great deal more calculation. Here is revealed a certain weakness of FFT analysis for logarithmically perceived signals. It is in the nature of human hearing that the resolution required to distinguish low frequencies is unnecessarily fine for the high frequencies. This derives from the linear nature of FFT analysis, but we must live with it. However, an FFT size of 8192 points results in a resolution of 5.86 Hz, which is fully sufficient for most situations.

Window functions

Window: Determines the particular window function to be used. The window should be selected depending on the particular measurement problem at hand. All window functions are a compromise between frequency resolution and amplitude precision.

None: No window (Rectangle). If a complete period (or multiple) of the waveform fits within the exact signal frame, then no discontinuity emerges at the section boundaries. This allows for the greatest frequency resolution. In the continuous analysis of musical signals this is virtually never the case.

Hanning: A good compromise between selectivity and leakage suppression in the 'side lobes'. However, it has a rather broad bell shaped curve around the target frequency

Rife-Vinc 1, 2, 3: These windows exhibit very good side lobe leakage suppression. With each increasing window number, the bell width is reduced at the bottom but increases at the top. The different compromises between frequency resolution and supression of adjacent frequencies are adjustable with the different variations.

Blackman-Harris: This is marked by a very steep bell, which supresses adjacent frequencies to about -80 dB. However it is also characterized by rather strong leakage.

Hamming: The Hamming window has little importance, but is included for the sake of completeness.

Flat-Top: This window has the characteristic that around the signal line at least two adjacent frequency ranges will also be stimulated by a single sine component. The amplitude of the signal can be read quite exactly, but the selectivity suffers.

FFT No Calc: Depending on the selected FFT size, the display is refreshed more often than you need. In this case you can specify the time interval of the display refresh.

Display parameters

dB Range: The left value determines the dB range of the FFT display while the right value determines the maximum of the displayed range. These parameters are determined automatically if **Auto Range** is enabled.

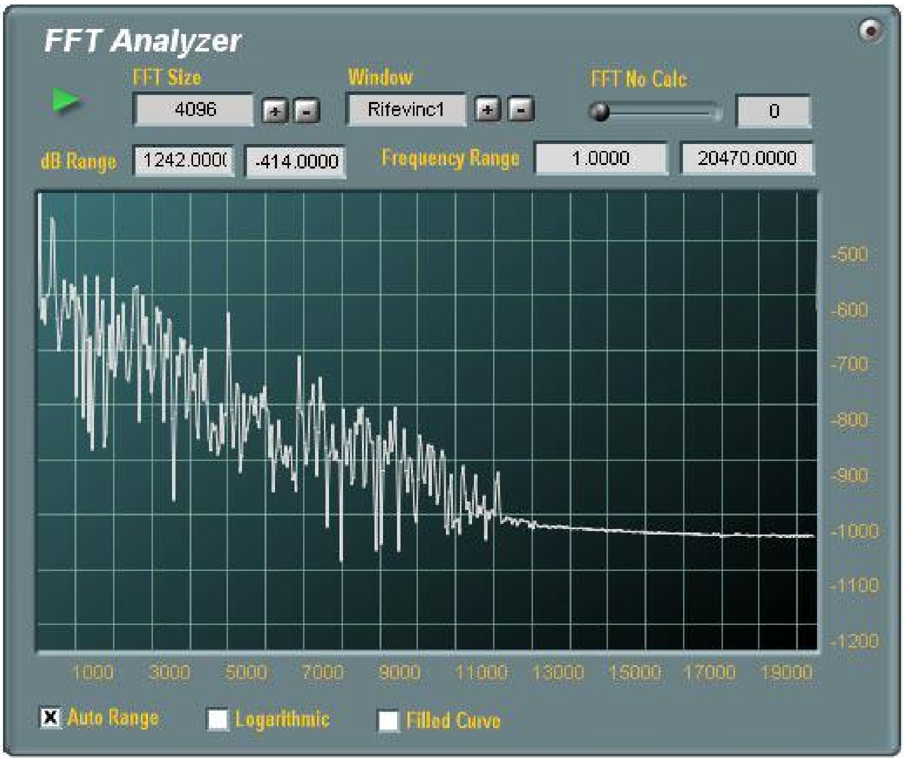
Frequency Range: Here you can specify the minimum and maximum values for the frequency range of the display. These parameters are determined automatically if **Auto Range** is enabled.

Logarithmic: Switches between linear and logarithmic scaling of the horizontal (frequency) axis. In log mode the display of the lower frequencies in a range are expanded, especially at small FFT Size values. This does not change the calculations - they remain linearly distributed. Only the display scaling changes.

Filled Curve: Changes the display so that the area under the curve is filled with color

Practical Guidelines

System-induced errors can emerge depending on the type of window function chosen, and the characteristics of the particular signal being analyzed. Readings can be off by as much as 3 dB. These errors are most pronounced when the chosen window is characterized by having a narrow ‘center lobe’ (e.g. a rectangle window) and the source frequency lies between two FFT frequency lines. A good general rule is: Choose a window appropriate to the value you want to ascertain. If your primary interest is in determining the frequency accurately use a Hanning or Rife-Vinc 1. If you are examining the amplitude, use a Flat-Top or Rife-Vinc 3.



FFT Size 4096

Window Rifevinc1

FFT No Calc 0

dB Range 1242.000(-414.0000

Frequency Range 1.0000 20470.0000

☒ Auto Range □ Logarithmic □ Filled Curve

FFT Analyzer surface

Quick Tour For The Designer

Return To Main Table Of Contents

Chapter Overview

This chapter takes you on a quick tour through the SCOPE workspace and its main features. There are no lengthy explanations or discussions here. This chapter will not teach you how to use SCOPE. The goal here is simply to familiarize you quickly with the essentials of the SCOPE environment, so that you have the knowledge you need to proceed further with learning how to develop circuits and devices in SCOPE – for which, by the way, the tutorials are the logical next step.

We recommend that you run SCOPE while going through this chapter (and later chapters as well) and directly examine and try out each feature or technique as it is introduced. By doing so, you'll become familiar and comfortable with the SCOPE environment in the shortest possible time. This will vastly speed up the learning process as you progress into the more advanced features. Hint: expand Acrobat Reader to full-screen – the manual is designed to be comfortably viewed in this format.

In creating this manual, we've tried to avoid lapsing too quickly into "SCOPE-speak". Instead, new terms are introduced as needed, and are not used before they've been "properly introduced". There are a few exceptions. For example, early on in this chapter you'll encounter the terms *device* and *module*. You don't need to know much about them at this point, other than that they are screen objects which you can manipulate in SCOPE.

So – here goes ....

Workspace Overview

This section lists and briefly describes the central features of the SCOPE workspace.

The Mouse

SCOPE is, by and large, a mouse-driven program. Apart from entering the occasional name or numerical value (and ignoring keyboard shortcuts for the moment), everything is done (or can be done) using the mouse.

Just for the sake of clarity, the "mouse rules" are reviewed below. They're basically the same as those you are already used to as a Windows user:

The **left** mouse button is always used, *unless* the right button is specifically mentioned. *Click*, *double-click*, *click-and-hold*, *click-and-drag*, *drag-and-drop*, etc. all involve the left mouse button.

The term *Click on (object)* signifies a single click on an object and is used interchangeably with the term *Select (object)*.

The **right** mouse button is used *only* to open context menus – i.e., menus belonging to individual objects (tools, modules, etc.), as opposed to the fixed program menus.

The Menu Bar

This is always present at the top of the SCOPE screen. The two buttons at the far right end of the menu bar perform the familiar functions of **minimizing** the SCOPE screen to a button on the Windows Task Bar and **closing** the program.

The Menu Bar provides centralized access to most of the features of SCOPE. Each individual menu function is described in full detail in a later section of this manual. Here is an overview of the menus:

The **File Menu** provides functions for loading and saving projects as well as for clearing the SCOPE project space and exiting from SCOPE. All SCOPE sessions begin here.

The **Edit Menu** includes standard program functions such as Copy, Cut, Paste, and Delete which operate on a selected item or using the contents of the Clipboard, along with several SCOPE-specific circuit editing commands.

The **Window Menu** provides access to various SCOPE windows and configuration tools, allowing you to hide them or make them visible on an individual basis. Many of these are the same as those provided by Pulsar. They deal mainly with program options and the *use* of finished devices, rather than with device construction.

The **Design Menu** lets you show or hide the various tools used in device and module construction and modification, so you see only the ones you need at the moment (there are quite a few!). These tools are powerful

but easy to use, *after* you've become familiar with the basic features and underlying concepts of SCOPE.

The **Help Menu** provides access to the Tool Tips and Help Window options – the SCOPE online help system.

The Project Window

This is the primary SCOPE construction area. The components of a project – the individual devices and modules – appear in this window when a project is loaded. Connections can be added and removed here, and creation and modification of modules and devices – on both the circuit and graphical levels – is also carried out primarily via this window. The name of the current project file appears in the top edge of the Project Window frame.

The File Browser

This special-purpose window lets you navigate file directories and access files of interest in SCOPE – modules, devices, samples, bitmaps etc. Items are dragged directly from the File Browser and dropped – usually into the Project Window or onto an object within the window, sometimes onto other tools. Common file operations such as copy, paste and delete and rename are also possible here. The File Browser is covered in depth later in this chapter.

The Toolbar

The Toolbar appears directly below the Menu Bar when SCOPE is started. It is a compact control panel which provides buttons for switching the most commonly-used design tools and options on and off. The selection includes all of the items in the Design Menu and the Help Menu. Whether you use the menus or the Toolbar to access these items is a question of personal preference, although the Toolbar is faster as a rule. The Toolbar can itself be displayed or hidden via the Window Menu.

Tools

These are control-panel objects which provide access to program settings or serve as tools for information display and object or circuit editing and other operations in SCOPE. Tools can be opened and closed via their entries in the Window, Design and Help Menus (and in most cases also via buttons on the Toolbar).

Any combination of SCOPE tools can be open at one time. With very few exceptions, actions performed using these tools take effect immediately (or are interactive), and the tools don't need to be closed or otherwise exited before other operations can be performed.

Tool Tips

This is part of the SCOPE online help system. It functions as an "information probe" for smaller details such as buttons and faders, or more importantly, individual module connection pads. Simply position the cursor over one of these items, and after a brief delay, a short informational message "bubble" will pop up next to the cursor.

Tool Tips can be helpful to you as a beginning SCOPE user by reminding you, for example, what each of the buttons on the Toolbar or the File Browser is for (since these don't have text labels). However, its main purpose is the display of long-format names for module connection pads – an important function indeed, as the short-format names which appear on the modules themselves are often rather cryptic and, especially with special-purpose signals, relatively difficult to remember.

Tool Tips is normally active when SCOPE is started. It can be switched on or off via the Toolbar or the Help Menu.

The Help Window

This is the "big brother" to Tool Tips – a tool featuring a window in which extended text messages are displayed. As with Tool Tips – but faster – these messages appear instantly when the mouse cursor is positioned over modules and module connection pads.

The Help Window can be considered as one of the design tools. It is not a general-purpose program help feature, but instead (like Tool Tips) provides information about individual circuit modules and their connection pads. In effect, it serves as an online module reference guide. We recommend that you leave it open while working with SCOPE and observe the information which appears there.

Utmost Basics

Starting SCOPE

Mentioned here just for the sake of completeness: Double-click the SCOPE icon deposited on the Windows desktop by the SCOPE installer.

SCOPE projects and project files

A SCOPE **project** is simply a SCOPE setup: the objects loaded into SCOPE at the moment, all of the connections between these objects, and all other information needed to fully describe the setup (screen positions of the objects, their hierarchical relationships, etc.).

A **project file** is a file containing all of this information. These files are identified by the extension ***.pro***. They allow the *complete* setup (but not its individual components) to be saved and later reloaded.

All operations which cause the current SCOPE setup to be discarded (*Open/New/Close Project* and exiting from SCOPE, as described below) give you a chance to save the setup first, if it has been modified since it was loaded or since the last time it was last saved.

Loading an existing SCOPE project

Click on *File* in the Menu Bar to open the File Menu, then on *Open Project*. A standard Windows *Open File* dialog appears, showing all *.pro* files in the Projects directory. Double-click on a file name or select it and click *Open*.

(**Note to Pulsar users:** loading projects by dragging them directly from the File Browser into the Project Window is *not* possible in SCOPE – you must open them via the File Menu.)

Exiting SCOPE

Open the File Menu as described above, then click *Close Program.* You can also quit SCOPE by clicking the *Close* button at the far right end of the Menu Bar, or using the standard Windows keyboard shortcut *Alt-F4*.

Other File Menu operations

Save Project saves the current SCOPE setup into the current project file – the *.pro* file from which it was loaded, or into which it was last saved.

Save Project As opens a standard Windows *Save As* dialog, allowing you to save the setup under a new or different name or in a different directory.

New Project clears the current SCOPE setup and loads the *default* project. This is simply the set of modules which appear in the Project Window when you start SCOPE. It provides a basis for starting a new project from scratch. You can customize the default project to suit your needs.

Close Project clears the current SCOPE setup, without

Handling Screen Features

Opening and closing windows and tools

Windows and tools are opened by clicking on the associated items in the Window Menu, Design Menu or Help Menu. These items normally appear dark in the menus. If a particular window or tool is already open, the corresponding item appears highlighted in the menu. Clicking this item then closes the window or tool. Some windows and tools also have associated keyboard shortcuts – these are indicated in the menus.

Windows and tools can also be closed by clicking their *Close* buttons (usually in their upper right corner).

Tip: SCOPE tools, windows and control surfaces can be left open indefinitely, in any combination. The smaller ones can sometimes become hidden behind other screen features. Often, the fastest way to get to a particular tool on a crowded screen is to simply close it and reopen it via its menu entry or via the Toolbar. This brings it all the way forward again.

Using slide-out drawers

Some SCOPE tools and device surfaces include "drawers" containing additional controls in logical groups. These stay tucked away out of sight when not needed, to keep the surface from becoming overly large or cluttered. In some cases, especially with Pulsar synths, drawers may also contain drawers of their own. When first getting acquainted with a new tool or device surface, be on the lookout for drawers and take a look inside any you find, so that you know what each drawer contains.

Drawer "handles" appear as dark grey tabs with labels on them to indicate what's inside the drawer. Click on the handle to pop the drawer open or slide it shut again.

Moving windows, tools and device surfaces

All SCOPE screen features other than the Menu Bar can be moved about onscreen as desired. This is done by simply dragging them with the mouse. You do need to make sure to click on a spot which has no other mouse-activated function. For example, clicking and dragging directly on a knob or fader will move *it*, but not the object it is a part of.

Tools and control surfaces can be moved by dragging any "featureless", non-functional point on the object. To move a window, you must drag it by the *top* edge of its frame. Dragging the other edges of the window frame will resize the window rather than moving it. Dragging *inside* the window frame often produces a window-specific action and is not effective for moving the window.

Tip: Objects become transparent while being dragged. This lets you see what you're about to cover up. It can also be a quick way to see what's currently hidden behind an object.

Resizing and scrolling windows

SCOPE windows, including the File Browser, can be resized by dragging the side or bottom edges of the window frame, or its lower right-hand corner. Other SCOPE program features, as a rule, are not resizable.

SCOPE windows, as with windows in any normal Windows program, can have horizontal and vertical scroll bars. These appear or vanish automatically as a window is resized (or as its content changes), depending upon whether the window content is too large to be viewed all at once with the current window size.

Minimizing and restoring windows and device surfaces

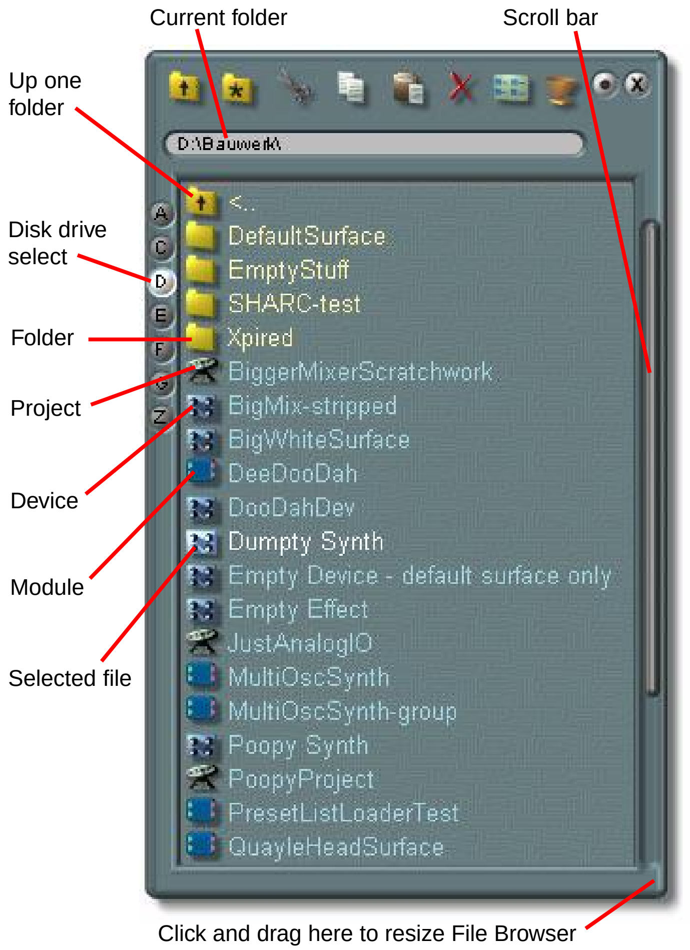
SCOPE windows and device control surfaces can be minimized, so that they appear only as small icons onscreen, by clicking on their *Minimize* buttons. The icons, which (optionally) are instantaneously-generated shrunken versions of the actual window or surface, show up at the bottom of the screen and can be dragged elsewhere as desired. The full-sized object is restored by double-clicking on its icon.

Tools are either open (visible) or closed (hidden) – they cannot be minimized.

Maximizing the Project Window

The Project Window features a *Maximize* button which expands it to fill the SCOPE screen, or restores it to its normal state if already maximized. Normal moving and resizing of this window is disabled while it is maximized.

Using The File Browser



The File Browser is a resizable window used to bring devices, modules and other types of objects into a project. (Don't confuse the File Browser with the File Menu, which is used for loading, saving and closing projects.)

Getting around in the File Browser

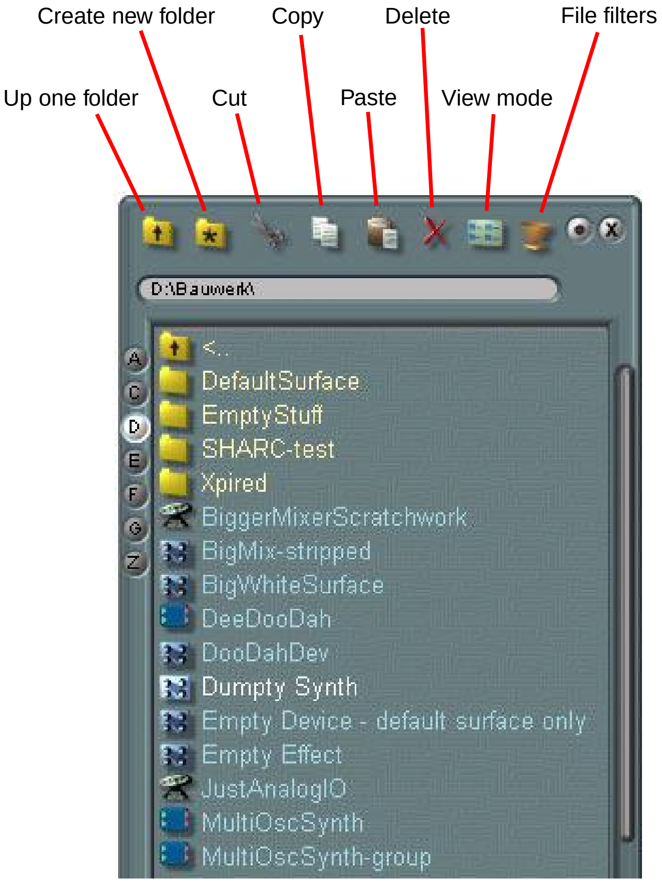
The browser displays folders and files in the currently selected folder. The name of this folder, including the full path which leads to it, appears in the field directly above the display window. If you like, you can click on this field at any time and type in a new path (followed by the *Enter* key) to switch directly to a specific folder.

If the current folder contains folders, these come first in the display, followed by the files, with each category sorted alphabetically. Folder names are shown in yellow, with a folder icon next to each. File names are blue and have various icons next to them, depending on the file type. To go "down" into a folder, double-click on its name. Double-click on the yellow "**<..**" at the very top of the list to go back "up" one level in the directory tree (to the "parent" folder of the current folder). As you navigate, the folder path name display above the display window is continually updated to show you where you are.

Tip: Instead of double-clicking on folder names, you can navigate more easily by single-clicking on the folder *icons* next to the names. However, to *select* a folder, you must click on the folder name and not on its icon.

The row of buttons along the left edge of the browser corresponds to the available disk drives in your system. Click on these buttons to switch directly from drive to drive.

File Browser function buttons



These buttons at the top of the File Browser (refer to the illustration at right) are similar to those in Windows Explorer and other Windows programs. They let you perform the most common file management tasks directly within SCOPE, as described below.

Tip: These buttons have icons rather than text labels (as is generally true of tools in SCOPE). If you have trouble remembering what a particular button is for, simply hesitate – with the mouse cursor over that button – and Tool Tips will automatically remind you.

Up One Folder takes you "up" one level in the directory tree to the "parent" folder of the current folder.

Create New Folder creates a new folder within the current folder.

Cut and ***Copy*** work together with ***Paste***. Use *Cut* and *Paste* to move a file or folder as follows:

– Select the file or folder in the File Browser.

– Click the *Cut* button.

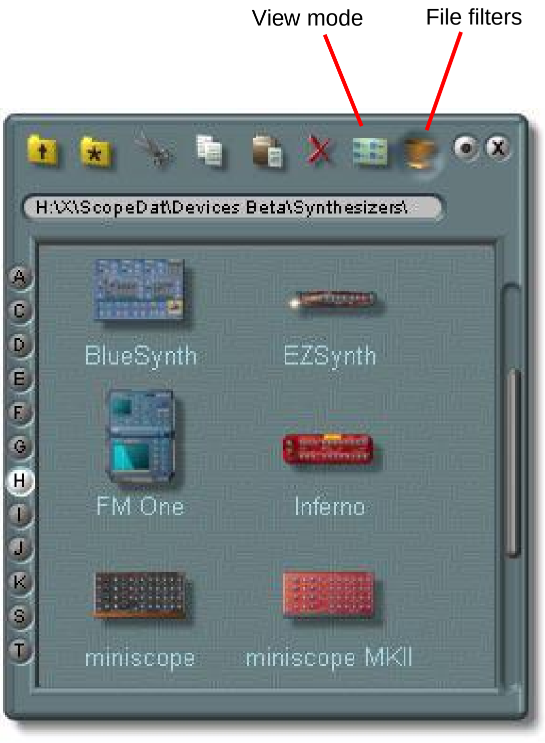
– Switch to the destination folder and click *Paste*.

If you click *Copy* instead of *Cut* in the above procedure, clicking the *Paste* button will copy the file or folder to the destination directory, leaving the original intact.

Delete deletes the currently selected file or folder.

The ***View Mode*** button changes the way the contents of the current directory are displayed in the File Browser. The available display modes are similar to those offered by Windows Explorer: *name list only*, *name list with small icons*, *name list with large icons*, and *auto-arranged large icons*. Click on this button to switch from one mode to the next. The default mode is *name list with small icons*, as depicted in the foregoing illustrations.

In the large-icon modes, SCOPE files are represented by icons which, in the case of devices with control surfaces (e.g., synthesizers), are miniature versions of the surfaces themselves. These are similar to the icons used in SCOPE to represent a minimized device or surface. An example of one of these view modes is shown at right. The same applies to graphical modules such as the control and display element and surface component modules found in the SCOPE module library, as well as to libraries of colors, gradients and surface textures, which makes this mode extremely useful for browsing these libraries.



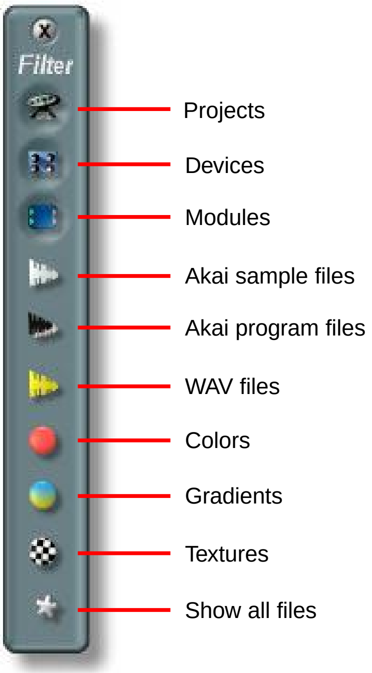
H:\X\ScopeDat\Devices Beta\Synthesizers\

View mode: *auto-arranged large icons*

The ***File Filters*** button opens an auxiliary button bar with buttons for switching various file types on and off in the File Browser display. This is discussed in detail below.

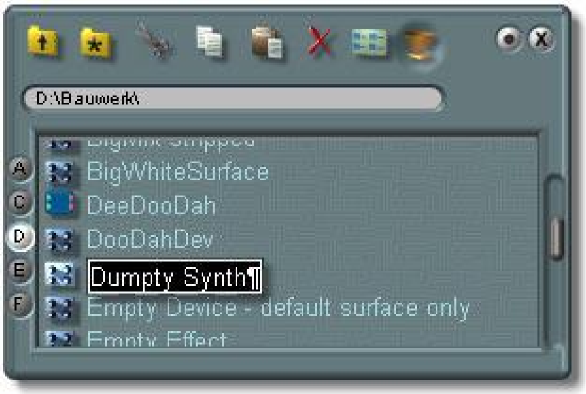
SCOPE file types – filtering the view

SCOPE uses several SCOPE-specific file types, which are distinguished via their filename extensions. You'll see these extensions if you look at SCOPE directories in Windows Explorer. However, they are *not* shown in the File Browser. Instead, a corresponding small or large icon appears along with each file name to signify its type.



By default, the File Browser view is already pre-filtered. It shows only SCOPE project, device and module files, plus folders. At times you may want to restrict the view further (e.g., to view modules only). This can be done via the buttons on the *Filters* button bar, which is opened by clicking on the File Browser's *File Filters* button. Click the file type buttons in the *Filters* button bar in and out to control which file types appear in the list (buttons appear indented when switched on). The icons on these buttons correspond to those displayed next to file names in the File Browser when the list with small icons view mode is selected (and as usual, Tool Tips can help jog your memory if you forget what a particular icon stands for).

The button at the bottom disables all filters and permits files of all types (including non-SCOPE files) to be displayed in the File Browser. This is a "temporary override" of the other buttons on the *Filters* button bar and does not affect the states of the other buttons.



Renaming files and folders

Files and folders can be renamed "in place" in the File Browser. This appears as shown at right. Simply select the file or folder and press *F2* to begin editing its name – or, right-click on the file and select *Rename* from the context menu which pops up. You can position the cursor as desired using the cursor keys to modify an existing name, or just begin typing to enter a new name. Save the name edit by hitting *Enter*, or cancel it by hitting *Esc*.

Bringing objects into a project via the Browser

Last but by no means least! This is done by dragging the name of the desired object – a device or module, a sample, a bitmap, a color file, etc. – directly from the File Browser window. The name itself doesn't move, but the cursor changes its shape when you drag it away from the name. This shows that it's "loaded" and ready to drop the object in any location which is appropriate for the particular type of object. Most often, you'll use this technique to drop devices and modules into the Project Window.

Tip: If the Project Window is empty and SCOPE

The Toolbar And The Design Tools

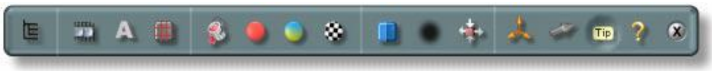
The Toolbar

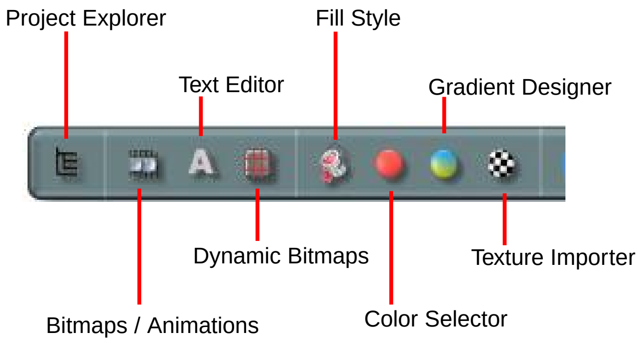
The Toolbar is a compact control panel whose buttons permit opening and closing (i.e., showing and hiding) the most often-used design tools, including Tool Tips and the Help Window, both described earlier in the Quick Tour.

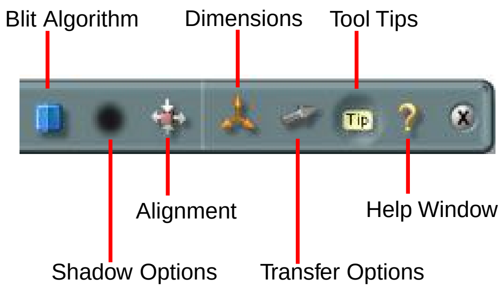
The Toolbar appears directly below the Menu Bar when SCOPE is started. The Toolbar is itself a tool which can be displayed or hidden via the Window Menu and repositioned onscreen as desired.

The button functions are indicated in the accompanying illustrations. They include all of the options in the Design and Help Menus, plus a few from the Window Menu. This makes the Toolbar a handy shortcut for these functions. The icons on these buttons correspond to those displayed next to the corresponding items in the various menus (and of course, Tool Tips is there to inform you – *if* you haven't switched it off). Buttons appear indented when their associated options or tools are switched on (in the images here, only the Tool Tips option is on).

Tip: SCOPE tools can be left open indefinitely, in any combination. Tools always appear on top of other screen features (e.g., windows and device surfaces), but they can become hidden behind other tools. If you're having trouble finding a particular tool on a crowded screen,





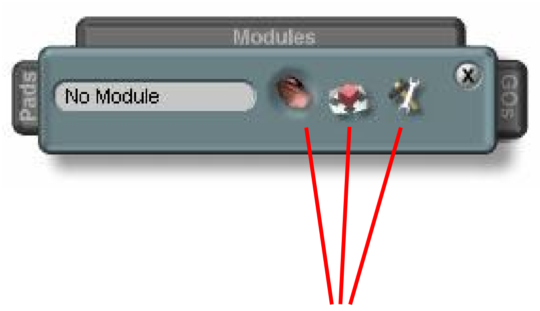


simply close it and reopen it via the Toolbar to bring it back to the top.

The Project Explorer tool and SCOPE modes

The Toolbar provides quick access to the design tools. Most of these are visual design tools – to which an entire separate chapter of the manual is dedicated. Aside from Tool Tips and the Help Window, both described earlier, there is one other Toolbar function which you should take good note of now: namely, the *Project Explorer* tool.

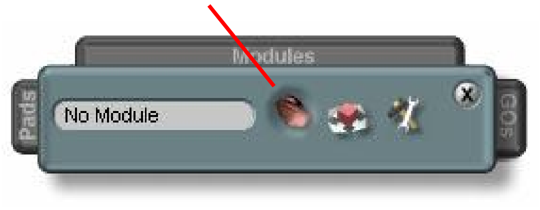
This tool is the central resource for the examination and modification of module circuits, devices and surfaces. It's deceptively simple-looking when first opened. However, it contains many drawers which display exhaustively complete information about the structure of a device – its internal hierarchy, the modules it contains, their connections and so on. As a rule, these displays are interactive – you can modify virtually any aspect of a device or circuit via one or another section of this tool. Owing to its extensive capabilities, the Project Explorer tool is described in detail in its own chapter.



Important to note now, however, are the three mode buttons on the small main surface of the Project Explorer tool. These modes – one of the three is always in effect – determine the behavior of objects in the Project Window and the operations on them which are possible at any time. You'll need to know how to use these buttons and modes as soon as you begin working in the Project Window. A brief description of these modes follows. Note that each one has an associated keyboard shortcut.

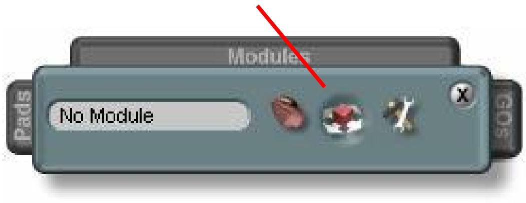
Use Mode (Ctrl-U)

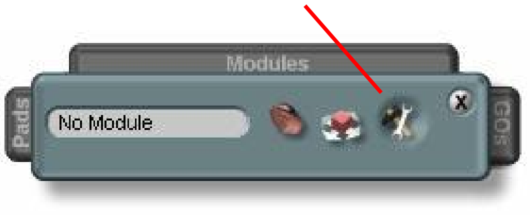
The default "user" mode. Modules and devices can be added and deleted via the Project Window, and connections between them can be modified, but they cannot be repositioned within the window. Faders, buttons and other graphical controls can be *operated* using the mouse, but graphical elements (controls, displays, labels etc.) cannot be repositioned on the surfaces they are attached to, nor in the Project Window, if not attached to a surface. It is not possible to go inside a device or module.



Move Mode (Ctrl-M)

Modules and devices can be added and deleted via the Project Window, and connections between them can be modified. Permits repositioning of modules and "unattached" graphical elements within the Project Window (but not, therefore, the *operation* of graphical controls), as well as entering a device or module to view and modify its internal structure. Device surfaces can still be operated normally, but Move Mode also permits the repositioning of graphical elements on a device surface, when working *inside* a device.





Edit Mode (Ctrl-E)

Permits direct manipulation of the individual graphical components from which screen objects are built. Most Use/Move mode operations are not possible in this mode.

The graphical design tools

The other tools accessible via the Toolbar buttons are concerned with graphical design and editing. It's not important to know much about them right at this point. As mentioned earlier, they are covered in depth in their own chapter of the manual, in addition to being briefly introduced in the tutorials. The overview below will give you an idea of the possibilities they offer.

The **Text Editor** is used for editing the text content of surface text elements, including labels on device surfaces, module and module pad names, and so on. It also provides extensive control over text appearance: size, font, rotation, alignment modes, etc.

The **Bitmaps/Animations** tool is used to apply single bitmaps and animation bitmap sequences to objects of the corresponding types.

The **Dynamic Bitmaps** tool is used to apply bitmaps to dynamic bitmap (resizable) objects and to adjust their appearance parameters, including whether or not they are used as "bump" or contour maps upon which colors, color gradients or texture maps can be overlaid.

The **Fill Style** tool permits selection of the fill style (single color, gradient or texture map or no fill style) for a dynamic bitmap object. It therefore determines whether or not any effect is produced through the use of the next three tools described following.

The **Color Selector** tool is a color-wheel color selector used to select the color of objects (including text) in single-color mode, or to select the colors with which a color gradient is produced.

The **Gradient Designer** tool permits design of custom color gradients (color fading pattern) parameters. These gradients can be applied to graphical text and dynamic bitmap objects. The Color Selector tool is used in conjunction with this tool for selection of gradient colors.

The **Texture Importer** tool is used to select texture maps to be applied to the surfaces of dynamic bitmap objects which are in texture-map mode, and to control the way in which the texture maps are applied. Specific types of bitmap files can be imported via this tool and optionally saved as SCOPE texture map files.

The **Blitter Options** tool is to control the manner in which an object is drawn onscreen. Among other things, this includes illumination mode effects and transparency values.

The **Shadow Options** tool is used to activate or deactive shadowing effects for individual objects, as well as to control shadow intensity, sharpness, and orientation for objects whose shadow option is enabled.

The **Alignment** tool is used to adjust the alignment and alignment behavior of objects with respect to their background surfaces – for example, to keep a button in the lower right corner of a resizable control surface when the surface is resized.

The **Dimensions** tool permits direct control over the dimensions and position of an object.

Finally, the **Transfer Options** tool permits single-click copying of any of the aforementioned graphical parameters of an object, individually or in any combination, to one or several other objects.

Basic Techniques

About This Tutorial

The goal: a quick (and easy) start

This tutorial is strictly hands-on. Rather than explaining things, it simply takes your hand and walks you through some basic exercises, followed by the step-by-step construction of a simple synth device. There are no pictures – we want you to create them yourself, *in* SCOPE. The procedures in the tutorial may seem a bit pedestrian at first, but they'll quickly become more interesting. If you follow them closely, you'll pick up quite a bit about how SCOPE works – without lengthy explanations.

To avoid loading you down with a lot of information before you can use it, this first tutorial focuses exclusively on the creation and modification of device circuits at the simplest level. The circuits constructed in this tutorial *do* function, but are not refined or especially practical. They're designed to illustrate the fundamental concepts of the SCOPE environment and SCOPE devices and the basic techniques for working with them. More advanced issues and techniques, in particular device surface construction and the extensive graphic design possibilties, are presented the next tutorial and in reference chapters.

The tutorial assumes that you've read the *SCOPE Quick Tour* chapter of the manual and have a basic familiarity with the main features of the SCOPE workspace – what they're called, what they look like and where to find them. In the course of taking the *Quick Tour*, you should also have spent a few minutes exploring the File Browser.

The tutorial also assumes that you have SCOPE installed and running, with some sort of audio monitoring system connected to one of the ADAT outputs of the SCOPE board and a MIDI keyboard or other MIDI controller connected to one of its MIDI inputs. We'll refer to these in the text as *ADAT Out* and *MIDI In*, although there is more than one of each – which MIDI or ADAT port you actually use is up to you.

On the *ADAT Out* port, we'll be using the *S1* and *S2* connections as "left audio output" and "right audio output" – your monitoring system should be adjusted accordingly.

A friendly but IMPORTANT WARNING!

In the tutorial, you will be constructing circuits which in some cases will produce sound immediately when the last connection is made. Up until this point, the SCOPE outputs will remain utterly silent, giving you no warning as to how loud the sound will be. This can naturally lead to some rather unpleasant or even nasty surprises. Therefore, we *strongly* recommend the following:

Before you begin the tutorial: For the protection of your ears (and your monitors), TURN DOWN THE VOLUME ON YOUR MONITORING SYSTEM NOW!

This warning will be repeated at various points in the tutorial where sudden loud noises could ambush you.

Basic Exercises

New Project: starting clean

Start SCOPE. If you've already been working in SCOPE (for example, if you've gone partway through this tutorial and want to skip ahead or start over), open the File Menu (click on *File* in the Menu Bar at the top of the screen) and select *New Project*. (or use the keyboard shortcut *Ctrl-N*). This clears the Project Window and loads a default set of input and output modules into the window. This is the operation to perform when the tutorial instructs you to open a new project.

Other initial details:

Make sure that the *Highlight Selected Object* option is enabled: click the corresponding Toolbar button in (this is the lightbulb icon at the right, between the *Tool Tips* and *Help Window* buttons).

Where to find modules and other stuff

Go to the File Browser and navigate into the folder ***SCOPE\Data***. (File Browser navigation is explained in the SCOPE Quick Tour). Here you'll find folders containing all SCOPE data files – synthesizers, effects, I/O devices, etc. Of interest for the device designer, however, are the folders *Circuit Design*, *Device Design* and *Surface Design*. Practically everything needed for creating SCOPE devices is to be found within one of these three folders.

Of particular interest for the tutorials are the folders *Circuit Design* and *Examples*. The *Examples* folder contains useful device examples which are well worth a long and close examination once you've gone through the tutorials (if not sooner). *Circuit Design* is where the circuit module library resides, again subdivided into folders according to category. In the tutorial, these folders are referred to using shortened names, e.g., *Envelopes*. Longer paths will be specified only for items not located in *SCOPE\Data\Circuit Design*.

Move Mode: the basic SCOPE working mode

Locate the Toolbar (at the top of the SCOPE screen, just below the Menu Bar) and click the button at its far left end to open the Project Explorer tool. If any of its drawers are open, click on their handles to close them. On the main surface of the Project Explorer tool, click the middle button to select *Move Mode*. This is the mode in which most circuit construction work in SCOPE is done. Move Mode is also conveniently activated at any time by hitting *Ctrl-M*.

Adding modules to a project

Navigate into the *Oscillators/Noise* folder. Drag the *White Noise* module into the middle of the Project Window and drop it there. That is: drag the filename "White Noise" from the File Browser. The name will not actually move from its position in the File Browser list when you do this, but the shape of the cursor will change to show that it has picked up this file, which you can then drop into the Project Window.

Making a connection between two modules

The *White Noise* module now appears in the Project Window. Position the cursor over the red output pad on this module – or, better still, over the *out* label next to this pad. The cursor shape changes to a black audio plug icon. Click once to select this pad. A red "X" now appears over the plug icon. This means: SCOPE wants you to move the cursor elsewhere – i.e., to another pad which can be connected to this one.

Move the cursor away. The black audio plug cursor changes to a red one, indicating that SCOPE is ready and waiting to complete a connection to the pad you selected.

Once again, **make sure the volume control on your monitoring system is turned all the way down!**

Position the cursor over the *S1* pad on the *ADAT Out* module (this is in the middle of the group of default I/O modules along the right edge of the Project Window). Again, you can place the cursor over either the pad itself, or its label. The cursor changes back to the black audio plug shape, indicating that this is a pad which can be connected to the first pad. Click again. A cable connecting these two pads appears, and the cursor returns to its normal shape. With this action, you've made a connection to one of the channels of the SCOPE board *ADAT Out* port.

Quick sound check

Now *slowly* bring up the level on your monitors. You should hear white noise – from the left side only.

Repeat the above connection procedure, this time connecting the *S1* pad of the *ADAT Out* module to the *S2* pad on the same module. Now you should hear white noise from both sides.

Tip: Input pads of compatible types can always be connected to one another.

Repositioning a module onscreen

You can move the *White Noise* module to a different position in the Project Window by dragging it there with the mouse. Note that its connection to the *ADAT Out* module follows it around as you drag it.

If the module doesn't move when you try to drag it, recheck the Project Explorer tool to make sure that the *Move Mode* (center) button is clicked in. Or, simply hit *Ctrl-M* to switch immediately into Move Mode.

You can also use the arrow keys on your keyboard to move a module around while in Move Mode – simply select it with the mouse first (the body of a module "lights up" when the module is selected). This technique is slower than using the mouse, but lets you position modules with more precision – sometimes resulting in a much cleaner layout of the onscreen connecting cables than is otherwise possible. (If this doesn't make sense right away, try it out – you'll quickly see what this is about.)

Increased precision of movement can be also achieved by holding the *Shift* key down while dragging modules around with the mouse. This defeats the "snap-to-grid" behavior which normally applies when moving modules using the mouse. You can also disable this grid completely via the Toolbar or the Design Menu.

Deleting a connection

Select the cable connecting the *White Noise* module to the *ADAT Out* module. It becomes highlighted to show that it is selected. Now, hit *Delete* on your keyboard. The connection vanishes – as does the white noise coming from your monitors.

If you prefer not to take your hand off your mouse, you can also delete a connection between two pads by clicking first on one pad and then on the other – i.e., by repeating the actions you performed to create the connection. SCOPE detects the existing connection between the two pads you've selected and removes this connection, rather than adding another one.

Deleting a module via the keyboard

This works the same as deleting a connection: select the module (it becomes highlighted) and hit the *Delete* key. A dialog will pop up, asking for your confirmation before carrying out the delete.

Do this now with the *White Noise* module, after first restoring the connection to the *ADAT Out* module which you deleted in the last step. Note that when you select the *White Noise* module, both the module and the cable connecting it to the *ADAT Out* module become highlighted. This serves to remind you that when you delete the module, all connections to it will be deleted as well. The highlighting of a module's connections when the module is selected can also be a very useful tool for tracing connections through a complicated circuit.

Deleting a module via its context menu

Drag a new copy of the *White Noise* module into the Project Window and connect it as before. Right-click on the module. A small menu pops up next to the cursor. (Note that this also selects the module – the module and its connection become highlighted, as in the previous step.) Now select *Delete Module* from the pop-up menu (via normal left-click). The module and its connection are deleted. The options in the module context menu may vary from module to module, but *Delete* is always one of them.

Selecting / moving / deleting multiple modules

You can select modules one at a time via simple clicking. If you hold down the *Shift* key, you can proceed to select any number of additional modules – or deselect modules you've already selected. This is, of course, a standard Windows technique for making selections. Another standard technique can be useful if the modules you want to select are grouped together onscreen: click and drag on an empty spot in the Project Window to open and enlarge a *selection rectangle*. Any module which is partially or completely inside the rectangle when you release the mouse button becomes selected.

You can then click on any one of these modules and drag the whole set around as a single unit. The multiple selection remains in effect until you click somewhere else. Try this now using any combination of the modules currently onscreen.

The arrow-keys method of moving a module with finer precision also works with multiple selected modules, as does the *Shift-drag* method.

A selected set of modules can be also be deleted in a single stroke, via the *Delete* key or via the context menu of any module in the set (or the *Delete* item in the Edit Menu). Try this out now as well. Delete any or all of the modules still in the Project Window. You don't need them anymore – you're about to begin a new project anyway.

From Simple Oscillator To Synth

Select *New Project* from the File Menu to clear the Project Window. Hit *Ctrl-M* to select Move Mode (or click the middle button on the Project Explorer tool). Once again, **make sure your monitors are turned down.**

Sine oscillator with pitch control

Go to the File Browser and navigate into the *Oscillators* folder. Drag the *Sine Oscillator* module into the Project Window. In the same manner as in the previous section, connect its output pad to the *ADAT Out* module *S1* pad, and the *S1* pad to the *S2* pad on the same module.

Now drag *Fader Vertical* into the Project Window from the *Circuit Controls* folder. A vertical fader control appears in the window. As with the modules, you can drag this control around freely in the window while Move Mode is active. However, it cannot be adjusted while in this mode.

The fader has only one pad. Connect this pad to the *f* input pad on the *Sine Oscillator* module.

Use Mode: operating the pitch control

Turn your monitors up *just barely*. You will in fact hear nothing yet. The fader is all the way down, forcing the oscillator frequency to zero. To operate the fader and raise the oscillator frequency, hit *Ctrl-U* to switch into Use Mode (or click the leftmost button on the Project Explorer tool). Now you can drag the fader knob up and down. Do this *very* gradually – as you can hear, the oscillator frequency rises quite quickly as the fader is moved away from its zero position.

You'll notice that the fader adjustment is rather coarse – it changes the oscillator frequency in large steps. With this particular variety of fader, you can achieve finer, smoother control by moving the cursor away from the fader to one side or the other while "holding onto" the fader knob. The mouse retains its "grip" on the fader as long as you keep the mouse button held down. The fader responds to increasing horizontal mouse distance by producing slower and more gradual changes in response to vertical mouse movements.

Adding a level control

Add the *Linear Volume Attenuator* module to the project from the *Envelopes* folder. Delete the cable connecting the oscillator to the *ADAT Out* module. Then connect the oscillator output to the *in* pad of the attenuator and the attenuator's *out* pad to the *ADAT Out* module.

Copying modules in the Project Window

Now we need another fader to control the attenuator. Instead of dragging one in again from the File Browser, we can simply copy the fader which is already in the project, using standard keyboard copy and paste commands.

Click on the fader to select it. Hit *Ctrl-C* to copy it. Now click in the Project Window and hit *Ctrl-V* to perform the paste. A new copy of the fader appears at the spot where you clicked. (The *Ctrl-V* keystroke could now be repeated as many times as desired to paste additional copies of the fader into the project).

Connect the new fader's pad to the *Vol* input pad on the attenuator. Hit *Ctrl-U* to go into Use Mode, if this mode is not already selected. Now, both the pitch and level faders can be operated.

As with the pitch fader added in the previous step, the level fader should be adjusted slowly. It will raise the output level quite quickly. (This is because it is controlling a *linear* attenuator and is itself a linear fader. The linear attenuator module is not actually intended for use with a linear fader – we'll come back to this later.) Also note that this fader may not be set to zero when it shows up in the project, since it was created by copying another fader – and *that* fader have not been set to zero when it was copied.

Adding an envelope and a MIDI input connection

Delete the *Sine Oscillator* module and the two faders.

Turn your monitors down again!

Add a new copy of the *Sine Oscillator* to the project and connect its output to the attenuator input as before.

Now add two new modules to the project: *ADSR Envelope Generator* from the *Envelopes* folder, and *MIDI Voice Control Easy 16,* which can be found in the *MIDI* folder.

Each of the two new modules has a small control surface with tiny display windows and knobs. These will also appear onscreen when you add the modules. But at the moment, they're just in the way. For now, drag them off to the side. Note that control surfaces can be moved even in Use Mode. Also note that they're not restricted to the Project Window, but can be moved anywhere onscreen. If you want, you can also close (hide) them by clicking their *Close* buttons.

Now make the following connections:

1. Envelope generator *Vol* output to attenuator *Vol* input.
2. Voice control *Gate* output to envelope generator *Gate* input.
3. Envelope *ESYN* output to voice control *ESYN* input.
4. Voice control *f* output to oscillator *f* input.

Finally, make a connection from the *MIDI Input* module (at left in the Project Window, in the set of default modules – it has only one pad) to the voice control input pad *MIDI*. Both the envelope and voice control modules have several additional pads. These can be left unconnected for now.

MIDI sound check

While playing notes on your MIDI keyboard, bring your monitors up *slowly* (remember, we've just done away with the output level control again). You'll hear enveloped, musically tuned, monophonic sine tones. In other words: you've created a synth – admittedly a very simple one.

Opening and closing control surfaces

You've already seen the control surfaces of the voice control and envelope modules. If these are still onscreen, close them now by clicking their *Close* buttons.

Now (re)open the control surface of the envelope generator module: right-click on the module to open its context menu (make sure to right-click on the module *body* and not on one of its pads – otherwise you'll pop up a *pad* context menu). At the bottom of this menu you'll see the option *Surfaces*. Position the mouse cursor over this option to open a cascaded submenu. From this submenu, select the option *Open ADSR Envelope Surface* (it's probably the only option in the submenu). Open the control surface for the voice control module in the same way.

Tip: Control surfaces can also be *closed* via the module context menu – when they are already open. These surfaces can become hidden by SCOPE tools and other onscreen objects, or you may simply have trouble spotting them on a crowded screen. Sometimes the quickest way to locate a control surface is to close it and reopen it via its module's context menu.

Another Tip: Control surfaces can also be opened and closed by double-clicking on the associated module while holding down the *Shift* key.

The special characteristics of control surfaces were mentioned earlier: 1) they can be moved anywhere within the SCOPE screen, not just within the Project Window; 2) repositioning of a control surface and operation of its controls are possible in both Use Mode and Move Mode.

The controls on the surfaces in the current example are mostly self-explanatory. Readouts are included, showing the current settings of each control. Simply click and drag on one of the tiny knobs and move the cursor in circles around it to adjust it. Similarly to the fader control discussed earlier, making wider circles around a *rotary* control results in smoother, finer adjustments of its associated parameter.

The MIDI Channel control is perhaps *not* entirely self-explanatory. This is a fader with no knob: click and drag *directly* on the displayed channel number and move the mouse upward or downward to change the setting.

Also not immediately obvious is that the readouts which display the current settings of the knobs are themselves data entry fields: simply select a field, type in a value directly via the keyboard, and hit the *Enter* key. Note that the associated knob changes its setting to match the entered value.

Take a few minutes now to become familiar with the behavior of these controls, and their effects on the synth.

Saving the project

Select *Save Project* from the File Menu. A standard *Save As* ... dialog appears, since this is a new project which has no name as yet. Enter a name for the project (such as *FirstSCOPEsynth* – the *.pro* extension will be added automatically). The dialog suggests a folder to save the project into. If you want, select a different folder. Then click *Save*.

Reloading the project

Now you have a very simple functioning synthesizer which can be reloaded as a complete SCOPE project. Try this now. In the File Menu, select *New Project* to clear the Project Window, then select *Load Project*. Choose the name of the file you just saved in the *Open* dialog which appears, then click *Open*.

Devices: the next level

A circuit such as *FirstSCOPEsynth*, stored in the form of a complete SCOPE project, is not particularly versatile. It doesn't lend itself easily to being combined with other circuits. In general, what you'd really like to have instead is a synthesizer *device* which can be *added into* a project. The next section of this tutorial shows how this is done.

Creating The TinySynth Device

This section leads you through the creation of a TinySynth device based on the simple sine synth which you created in the previous section. This time, however, instead of a SCOPE project containing a synth circuit, the end result is a separately stored *device*. This device can be added as a unit into a SCOPE (or Pulsar) project and connected to other devices in that project. In the process of creating the device, you'll learn how to work *inside* modules and devices.

Introducing the *Empty Synth*

The *Empty Synth* module serves as a "container" for a synth circuit such as the sine synth. It packages the "raw" circuit into a loadable device with a device surface and icon. It also builds in the important functionality of MIDI controller assignments, preset storage/recall and adjustment of the number of voices a synth has. The elements contained within the *Empty Synth* are exactly those found in any Pulsar synth device.

Open a new project and drag *Empty Synth* from the *Basics* folder into the Project Window.

In addition to the module which appears in the Project Window as expected, you'll also see a small icon pop up in the lower part of the screen just below the Project Window. Double-click on this icon. It disappears and is replaced by its full-size counterpart, which you can now see is a device control surface with almost nothing on it. This is a blank-slate surface upon which a custom front panel for a synth device can be built. We'll come back to it later. For now, click its *Close* button to shrink it back to an icon.

Going inside the *Empty Synth* ....

Make sure Move Mode is selected. Double-click on the *Empty Synth* module in the Project Window (*don't* hold down the *Shift* key here). This takes you inside the *Empty Synth* module. A completely different circuit is now displayed in the Project Window – namely, the circuit contained at the first level *inside* the *Empty Synth*.

.... and getting back out

Now that you know how to go inside a module, practice the equally important and simple technique for getting back out: double-click on any empty spot in the Project Window.

Note that "going inside" devices and modules in the Project Window is possible *only* in Move Mode. If nothing happens when you double-click on the *Empty Synth*, recheck the Project Explorer tool to make sure that Move Mode is selected.

Empty Synth internal components

Go back inside the *Empty Synth*. You'll find an assortment of modules inside. You'll also see some simple instructions posted in the form of plain text. At the right, a module provides hookup points for the audio outputs of your synth – it routes the audio signals from there to the "outside".

The group of modules on the left is responsible for all of the built-in SCOPE device functionality described above. This group is a bit cryptic-looking – but everything you need to know about it at this point can be summed up as follows:

1) You'll need to make *one* connection to *one* of these modules (to feed MIDI messages into your synth circuit – as the built-in instructions clearly indicate), and:

2) These modules appear in every synth device *exactly* as they do here. Don't give this point any further thought.

If you want, you can use the same technique you learned just above to go into and come back out of some of these modules. You won't see much inside them – nor anything of interest for our purposes. In some cases you won't see anything at all. This merely demonstrates that modules contained within a module can also contain modules (and so on), as well as illustrating how you get to them. It also highlights the fact that you generally don't need to concern yourself with what's *inside* a module in order to use it.

Constructing a synth using the *Empty Synth*

For all that it offers, the *Empty Synth* is easy to use. You begin by dragging it into the Project Window and then going inside it – as you've just done. Then, you drop your synth circuit directly into the *Empty Synth* and hook it up to the internal connection points provided there.

In principle, you could construct a synth circuit inside the *Empty Synth* exactly the same as was done in the previous section – by dragging modules from the File Browser into the Project Window and hooking them up. Recommended SCOPE practice, however, is to build the synth circuit into a separate module – referred to as a *circuit layout* module – and then to place this layout module inside the *Empty Synth* and hook it up.

Building the *TinySynth*

For this example, the step of creating a layout module has already been done for you. The *Sine Synth* module, located in the folder *SCOPE\Data\Examples*, contains a circuit which is largely the same as the sine synth circuit you constructed in the previous part of this tutorial.

Make sure you're just inside the *Empty Synth* where you can see the text instructions and modules discussed previously. Also make sure that Move Mode is selected. Drag the *Sine Synth* module from the File Browser into the Project Window. For the moment, ignore the control surfaces which pop up. Move them aside if you need to, but leave them open.

Connect the MIDI input pad of the *Sine Synth* module to the pad indicated by the built-in text as the MIDI input connection point. (The pad itself may be a bit crowded – there are already other connections to it – but, as always, clicking on the pad's label yields the same result.) Likewise connect the audio output pad of the *Sine Synth* module to one of the two indicated audio output hookup pads. Then connect these two pads to one another, as you did in the earlier example, so that both left and right outputs of the *Empty Synth* will produce a signal.

Saving the *TinySynth* as a device

Basic construction of the *TinySynth* is complete. This would be a good time to save it! Double-click on an empty spot in the Project Window to exit from the *Empty Synth* (which is in fact no longer empty) and return to the top level of the Project Window. Pop up the context menu of the *Empty Synth* by right-clicking on the *Empty Synth* module in the Project Window. Select *Save as new device* from this menu. In the *Save as new device* dialog which appears, enter a name for the device (how about ... *TinySynth?*) and select a new directory into which to store the device file (we *don't* recommend saving your own creations in the module library directories!) Finally, click *Save*.

You can confirm the results of your work by deleting the *Empty Synth* from the Project Window. The various module control surfaces as well as the device surface icon will disappear along with it. Now add the device back again via the File Browser – this time as *TinySynth* – from the folder where you just stored it. The surfaces and icon reappear as they were before, and if you look *inside* the *TinySynth*, you'll find that everything there is likewise exactly as it was when stored. It will also still appear onscreen as *EmptySynth*. The technique for changing the device name label is demonstrated below.

Using the *TinySynth*

That's it – *almost*. Like any SCOPE synth device, the *TinySynth* device must *itself* be connected to the SCOPE inputs and outputs before it'll play. This means: connect a *SCOPE MIDI Input* to the MIDI input of the device, and the device audio outputs *outL* and *outR* to the *S1* and *S2* inputs of the *SCOPE ADAT Out* module, respectively. The device's MIDI output pad does not need to be connected.

Now you can turn up your monitors and play.

You can make adjustments while playing via the module control surfaces which popped up out of the *Sine Synth* module when you added it to the *Empty Synth*. These are still at hand, even though you've exited from the device itself. Control surfaces such as these are indeed attached to a specific level of a device or module, but while in normal use, they pop completely outside of the device and float around "above it all". You can adjust coarse and fine tuning and pitch bend range via the *MIDI Voice Control Easy 16* control surface. The envelope parameters can be adjusted on the amp envelope control surface. Master volume can be adjusted on the *Poly Mix 16* control surface (careful here! – its default setting, which produces a reasonable output level, is far below its maximum setting).

Opening the device surface

Open up the device surface of your new synth by double-clicking on the *Empty Synth* device surface icon you left sitting at the bottom of the screen earlier. As mentioned earlier, you can also do this via the context menu of the *Empty Synth* module in the Project Window (right-click on the module body, then select *Open Empty Device Surface* from the *Surfaces* submenu) or by double-clicking on the module while holding down the *Shift* key.

Adjusting the polyphony of a synth

Click the *Presets* button on the device surface to open the *Empty Synth* Preset List control panel. (By the way, a synth's Preset List can also be opened and closed via the module context menu, even when the device surface is not open.) Click and drag upward on the number which appears in the *Voices* display field in the lower right corner of the panel to increase the polyphony of the *TinySynth* from its initial default value of 1 to a more interesting value such as 6 or 10 (or click several times on the neighboring "+" button). In the next section we'll discuss the details of the *Sine Synth* circuit and the *TinySynth* device which permit adjustable polyphony to be implemented so easily.

Creating and recalling presets

Creation and recall of presets is likewise already possible. The *TinySynth* doesn't offer much in the way of control, but what few controls it has – those on the individual module control surfaces – are automatically swept into any presets you create in the Preset List control panel.

Do this now. Click the *Save* button on the Preset List control panel. The new preset appears in the list as *Untitled.* Its "outstanding" appearance and the ¶ symbol next to the name indicate that the name is open for editing. You can type in a new name right away if you choose, followed by the *Enter* key. Or, just hit *Esc* (or click anywhere else in the Preset List window) to leave the name *Untitled* in place. You can come back at any later time to edit the name as follows: 1) select the preset, then 2) hit *F2*, or wait briefly and click on it a second time.

Twiddle some knobs on the module control panels and save another preset. Recall the presets by double-clicking on one preset name and then the next, and watch the knobs jump back and forth. This would work in exactly the same way with any other synth circuit you might construct, no matter how large and complex, once it was packaged as a device via the *Empty Synth*. All of this is handled by the "standard equipment" modules contained in the *Empty Synth*.

The only other feature on the default surface – apart from the *Close* button – is an LED which indicates incoming MIDI activity. It's driven by a special-purpose display module included with the group of "standard equipment" modules built into the *Empty Synth*. As mentioned earlier, the default surface is a foundation upon which a complete device surface can be built. This process – and the techniques for making connections between the surface and the device – are demonstrated in the next tutorial.

Renaming the device and device surface

Here's a simple first exercise with the Project Explorer tool. (In the next tutorial, we'll explore this tool in much more depth.) Open this tool via the Toolbar or the Design Menu, if it is not already open. Click on its *Modules* drawer handle to open this drawer and display the *module tree*. The module tree is a hierarchical list of the modules and devices contained in the current project.

Now, back in the Project Window, click on the *Empty Synth* module. The module tree display in the Project Explorer tool is updated and a corresponding entry – likewise shown as *Empty Synth* – now appears highlighted in the list of displayed objects. Note that *Empty Synth* is also displayed as the selected object in the small field to the left of the Project Explorer's Use/Move/Edit mode buttons.

Click on the highlighted name in the module tree to select it there. Then, click on it again, or hit *F2*. A tiny edit box which contains the name displayed in white-on-black will "pop out" in front of the module tree.

Tip: The same technique for initiating name editing – select the object, then hit *F2* or wait briefly and click on it again – applies in virtually every place in SCOPE where objects with assigned names are displayed.

Now type in the new name (... *TinySynth*, perhaps?) and hit the *Enter* key. The new name is transmitted immediately to the device in the Project Window. You'll see this name atop the Preset List control panel as well, the next time you open it.

However, if you pop up the device context menu, you'll find that the item *Open Empty Device Surface* still appears there. The surface is a distinct element – a module, in fact – within the device. As such, it has its own name, which must be changed separately.

First, you must make the label for the surface module appear in the module tree. The most direct way to do this is simply to click on the surface itself. This brings up *Empty Device Surface* highlighted in the tree. (You may need to double-click on the surface icon first to make the full-size surface visible.)

Alternatively, you can go directly to the module tree and click on the "+" symbol next to *TinySynth*. A sub-branch of the tree drops open below *TinySynth*, showing you the components of the synth, which include *Empty Device Surface*. (If you no longer see *TinySynth* in the module tree, select the device again in the Project Window, as you did above, to bring it back into view.)

However you get there, rename *Empty Device Surface* now to *TinySynth Surface* using the same method you used above to rename the device itself. The new name for the surface now appears in the *TinySynth* device context menu the next time you open it.

Saving modifications to the device

Once you've created a device, you can of course continue to modify it – for example, by changing the name of the device, as you've just done. However, you must save the modified device to make the modifications permanent.

Do this now. Use the *Save as new device* option in the device context menu – the same function you used to save the device for the first time. To *resave* the device, simply select the existing device file in the *Save As* dialog and direct this dialog to overwrite the existing file. From now on, the new name will be there the next time you add the device to a project via the File Browser.

Any and all changes you've made are likewise saved: any presets you've created or modified, the current setting for the number of voices, whether the device surface, other control surfaces and Preset List panel are open and if so, where they initially appear onscreen – but in addition, and more interesting for our present purposes: circuit changes, graphical alterations, module positions, internal control settings, etc.

A Closer Look At The TinySynth

Before going on to the next tutorial, there are a few aspects of the *TinySynth* which are worth taking a closer look at.

(Author's warning: This section contains actual explanations and may be hazardous to your attention span.)

An inside look at the *Sine Synth* module

The synth circuit inside this module is essentially the same as the circuit you constructed in the previous section of the tutorial, but not identical. Switch into Move Mode and go inside the *TinySynth* device. From there into the *Sine Synth* module, to have a look at the differences.

First, note the things which *aren't* different – namely, the modules *Sine Oscillator* and *MIDI Voice Control Easy 16*. These appear in this circuit just as they did in the other one, with the same single connection between them.

In place of the modules *ADSR Envelope Generator* and *Linear Volume Attenuator*, a single module called *Envelope + Amp* appears. However, take a look inside this module. You'll find that *ADSR Envelope Generator* and *Linear Volume Attenuator* are in fact contained within it, likewise with a single connection between them corresponding to the one you made in the circuit you constructed earlier.

Thus, you may already be able to guess – with a little bit of imagination, perhaps – that although this circuit *looks* rather different from the earlier one, it is actually pretty much the same circuit. For the most part, only the layout is different. However, these layout differences are significant for the SCOPE circuit builder. We'll explore them further in the next tutorial.

Variable polyphony and the *Poly Mix 16* module

With respect to the circuit you built previously, the only new module in the *Sine Synth* circuit is *Poly Mix 16*. The oscillator output passes through the *Envelope + Amp* module and then through *Poly Mix 16* on its way out of the *Sine Synth* module. As you've already seen, the *Poly Mix 16* module provides a master volume control by way of its built-in control surface – so it's logical for it to appear at the end of the line. But what else does it do, and what does the name mean?

The *Poly Mix 16* is in fact a special-purpose mixer. How can it be a mixer when it has only one input? The answer is simple enough: the single input can handle up to sixteen distinct incoming signals (that's what the "16" stands for). All of these signals are delivered via the single connected cable. (Why not? After all, it's only an image on a computer screen.) *Poly Mix 16* attenuates each of these signals according to the setting of its single gain control and mixes them together into a single, monophonic output signal.

These multiple-identity signals are present in the rest of the *Sine Synth* circuit. Each of the modules in this circuit is capable of generating or processing a variable number of distinct signals simultaneously. The exception is the *MIDI Voice Control Easy 16* module. This module is classified as monophonic, but it is capable of *managing* the voicing in a polyphonic circuit, up to a maximum of 16 voices. The *Poly Mix 16* module, serving as it does as the point of transition from polyphonic to monophonic in the *Sine Synth* circuit, is likewise classified as a monophonic module.

The number-of-voices value is passed down from the *Voices* setting on the Preset List control panel – by way of the *Num Voices Of Parent* module, one of the esoteric group of "standard equipment" modules built into the *Empty Synth*. This module exists for precisely this purpose and no other. The value which is fed into it is relayed automatically to each of the modules in the *Sine Synth* and lets them know how many voices they should be ready to synthesize, process or control.

Many of the modules in the *Circuit Design* library are polyphonic. They will adjust themselves automatically to the value sent out by the *Num Voices Of Parent* module whenever that module appears somewhere "higher up" (in the module hierarchy). As to which modules are polyphonic and which aren't – the Help Window displays this information just above its help text field.

All *you* need to attend to when constructing a polyphonic synth is to make sure that the *Poly Mix 16* module or its equivalent is wired into the synth circuit at the end of its *polyphonic* signal chain (just before the output, or before any built-in *non*-polyphonic modules, such as a chorus effect) – and to avoid using non-polyphonic modules in the middle of the circuit. Deeper thought is in general not required.

About the voice control and envelope modules

This module combination, or variations upon it, shows up time and again in connection with polyphonic, MIDI-driven synthesizer device construction in SCOPE.

The *MIDI Voice Control* modules are mandatory for building polyphonic, MIDI-driven synths in SCOPE. They interpret incoming MIDI signals and produce outputs in response to them for controlling synth circuits. One of these is the oscillator frequency control signal *f*. You used this signal to control oscillator pitch in the *TinySynth*. The value of *f* is derived primarily from MIDI note numbers, so that the oscillator produces the pitches of the chromatic scale in response to MIDI note messages. The *Coarse Tune* and *Fine Tune* settings on the control surface of this module of course affect the *f* value as well. MIDI Pitch Wheel messages also contribute, to an extent regulated by the *Pitch Wheel Range* setting.

You also used the *Gate* signal, which delivers trigger messages to envelope generators when MIDI Note On or Note Off messages are received. The *ESYN* (Envelope SYNc) signal works in the opposite direction, feeding envelope status information back to the voice control module. This module is also in charge of assigning and reassigning the available synthesizer voices to note events as they come in. The info it receives via the *ESYN* line enables it to handle this task optimally.

The voice control / envelope module combination and connections as used in the foregoing examples are basically all that is needed to form the core MIDI portion of a synth device. It's not essential to know much more about the voice control module than you already know in order to use it. Many (but not all) of the signals which we left unconnected on this module are optional – outputs for creating MIDI-driven modulation effects (e.g., velocity and aftertouch response), or inputs for tables (for contouring velocity response, creating alternate tunings, etc.). Connections to these signals can be added later as corresponding circuits are added to the synth.

Incidentally, the initial steps of this tutorial showed that it is also possible in SCOPE to create functioning synthesizer circuits *without* MIDI connections.

The voice control and envelope module control surfaces

These two modules (and a good many others in the module library) have been outfitted with their own built-in control surfaces. The controls on these surfaces are already hooked up to the appropriate module inputs, internally to the module. On the voice control module, these include the coarse and fine tuning, MIDI channel and Pitch Wheel Range pads, and on the envelope module, the A-D-S-R and slope control inputs. This is why we were able to use these modules in our examples without making connections to these inputs.

Normally – that is, in a full-blown synth *device* – these pads would be connected to graphical controls mounted on the *device surface*, which unifies all of the synth's controls and brings them "outside" the device where they can easily be used. The front panels of Pulsar synthesizer and effect devices are examples of such device surfaces.

However, the control surfaces built into these two modules already incorporate the essential controls needed to make the modules usable. They provide a real convenience factor for experimentation, in that they permit you to operate the control functions of these modules immediately upon building the modules into a circuit, *without* needing to first construct a control surface for the device and make connections to it from these modules. They also serve a useful tutorial or documentation-by-example function, showing which inputs on a module require connections and what kind of values they expect.

At the same time, these built-in control surfaces present no obstacle to going the whole way with a device surface. Their presence can simply be ignored. External connections to these module pads can be made, even though these pads are also connected internally to components on the control surfaces. Furthermore, unlike typical device surfaces, these built-in module control surfaces do not leave icons onscreen when closed. Therefore, they can be kept completely hidden away inside their modules if not needed.

As mentioned above, having control surfaces built directly into modules is not the typical SCOPE way. With complex devices, it's not particularly comfortable to have a separate control surface for each module. Also, these surfaces would add needless "bulk" to a device – for example, if it contained multiple instances of an envelope module, each with its own unused control surface. However, surgical removal of module control surfaces is possible – as is wholesale elimination. You will shortly learn the method for surgical removal.

Further Techniques

About This Tutorial

This tutorial picks up where the first one left off.

It begins with a look at some further interesting details of the *TinySynth* device created in that tutorial. Then, via a series of simple exercises, an extensive set of additional useful and powerful circuit handling tools and techniques is demonstrated.

Most important, as with the first tutorial, is that you actually work through these exercises yourself, step by step. In doing so, you'll learn some of the many capabilities of the Project Explorer tool – the central resource for SCOPE circuit builders. In addition, you'll pick up the essential basics of constructing device surfaces.

As in the first tutorial, however, the emphasis is on the circuit-building tools and techniques themselves, rather than on practical device design or the functionality of specific modules.

Creating A Stereo Attenuator Module

Exported pads

Take another look at the circuit inside the *Sine Synth* module. There are four interconnected modules, seemingly floating unconnected in space. What's going on here?

Look again closely. Note that two of the pads in this circuit have small metallic-looking arrowheads extending away from them – namely, the MIDI input of the *MIDI Voice Control Easy 16* module and the output of the *Poly Mix 16* module.

The arrowheads indicate that these pads have been *exported* – meaning that they have been projected to the exterior of the module which contains them. They are visible there as pads to which connections can be made.

Exporting of pads is an important SCOPE technique for allowing circuits encapsulated within a module to exchange signals with the "outside". If you look again at the *Sine Synth* module itself, you'll see that the two pads it has – *MIDI* (an input) and *out* – correspond exactly to the exported pads inside the module.

Look now at the *Envelope + Amp* module. You'll see that the same thing has been done there. The connection pads on its exterior correspond to the pads in its internal circuit which bear the "exported" arrowhead.

In each of these cases, the pad appearing on the module exterior has all of the properties of the exported circuit pad – it's an input, output or I/O pad, just like the exported pad, has the same signal type and the same name. For all practical purposes, it can be considered to be the same pad, made externally visible.

A look at the circuit inside the *TinySynth* device – specifically, at the *MIDI* input pad of the *MIDI Controller Linker* module – shows that connections can still be made *inside* a device or module to an exported pad. In this circuit, the external pad serves as the MIDI input for the device as seen from the outside. When a MIDI connection is made to this pad on the "outside", incoming MIDI messages appear inside the device on the original pad. In addition to directly feeding the *MIDI Controller Linker* module, they are "tapped off" from there to additionally feed the the *MIDI LED Timer* module (which controls the MIDI LED on the *TinySynth* device surface), as well as the *Sine Synth* module.

The techniques for exporting pads are demonstrated a bit further on in the course of the following exercises.

Exploring device structure via the module tree

Open a new project. Open the Project Explorer tool as well, if it isn't already open, and select Move Mode. Open the *Modules* drawer to display the module tree.

In the File Browser, locate the module *Amplifier 12dB* in the *Modifiers\Level* folder and drag a copy into the Project Window. Along with the module itself, a small module control surface will appear.

Note that *Amplifier 12dB* immediately appears as a highlighted label in the module tree. Also note the "+" symbol next to this label. This indicates that this object (in this case, a module) contains other objects which can also be displayed in the tree. These are referred to as *child* objects, or simply *children*.

Click on this "+" (which now changes to a "-") to view the children of the *Amplifier 12dB* module. You'll find that it has only one "immediate" child: namely, the *Amplifier 12dB Surface*. However, you'll also find a "+" next to *this* label. Click on this "+" to display the *surface's* children. These are the components visible on the surface – the tiny knob, the display field and the text labels, each of which is likewise a module.

Locating objects in the module tree

By now you may have noticed that merely looking at what is shown in the module tree does not always make clear which label belongs to which object. For example, among the children of *Amplifier 12dB Surface* are two instances of the graphical text object. Both of these appear in the module tree with the same label. The same would be true of the *Amplifier 12dB* module itself, if you had more than one of them in your project (as you soon will). Furthermore, the label for a particular object may not even be visible in the module tree at any given moment – as will often be the case with complex devices and surfaces containing dozens of child components.

To clear up the confusion, simply click in the Project Window directly on the object in question – a device or module, a control surface, or an individual surface component. The module tree display automatically adjusts itself so that the label comes into view there – opening child branches of the tree if necessary – and highlights the label.

Deleting objects via the module tree

We don't need the module control surface – we're going to build a device with its own surface. We could just click on its *Close* button and hide it, but it's bad form to build unnecessary components into a SCOPE device. More to the point: this is a good opportunity to demonstrate how any device or module component can be eliminated.

Tip: "Gutting" is a generally useful technique in SCOPE. Starting with a copy of a device or module which already contains some or all of the main structures you need and then stripping out the unwanted parts is often quicker and easier than building the same thing up from scratch. And copies of devices and modules are free, of course.

You might guess that the control surface could be deleted in the same way as a device or module: i.e., click on it and then hit *Delete*, or right-click on it and select *Delete* from its context menu. At the moment, however, this won't work – precisely because of the special property of control surfaces which enables them to float around outside of their "parent" devices or modules.

There *are* a couple of ways to get rid of a control surface. The more "sophisticated" method – deleting it via the module tree – is often also the easier one:

Click on a blank spot on the control surface. In the module tree display you'll now see the highlighted label *Amplifier 12dB Surface*. Click on this label, and then hit *Delete*. The label in the module tree vanishes – as does the actual surface. (Note that this is done without a request for your confirmation.)

Practically anything which appears in the module tree can be deleted in this way. For example: while you're at it, click on the label *Amplifier 12dB* and hit *Delete* again. This time, the module itself vanishes.

Now, just for practice, add a new copy of *Amplifier 12dB* to the project and strip the control surface clean, piece by piece. (A practical use for this skill will come up shortly.) This works just like deleting a complete surface or module via the tree – click on the label representing a surface component, then hit *Delete*, and watch that component vanish from the surface. Work at deleting *precisely* the component you want to, using the technique of first clicking on the object itself – a knob, a display field, etc – then locating its highlighted label in the module tree, and finally clicking on that label and hitting *Delete*.

Using context menus in the module tree

By now, it shouldn't surprise you to learn that context menus are also present in the module tree. These menus are opened in the typical context menu way: by right-clicking on an object's label in the tree. From the menu which opens, you can then select one of several operations to be performed on that object.

For example, the entire preceding exercise (deletion of a module, a control surface or individual surface components via the module tree) could be carried out using the *Delete* entry in the module tree's context menus, instead of hitting the *Delete* key. But – let's move along:

Right-click on the *Amplifier 12dB* label in the module tree (if you still have one of these modules!), then select *Delete* from the menu which pops up. The module disappears.

Now scroll to the top of the module tree and right-click on *New Project*, which in fact represents the Project Window. Select *Load > Module* in this menu. A Windows *Load Module* dialog opens. Double-click on *Amplifier 12dB.mdl*. A new copy of this module appears in the Project window.

Use the context menu technique to repeat what you did just before: click on the new *Amplifier 12dB* control surface, then right-click on the corresponding *Amplifier 12dB Surface* label in the module tree and select *Delete* to remove the surface from the module.

Now right-click on the *Amplifier 12dB* label in the module tree (which no longer has a "+" or "-" next to it, since the module no longer contains any child objects). Select *Copy*. Scroll back to the top of the tree, right-click on *New Project* (which is really the Project Window) once again, and select *Paste Amplifier 12dB*. A new copy of the childless *Amplifier 12dB* module appears in the Project Window – directly on top of the original. Move it down so you can see both.

Note that the above module tree context menu methods of adding, copying and deleting modules are functionally no different from the other methods you've already used to accomplish equivalent tasks (dragging modules from the File Browser into the Project Window, copy-and-paste using *Ctrl-C/Ctrl-V*, selecting a module and hitting *Delete*, etc.). They're just alternative methods of doing exactly the same things.

Connect the *Gain* pads of the two surfaceless *Amplifier 12dB* modules to one another. This simple circuit will form the basis of the *Stereo Attenuator* device.

Creating new modules from circuits

Select both *Amplifier 12dB* modules. If you haven't yet used the selection rectangle technique, try it now: draw a rectangle by clicking and dragging on an empty spot in the Project Window. Release the mouse button when the rectangle encompasses both *Amplifier 12dB* modules. Both modules become highlighted.

Now select the *Fold* item in the Edit Menu (or hit *Ctrl-F*). The two modules disappear. In their place – actually, in the upper left corner of the Project Window – a *very* small module appears, whose only feature is the label *untitled*. Drag it into the middle of the window and double-click on it. Inside, you'll find the two *Amplifier 12dB* modules – with the single connection between them still in place. The module tree correspondingly shows a new *untitled* object with child objects: the two *Amplifier 12dB* modules.

Folding is the method which was used to package a "raw" synth circuit as the *Sine Synth* layout module used in the previous tutorial – and likewise, to create the *Envelope + Amp* module inside it. Arbitrarily complex circuits containing any number of modules and connections can be "modulized" in this way – after which, of course, you can modify them further, working inside the new module.

The *Fold* operation has an inverse. Via *Unfold* (shortcut *Ctrl-Shift-F*), a selected module vanishes (permanently), and the circuit which was previously inside "pops out".

Manually redrawing windows and tools

As you continue to extend the number of different methods and tools via which you can display and/or modify a circuit or its components, you are more likely to encounter situations in which the various active tools are a bit "out of sync" with one another. (This amounts to a confession of sorts. Ideally, all screen representations should be automatically updated in response to any change made to a circuit or object, and should always fully agree with one another. In fact, this is pretty much the case, but it's not one-hundred percent – yet.) By clicking on any window or tool and hitting *F5*, you can force it to be redisplayed "from scratch". We encourage you to do this any time you suspect that the information presented by a particular window or tool is not completely "current".

The Pad List in the Project Explorer tool

Now it's time to get a bit more surgical – and to pop open one of the drawers of the Project Explorer tool we haven't looked inside yet. Click on the *Pads* tab to open the Pad List drawer, then click in the Project Window on one of the *Amplifier 12dB* modules.

The Pad List shows a list of the connection pads on the selected device or module. In this case, the list contains three entries, corresponding to the three pads on the *Amplifier 12dB* module. A great many operations are possible via the Pad List. We'll try out a few of them now.

Renaming pads via the Pad List

Prominently displayed down the middle of the list are the pad names. By default, following program startup, the Pad List displays *long names*. These are the same names which pop up as Tool Tips when you place the cursor over a module's pads.

Long names are more intelligible for the average user than the *short names* (maximum length 4 characters) which appear on the pads themselves. However, long names are too long for use as pad labels. Therefore, each pad on a SCOPE module has both long *and* short names. In practice, short names can be considered as the "real" pad names. The long names play a more cosmetic role.

We want to export the *in* and *out* pads on the *Amplifier 12dB* modules to the exterior of our new module. However, since we have *two* of these modules, we also have two of each of these pads, with identical names. In order to keep the pads distinct from one another – i.e., in order to know which exported pad an external pad actually represents, and avoid cross-wiring the signals – we need to rename them. We'll want to do this *before* exporting them, since their names will be applied directly to the external pads which are created by the exporting process.

For this purpose, we are interested in the short names. Right-click inside the Pad List – in this case, it doesn't matter where – and select *Short Names* in the context menu which pops up. Now the displayed names are exactly those appearing on the pads themselves.

Select one of the modules. Then, in the Pad List, select the *in* label, pause momentarily, then click on it again to begin editing it, just as you did with object names in the module tree (and preset names in the Preset List control panel). Rename this pad to *inL*. Hit *Enter* when you're done. Since you're now working in the Pad List, you can use the arrow keys to step down to the *out* pad and then hit *F2* to begin editing it (as always, either method of initiating name editing can be used). Rename the *out* pad to *outL* in the same way.

Repeat the same procedure with the other *Amplifier 12dB* module, renaming its *in* and *out* pads to *inR* and*outR*.

Exporting pads via the Pad List

There are three different ways to do this in the Pad List alone – we'll try 'em all.

Select the module with pads *inL* and *outL*.

Using method 1: Click on the *inL* entry in the Pad List, and then on the *Export Pad* button (this is the pad-with-up-arrow icon at the far right end of the row of icons along the top of the Pad List – ToolTips can help you out here if you're still not certain as to which button this is).

Using method 2: Right-click on the *outL* entry in the Pad List, then select *Export* from the context menu which pops up.

Using method 3: Click on the *Gain* entry in the Pad List, then hit the *E* key on your keyboard.

Exit from the *untitled* module and have a look at it from the outside. There you'll see the three pads you just exported.

Unexporting ("deleting") exported pads

Each of these three methods of exporting a pad is instantly reversible by simply repeating the same action, thereby "unexporting" the exported pad. This is also referred to as *deleting an exported pad*, although the pad itself remains in place – only its external "representative" is deleted.

Try this now with one of the pads you just exported. You can do this without going back inside the *untitled* module. Instead, select one of its child *Amplifier 12dB* modules in the module tree. Its pads will be displayed in the Pad List. The pad names in the list will let you know if you've got the module whose pads you've already exported – if not, select the other *Amplifier 12dB* module. Select a pad, and then (per method 3 above, for example) hit the *E* key repeatedly. Each time you do this, the corresponding pad on the *untitled* module exterior disappears or reappears.

Tip: This is, by the way, one of the "power" aspects of using the Project Explorer tool – namely, that you don't have to actually be looking directly at a module or device in the Project Window in order to do something with it. The entire hierarchy of a project is accessible at any time via the module tree, regardless of what is currently being displayed in the Project Window.

Note that the *Export / Unexport Pad* button in the Pad List also functions as a status indicator – it pops in and out as you select different pad entries in the Pad List, depending upon whether or not that particular pad is currently exported.

Exported pads can be unexported at any time – even when external connections to them are present. Of course, any external connection to an exported pad is irretrievably lost when the pad is unexported. You should therefore be careful when doing this in your own circuits, if you're modifying a module which is already wired into a circuit.

Exporting pads via pad context menus

This is as good a place as any to mention pad context menus. By right-clicking directly on a device or module *pad*, you pop up the context menu for that pad. (You may well have already accidentally discovered this for yourself.) In this menu, you can select *Export Pad* or *Unexport Pad*, among other possibilites. Go back inside the *untitled* module and use this method now to export the *inR* and *outR* pads of the second *Amplifier 12dB* module, as well as any of the three pads from the first module which are not currently exported.

Look at the *untitled* module again from the outside. You should now see five pads: *inL* and *inR* on the left, *outL* and *outR* on the right, and one *Gain* pad at the top. These pads are arranged on the module exterior in the same way as the exported pads they represent. For example, the input pads are on the left edge, since the exported inputs pads were also on the left edges of their respective modules.

Note carefully, however, the ordering of the external input and output pads. Although they are now distinctly labelled, these pads should ideally also have the same sequence – e.g., *L* above, *R* below – to avoid confusion when hooking the module up. The ordering of the pads along each edge of a module exterior depends on the order in which the pads were exported – so it's possible, after all of the exporting and unexporting you've been doing, that your input and output pads are not in the same sequence. To correct this, simply unexport and re-export the pad which is ahead of its desired position. For example, do this to *outR*, if necessary, to move it from above to below *outL*.

Rename the module now via the module tree (you already know how to do this!) to *Stereo Amp 12dB*. Finally, save it as a new module. This time, just for variety's sake, use the context menu for the *untitled* label in the module tree. Select *Save as ...* from this menu. In the *Save as new module* dialog which opens, select the directory where you want to put the file and then simply click on *Save* – the module name is used automatically as the file name.

Moving modules out of a module

The *Move Out* operation is introduced here somewhat out of the "logical" order of succession, since it is the reverse of the *Move Into* operation discussed following. *Move Out* is *approximately* the reverse of the folding operation: a selected module or set of modules is "kicked out" of the module or device which contains it, and takes up residence at the same place (in the project hierarchy) as that module or device.

Try this with your new *Stereo Amp 12dB* module. Exit from any module you happen to be in, make your way back to the top level of the Project Window and drag a new copy of the *Stereo Amp 12dB* module into the window. Go inside this module, select both *Amplifier 12dB* modules, and select *Move Out* in the Edit Menu (or hit *Ctrl-Shift-I* on your keyboard). The modules disappear from the project window.

Now double-click on the window to exit from the *Stereo Amp 12dB* module. There, you'll find your original two-module circuit. The pads in this circuit which were previously exported have been automatically unexported.

Note that the *Stereo Amp 12dB* module is also still there – devoid of pads. It is now, in fact, a completely empty module whose only remaining feature is its name. Leave it where it is for the moment – we'll come right back to it.

Folding portions of a circuit

Make the following connections to the two *Amplifier 12dB* modules from the preceding step: *S1* and *S2* outputs of a *SCOPE ADAT In* module to the *inL* and *inR* pads, *S1* and *S2* inputs of a *SCOPE ADAT Out* module to the *outL* and *outR* pads. Finally, drag in a new copy of the *Fader Vertical* module (from the folder *Circuit Controls*). Hook this fader up to the *Gain* pad on one of the *Amplifier 12dB* modules.

Now, select the two *Amplifier 12dB* modules and create a module around them using the *Fold* item in the Edit Menu. As before, the two modules vanish and a new *untitled* module appears in the upper left corner of the Project Window. This time, however, the new module already has pads. These pads correspond to the pads in your two-module circuit which, when you folded it, already had "external" connections – i.e., connections to other modules in the Project Window which were *not* selected for folding into a new module. These already-connected pads have been automatically exported as a "side effect" of the folding operation.

In addition, the preexisting connections are still intact – via the new external pads. To make this new module equivalent to the *Stereo Amp 12dB* module you created previously, all you'd need to do is rename it.

However, "modulizing" selected *portions* of a circuit is not intended specifically as a shortcut method of exporting pads, however convenient that may be. Its true importance is that it permits you to develop a circuit without knowing in advance what shape its structure will eventually assume. Later, you can modulize sections of the circuit as you see fit – without altering the circuit structure. The optimal logical groupings are often easier to identify when you've got the finished circuit in front of you – and it can be easier to develop the circuit "all in one place", as opposed to working inside of several different modules.

Packaging the circuit into a smaller number of complex modules, done astutely, can make the circuit easier to understand and manage. Furthermore, it permits useful function blocks which you've developed in that circuit to be saved as new module files which can easily be deployed in other circuits.

This operation is of course also reversible, using the *Move Out* operation discussed previously. Go inside the new *untitled* module, select the two modules inside, and then select *Move Out* from the Edit Menu (or hit *Ctrl-Shift-I*. When you exit the *untitled* module again, you'll find the two-module circuit you started with, along with all of the connections to other modules which it had before.

Moving modules inside an existing module

The *Move Into* operation is largely the same as the *Fold* operation, but instead of creating a new module from the selected modules, the selected modules are moved inside an *already existing* module. In fact, it is the reverse of the *Move Out* (*Ctrl-Shift-I*) operation discussed earlier (or the other way around).

Select one or more modules, then hit *Ctrl-I* (or select *Move Into* from the Edit Menu). When the cursor changes to its "ready to drop" shape, click on the destination module to send the selected modules inside it. As with the *Fold* operation, any external connections to these modules are preserved by way of new exported pads which appear on the exterior of the target module.

Practice moving one or both of the externally-connected *Amplifier 12dB* modules inside (*Ctrl-I*) and back out of (*Ctrl-Shift-I*) the *untitled* module you created in the previous step. Leave them outside when you're done.

Using drag and drop in the module tree

What about that empty *Stereo Amp 12dB* module? We'll put it back together now, by way of demonstrating another very useful and powerful circuit manipulation technique.

Cut all of the "external" connections to your two-module circuit, leaving only the single connection between the *Gain* pads of the two modules as before. Click on one of these modules to bring up its label in the module tree. Now *drag* this label and drop it onto the *Stereo Amp 12dB* label, which should be just above it in the tree. (Tip: release the mouse button when the *Stereo Amp 12dB* label becomes highlighted).

The "-" symbol next to the *Stereo Amp 12dB* label reappears, indicating that this module once again has a child, and the *Amplifier 12dB* label you dropped on it takes up a new position in the open child object branch below it. This change is reflected in the Project Window, where one of the *Amplifier 12dB* modules has vanished (into the *Stereo Amp 12dB* module, of course).

Invisible connections – a first look

The remaining *Amplifier 12dB* module has grown an odd-looking appendage which extends from its *Gain* pad. This *antenna tip* symbol indicates that there is a "wireless" or *invisible* connection to this pad – i.e., one which is not represented onscreen by a cable. This is, of course, the connection to the *Gain* pad on the other module, which is still intact. You'll find an antenna tip on that module as well, if you look inside the *Stereo Amp 12dB* module.

The two *Amplifier 12dB* modules are now located at different points in the project hierarchy, and therefore can no longer both be displayed at one time in the Project Window. Consequently, the connection between them cannot be indicated by a cable – but it is otherwise no different from any other connection. (Remember, those cables are only a screen illusion, anyway).

In contrast to exported pads (merely another screen trick), invisible connections (ditto) can be used to represent direct connections between *any* two pads anywhere in a project, regardless of how distant they are from one another in the project hierarchy – just as, in the module tree, you could have dropped the *Amplifier 12dB* module *anywhere* in the project. This is why the antenna tip / invisible connection representation is used in this case. Invisible connections are of course quite a powerful tool – we'll be working with them a bit further on.

Drag and drop in the module tree (continued)

Go back to the module tree and drag the other *Amplifier 12dB* module into the *Stereo Amp 12dB* module. Look inside and you'll find that the antenna tips on both modules are gone, replaced by a normal cable between the two modules – now that they're "siblings" again.

Now try out the reverse operation: dragging a module *out* of another module. Actually, it's the same operation: dropping an object onto another object in the tree. The dropped object becomes the child of the object it is dropped onto – or becomes its sibling, if you hold the *Shift* key down while *dropping* it (note: *not* while dragging).

Thus: To drag an *Amplifier 12dB* module out of the *Stereo Amp 12dB* module so that it becomes the sibling of that module, you can either 1) drop the *Amplifier 12dB* module onto the *parent* of the *Stereo Amp 12dB* module, or 2) hold down the *Shift* key while dropping the *Amplifier 12dB* module directly *onto* the *Stereo Amp 12dB* module. In this example, the parent of the *Stereo Amp 12dB* module is actually the Project Window – so to use method 1, you would drop the *Amplifier 12dB* module onto the *New Project* label at the very top of the module tree.

Reassemble the *Stereo Amp 12dB* module and finish by exporting pads from inside as you did before, via whatever method(s) you choose. Or, better still: delete the whole mess – you've already saved the module (haven't you?)

Module tree: drag-and-drop vs. cut-and-paste

To pull all of this together a little bit: drag-and-drop in the module tree is the graphical equivalent of cut-and-paste. The module movements you've just performed in the module tree using drag-and-drop can also be accomplished there via the *Cut* and *Paste* entries in the module tree context menus – or using the standard *Ctrl-X* and *Ctrl-V* keyboard shortcuts, which also work in the module tree. To do this, first select the label in the module tree corresponding to the module to be moved and select *Cut* from its context menu (or hit *Ctrl-X*). Next select the destination module in the module tree and select *Paste* from its context menu (or hit *Ctrl-V*). Just as with direct drag-and-drop, the first module gets moved inside the second, while retaining any connections it had before, possibly now as invisible connections.

Cut-and-paste: module tree vs. Project Window

The cut-and-paste method of moving modules also works in the Project Window. Select the module to be moved and hit *Ctrl-X*. Then select another module and hit *Ctrl-V*. Between the first step and the second, you can go inside and/or exit from modules any number of times – the destination can be far removed from the original location in the module hierarchy. Also: as an alternative to clicking *on* the destination module, you can also simply click on the Project Window background while *inside* that module (similar to what we did when copying a fader module in the first tutorial) – this is handy when moving a module upward in the module tree.

Creating The Device And Its Surface

Preparing the device

This part is easy – you've already done basically the same thing in Tutorial 1. This time we'll practice the "gutting" technique mentioned earlier, since we're building a *very* simple device. We'll pick up the pace a little bit, since we're building upon the basic skills you've been practicing up to this point.

Open a new project and add the *Empty Synth*. Locate it in the module tree and open its child object branch. Now, delete *everything* you find in this branch, except for *Empty Device Surface*. The result in the Project Window is an *Empty Synth* device with no pads. (You can also do this by going directly inside the device in the Project Window and deleting everything you see there, since only the device surface is not in there – yet.)

Next, expand the *Empty Synth* control surface (via its module context menu, or by double-clicking on the minimized surface icon, or by holding down the *Shift* key and double-clicking on the module). The child object branch of the *Empty Device Surface* may already be open in the module tree (if you've just double-clicked on the *Minimized View* object, which is one of its children). If not, open it. Delete the child objects *@Preset Button* and *@MIDI In LED* and note that these objects vanish from the surface. Rename *Empty Device Surface* to *Stereo Attenuator Surface*. Finally, rename the device itself to *Stereo Attenuator*.

Drag & drop from the File Browser into the module tree

As you already know, you can add the *Stereo Amp 12dB* module to the *Stereo Attenuator* device by going inside the device and then dragging the *Stereo Amp 12dB* module from the File Browser into the Project Window. However, files (i.e., modules and devices) can also be dropped directly into the module tree.

Practice this method now: drag the *Stereo Amp 12dB* module from the File Browser and drop it onto the *Stereo Attenuator* label in the module tree. This adds it as a child of the *Stereo Attenuator* device. Or, if the device's child object tree happens to still be open, you can also drop the module onto *Stereo Attenuator Surface* while holding down the *Shift* key. This adds it as a sibling of *Stereo Attenuator Surface* – the equivalent result.

Export the input and output pads of the module, taking care to use the same sequence for left and right at both ends. This part of the device is now complete. Leave the *Gain* pad unexported – we're coming to that next. Save the device under its new name.

Device surface states: *flipped in*, *flipped out*

As you've seen, a control surface object has a specific location in the module tree – typically as a direct child of a device. However, as you've also seen, surfaces float around continually onscreen, regardless of which circuit branch is displayed in the Project Window (or whether that window is even open). A surface can also be placed anywhere onscreen, not just within the Project Window.

The surface *object* is a type of module with some special properties. One of these is the ability to be cut loose from its position in the module hierarchy and "pop outside" of its owner device as well as the Project Window itself, so that the user controls of a device can be kept "on top" and readily accessible. This is one of the two basic states which a surface module can be in. It's referred to in SCOPEspeak as the *flipped-out* state.

The other state is, of course, the *flipped-in* state. In this state, the surface is confined to its device. It appears *inside* the Project Window, when the module tree branch where it resides is being displayed there. In this respect (and in general, overlooking for the moment a couple of other special properties), the surface behaves just like any other module when flipped in. Most importantly, it comes fully under the influence of the *Use*, *Move*, and *Edit* mode buttons. The flipped-in state can therefore be used for making structural modifications to a surface object – as we're going to do now.

Adding a fader to the device surface

Right-click on the surface (the full-size one, *not* the minimized icon) and select *Flip In* from the surface's context menu. If the Project Window was still displaying the first child level inside the device – i.e., the level where the *Stereo Amp 12dB* module is located – then you'll still see the surface now, *inside* the window. Otherwise, the surface vanishes altogether – you must maneuver in the Project Window to this point in the circuit (and possibly also scroll the window around a bit) to see it again.

In the File Browser, locate the file for the *Fader Horizontal* module, drag it into the Project Window and drop it directly onto the surface. Select Move Mode if it isn't already selected. Note that you can now drag the surface around inside the Project Window and the fader will move along with it – but also that you can *still* drag the fader itself around and reposition it on (or even off) the surface as desired. The same is true of the graphical text object ("Device Surface") which is still present on the surface.

This demonstrates another special property of the surface object: modules (and graphical objects in general) can be dropped directly onto a surface in the Project Window. They thus become children of the surface object. Their graphical representations become attached to the surface graphic but remain visible on top of it, and can be individually repositioned with respect to it.

Connecting the fader via cable

This method of connecting control surface elements is not the usual method (we're coming to that next), but it is legitimate, and illustrates some useful points.

Connect a cable from the *Val* pad on the fader to the *Gain* pad on the *Stereo Amp 12dB* module using the "usual" method of clicking on one and then the other. The fader's connected – but you can't adjust it, because Move Mode is still selected. All you can do is drag it around.

Now right-click on the surface again and select *Flip Out*. Note that the fader can now be operated, even though Move Mode is still selected – as you can easily confirm by moving the module in the Project Window. Flipping a surface out forces it into Use Mode, regardless of which mode is effective within the Project Window. In keeping with this, the fader can no longer be repositioned on the surface. The text object is likewise now "glued" into place.

Also note that the *Val* pad on the fader module is no longer visible. The cable connecting to it is likewise gone, but an antenna tip on the *Gain* pad of the *Stereo Amp 12dB* module testifies to the presence of a connection. This is one more special property of surfaces: all connection pads, antenna tips, etc. on a surface's child objects become hidden when the surface is flipped out, thereby automatically "cleaning up" the surface to make it presentable to the user.

The *MinimizedView* and the *Minimize* button

Minimize the surface by clicking on its *Minimize* button (or via the surface context menu). Note that the minimized icon now includes the fader you just added (the miniaturized fader even reflects the current position of the real one).

This bit of trickery is performed by the *MinimizedView* child object of the device surface (visible in the module tree, although not elsewhere). This module is intended specifically for use as a child of a surface object. It creates the minimized icon directly from the current device surface image whenever the surface is minimized, eliminating the need for manual creation (or permanent storage) of bitmap icons for this purpose.

By the way, if you were paying *very* close attention earlier, you may have noticed that the *Minimize* button does not appear in the module tree. It is not, as you might guess, a child module of the surface object, but is in fact a fixed feature of the object, with its own graphical attachments. This may at first seem like a special case, but it is actually just one example of a very general aspect of SCOPE graphical control objects. Another example is the handle on a fader – likewise a fixed feature of the fader object, with pre-defined functionality (sliding up and down or side to side and generating values accordingly) and its own graphical attachment (i.e., the handle bitmap).

Connecting the fader via the Pad List

This is the preferred technique for making connections between modules in a device circuit and control elements on its surface. Expand the surface again, flip it in, delete the connection to the fader, and flip the surface out again. Open the Project Explorer tool and its Pad List drawer.

Click on the *Stereo Amp 12dB* module. This displays its pads in the Pad List. Select the *Gain* pad in the list, then click on the *Store Pad* button (the black audio plug icon directly above the list). Now click on the fader, select its *Value* pad in the Pad List, and click on the *Connect To Stored Pad* button (the red audio plug just to the right of the *Store Pad* button). This completes the connection – accordingly, an antenna tip appears on the *Gain* pad of the *Stereo Amp 12dB* module.

You may recognize the black and red audio plug icons – they're the same ones which appear as mouse cursors when stringing cables point-to-point in the Project Window. The two processes are essentially one and the same. In the Project Window, selecting a pad "stores" it as one end of a connection to be made (or deleted). The next pad to be selected becomes the other end – and the connection is made (or deleted, if it already existed).

The important difference, as illustrated above, is that the Pad List permits connections to be added or deleted in situations where the graphical point-to-point cabling method of the Project Window *can't* be used – either because the two pads in question don't appear together in the Project Window, or because one or both of the pads is hidden. The above example illustrates both cases – sort of.

A more realistic example is a device surface whose control elements are connected to modules at various points in the module hierarchy of a device. In this case, which is entirely typical (the *TinySynth*, with a fully developed device surface, would match this description), merely flipping the device surface in does not permit all necessary connections to be made by stringing cables. No matter where the device surface resides, some of the connections to it must come from elsewhere in the module hierarchy.

Via the Pad List, on the other hand, connections can be made to controls on a surface even when it is flipped out. The surface can thus always be right at hand, regardless of where you navigate in the Project Window – sparing you a lot of jumping back and forth in the window.

Finally, while the control elements in *Circuit Controls* all feature a visible *Val* pad, most of the modules in the much larger control element library in *Surface Design\Controls* have *no* visible pads. These modules can be connected *only* via the *Store Pad / Connect To Stored Pad* method.

Connections via pad context menus

The *Store Pad* and *Connect To {stored pad}* functions are also accessible as options in the pad context menus accessible in the Project Window. They are used in exactly the same way as the *Store Pad / Connect To Stored Pad* buttons in the Pad List – but this method spares you the detour via the Pad List and can thus be faster and easier.

In these menus, "{stored pad}" is replaced by the actual name of the currently stored pad, to remind you which pad you're connecting to (or disconnecting from).

Connections to controls via module context menus

With graphical controls requiring only one external connection (almost all faders, buttons, displays etc.), the *Store Pad* and *Connect To {stored pad}* options appear in the *module* context menu, with the *Value* pad automatically selected as the store/connect target. Connections can thus be made via the module context menu to control and display modules situated on a flipped-out surface (with the associated convenience factor), even though their pads are then hidden. More generally, this method can be used to make connections to control modules from the large library in *Surface Design\Controls*, whose pads are (by default) always hidden.

Tracing and deleting connections via the Routings List

Before continuing: locate the fader in the module tree and rename it to *Stereo Atten Fader*.

Now click on the button at far left above the Pad List to open the Routings drawer (ToolTips will indicate this button as "Show Routings"). Click on the fader and observe the action in the Project Explorer tool:

- The selected-object field (next to the mode buttons) reads "Stereo Atten Fader".
- The module tree display highlights the *Stereo Atten Fader* object, showing it as a child of the *Stereo Attenuator Surface*.
- A long list of pads appears in the Pad List, with the pad *Val* at the top, highlighted.
- In the Routings drawer, *Val* likewise appears in the selected-pad field at the top. Below this, in the Routings List window, you'll see the entry *Stereo Amp 12dB*, with a subordinate entry *Gain* directly below it.

These entries in the Routings List indicate the module (as well as the specific pad on that module) to which the fader's *Val* pad is connected.

Double-click on either of these two entries. Now the Routings List shows you the other end of the connection: *Stereo Atten Fader – Val*. The other Project Browser displays have been updated accordingly: *Stereo Amp 12dB* is now shown as the selected module, it appears highlighted in the module tree, and its list of pads appears in the Pad List, with the *Gain* pad highlighted.

In this manner, connections in a project can be traced point-to-point, even when there are no visible cables to show you the way.

Connections can also be directly deleted via the Routings List – in this case by selecting *Gain* or *Val* (whichever end of the connection happens to appear there at the moment) and hitting *Delete*. This is often more convenient than deleting an invisible connection by retracing the steps used to create it.

Adjusting the fader characteristic

Make sure that the connection between the fader and the *Gain* pad is in place, and that the surface is flipped out (and for the sake of form, minimized as well). Resave the device.

The device is now (after only twenty-one pages!) a fully-functioning stereo attenuator. Test it out by loading a synth device (e.g., Inferno) and connecting the attenuator inline between the synth outputs and the *S1/S2* pads of the *Scope ADAT Out* module you're using to send audio to your monitors. Connect the MIDI input of the synth (*MIn*) to the *SCOPE MIDI In* module for the MIDI input where your keyboard or other MIDI controller is connected. Or, if you don't have a keyboard connected, add the *MIDI Keyboard* module from the *MIDI* folder and connect it as your MIDI source, using its onscreen keyboard to play notes. Open the synth and attenuator device surfaces. **Turn the volume on the attenuator all the way *down* (full left)**.

Turn up the main volume on the synth all the way and, while playing, *slowly* raise the gain on the attenuator. You'll find that the output volume jumps up rather abruptly from zero, and that most of the useful adjustment range is concentrated near the left end of the fader range. This is because the fader, by default, has a linear characteristic, as does the *Stereo Attenuator* – not a good combination for a volume control device.

However, the fader characteristic can easily be changed. Via the Window Menu, open the Control Ranger, and then click again on the fader. Or (of course) right-click on the fader and select *Open Control Ranger*.

A text field at the bottom of this tool reads *Stereo Atten Fader* to let you confirm that you're adjusting the intended control (you can also rename a control module here, instead of in the module tree). In the display window in the middle, you'll see a curve which is distinctly linear. Adjust the *Form* fader directly below this display so that the text field next to it changes from *lin* to *log*. The curve shape will change correspondingly. The default value of 100 for *Intensity* should do nicely – or, you can dial in a slightly higher one (150 or so) using the nearby vertical fader or by editing the *Intensity* value directly.

You'll now find the action on the fader much more comfortable. The control surface itself is another story – it needs some serious *cosmetic* help! Save the device again. Now we'll work on appearances a little bit.

Modifying the surface label: the Text Editor tool

First of all – calling a device surface *Device Surface* does not do us a lot of good.

Open the Text Editor tool (via the Design Menu or the Toolbar). Click on the *Device Surface* label. The same text appears in the text field of the Text Editor. Modify the text in this field to *Stereo Attenuator* in the usual way (hit *Enter* when done). The label on the device surface changes accordingly.

With this step, you've edited the text content of a *graphical text* object. In the module tree, you'll see this object as a child of the surface, identified as *Graph Text*. Modifying the content of this object has no effect on its name, and as with other types of object, having several different objects called *Graph Text* in the same branch of a circuit can become confusing. Therefore, in the module tree, rename this object now to *Graph Text "Stereo Atten"*. Renaming it likewise does not affect its content, but now it is clearly identified.

Adding and duplicating labels

Labels on the fader indicating its action certainly couldn't hurt – one at each end will do.

Flip the surface in. In the File Browser, go into the directory *Surface Design\Graphic Components*. Drag the object *Graphical Text* from the Browser and drop it into an open area on the surface. A new graphical text object appears on the surface, with a random text content such as *Channel* as a placeholder. Click on the new graphical text object and use the Text Editor to modify it to *-Inf*.

Now, click on the graphical text object again, hit *Ctrl-C* to copy it, then click elsewhere on the surface and hit *Ctrl-V*. A copy of the *-Inf* graphical text object appears there. Use the Text Editor to change the second graphical text object to read *+12dB*, corresponding to the maximum gain of the *Amplifier 12dB* modules at the core of your device.

In the module tree, you'll now see two new *Graphical Text* objects as children of the surface. Click on the *-Inf* text object to highlight the corresponding object label in the module tree, and change the name of that object to *Graph Text "-Inf"*. Rename the other *Graphical Text* object to *Graph Text "+12dB"*.

Note that renaming the text objects is is a purely superficial measure and not at all mandatory – but can prevent confusion and needless mistakes in a complex project.

Hiding, unhiding and reorienting module pads

Now we'd like to slide these labels over next to the ends of the fader – but that unsightly *Val* pad at one end of the fader, with its attached antenna tip, will make it difficult to see if we've got the label correctly positioned at that end. However, as we now know, we really don't need to see this pad at all – so let's get it out of sight.

Click on the fader. In the Pad List, select the *Val* pad. Now click on the *Change Pad Orientation* button above the list (it's the small green pad symbol with the arrows running in circles around it). On the small square grey "pill" next to the *Val* pad entry in the Pad List, the red dot which indicates the current pad orientation jumps from the left side to the top – as does the pad on the fader itself (along with its antenna tip).

Click the button three more times. The pad will jump to the right end of the fader, then to the bottom side, and finally, after the third click, disappear completely. The grey "pill" next to the Pad List entry for the *Val* pad likewise no longer has its red dot, and now looks like the ones for all of the other pads on the fader, which are likewise hidden.

As should by now be clear, module pads are just one more screen trick, and reorienting them or hiding them has no effect upon the functionality they represent. Hiding a pad merely makes it impossible to string cables to it (or to obtain information about that pad via the Help Window).

Pad context menus for hidden pads are still accessible, by right-clicking on the associated Pad List entries. Of course, you can reorient a pad using this context menu instead of the *Change Pad Orientation* button – and there are keyboard shortcuts as well, which can be used freely after selecting a pad in the list: L - R - T - B - H for Left - Right - Top - Bottom - Hide.

(In addition, as you can easily see for yourself, the Pad List context menus include all of the other options found in the "regular" Project Window pad context menus, plus a good many others in addition.)

Tightening up the device surface layout

With the *Val* pad out of the way, you can drag the new labels over to the appropriate ends of the fader (*-Inf* at left, *+12dB* at right). Remember that you can use the arrow keys as well as the mouse to precisely position these objects, and also that you can defeat the "snap-to-grid" behavior of the mouse temporarily by holding *Shift* down while dragging – or completely, by disabling the grid via the Toolbar or the Design Menu.

Now we can finish the task of arranging the objects on the surface with an eye towards space-efficiency (and aesthetics). Drag the *Stereo Attenuator* text object all the way to the top of the surface and place it directly to the left of the minimize button. Next, select the fader, then use *Shift*-click to select both of its labels as well. As with "normal" modules, this multiple selection enables you to move the fader and its labels together as a single unit. Place this unit directly below the *Stereo Attenuator* text object and minimize button. Flip the surface out, minimize it, and save the device again.

Now the layout is nice and tight – way up in the corner there. The surface is needlessly large. To fix this requires Edit Mode.

Resizing the surface: a first look at Edit Mode

Expand the surface and flip it back in. On the Project Explorer tool, click the Edit Mode button (or hit *Ctrl-E*) to switch into Edit Mode.

Click on the device surface. A "pane of glass" appears behind the surface (or is it in front?) The perimeter of this pane encompasses the surface and indicates its true boundaries. (These will appear to extend beyond the surface on the bottom and right sides as a result of shadowing effects which are enabled on the surface.)

Move the cursor over this pane and note the changes in the shape of the cursor. These shapes indicate the size or position modification which is possible at any point via click-and-drag: you can reposition any edge or corner of the pane, or you can move the entire pane without changing any of its dimensions.

Take hold of the lower left corner of the pane and shrink the surface until it encompasses the "tightly arranged" set of surface objects symmetrically and without wasted space. Switch back to Move Mode (*Ctrl-M* or *Esc* are both good shortcuts here), flip the surface out, minimize it and save the device again.

As you'll note, the surface itself is now so small that the minimized view of it is not all that much smaller – in fact, you'll probably find yourself trying to operate the minimized view!

Modifying surface appearance: the basics

Small though the surface may be, there's no reason not to indulge in a bit of restyling (this *is* a tutorial, after all). The surface object is a blank slate not merely in the sense that other objects can be freely arranged upon it – the surface appearance can also be extensively tailored.

The following modifications can be done with the surface either flipped in or flipped out, but it must of course be expanded to full size. Via the Toolbar, open the following tools: Fill Style, Color Selector, Gradient Designer and Texture Importer. Arrange them onscreen where you can see all of them at once.

Click on the *Stereo Attenuator* surface (note: click on an empty spot on the surface, *not* on a label, fader or button). Note that the *Fill With Gradient* button on the Fill Style tool has popped itself in. The Gradient Designer tool is now likewise displaying the color gradient which is currently applied to the device surface.

Solid colors

Click the *Fill With Color* button on the Fill Style tool. The *Fill With Gradient* button pops out, since only one fill style at a time can be applied to an object. The surface now takes on a solid color – whatever color happens to be selected in the Color Selector tool at the moment.

The dot in the color ring on this tool indicates the spectrum point of the color. You can drag this dot around in the ring to change the spectrum point, or double-click anywhere in the ring to select the corresponding point directly. Since the surface is currently selected, these changes in the spectrum point are relayed immediately to the surface itself, so you can see right away how the new color looks.

The large square inside the color ring shows the range of colors obtainable by mixing the spectrum point color with black or white in varying proportions. (Hint: along the top of this square, the color is completely "mixed out" and all shades of grey between black and white are available). The dot in this square indicates the currently selected mix. The small square beneath the large one shows the result of the current mix as a small patch of solid color. You can adjust the mix in the same way as the spectrum point, with changes relayed immediately to the surface.

For a weird effect, you can also reduce the opacity of the color via the *Opacity* fader. With values lower than the default maximum of 255 (corresponding to 100% opacity), it gradually becomes possible to see other objects through the device surface when the device surface is positioned over them. But BE CAREFUL HERE – don't set this value below 128, or you will no longer be able to select the surface! (If you happen to do this anyway, the easiest way out is simply to delete and reload the entire device.)

You might decide that you like the color of an existing object – for example, the *Stereo Attenuator* module itself.

Click on this module (again, not on one of its labels or pads). Note that the Color Selector now indicates the color of the module. Next, click on the *Transfer To Next* button – the small, transparent arrow "bump" at the top of the Color Selector tool, which then "fills in" and turns red. Now click on the attenuator surface. The color in the Color Selector is transferred to the surface (and the *Transfer To Next* button switches itself off).

Important to note is that all of these changes affect *only* the color of the surface. The objects on the surface are not affected (apart from the background of the fader, which is transparent and merely allows the surface color to show through). However, you can just as easily change the color of the graphical text objects by clicking directly on *them*, instead of on the surface, and applying the Color Selector tool as described above. (The colors of the fader knob and the minimize button can also be adjusted, but not as simply.)

Color gradients

Instead of having a solid color, an object can be "painted" with two or more colors, with transitions between them according to settings made in the Gradient Designer tool.

Click on the surface again, and click the *Fill With Gradient* button on the Fill Style tool. The surface is now colored with a gradient which is also visible in the Gradient Designer tool. This tool is most easily understood through experimentation, once you're aware of a few details.

The dots at far left and far right directly below the gradient display window indicate the existing gradient points. A color is associated with each point. If you select one of these dots, its color appears in the Color Selector tool and can be adjusted. As always, the changes are relayed immediately to the surface.

By double-clicking in the gradient display window, you can add gradient points at any intermediate position. The colors associated with these points can likewise be adjusted. In addition, the points can be dragged from side to side as desired. A selected gradient point can be deleted by hitting *Delete*. (The points at extreme left and right have fixed positions and cannot be deleted).

The remaining controls are left for you to discover. Note that gradients, like solid colors, can also be applied to graphical text objects, and that the Gradient Designer, like the Color Selector, also has a *Transfer To Next* button (as do all of the Graphical Design tools). Via this button, you can, as one example, transfer any gradient you design on the device surface directly to the *body* of the *Stereo Attenuator* module (whose fill style is automatically switched from *Fill With Color* to *Fill With Gradient* in the process) – or to device surfaces you develop in the future.

You may now take the rest of the day for experimentation with gradients.

Texture maps

As a third alternative to solid colors and color gradients, surfaces can also be "painted" by mapping other bitmaps onto them.

Click on the surface and click the *Fill With Texture* button on the Fill Style tool. The surface "goes grey" as there is currently no texture bitmap assigned to it (and the *Fill With Color* and *Fill With Gradient* options are deselected).

Next click the Import Bitmap button on the Texture Import tool. In the *Select Pattern* dialog which opens, select the file *PurpleBlur.bmp* from the *Examples* folder. The texture is immediately "sprayed" onto the surface. Any 24-bit Windows RGB bitmap or uncompressed TGA bitmap can be imported as a texture bitmap.

You can click and drag on the texture displayed in the tool window to shift the texture around on the surface. To make it easier to do this with large bitmaps, or those which have been heavily "zoomed in" (see below), the drag operation can continue beyond the boundaries of the tool window and go anywhere on the computer screen.

Activating the *Enhanced* option on the tool permits you to rotate the map via the ring control around the tool's display window, or change its magnification via the *Zoom* fader. For these adjustments, activating the *Tiled* option is also recommended, to help avoid losing "sight" of the pattern while shifting it around.

As this tool (like the other Graphical Design tools) also has a *Transfer To Next* button, you can also transfer texture maps from object to object. Try this now. Load the *Inferno* synth device from the *Examples* folder. Expand its device surface and click on the surface, then click the *Transfer To Next* button on the Texture Import tool to turn it red, and finally, click on the *Stereo Attenuator* surface to transfer the *Inferno* surface texture to it.

General remarks

In addition to creating colors and gradients from "scratch", importing texture maps from files, and copying any of these directly from one object to another, you can also compile them in your own libraries via the *Save* button on each of these tools. The resulting disk files are then accessible via the *Browser* drawer built into each tool.

Of course, we've only scratched the "surface" here (sorry!) The graphical design possibilities in SCOPE are enormous, and the ability to import graphics files which have been created and/or edited in other programs extends beyond texture mapping to include virtually any graphical element – controls and their components, displays, surfaces, etc. The use of the Graphical Design tools, as well as important relevant background information, is covered in depth in the reference chapter *Graphical Design*.

The *TinySynth* Revisited

A (very) quick-and-dirty *TinySynth* surface

This section shows how you can take advantage of the control surfaces built into many of the SCOPE library modules – and benefit from the considerable effort which has already been expended in creating them – to throw together a basic but complete control surface for the *TinySynth* device. In the process, you'll exercise many of the techniques you've learned up to this point. You'll also get a bit of a whirlwind tour of some of the Graphical Design tools we haven't looked at yet, get a quick glimpse at the GO (Graphical Object) Tree and GO Properties sections of the Project Explorer, and learn some of the concepts and techniques associated with SCOPE GOs.

Open a new project and load the *TinySynth*. Go inside it, and then inside the *Sine Synth* module. For each of the modules *MIDI Voice Control Easy 16*, *Envelope + Amp* and *Poly Mix 16*, open the module control surface and flip it in. Exit from the *Sine Synth* module and drag it down toward the bottom of the Project Window – we're going to need working space at the top.

Exit from the *TinySynth*. Expand its device surface and flip it in. Go back inside the *TinySynth* and drag the device surface into the upper right corner of the Project Window. Switch into Edit Mode, select the surface, and enlarge it by dragging its left edge all the way to the left edge of the Project Window. Switch back out of Edit Mode.

Now, open the Project Explorer, its Modules drawer and its GOs drawer. In the GO drawer, click the *Show GO Properties* button to open this sub-panel.

We'll begin with the *MIDI Voice Control Easy 16* module. The same steps should be repeated with both of the other two modules in the *Sine Synth* which have control surfaces.

Go inside this module, where you'll find its control surface and nothing more. Click on the surface. At the top of the GO tree display you'll see the label *SurfaceGroup*, which is the parent of the *graphical* object tree for the surface module. The graphical object tree (or GO tree) is analogous to the module tree itself, but contains only the *graphical* components of the currently selected module.

In this case, the selected module is the device surface, whose only GO component is *SurfaceGroup* – a *container* which can hold other GOs. The only child GO of the *SurfaceGroup* is *Background* – the visible dynamic bitmap element itself, which we resized and decorated in various ways in the preceding sections of this tutorial.

Click on the "+" next to *Background* to display *its* child GO, *CloseButtonBack*. This GO also has a child – *CloseButton* – but we're interested in the button background at the moment. Select *CloseButtonBack*, and then, in the GO Properties panel at right, click the *Visible* check box to extinguish the check mark.

Note that the *Close* button on the surface has vanished. It's still there, but no longer to be seen. And as a result, it is no longer accessible via mouse click. The same is true of any invisible GO in SCOPE (or even, as mentioned earlier, a GO which is more than halfway transparent). In effect, it's as good as deleted – which is exactly what we want. Note that we didn't need to do this to the *CloseButton* GO itself. Since it's a child of the *CloseButtonBack* GO, it becomes invisible when the *CloseButtonBack* GO is made invisible.

(Why didn't we just delete the *CloseButton* GOs if we don't want them? Try this via the GO tree, using either the *Delete* key or the GO context menus. As you will quickly be informed, this is not possible – the *Close* button and its components are fixed features of a surface module.)

Next, go into the *Surface Design\Graphic Components* folder via the File Browser and drag the module *Border* onto the surface. A transparent "frame" appears. This is a module whose GO, also called *Border*, has no color of its own and merely highlights or darkens the colors of whatever is behind in such a way as to create the illusion of a recessed rectangle. Since the interior of this GO is completely transparent, the *Border* module can be selected *only* via its visible edges.

Via the Toolbar, open the Dimensions tool. Select the surface – its dimensions (and position) appear in the tool. Now click the *Transfer To Next* button on this tool, and click on any visible edge of the *Border* module. It immediately expands to the same size as the surface.

However, it is probably sitting off to one side of the surface, which is not what we're after. To correct this, you could simply drag it (again, by one of its visible edges) until it is lined up with the surface itself. But that would be too easy – this *is* a tutorial. Therefore, open the Alignment tool and select the border by one of its edges. Click the *Alignment On/Off* button in, followed by the *Align To Center* buttons for both vertical and horizontal alignment, and finally, make sure that all of the *Offset* fields contain zero.

Now, go into Edit Mode, resize any edge of the surface slightly, then put it back where it was, and exit Edit Mode. Or, to make full use of the tools: stay in Move Mode, select the border, click the *Transfer To Next* button on the Dimensions tool, and then click on the surface. Whichever way you choose, this resizing operation on the surface (regardless of whether the size of the surface was actually changed) causes the border to jump to the fully-centered position you specified for it via the Alignment tool.

One more thing: open the Shadow Options tool, click on the surface (*inside* the border, so you don't select the border) and click the *Shadow On/Off* button out to disable shadowing effects for the surface.

Now, select the surface and hit *Ctrl-X.* Exit the *MIDI Voice*

Control Easy 16 module and *Sine Synth* modules to get back to where the *TinySynth* device surface is. Click near the upper left corner of this surface and hit *Ctrl-V*. The *MIDI Voice Control Easy 16* surface appears on this surface. Since it has been moved from its original position inside the *MIDI Voice Control Easy 16* module using cut and paste, its control and display elements are still connected to that module as before and are still functional – but they are all now also part of (and attached to) the *TinySynth* surface.

Click on the *MIDI Voice Control Easy 16* module surface and drag the whole unit – controls, display fields, labels and border – to a suitable place near the left end of the *TinySynth* surface. Now open the Color Selector tool, click again on the *MIDI Voice Control Easy 16* module surface to make sure you've got it selected, and dial the *Opacity* parameter all the way down to zero. The end result is a framed MIDI control section on the *TinySynth* surface.

Note that you can now select, move and modify all of the components in this section individually – including the border, since it is merely another module attached to the *MIDI Voice Control Easy 16* module and therefore does not drag the rest of the components around with it. Note also, however, that you can no longer select and move the *MIDI Voice Control Easy 16* module as a unit, since its background is now transparent – if you try this, you'll select and move the *TinySynth* surface instead.

This is actually what we want. However, you might find that you aren't satisfied with where you put the MIDI control section. To reposition it, you need to be able to select it again, which means you must change its color back to non-transparent. But to change its color, you must select it first – and you can't select it in the normal way anymore, precisely *because* it's now transparent.

The GO tree provides a solution. Select the *TinySynth* surface. In the module tree, locate and select its child *MVC Easy 16 Surface*, and then select *Background* via the GO tree – you can now modify the color back to non-transparent and proceed to move the MIDI control section wherever you like.

To complete the *TinySynth* surface, repeat the same procedure with the control surfaces of the *Envelope + Amp* and *Poly Mix 16* modules. Other changes you might want to make: move the *Presets* button up next to the minimize button, and move the MIDI activity indicator lamp over next to the *MIDI Channel* field in the MIDI control section. And of course, the whole range of graphical changes is possible, if you want the surface to be not so quick and dirty.

Along the same lines, you can proceed to replace the controls on this rough-cut surface one at a time with alternate controls of your choice from the large library of control modules provided with SCOPE (or ultimately, with controls of your own design). If you choose to do this, you will need to pay close attention to the settings of the individual controls – as revealed by the Control Ranger – since the default values they are loaded with when dragged in from the library are not appropriate for all of the signals needing to be controlled in the *TinySynth*.

However far you pursue this process, you will learn quite a bit about the way in which the various modules need to be controlled. You can likewise learn from the information which the Control Ranger provides about the display modules – since the distinction SCOPE between "control" module and "display" module is very fine indeed.

Final remarks – what to do next

With what you've learned up to this point, you're ready to proceed beyond the tutorials and start creating devices.

When you're ready to learn more, consult the *Circuit Design* and *Device Design* chapters of the manual, which provide extended, in-depth information regarding all aspects of these topics, including much more background information about SCOPE and its components.

The *Project Explorer* chapter describes this tool exhaustively in reference manual format, while the *Graphical Design* chapter does the same for the Graphical Design tools, and also discusses aspects of custom control creation. Last but not least, full-sized devices are provided in the *Examples* folder for your inspection.

Good luck and have fun!

Best regards,

The Creamware Team

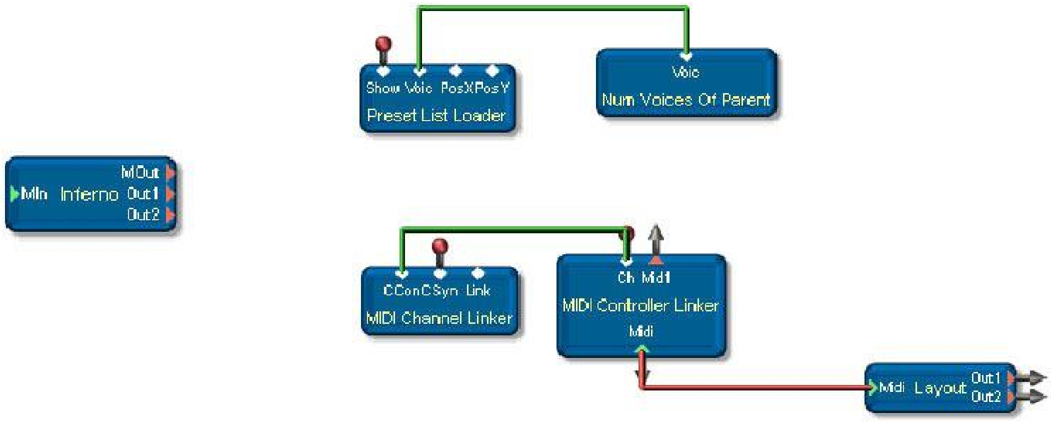
Inferno Example

Return To Main Table Of Contents

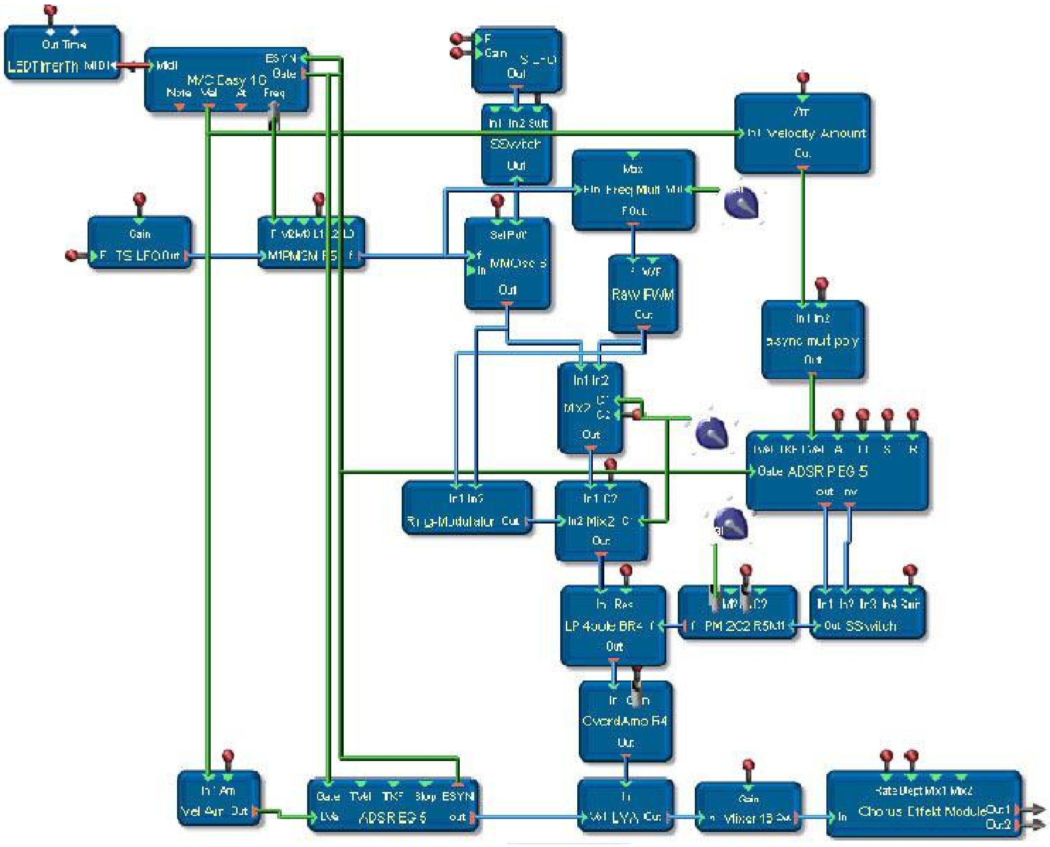
Inferno Example

To help you better understand how to efficiently build your own synths, we've included information on basic construction of the Inferno and Blue Synth synthesizers.

We'll start with the Inferno.

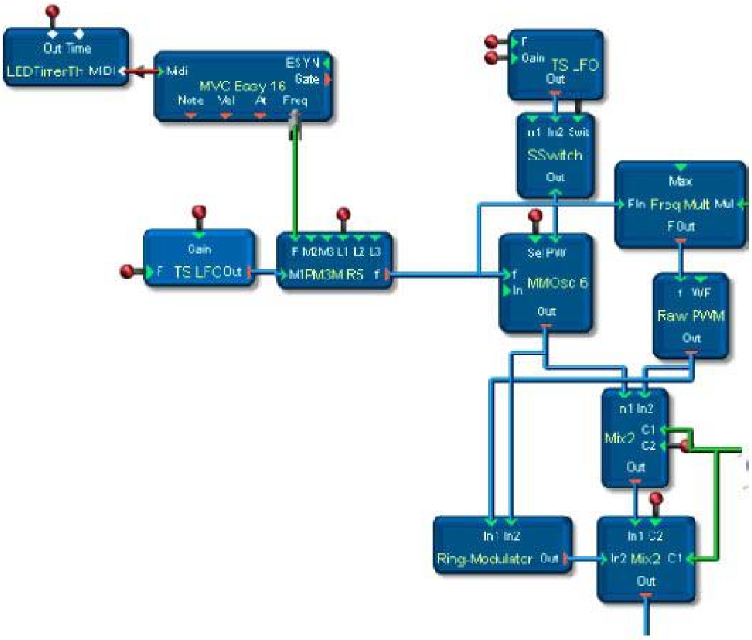


When you first double-click on the Inferno module, you should see an arrangement of modules as shown above. The module you want is called ‘Layout’. Double-clicking on this will get you to the innards of the Inferno synth:



It may look like a lot of modules, but, the Inferno Synth is actually a fairly simple design. Also, we have arranged the pads so that the graphical flow is easier to see. To do this, the standard convention of Inputs on the left and Outputs on the right was ignored, as otherwise the connections would be crossing at places where it would be hard to read, etc. By changing the location of the pads (connections) on certain modules, we tried to arrange the visual presentations of the Inferno and Blue Synth so that you could more easily see the signal flows of the layout. You can easily change these by using the ‘Change Pad Orientation’ icon in the Module Properties Design window, Pads drawer.

Now let’s take a look at some of the sections of this synthesizer. We’ll start with the oscillator/signal generating section:



The Inferno synth uses the Multi Mode Oscillator (MMOsc 6) for its main waveshape source, but there is also a ”sub” oscillator, which is created by taking the Frequency control (sent from the MVC through the Pitch Mod block), routing it to a Frequency Multiplier (Freq Mult) and, by using a value of 8388607 as the multiplier, deriving a value one octave below to send to the Raw PWM oscillator. The Raw PWM oscillator generates a square wave when there is a value of zero or no input to the waveform (WF) pad, so this oscillator in the Inferno synth is only a square wave. However, you could easily attach a front panel control knob to the PW pad, and make it variable!

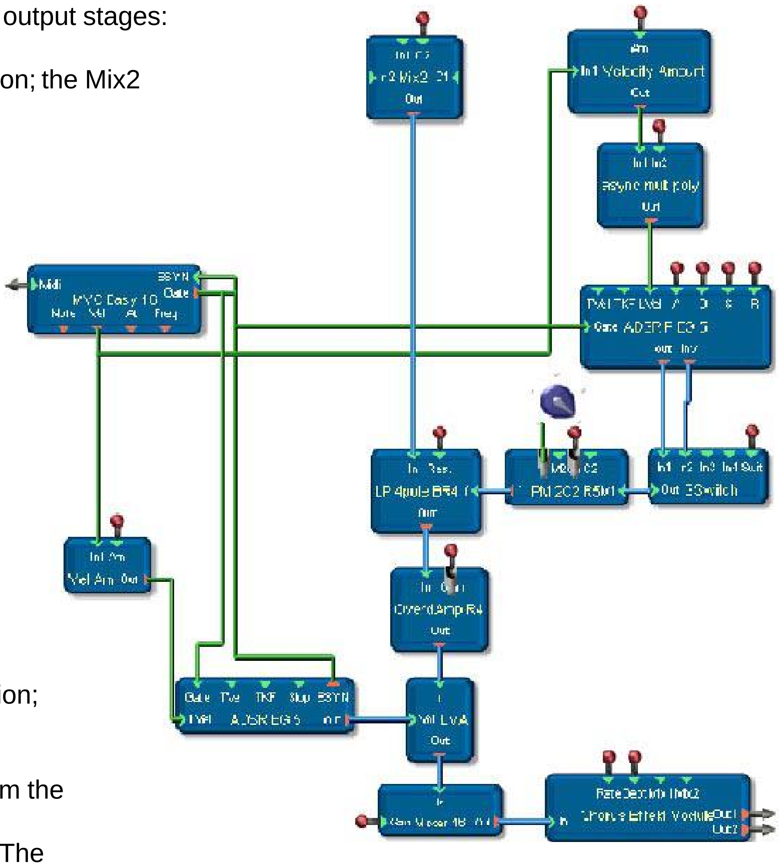
The outputs of these 2 oscillators are then fed into both a Mix2 mixer and a ring modulator. Those outputs are then sent to another Mix2 for final output to the next section. Notice that there is a standard control pot (blue), set at maximum value, attached to both of the mixers' C1 control inputs. This guarantees constant full level output of Input 1 for both mixers. The C2 pads are attached to knobs on the Inferno's front panel to vary the level of the ring modulator and the sub-osc.

There are also 2 LFOs in this picture, one for vibrato of both oscillators, the other for pulse width modulation of the MMOsc. They are the Tiny Sine (TS) LFOs (notice that they are the same module, but with different pad orientations, as explained above).

There are several Pitch Mod blocks one can use to route and control modulation signals; we used the Pitch Mod 3M R5 (PM3M R5), which provides 3 mod inputs, each with their own level control. It takes the Frequency signal from the MVC and modulates it exponentially by the signals coming in at M1, M2 and M3, sending out a combined frequency control signal for the oscillators. The Tiny Sine LFO to the left of the PM3M provides the oscillators' vibrato, and is controlled by the Pitch Mod section of the Inferno's front panel. The Pitch Mod Depth is attached to the TS LFO Gain input, but could have also been attached to the PM3M Level 1 (L1) control input. In this case, we are using L1 to receive MIDI Mod Wheel Control information to additionally control the LFO depth. The other TS LFO is routed through a Switch module (SSwitch), so that we can switch the control of the pulse width from LFO to Manual control (Input 1 of SSwitch is attached to the Width pot on the front panel).

Now let's take a look at the filter, shaping, and final output stages:

(The layout has been edited a little for this description; the Mix2



Now let's take a look at the filter, shaping, and final output stages:

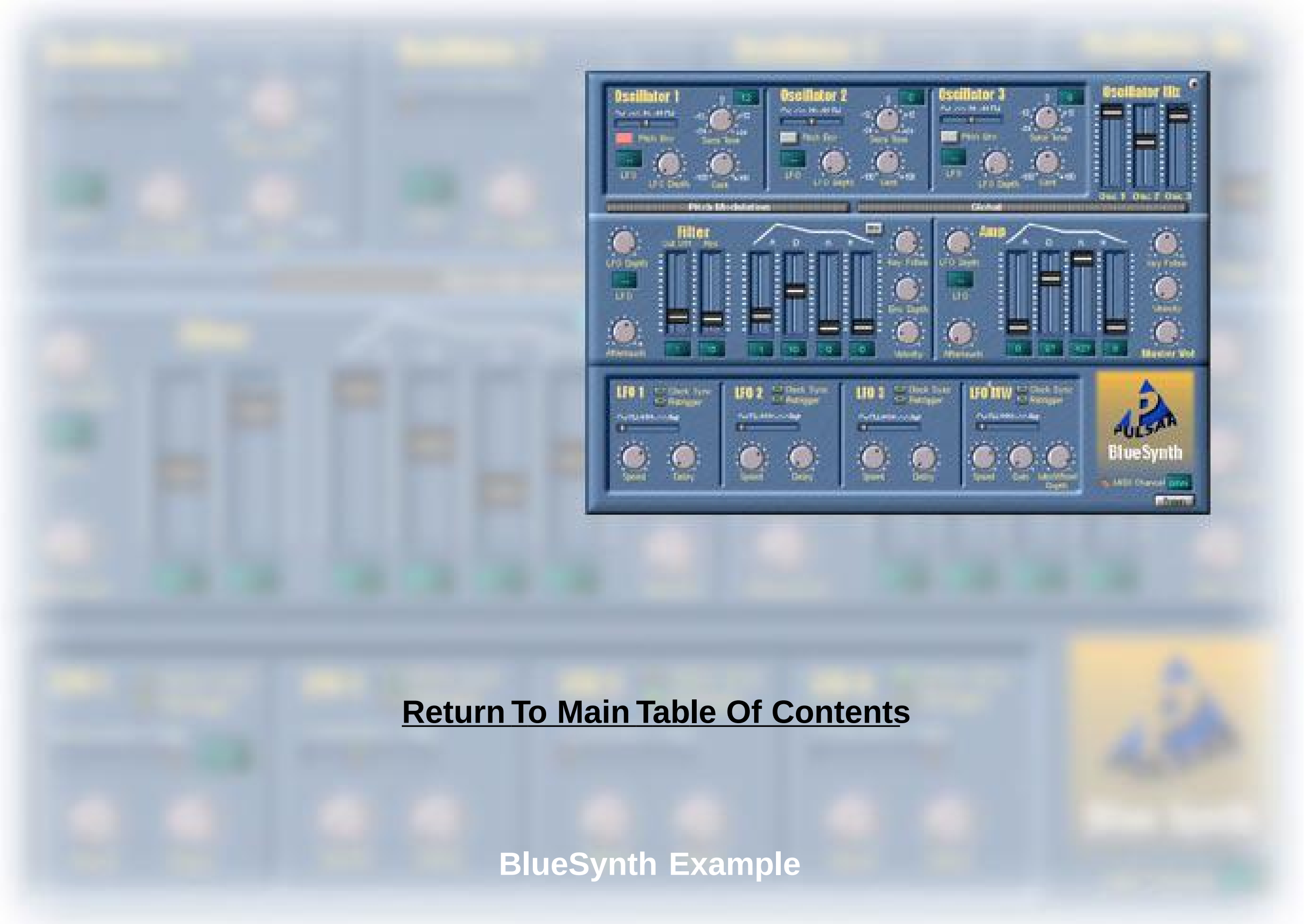
(The layout has been edited a little for this description; the Mix2

Here the signal path flows in a very typical way, from the Mix2 through a Low Pass Filter (LP 4pole BR4), Overdrive Amp, LVA, Mixer 16, and finally, Chorus. The only items demanding special comment are the ADSRs and filter modulation.

There are 2 ADSRs (envelope generators) providing a ”shape” to the sound. The overall shape of the sound (Amplitude) is provided by the ADSR EG 5 attached to the Linear Volume Attenuator (LVA). This ADSR’s level is controlled by Velocity (Vel Amt is connected to the front panel Amplitude – Vel pot). It’s output is uni-polar, which is required by the LVA’s Volume input.
The other ADSR is a Pitch EG version (ADSR PEG 5), which means the envelope output is scaled around zero (bi-polar). It also has an inverted output. You want to use a Pitch EG when controlling the frequency of something, such as an oscillator, LFO, or Filter (Cutoff).
In this case, both ADSR outputs are routed to a switch, then to another Pitch Mod block (PM 2C2 R5) which is connected to the filter’s frequency control pad for modulation.
The level of this ADSR PEG is controlled by both Velocity and the Env pot in the Filter section on the front panel. The ‘async multi poly’ module is needed to combine the 2 asynchronous signals (Velocity and the Env pot) for the ADSR Vel level control input. The Filter’s Cutoff is actually controlled by a pot connected to the C1 pad of the Pitch Mod block.

7

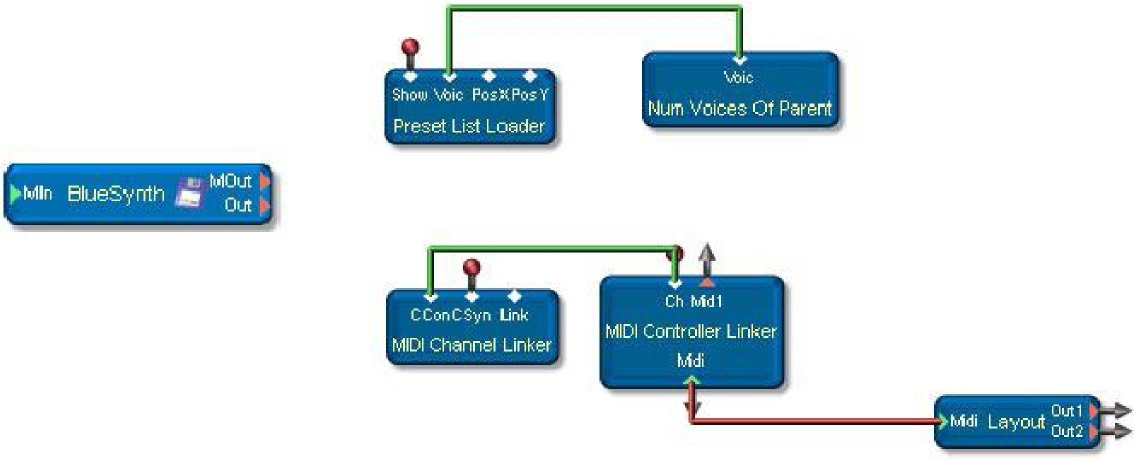
BlueSynth Example



Return To Main Table Of Contents

BlueSynth Example

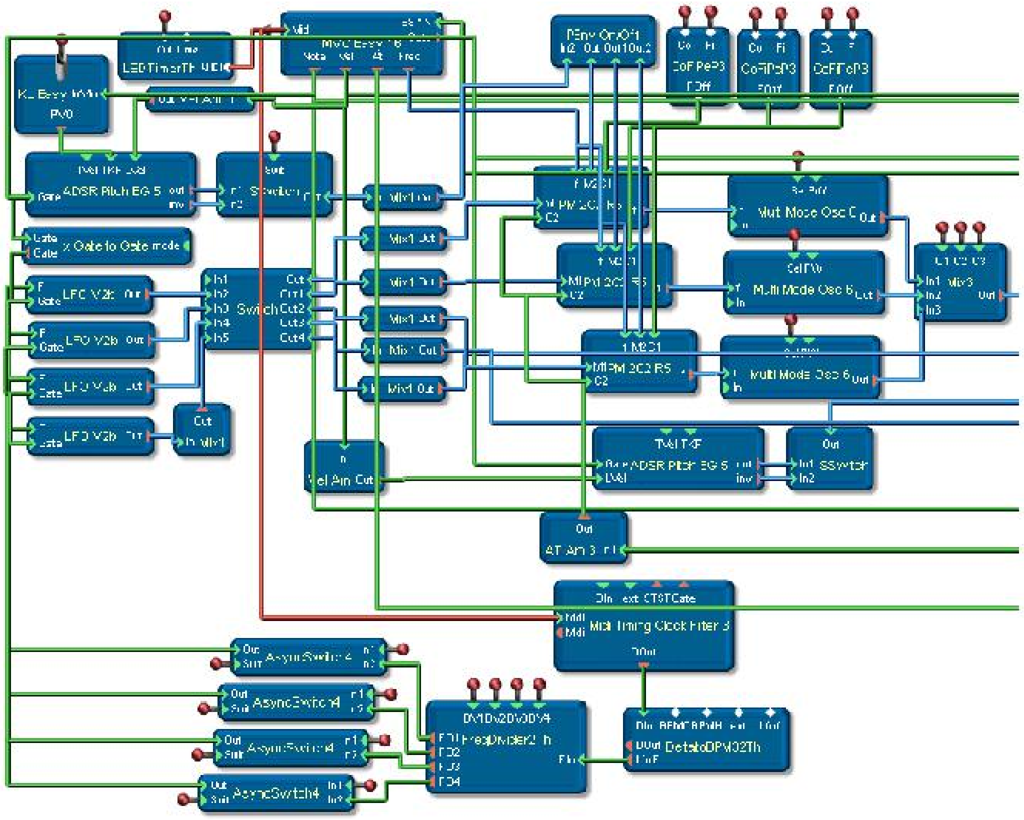
To help you understand how some of Creamware's synths were designed, we've included basic information on elements of the Blue Synth and Inferno synthesizers. Here's some comments on the Blue Synth.



When you first double-click on the BlueSynth device, you should see an arrangement of modules as shown above. The module you want is called 'Layout'. Double-clicking on this will get you to the entire layout of the BlueSynth. It's too large for a one-page overview, so we will look at it in two sections. As we go through the explanation, we suggest you keep the front panel (BlueSynth surface) open for frequent referral.

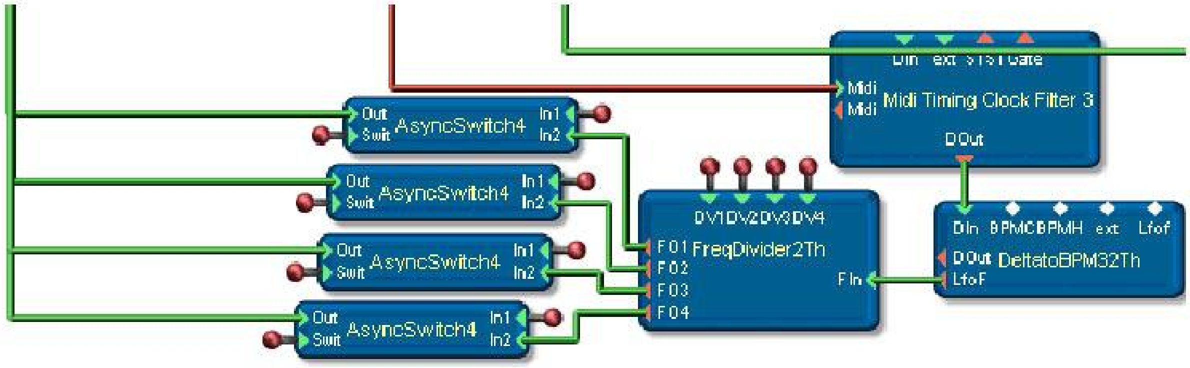
2

The BlueSynth has the following architecture: 4 LFOs, 3 multimode oscillators, 3 envelopes, and 1 filter. So, in this picture you will see a lot of repetition of modules. Let's try to break down this view into smaller sections. The section you see at the bottom with 4 AsyncSwitches, etc., is for running each of the 4 LFOs from a MIDI Clock source, with variable divisions of clocks per LFO possible.



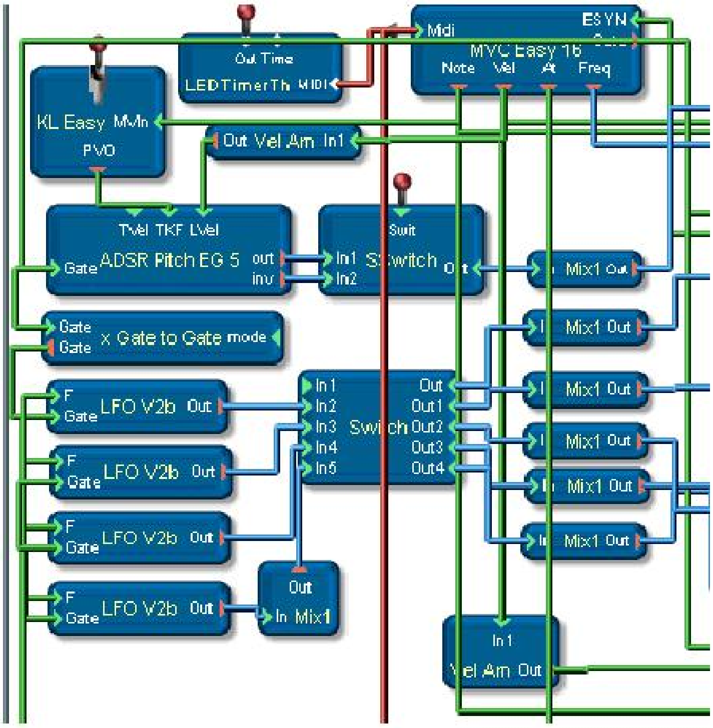
3

The Midi Timing Clock Filter 3 strips the MIDI stream of all but clock-related data (clock, start/stop, etc.). That data stream is sent to the DeltatoBPM32Th (converting pulses to Beats Per Minute), then on into the FreqDivider2Th, which sub-divides the pulses depending on the MIDI Clock Divider values selected in the GLOBAL drawer of the front panel. The AsyncSwitch4 determines if the attached LFO will get its frequency (rate) from a front panel knob or the Divider.

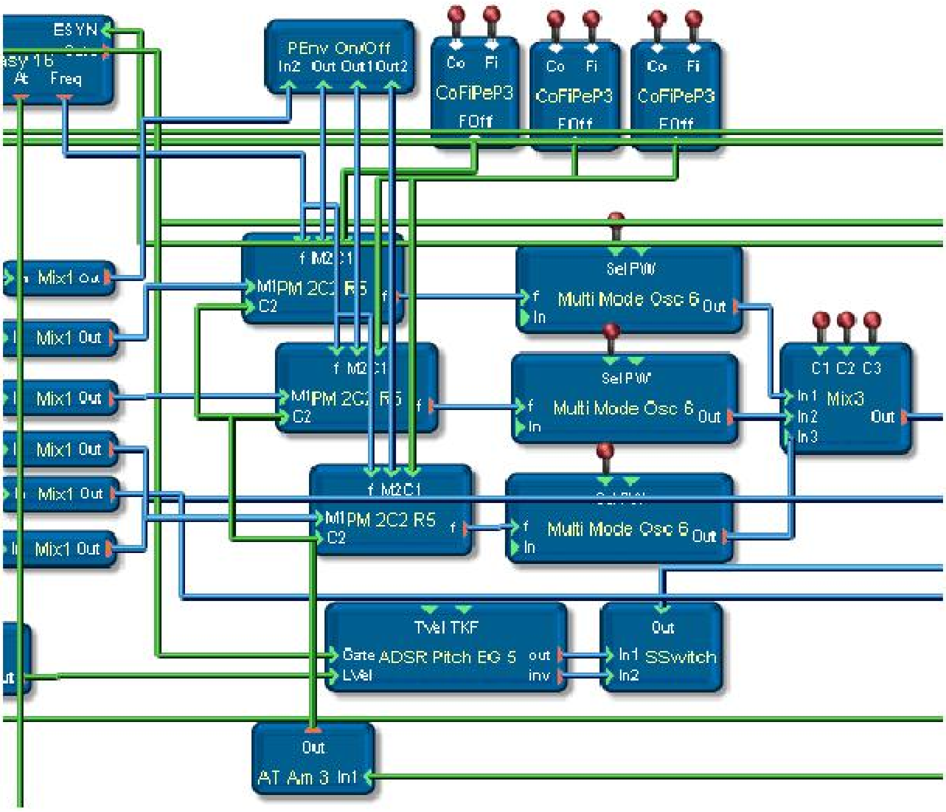


Here are the 4 LFOs on the left (LFO V2b), which are routed to a special matrix Switch allowing selection of any one LFO at each of 5 destinations. Notice only the bottom LFO has a Mix1 before going into the Switch. This LFO is called the LFO4 MW (Mod Wheel), as the Mix1 Gain is controlled by MIDI Mod Wheel controller data. Each of the matrix Switch's outputs is fed through a Mix1 module, providing for LFO Depth control at each of the 5 destinations.

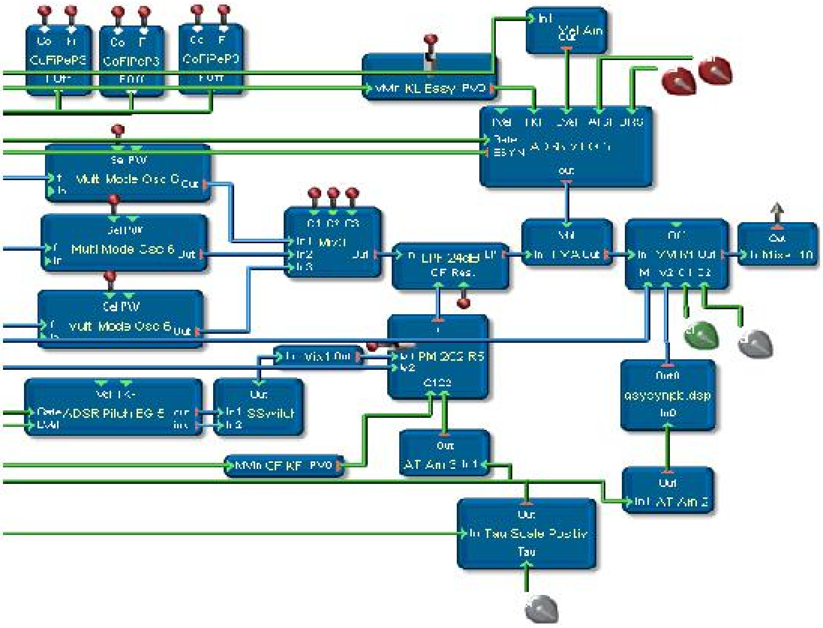
The module x Gate to Gate conditions the Gate signal from the MVC so that, when Retrigger for an LFO is set to ON, the LFOs' phase will reset correctly in polyphonic operation. The ADSR Pitch EG 5 is used for the Pitch Envelope for all 3 oscillators. It's connected to a Key Level Easy for keytracking, a Vel Amt for velocity response, a SSwitch to select normal or inverted ADSR, and a Mix1 for final amount. This next section shows the 3 Multi Mode Oscillators and their associated pitch modulation controls. Each oscillator has its own Pitch Mod block (PM 2C2 R5) and Coarse/Fine tuning control (CoFiPeP3). There is a single ADSR Pitch EG for pitch modulation, (which is pictured on the previous page, above the LFOs - the ADSR Pitch EG shown here goes to the Filter Cutoff). The Pitch EG for the oscillators is fed to a SSwitch module (here labeled PEnv On/Off) that determines whether or not the oscillator will receive ADSR pitch modulation (set by a button on the front panel).Each Pitch Mod block is set up as follows: frequency (f) is connected to the MVC Freq out, Mod 1 input receives the LFO modulation matrix, Mod 2 input receives the Pitch ADSR, C1 input receives the Coarse/Fine tuning offset, and C2 receives Aftertouch from the AT am 3 (Aftertouch Amount 3). Since M1 and M2 require synchronous signals, C1 and C2 require asynchronous controls, and we did not need scaling (as all of the mod sources are scaled before they arrive at the Pitch Mod block), the Pitch Mod block 2C2 R5 was selected for this application.



The outputs of the oscillators are fed into a Mix3 mixer.



Now let’s look at the last section:



To complete the BlueSynth, here is the filter, envelope (ADSR), and output section. The combined output of the 3 oscillators from the Mix3 goes into the Low Pass Filter 24 dB. From there it passes through the LVA for amplitude shaping, and then through the Volume Modulator 1 (VM R1), ending with the required module for polyphony, Mixer 16.

Volume Modulator 1 provides for a way to control the overall amplitude of the signal after being shaped with the ADSR v EG5 and LVA combination. The M1 input connects to the LFO mod source, and the M2 input connects to Aftertouch. The green and gray pots are set to maximum value for C1 and C2 to ”open up” the Volume

The ADSR v EG5 is the ”vintage” ADSR. It provides for independently variable slopes for both Attack and Decay/ Release (the ATSl and DRSl pads). Again, notice that both red knobs are set to the same maximum value* (so we could use only one here), but we wanted to have the ability to change the slopes separately for experimentation. The Attack slope varies from linear to log, and the Decay/Release slope varies from linear to exponential. Note: the values set on this level are also stored in the Preset, so as a Scope user, you could further customize your Presets with different slope settings for each.

There are also a few special modules of interest:
The Tau Scale Positiv is used whenever you need to smooth out a control signal (in the analog modular days it might have been called a slew or lag processor). The gray pot adjusts the amount of lag, and is typically most effective right towards the end of the pot’s range (just a small change at that range will be quite effective). In this example, we are smoothing out the Aftertouch signal, which is a positive-only (unipolar) control signal. Therefore, we need the Tau Positiv (there’s also a Tau bipolar).
Both the CF KF and Key Level (KL) Easy convert MIDI to keytracking information for the filter and ADSR, accordingly.

*Hold the right mouse button down over a pot, select ‘Open Control Ranger’, and you will see the current pot value.

Circuit Design

Circuit Design

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.

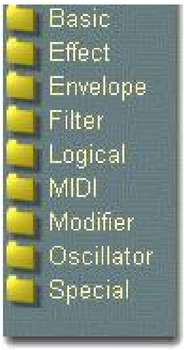
Connect a MIDI keyboard MIDI Out to the Scope MIDI Input. For this example we will use the ADAT 1 Destination channel 1 to monitor the audio.

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

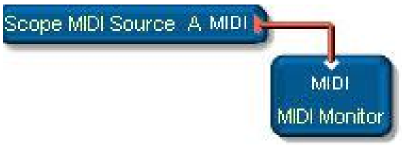
\\Scope\Data\Circuit Design

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.

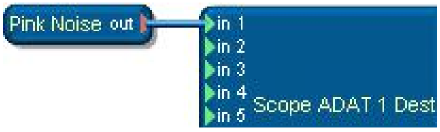
\\Scope\Data\Circuit Design**Modules**



Verifying the MIDI connection:



Verifying the audio output:



Delete the MIDI Monitor, and the Pink Noise generator.

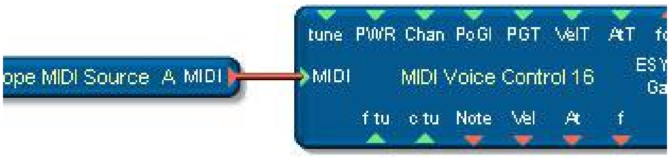
Drag the **MIDI Voice Control 16** module (\\Circuit Design\MIDI) into the Project window and connect the **MIDI** pad to the **MIDI** pad on **Scope MIDI Source A**.

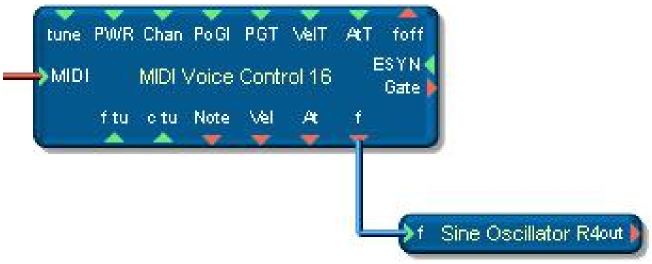
Now drag in the **Sine Oscillator** (\\Circuit Design\Oscillator) and connect the two **f** pads. The MIDI Voice Control sends frequency information according to the MIDI note number.

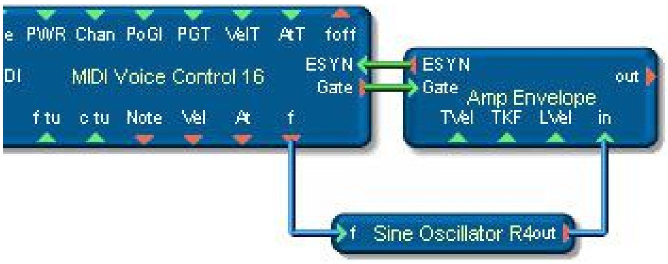
In the 'Envelope' directory you will find an integrated Envelope generator and Amplifier named **Envelope+Amp**. The envelope generator is triggered by information arriving at the **Gate** pad from the voice control. The voice control, in turm requires information from the envelope generator to handle polyphony properly (**ESyn pad** - for **E**nvelope **Syn**c). 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.

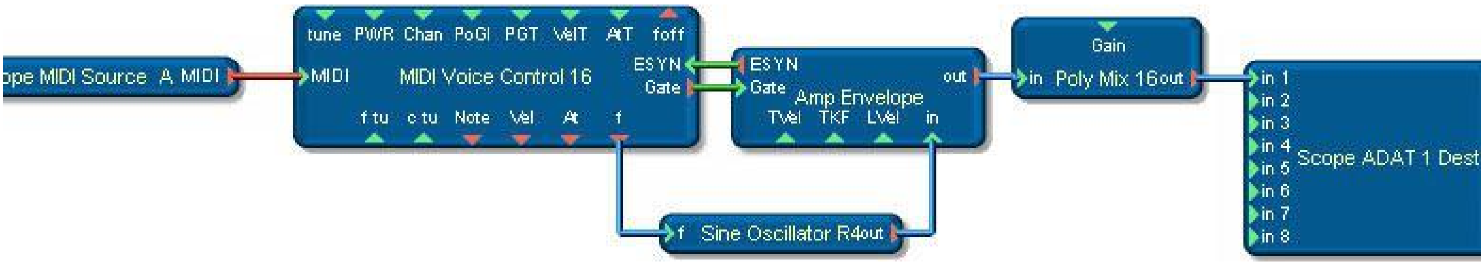
Make sure you are in Move mode (**<Ctrl><M>**) and double click on the 'untitled' module. Voila - the four floded modules appear. You are now in the first sublevel of this module.

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

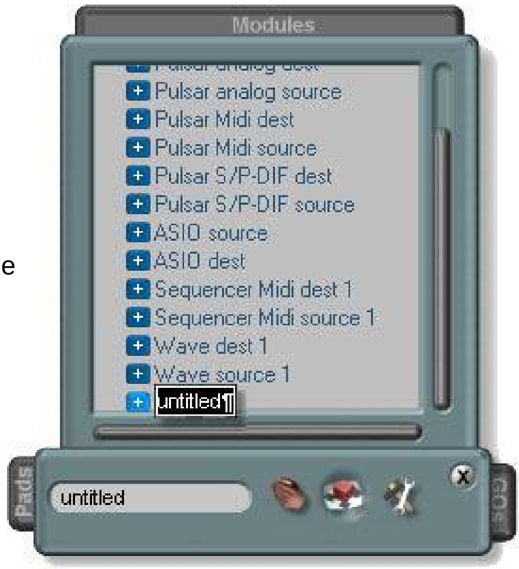
Click twice slowly over the 'untitled' entry, or use the **<F2>** key to enter edit mode for a tree entry. Type the name 'Sine Synth' and confirm with the **<Return>** key. The name of the module in the project will change appropriately.

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



- Pulsar analog dest
- Pulsar analog source
- Pulsar Midi dest
- Pulsar Midi source
- Pulsar S/P-DIF dest
- Pulsar S/P-DIF source
- ASIO source
- ASIO dest
- Sequencer Midi dest 1
- Sequencer Midi source 1
- Wave dest 1
- Wave source 1
- untitled¶

untitled

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.

13

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)

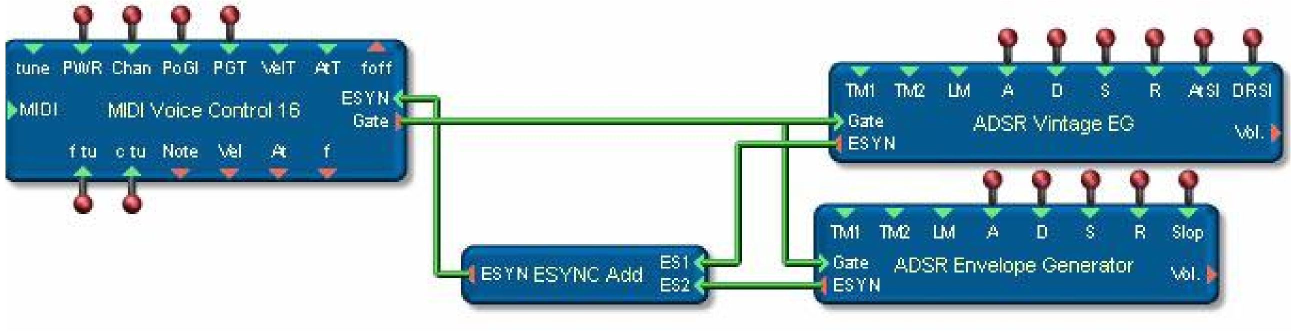
The two inputs are identical. The product TM1*TM2 controls the overall time compression or expansion.

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:

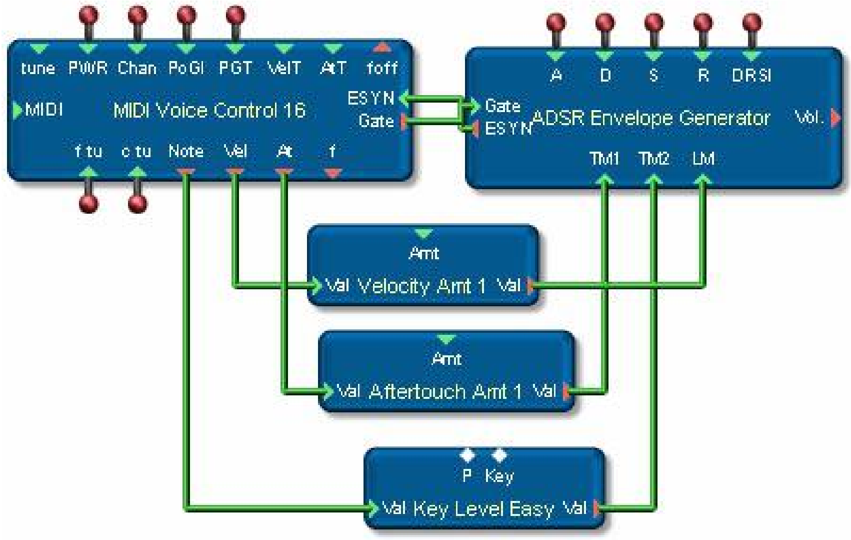
Here the overall envelope time and the sustain level are modulated. All modulation signals are asynchronous.

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

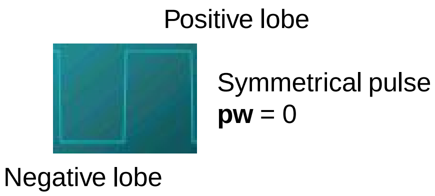
The value to use to set a 2s Fade-in time is

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.

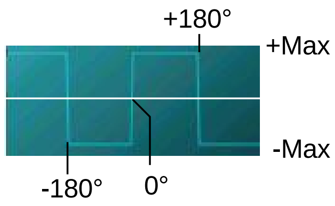


FIn/**FOut** pads (fade in and fade out control for LFOs)
(positive, asynchronous)
Fade in or fade out durations of 0 to 10 seconds result from the working range of values, 0 to 480,000. Input values greater than 480000 are reset internally to 480000.

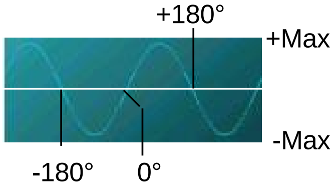
The following information is of particular importance when you are working with **LFOs** or **FM Operators**.

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.

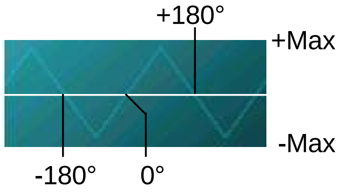
Rectangle



Sine



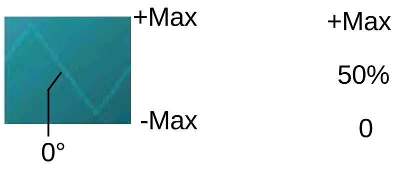
Triangle



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°.

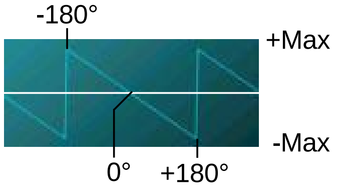
Volume Modulation:



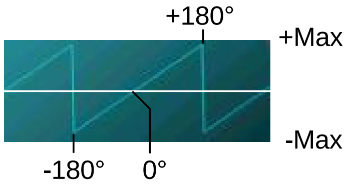
When using the ramp (sawtooth) waveforms in volume modulation undesireable audible effects can result from the characteristics of the wave shape.

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

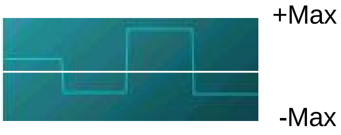


Sawtooth up



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

Random Number
Generator



Modulators

Volume Modulation

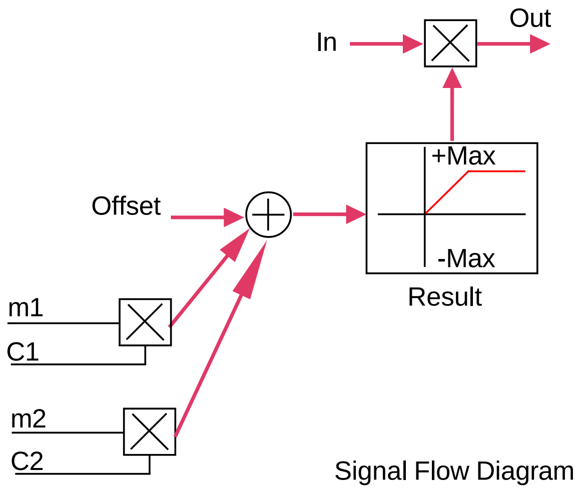
Volume Modulator 1 has three inputs - two modulation inputs and an input for a constant offset.

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

⊕ Adder

⊠ Multiplier

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).

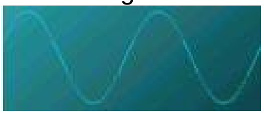
Fig. 2 shows the resulting modulation signal (transfer function). Negative values of the input signal have been ignored.

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.

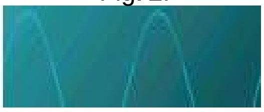
Fig. 1:



Modulation input

Transfer function 1

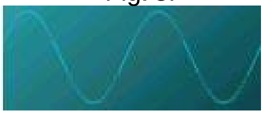
Fig. 2:



Modulation with
Offset=0
C=+Max

Transfer function 2

Fig. 3:

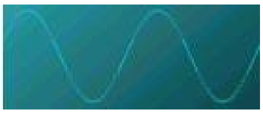


Modulation with
Offset=+Max/2
C=+Max

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



Modulation with
C=+Max

Modulation by key position:

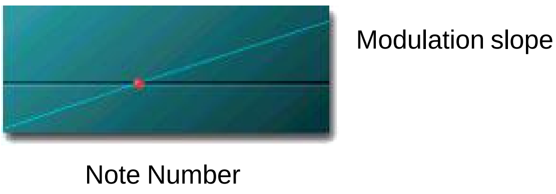
The **Key-Level Easy** module.

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

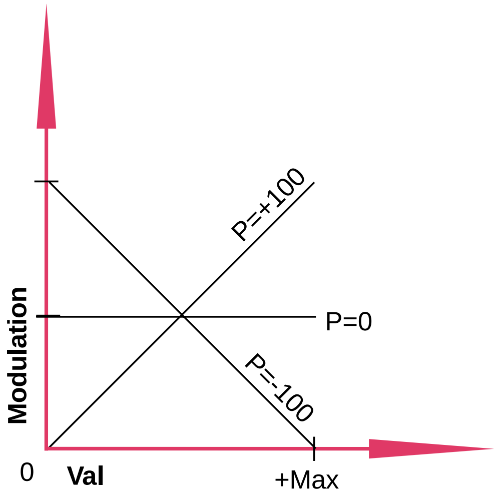
Val pad (controller input)
(positive, asynchronous)
You would normally connect this to the Note pad on the MVC. The MIDI keyboard serves then as a controller.

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).



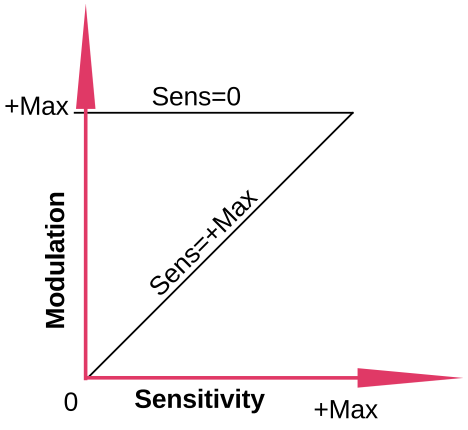
Modulation at Note Number = 64



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.

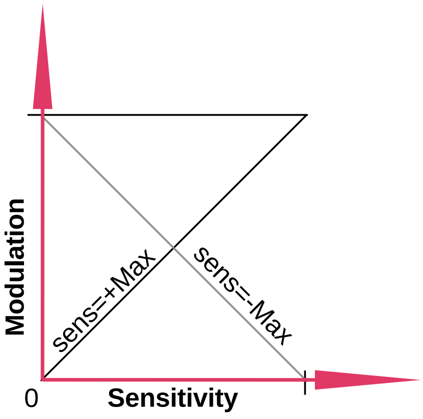
When Sens = 0, MIDI key velocity will have no influence and the notes will be played at maximum volume.

Maximum modulation is at Sens = +Max.

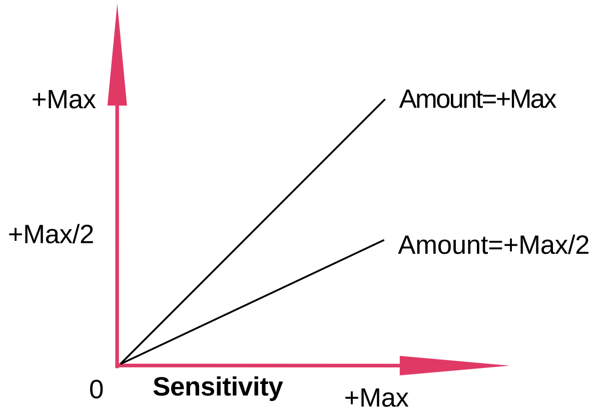


The **Velocity Sens 2** (Bipolar Velocity Sensitivity) module behaves like the **Velocity Sens 1** module, but permits inverse modulation - meaning high key velocities will produce weaker tones.

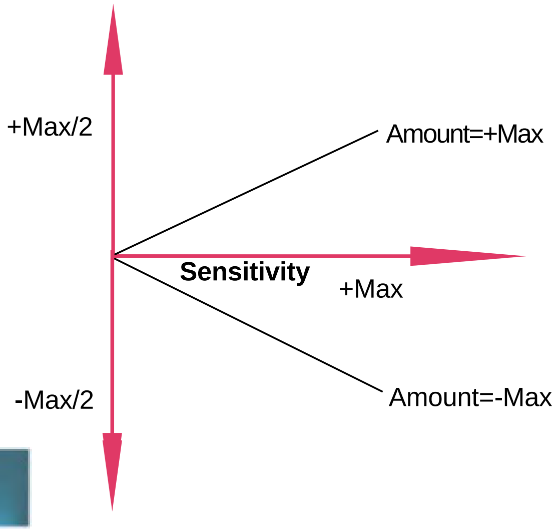
Maximum inverse modulation is at sens = -Max.



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.



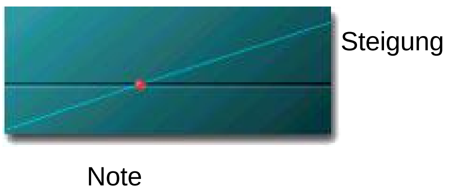
The **Cutoff Key Follow** module responds to controller data. Normally the input pad is connected to the Note pad of a MIDI voice controller so that cutoff frequencies become a function of the MIDI note number. However, any positive, asynchronous pad can be connected to the input. The output should be 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)



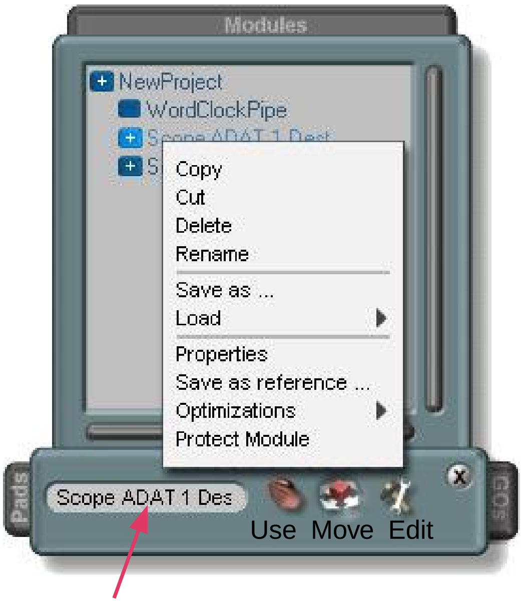
The **Cutoff Key Follow 2** module behaves like the Cutoff Key Follow 1 module, but with in increased range at the P pad of -200 to +200.

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



Selected module

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').

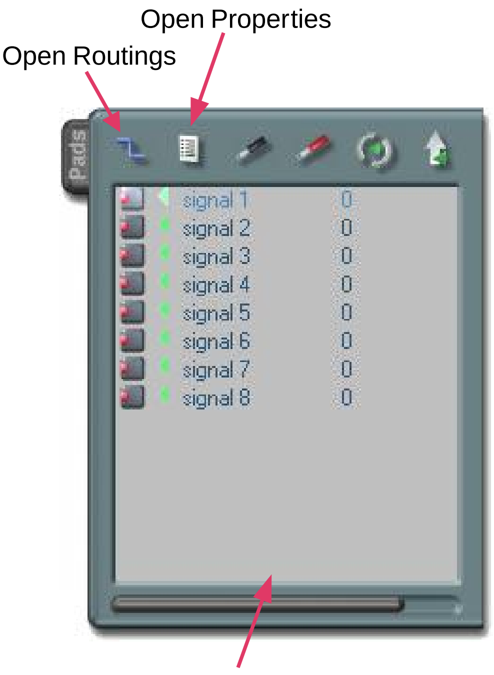
For details about the "**Protect Module**" option, refer to chapter "Device Design".

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).



signal 1	0
signal 2	0
signal 3	0
signal 4	0
signal 5	0
signal 6	0
signal 7	0
signal 8	0

The Pad List

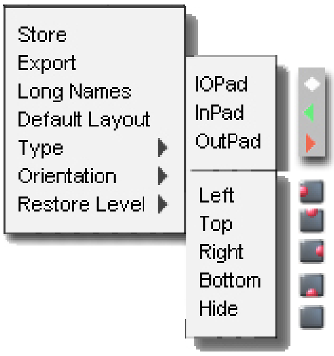
You can further organize your project for clarity by making some pads invisible, and arranging others so that connections are kept relatively short.

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.

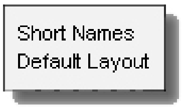
By clicking again on the **Export Pad** icon, the copy is removed from the parent module.

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.



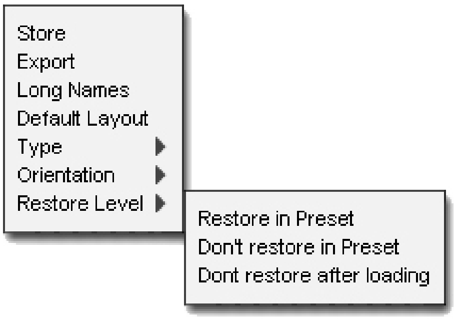
The Pad List's Context-Sensitive Menu

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.



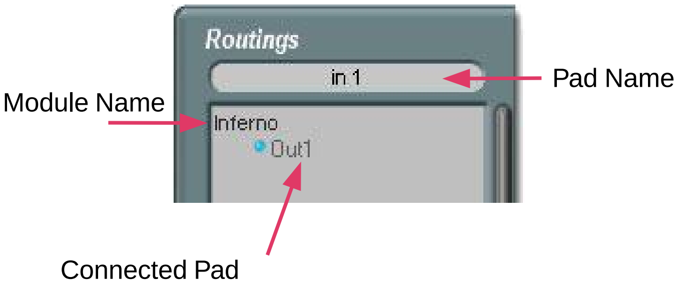
Select **Default Layout** to restore the module's original representation. Modifications made to pad types will not be restored, however.

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).



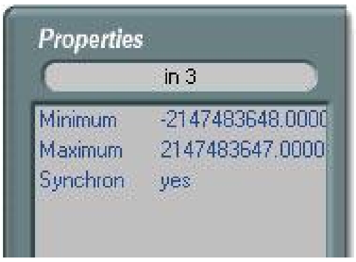
To edit a value or change a name in a list click twice slowly with the left mouse button while the mouse cursor is over the name or value.

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.



Minimum -2147483648.0000
Maximum 2147483647.0000
Synchron yes

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.

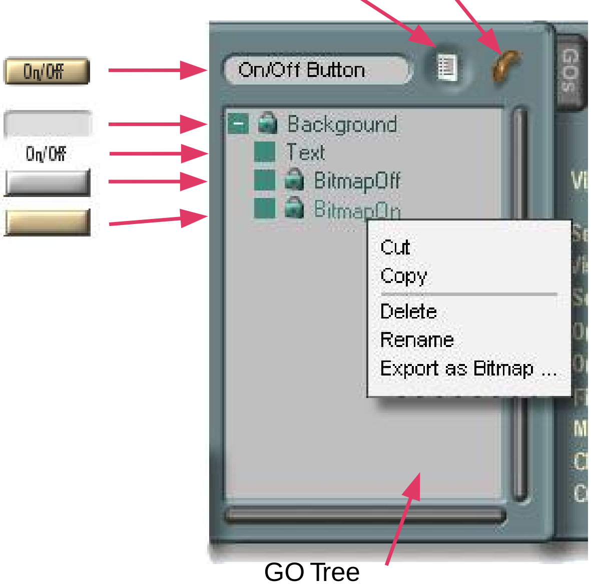
The **GO Tree** displays the graphical objects (GOs) for the selected module. Please see also the *Surface Design* section.

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.



GO-Properties
GO-Properties
GO Tree

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.

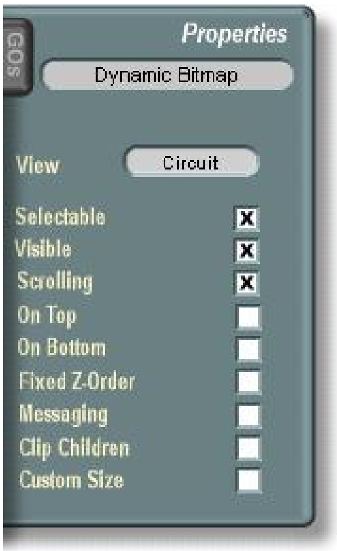
2. GOs at the same level are drawn in the order in which they appear in the list.

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.



Dynamic Bitmap

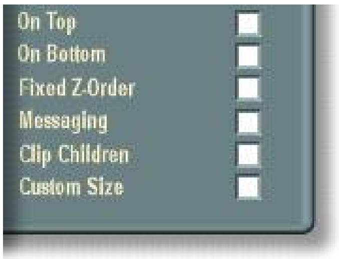
View: Circuit

Selectable ☒
Visible ☒
Scrolling ☒
On Top ☐
On Bottom ☐
Fixed Z-Order ☐
Messaging ☐
Clip Children ☐
Custom Size ☐

Art des GOs

GO-Properties

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.



On Top □
On Bottom □
Fixed Z-Order □
Messaging □
Clip Children □
Custom Size □

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

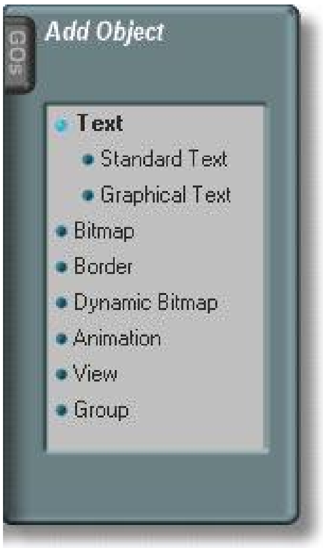
Fixed Z-Order means it will alwys be drawn according to the rules.

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.

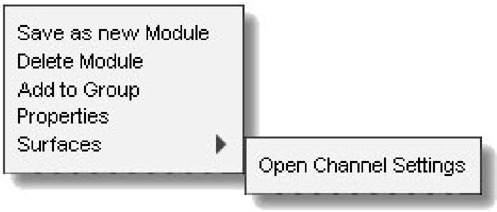
Context Sensitive Popup Menus in the Project Window

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

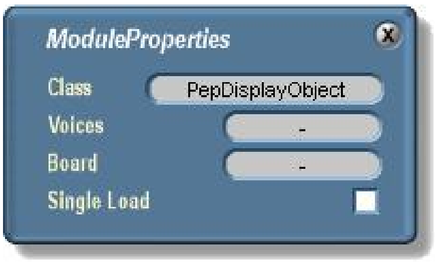
Surfaces brings up a list of all surfaces a module uses. A single left mouse click opens the surfaces.

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


ModuleProperties

Class PepDisplayObject

Voices -

Board -

Single Load □

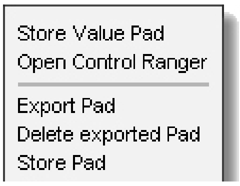
Module Properties Dialog

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.

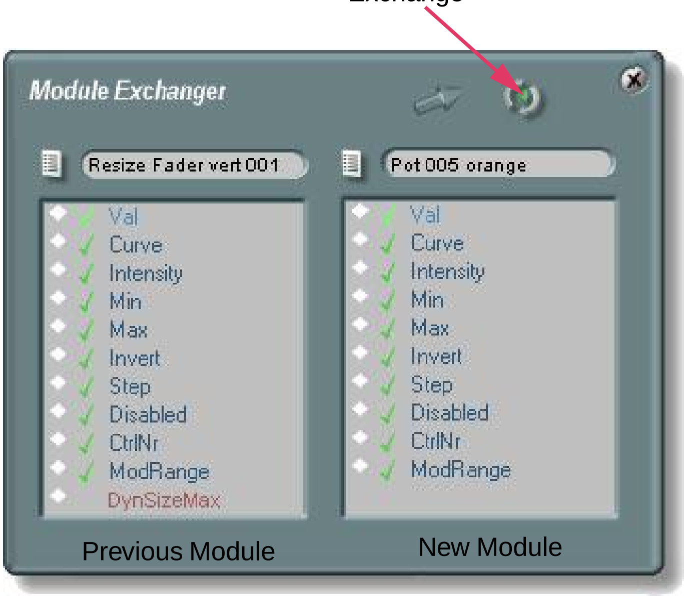
In the example to the right a fader is potentiometer is being exchanged in favor of a fader. All the connections that could be assigned are indicated with a green check mark. Connections that could not be assigned are indicated in red.

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.

Exchange



Old Module New Module

Control Ranger

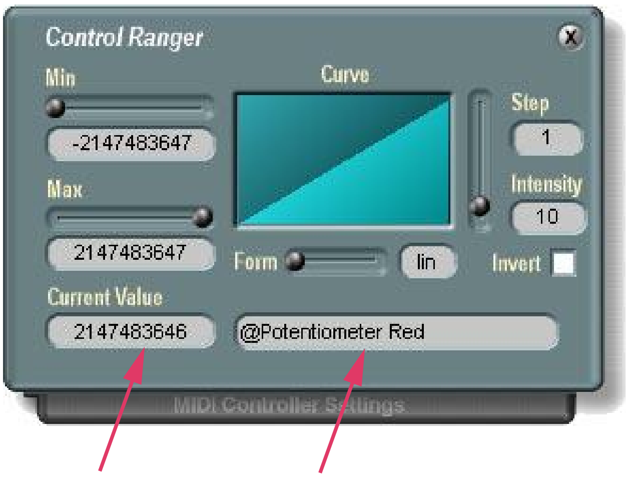
The Control Ranger is used to adjust the characteristics that determine the behavior of a control.

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.

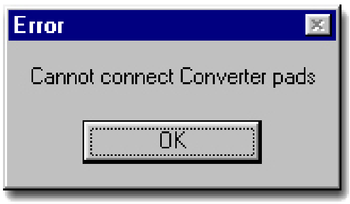


Current Value Controller Name

Miscellaneous

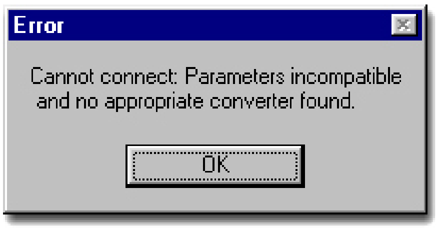
Error Messages

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


Error

Cannot connect Converter pads

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


Error

Cannot connect: Parameters incompatible and no appropriate converter found.

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

Device Design

Return To Main Table Of Contents

Device Design

Preface

In this chapter we'll lay the groundwork for turning a collection of interconnected modules into a working Device.

Please go through the tutorial - there is a lot of practical information there.

At the end of the chapter some final directions are given needed to fully finish a device.

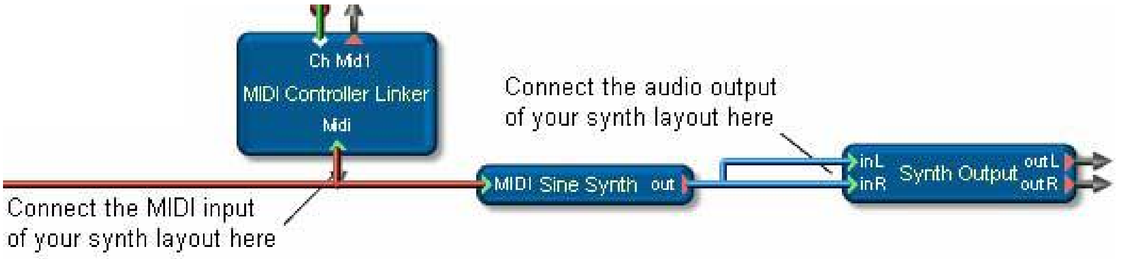
Sine Synth Device Tutorial

Here we will transform the Sine Synth module from the *Circuit Design* section into a finished **Device**.

First drag the Empty Synth (\\Scope\Data \DeviceDesign) into the Project window. Double click on the Empty Synth to go down one level. Now drag the Sine Synth module (from \\Scope\Data\Examples) into the Project window and make the appropriate audio and MIDI connections as shown below.

So that both output channels will be audible, connect the Sine Synth's **Out** pad to both the **inL** and **inR** pads of the **Synth Output** module.

Some modules in the Sine Synth have surfaces where you can adjust some of the parameters. For the Sine Synth to become a device, we first need to make only the current MIDI channel available with the **MIDI Channel Linker** module.



Connect the audio output of your synth layout here

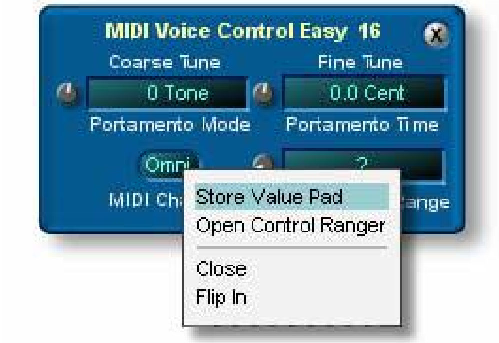
Connect the MIDI input of your synth layout here

Open the context sensitive menu for the **MIDI Channel Faders** as shown on the right. Select the entry **Store Value Pad.**

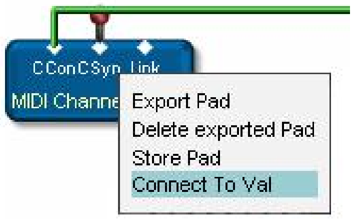
Now right click on the **CSyn** pad (below right) and select the entry **Connect to Val**. A connection symbol appears at the pad indicating that the MIDI Channel Linker is now connected to the adjusted MIDI channel.

As you know from the SCOPE synthesizer devices, you can assign a MIDI controller to a potentiometer or fader. In the **MIDI Controllers Settings** drawer in the **Control Ranger** you can select whether the control is to receive MIDI data from a MIDI channel or not. You must enable the **Link Channels** option for a control to read a MIDI channel.

MIDI Voice Controls Easy 16 Surface:



MIDI Channel Linker Module:



The next step describes how to quickly make a functioning control surface.

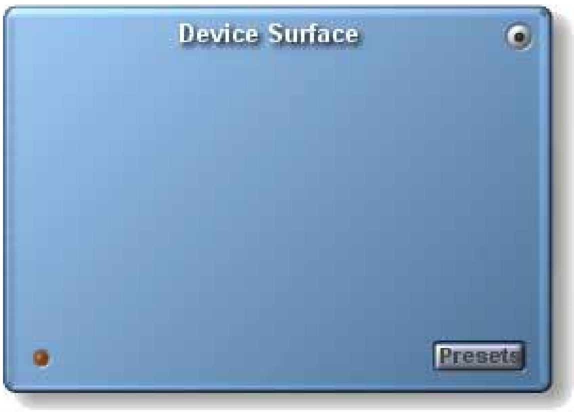
Double click on the minimized **Empty Synth** surface to open it to full size. In the upper right corner is a **close** button. At the top in the center is a graphical text **GO** that reads ‘Device Surface’. In the bottom right corner is an LED to indicate MIDI activity. And at the bottom right is a button to open the Preset list.

So that you can set the MIDI channel, your surface needs a text control. It is simplest to use the existing functional control on the MIDI Voice Control module already assigned to the surface.

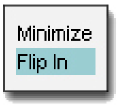
Using **<Ctrl><X>** you can, so to speak, cut the control from the MVC and assign it to the surface, but you first will need the help of the Project Explorer.

The device surface is in a special state called **Flip Out** mode. This mode is meant for using the surface, not designing it. For editing you must put it in **Flip In** mode. This will put the surface ‘inside’ the Project window. Select **Flip In** from the surface’s popup menu.

Minimized View



Empty Synth Surface



Popup menu for a surface

Open the **Project Explorer**, which will aid you in the selection of modules.

Go to the Module Tree and select the **MiditextFader** in the branch under the **MVC Easy 16 Surface**. With this module selected, click use the **Copy With Connections** function - **<Ctrl><X>**.

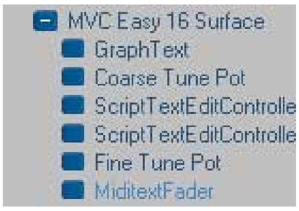
Now select the **Empty Device Surface** in the **Project Window** and its entry in the module tree will become highlighted. Use **<Ctrl><V>** and the text control is made a child of the Empty Device Surface. If you move the surface, the control moves along with it.

Now you can position the text control somewhere on the surface.

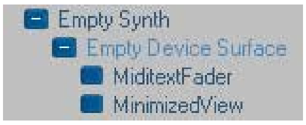
In the same way you can bring other elements onto your surface - for example, the potentiometers and text displays of the **Filter+Amp** surface.

Using the controls of the existing modules like this is probably the fastest way to learn how to equip a surface with controls.

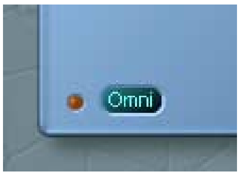
When you have finshed laying out the desired controls on your surface you can close all the various module surfaces and simply select the **Flip Out** option from



- MVC Easy 16 Surface
  - GraphText
  - Coarse Tune Pot
  - ScriptTextEditControlle
  - ScriptTextEditControlle
  - Fine Tune Pot
  - MiditextFader



- Empty Synth
  - Empty Device Surface
    - MiditextFader
    - MinimizedView



your surface's popup menu to enable it for use.

You can go on adding controls and making connections but one important consideration applies - make sure you connect the pads of the controllers directly to the pads of the modules, and not to their default surfaces. The reason for this:

When you optimize your synthesizer, all default surfaces and their connections are lost.

When you are 'finished' with your synthesizer, you should save a 'work version' of it. The next steps, **Optimization** and **Protection,** produce a device that can no longer be modified (described later in the manual, see pages 16-18).

Fundamentals

There are only a few basic things you should keep in mind when building devices. One thing - when you have a basic structure for a synth or an effect, you should fold the layout into a new module and give it a meaningful name (e.g. Synth Layout). Save this for security.

Empty Synth

Use this surface to quickly transform your layout into a synthesizer device. This bare-bones device provides MIDI controlability, a Preset list, and a surface with a minimized view.

The method for creating a working synth using this device is described in the tutorial section.

The Empty Synth module belongs to the script class **NewDevice** - thus it is a Device module. It's in the \\Scope\Data\Circuit Design\Basics folder and in the device Design file (\\Scope\Data\DeviceDesign). It is 'camouflaged' as a module, however simply by renaming the file with the .mdl extension.

By renaming it as a module it is ensured that it won't be detected as a device when it loads, and therefore is not loaded as a *reference*. This way you are prevented inadvertantly storing your Empty Synth project without modifications. (When an object is saved as a 'reference' only information about its name and location is saved with the project. The modified object is not included in the project's data).

The Empty Synth already contains some modules as described below.

Num Voices Of Parent

With the **Num Voices Of Parent** module you can determine the polyphony of your synthesizer. This value is set at the **Voic** pad. As the Empty Synth conatins this module, the Empty Synth is the module's parent. As the Num Voices Of Parent module value increases, so too does the number of voices for the Empty Synth.
The polyphonic modules which are contained in the device will inherit this value. Polyphony, therefore, is a property inherited by all its children.

For synthesizers the number of voices is limited to 16. This restriction comes from the Voice Controls themselves, as they cannot administer more than 16 voices.

MIDI Channel Linker:
Allows MIDI channels to be used for controllers.

MIDI Controller Linker:
This module ensures that MIDI controllers can be assigned. It makes MIDI input available to a controller. If a controller is assigned, moving it onscreen also provides MIDI output.

Preset List Loader:
Provides for the loading of parameters via Presets. This module must be a child of the Device module. You must connect the **Voice** pad of this module to the pad of the same name on the Num Voices Of Parent module in order Presets to adjust this porperty. If the value at the **Show** pad is 1, then this pad will be visible on the surface.

MIDI LED Timer:
Used to indicate when a MIDI signal is present. The module has an output for a controller. In the Empty Synth, this output of the MIDI LED Timer is connected to the red LED on the Device Surface.

Synth Output:
Routes the output of the synth layout to the the surface of the device. This is a placeholder used for pads you rename and export in your layout.

Empty Effect

The Empty Effect device is of the Script class EffectInsert. These devices can be used with the Big Mixer's mono and stereo inserts. For it to work correctly there is a consideration with regard to pad names. The system uses a naming convention to identify the inputs and outputs of an effect, so these names must not be changed.

If the effect is to be used as an aux effect only, then you don't have to worry about the pad names.

There is an option, **Use mono Input (Left)** that, when enabled, routes the signal arriving at the left input to both the left and right outputs.

The **Presets** button is enabled and connected ready to load a preset list for your effect.

The **Bypass** button is enabled and connected. When pressed, it routes the input directly to the output.

Disabled

Preset Button

Bypass Button

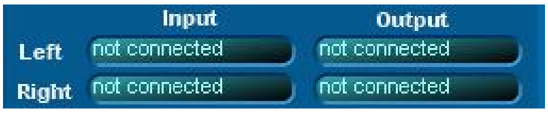
In the **Connection** drawer there are four text fields used to display the name of the connected module. For this to function, they must know the name of the effect. If you have finished your design, and have renamed your effect to, say, '4-D Audio', put this name in the field to the right of the 'Parent' pad in the text field pad list.

Since the pads of the four text fields are connected, you only need to enter the name in one of the pad lists.

You can also connect these fields. Simply click on the fields, and then, for example, click on the AUX input on the Big Mixer.

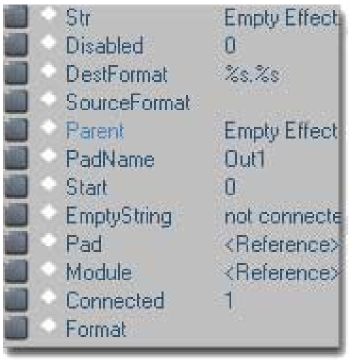
You can remove the Connections drawer entirely if you don't need routing text in your design.

Structure your design between the **Effect Input** and **Effect Output** modules, and connect your I/O pads to them. See the *Chorus Effect* example in \\Scope\Data\Examples.



	Input	Output
Left	not connected	not connected
Right	not connected	not connected

'RoutingText' Text Fields



Str	Empty Effect
Disabled	0
DestFormat	%s.%s
SourceFormat	
Parent	Empty Effect
PadName	Out1
Start	0
EmptyString	not connecte
Pad	<Reference>
Module	<Reference>
Connected	1
Format	

Pad list for RoutingText fields

Assigning Surfaces

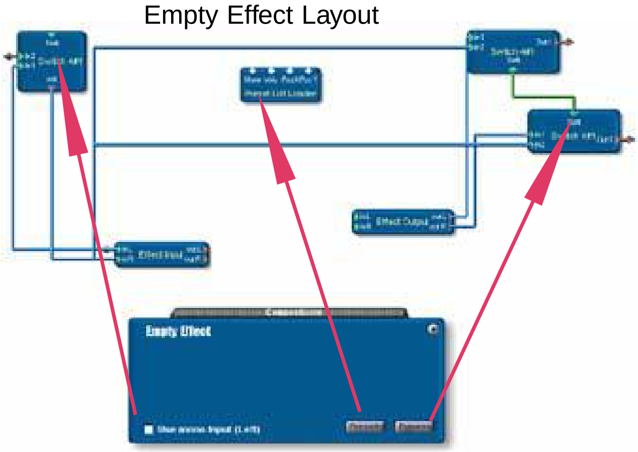
The surface of the **Empty Synth** has an associated minimized view. By doubleclicking on it, it expands to the full view.

The **Empty Effect** surface does not yet have an associated minimized representation. You can, however, drag the 'Empty Effect Surface M' (\\Scope\Data\Device\Design) surface to the first level under Empty Effect and delete the older surface.

The new surface does have a minized view attached. You must now restore the connections to the switches. Use the modules' popup menus (Store Pad, Connect to...). Connect **Use Mono Input (Left)** to the **Swit** pad on the module in the upper left part of the layout, the **Presets** button to the **Show** pad, finally the **Bypass** button to the **Swit** pad in the right side of the layout.

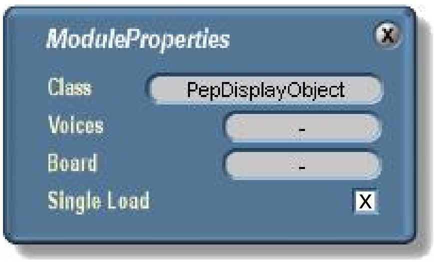
Minimized View of the Empty Effect

Empty Effect Layout



Gate & Sync

If you want to use the **Single Load** option for an LFO (which is more efficient in a polyphonic environment, and recommended) you will need to include another module for correct MIDI Note triggering of the LFO.



Class PepDisplayObject
Voices -
Board -
Single Load ☒

The **Gate To Gate** module receives and transmits a gate signal to a single-loaded LFO so that it will only retrigger the LFO after all keys have been released. This is the case when the value at the **Mode** pad is 1. If this value is 0, then the LFO triggers each time a key is pressed.

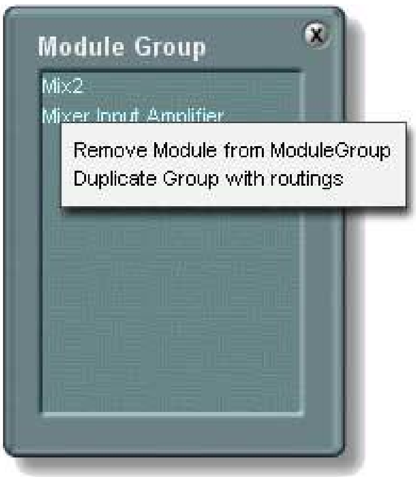
With FM Operators a synchronous trigger is required to guarantee phase-accurate triggering.

The **Gate to Sync** module converts an asynchronous **Gate** signal to a synchronous **Sync** signal. Using this, all FM operators will start at the same sample tick.

Dialogs

Module Group

Use the **Module Group Window** when you want to make several copies of a module, such as a mixer 'rail' while maintaining the connections between the surface and the layout. First fold the modules in the layout together, and then the group the graphic components together as desired.



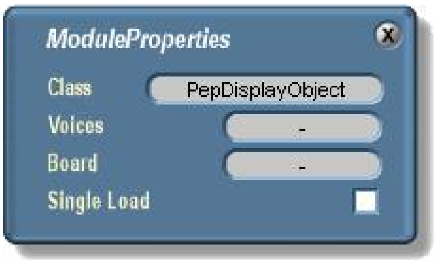
Click on **Add to Group** in the layout module's and graphic group's popup menus and open the **Module Group** window.

In the window you will see the names of the two module groups you added. Right click on one of the two modules and choose the entry **Duplicate Group with routings**.

You have now created an exact copies, with routings, of both modules. They are located directly over the originals and all you have to do is move them to the desired locations. Repeat for as many times as required.

This technique is similar to simply copying a group, but it copies many groups, on different levels, with their routings intact.

Module Properties



Class: PepDisplayObject
Voices: -
Board: -
Single Load: □

ModuleProperties Dialog

In the first field, the **Script Class** of a module is displayed.

The **Voices** scroll field value indicates the number of voices set for the module. If it is left blank, it inherits the value from its parent. It will also pass it down to its children.

The **Board** field lets you choose which hardware platform (Scope, Pulsar, Pulsar SRB) this module is to be loaded to. If nothing is explicitly indicated, it will load automatically to an appropriate board. Please use this only in exceptional cases.

Often it makes sense to enable **Single Load**. Normally in a polyphonic circuit, when you load an LFO, 16 LFOs are actually loaded (if you are using 16 voices). You don't see this, but that's how it works. With an LFO you don't need to load, LFOs don't produce or pass voices - they control modulators.

Enabling this option reduces the implied performance penalty.

Optimization

When you have finished a design you should follow the optimization procedures below. This 'finishes' your synthesizer and frees a great deal of your PCs working memory.

Make sure you store a work version of your device somewhere seperatly, as an optimized version is no longer capable of being edited.

Optimizing Layout Modules:

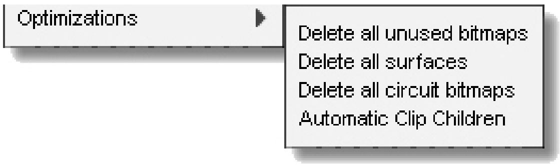
1. Select the layout module in the Module Tree.

2. Delete all surfaces:
This option deletes all layout module surfaces, and child-module surfaces. These surfaces are not required after the design is complete.

Make sure no connections to module surface controls exist on the device surface.

3. Delete all circuit bitmaps for the Device:
Deletes all bitmaps of the module's child modules. You will now only be able to access the modules in the Module Tree.

Popup menu for Modules in the Module Tree



4. Delete all Unused Bitmaps
This deletes all hidden pad bitmaps and their routings.

After these three optimizations, only the surface module for the Device itself remains.

Before continuing on to optimize the surface, we recommend you save the device again.

Optimizing the Surface:

4. Automatic Clip Children
This optimization speeds drawing of GOs of the types: ‘Back’, ‘Background’, ‘Integration’, ‘BKGround’, ‘Back1’ and ‘View’.

In certain cases Clip Children can lead to problems. If this happens, disable the Clip Children option.

5. Flat Surface
Combines all the graphic elements, including shadows and dynamic bitmaps into a static bitmap.

The surface cannot be changed in size, or shadowed any differently - it is as if it were a ‘screenshot’.

Independent modules, such as Controllers, are not affected by this optimization.

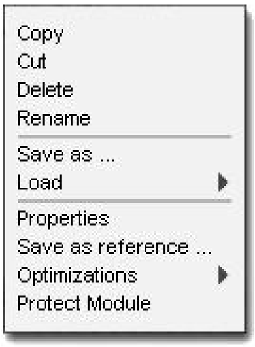
If a surface includes drawers, they are optimized the same way.

Last Step - Protecting Your Device

You’ve got an optimized, working device, and it’s time to perform the final step - protecting the device. Now your synth will becomes a fully functioning, independent SCOPE device.

Important: After performing this step, access to the internals of this device is absolutely impossible. It can never be opened again. Make sure you have made a safety copy of your device layout before protecting it.

To protect your synth, simply bring up the popup menu for your Device module and select **Protect**. That’s it! You’ve just given birth to a new SCOPE synth, and it didn’t hurt at all!



Control Design

Return To Main Table Of Contents

Design Tools

SCOPE is an environment in which you not only create sophisticated signal-processing circuits - you also build the graphical control surfaces required to operate them. The possibilities for graphic control design go far beyond what is technically required. From the beginning the SCOPE concept included the capability to develop professional looking, responsive control surfaces. The same modularity you find in SCOPE's circuit design extends to graphic design. Each graphical object can be considered to be a module. A potentiometer, for example, is not really much different than a DSP module. Graphical objects, like DSP modules, can contain inputs and outputs, and are therefore connected in a circuit quite easily.

You can begin designing surfaces immediately using the various control and display modules supplied with SCOPE. Simply drag the default 'surface' into the Project window, arrange some potentiometers, faders, displays etc. on it, give them some labels, and very quickly you have created a functional, custom control surface.

You can also create new controls by modifying the existing ones. Before attempting such modifications it is important to understand from what basic objects (GOs - Graphical Objects) a control is actually assembled. We will examine bitmaps, animations, dynamic bitmaps, standard text, graphic text, borders and groups in detail. From these seven elements you can construct most displays and controls - even quite complex ones. A control's functionality is determined by a script file. Think of it this way - a control is a collection of graphical objects designed to produce values according to a script. The values are then used to control another module.

A simple example would be a fader. A fader consists of at least two objects, the fader knob, and the rail on which it slides. In this case the fader knob and rail are both simple bitmaps. Through the script, the fader knob 'knows' that it moves vertically over another object called a 'rail', and that the dimensions of the rail restrict the area within which the knob can move. The script produces a value derived from the relative position of the knob to the rail. You don't have to know or worry about how the script does this. What you can do, though, is create an entirely new looking, fully functional fader by substituting different bitmaps. You can also add new objects to the fader. For example, you could add a bitmap to indicate the scaling (i.e. a graphic of log scaling). You can then save this new fader and use it as often as required in your project - simply pull it out of the browser onto one of your surfaces.

The first section of this manual describes the design tools available. Use these to edit or modify basic graphical objects. The second section details the standard controls - faders, potentiometers, buttons etc. and gives you some ideas on how to modify and customize them.

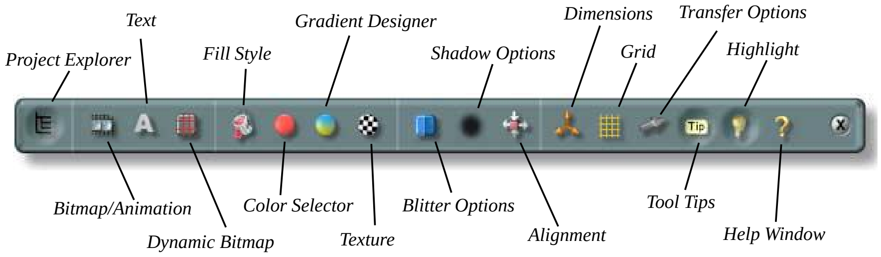
GOs and their Editors?

Each control consists of one or more Graphical Objects (GOs). GOs posses many ‘properties’ that can be modified using a set of GO editors. These editors are available in the Toolbar - clicking the mouse on their respective icons opens and closes them. When you select a GO, all the relevant editors are updated to reflect the properties of the object. For example, the Texture Editor will display the texture of the currently selected GO when you open it. If the current object is filled with a simple color, rather than a texture, the Texture Editor will display the texture of the last selected object that had a texture. In this case, changing the texture would have no effect on the object. You must first set an object’s Fill Style to Texture to use the Texture Editor.

Simply because an editor is open, then, does not necessarily mean that it will have any effect over the current object. For example, if you click on a bitmap and then try to use the Text Editor, nothing will happen. Or, if you click on Exchange Contents no File Open dialog will appear because this action applies only to Dynamic Bitmaps. This must be selected and attached in the Dynamic Bitmap Editor. If an editor does not produce the expected result on a GO then the object is probably incompatible with the editor.

The Toolbar

When working on the surface design portion of your project you should keep the Toolbar open. The Toolbar gives you quick access to all the GO editors. Open the Toolbar from the Window menu - Toolbar.



Project Explorer: Opens and closes the Project Explorer dialog - a central repository for the project’s overall design and routing information.

Bitmap/Animation: Opens and closes the Bitmap/Animation Editor in which you can exchange bitmaps or animations.

Text: Opens and closes the Text Editor for adding or modifying plain text.

Dynamic Bitmaps: Opens and closes the Dynamic Bitmap Editor.

Fill Style: Opens and closes the Fill Style Editor where you choose the fill style for dynamic bitmaps and text.

Color Selector: Opens and closes the Color Selector in which you select the colors for dynamic bitmaps, text and gradient nodes.

Gradient Designer: Opens and closes the Gradient Designer used to process color gradients for dynamic bitmaps and text.

Blitter Options: Opens and closes the Blitter Options dialog. The Blitter Options determine the manner in which a bit map is drawn on the display.

Shadow Options: Opens and closes the Shadow Options dialog where you assign shadows to a objects.

Alignment: Opens and closes the Alignment dialog where you can adjust an object’s position relative to its parent.

Dimensions: Opens and closes the Dimensions dialog in which the position and size of GOs and modules can be adjusted.

Grid: Enables or disables the Grid.

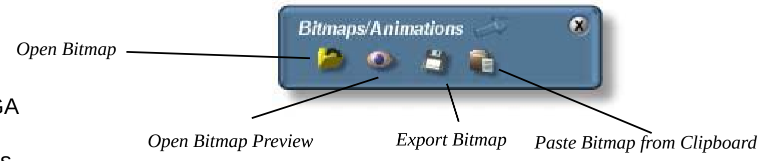
Transfer Options: Opens and closes the Transfer Options dialog which lets you transfer several properties from one object to another at one time.

Tool Tips: Enables and disables the Tool Tips feature.

Highlight: Enable this so selected modules will be visibly highlighted.

Help Window: Opens and closes the Help window where information on the selected module or pad is displayed.

Bitmaps/Animations



This dialog serves to provide content for bitmaps and animations. This editor supports 24-bit BMP and 32-bit TGA files. For rectangular bitmaps with no transparency component you can use the 24-bit BMP format. For objects with rounded or other transparent or semi-transparent regions (such as a potentiometer knob) you will need the alpha-channel. This is supported by only a few graphics formats, and in SCOPE it is supported only for the TGA format at present.

Animations also use BMP or TGA images. The 3d images used to animate the potentiometers were rendered as a sequence of TGA files (pot01.tga, pot02.tga, pot3.tga ...). The number of frames you use is arbitrary and does not influence the resolution of the values the control can produce. The more frames you use the smoother the animation will be - and the greater the storage requirements. The potentiometers use 32 images, which is quite sufficient to produce a smooth sequence. In general, the following rule applies: the larger the control, the more frames required to produce a fluid animation.

Open Bitmap: To open a BMP or TGA file for editing click on the Open Bitmap icon and select a file. Click on OK to confirm and the bitmap will be loaded into the editor.

Preview: Clicking on the Preview button (‘eye’ icon) opens a dialog that displays the currently selected GO. Since all graphic editing in SCOPE is in real time, this dialog isn’t strictly necessary. It’s useful, though, when you want to view the bitmap of a GO that is not visible. You may want to do this to check the button state of an invisible object while in the Project Explorer, for example.

Export Bitmap: Click on the Export Bitmap button to save a bitmap to disk as a TGA file. Animations are stored as consecutively numbered TGAs (bitmap0000.tga, bitmap0001.tga, bitmap0002.tga ...)

Paste Bitmap from Clipboard: Click here to import a graphic that has been saved in the Windows clipboard by another program.

Text

SCOPE uses two kinds of text - Standard Text and Graphical Text. These differ as follows:

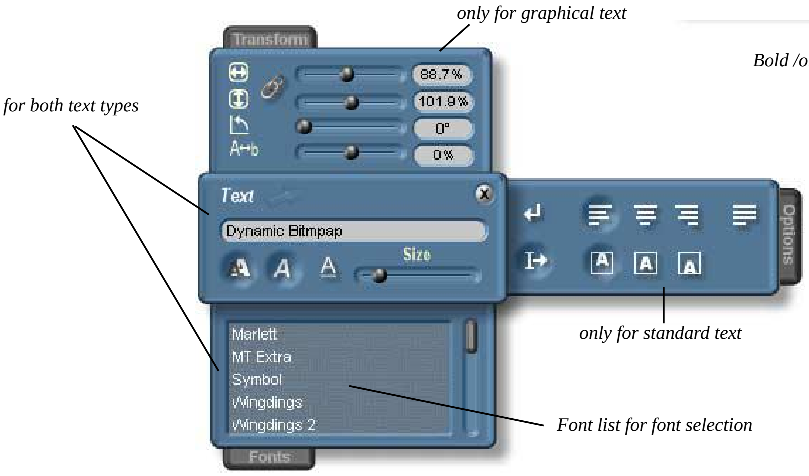
Standard Text: This refers to normal, Windows text. This is used for text that can change at run time, such as in a control’s text field. All controls and modules that use standard text default to the Arial font, since at the time of writing it is installed on all PCs running Windows. If the font is not available, another will be substituted. Standard text is not anti-aliased, so it is not really appropriate for static text used for labels or design elements. For this there is graphical text.

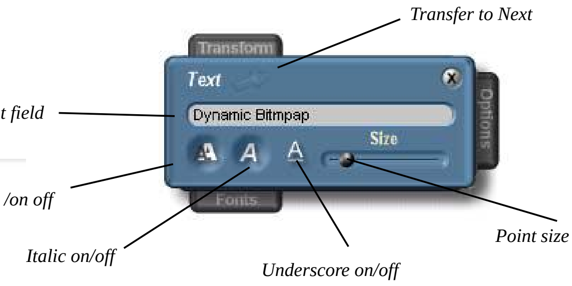
Graphical Text: Always use this text when its appearance is important, such as in controller names on synth surfaces or other labels that will not change at run time. Graphical text uses the same fonts as standard text, but it is anti-aliased (the edges are smoothed) and rendered immediately into a bitmap. The font used to render the graphical text does not have to be installed on the computer on which you want to use the object. However, the link between the font and the bitmap is preserved so the text can be modified at any time by adjusting parameters such as rotation and scaling. After each change, a new bitmap is produced.

If you produced some graphical text using, say, the Avalon font, and then removed the font from your system, the graphical text would still be displayed in the Avalon font. If you then try to modify the graphical text in SCOPE, you won’t be able to, since the Avalon font is no longer available. In this case, the next available font is substituted.

Editing Text:

Editing standard and graphical text differs only slightly. Although some options are available for both kinds of text, some apply only to standard text and others only to graphical. You can’t, for example, rotate standard text, or input multiple lines of graphical text.



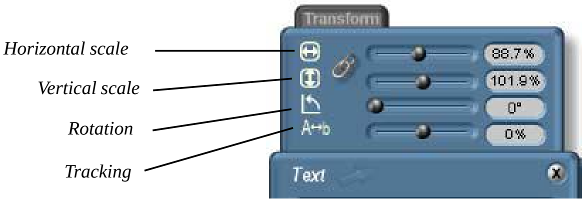


Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the text will be transferred to the next selected object. Use the multi-arrow to transfer it to several successive objects.

The **Bold on/off** function will have no effect on some fonts. The **Underscore** function works only with standard text.

Transform (graphical text only)

Modifications to graphical text beyond font selection and size are made using the contents of the Transform drawer.



Horizontal Scale: This adjusts the aspect ratio of the text on the horizontal axis (stretching, compressing).

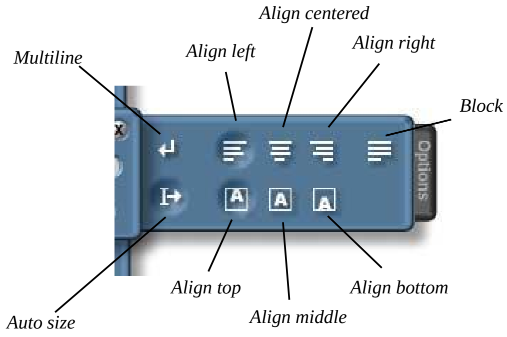
Vertical Scale: This adjusts the aspect ratio as above, but on the vertical axis.

Rotation: Text can be rotated between 0° and 360°..

Tracking: Increases or decreases the amount of space between letters, adjusting the length of the text without changing the aspect ratio of the letters.

Options (for standard text only)

Further options for standard text are found in the Options drawer, including multiline text.



Multiline: Enable this option to enter more than one line of text. If the text block is smaller than the text, the text is clipped (only partly visible).

Auto Size: The text will be automatically sized to fit a block. A line break can be forced by including the tilde (~) character.

Align Left: The text is aligned to the left side of the text block.

Align Centered: The text is centered in the block.

Align Right: The text is aligned to the right side of the block.

Block: The text is justified (each line adjusted to fill the block).

Align Top: The text is aligned to the upper edge of the block.

Align Middle: The text appears in the middle of the block.

Align Bottom: The text is aligned to the bottom edge of the block.

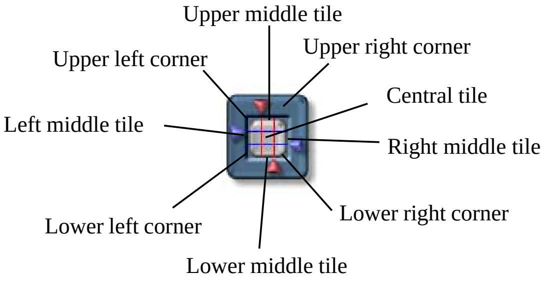
Dynamic Bitmaps

Dynamic bitmaps are a special kind of bitmap that can be resized at run-time. This is important when designing a surface, and you don't know yet what size it will be. The supplied default surfaces were all created as dynamic bitmaps. SCOPE uses the following technique to convert a normal bitmap into a dynamic bitmap:

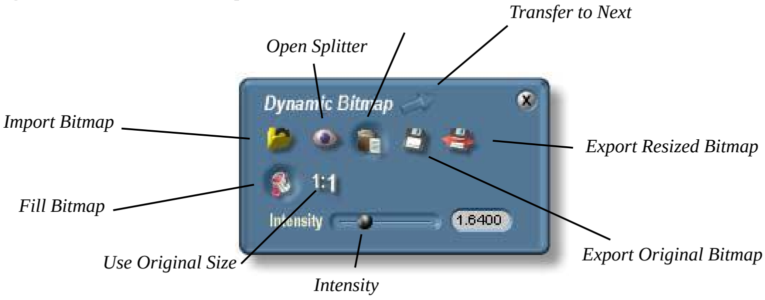
The bitmap is divided into nine tiles with two horizontal and two vertical 'splitter lines'. If a dynamic bitmap is horizontally adjusted in Edit mode beyond its original size, the upper, and lower middle tiles, and the central tile, are drawn repeatedly next to each other as required. The corners are shifted accordingly.

Vertical sizing is achieved in a similar way - by drawing the left and right middle tiles, and the central tile, repeatedly. Both dimensions can be adjusted at the same time. Also, if the object has a shadow, it is resized, too.

Dynamic bitmaps have another advantage over conventional bitmaps, as well. They can be filled with a color, a gradient or a texture if the **Fill bitmap** property is enabled.



The Dynamic Bitmap Editor



Import Bitmap: Click on this button to load a bitmap as a dynamic bitmap.

Open Splitter: Opens the Splitter Window where you can determine the regions to be used for resizing. The center tiles are always used as the corners are preserved this way.

Fill Bitmap: A dynamic bitmap can be used with its own color, or, if the Fill bitmap option is enabled, it can be filled with another color, a gradient, or a texture. The original bitmap then becomes a kind of container that describes only the form, and not the color.

Use Original Size: When this option is enabled, the dynamic bitmap can no longer be resized. Use this when you design something like a button that changes color depending on its status - you want to fill it with another color when it’s pressed - however you always want the object drawn at its original size.

Export Original Bitmap: Click on this option to store the original bitmap, from which the dynamic bitmap was created, back to disk as a normal bitmap.

Export Resized Bitmap: Click here to save the resized bitmap back to hard disk as a normal bitmap.

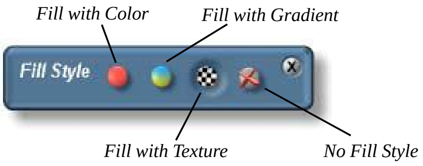
Intensity: If **Fill Bitmap** is enabled for a dynamic bitmap then this option will multiply the color intensity.

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the dynamic bitmap will be transferred to the next selected object. Use the multi-arrow to transfer it to several successive objects.

Paste Bitmap from Clipboard: Click here to use a bitmap you have copied to the clipboard from another Windows program as a dynamic bitmap. By default it is pasted with Fill Bitmap disabled, and at 1:1.

The Fill Style Dialog

IIn SCOPE, graphical text and dynamic bitmaps can each be filled using one of three different fill styles. By default, text is already filled with color, and can be adjusted with the Color Selector immediately. With dynamic bitmaps you must first enable its Fill Bitmap option before you can use the fill feature.



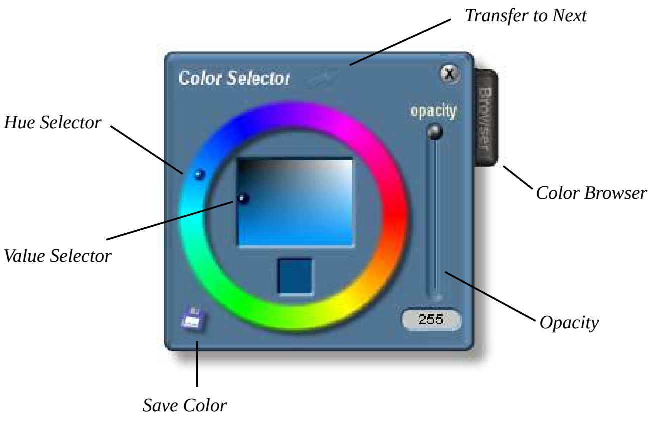
Example - you want to fill some text with a gradient. First select the desired text, then just click on the **Fill with Gradient** button in the Fill Style dialog. The text will be filled with the default gradient, which you can change in the Gradient Designer. Or, if you click on the **Fill with Texture** button, the text will be filled with the default texture. You can assign a new texture later in the Texture Import dialog.

The **No Fill Style** button is there as an indicator, not a pressable button. It will be shown in the pressed position if the selected object is a bitmap or and animation, or if a dynamic bitmap’s Fill Bitmap option is disabled. These objects cannot be filled.

Important: The Color Selector, Gradient Designer and Texture Import dialog will only work if the selected object is filled using the respective appropriate fill style.

The Color Selector

Use the Color Selector to modify the color of an object that is enabled for Color Fill.



Hue Selector: Move the ball to select the basic color.

Value Selector: Move this ball to select the intensity of the color - from white to black.

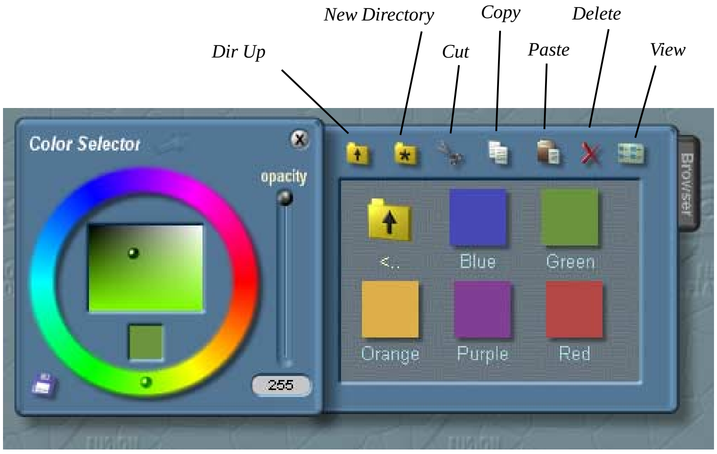
Opacity: Set the transparency/opacity with this control. The range is from 0 (completely transparent) to 255 (opaque).

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the color fill will be transferred to the next selected object. Use the multi-arrow to transfer it to several successive objects.

Save Color: Click on this button to save the new color in the Colorbrowser directory. It will be given a default name of new Color0001, new Color0002, new Color0003 etc. To rename it, select it in the browser and press F2.

Color Browser

The Color Browser is in essence another File Browser, but it deals only with colors. To use color stored on a hard drive, drag and drop it to a free space on the Color dialog surface.



Dir Up: Moves to the next higher level in the directory hierarchy.

New Direcrory: Creates a new directory.

Cut: Cuts the selected color (removes it to a copy buffer).

Copy: Copies a color or directory to a copy buffer.

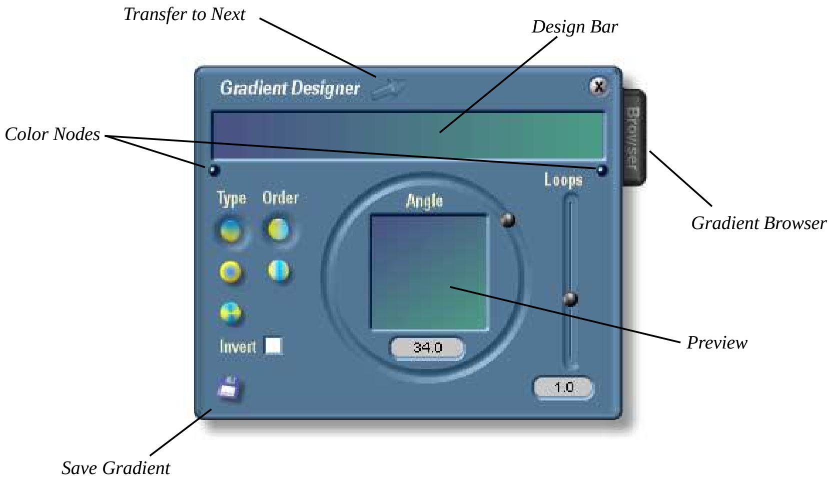
Paste: Pastes (inserts) a color or directory which has been put in the copy buffer by Cut or Copy.

Delete: Removes the color or directory.

View: Switches between large and small icons.

The Gradient Designer

SCOPE’s Gradient Designer is a powerful tool for creating gradients and processing color. A great number of effects can be obtained using an almost limitless number of colors.



General Remarks: A progression from one color to another (a gradient) always has at least two color ‘nodes’ - the initial color and the target color. Between these two nodes you can add other nodes. A node defines both color and transparency at a particular point in the progression. Therefore an object’s gradient can progressively cover or make visible another object lying beneath it.

Modifying a Color Node: When you select a node, the Color dialog appears in which its color is displayed, and which can be changed immediately. The change will be reflected in the Gradient Designer.

Adding a Color Node: Double click on a location in the Design Bar to add a color node at that point. The new node will contain the color and transparency information for the point you chose, so no modification has yet taken place.

Moving a Color Node: You can drag Color Nodes using the left mouse button.

Deleting Color Nodes: Select the node, and then strike the <Del> key.

Parameters

Type: The Gradient Designer can create three types of color progressions - Linear, Circular and Radial. Select the desired type with its respective ‘Type’ button.

Order: The Gradient Designer provides two ‘Order’ options - AB, or ABA. The AB option produces a gradient starting with the initial color and ending with the target color. The ABA option produces a gradient starting with the initial color, progressing to the target color, and then reversing back to the initial color.

Invert: The gradient fill is inverted when it is poured into the object. For example, if the gradient is black to white, the object will be filled white to black.

Angle: Set the rotation angle here. This affects only linear and radial gradients.

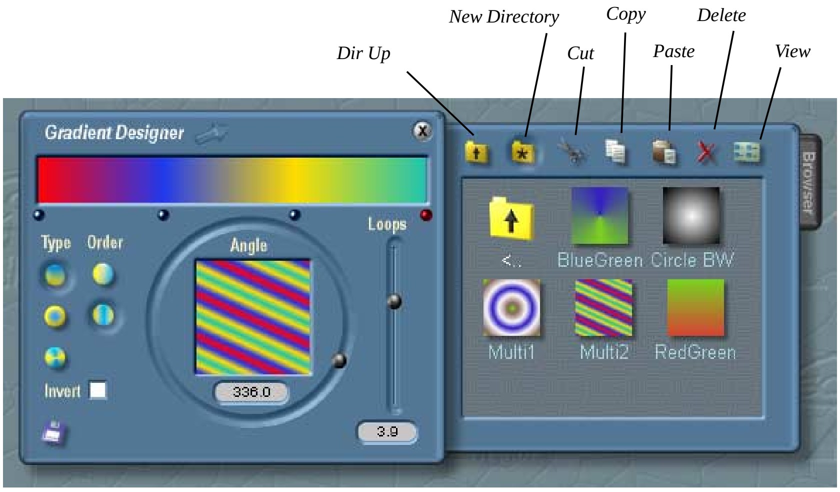
Loops: If this number is greater than 0, then the gradient will repeat itself that number of times, creating ‘gradient strips’.

Save Gradient: Click on this button to save a gradient in the Gradient Browser’s directory. A gradient is given a default name of new Gradient0000, new Gradient0001, new Gradient0002, etc. To change the name, select it in the Browser and press F2.

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the gradient will be transferred to the next selected object. Use the multi-arrow to transfer it to several successive objects.

Gradient Browser:

The Gradient Browser is really just another File Browser - but it handles gradient files only. Functionally it is almost identical to the File Browser. To edit a gradient stored on hard disk, drag and drop it to a free spot on the Gradient Designer surface.



Dir Up: Move to the next level up in the directory hierarchy.

New Direcrory: Creates a new directory.

Cut: Cuts (removes) the current gradient or directory and creates a temporary copy of it.

Copy: Creates a copy of the gradient or directory in a copy buffer or temporary file.

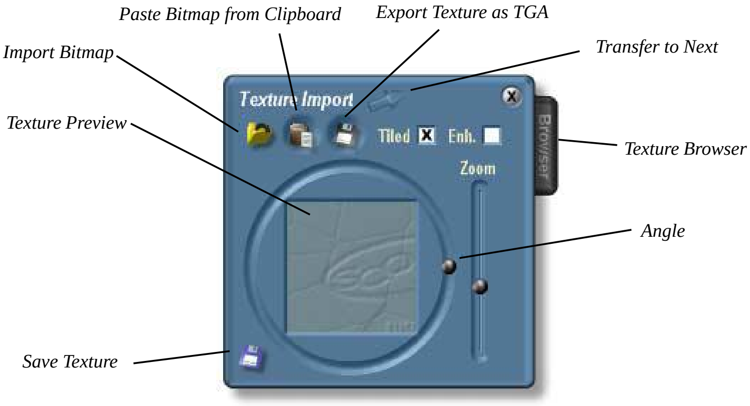
Paste: Inserts a Copied or Cut gradient into the selected object.

Delete: Remove the gradient or directory.

View: Switches between large and small icons.

The Texture Import Dialog

Graphical text and dynamic bitmaps can be filled with textures. Using tiling, small texture bitmaps can fill large objects.



Angle: If **Enh** is checked, the texture will be freely rotatable.

Zoom: If **Enh** is checked, you can adjust the zoom of the texture.

Shift: The texture can be shifted by dragging it with the left mouse button in the preview window.

Save Texture: Click here to save this texture in the texture browser directory. The texture will be given a default name (i.e. new Texture0001, new Texture 0002, new Texture003, etc.) which you can rename in the Texture Browser (F2).

Tile: Enable this option to ‘tile’ the texture. You will need to do this if the texture bitmap is smaller than the target GO.

Enh.: This option must be checked if you want to rotate or zoom the texture.

Paste Bitmap from Clipboard: Click here to use a bitmap you have copied to the clipboard from another Windows program as a texture.

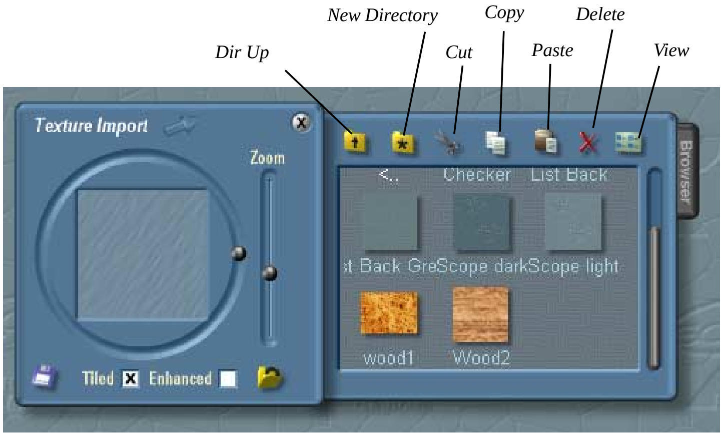
Import Bitmap: You can create a new texture from an existing 24-bit BMP or 32-bit TGA file. Click on this button and select the desired file in the File Selector dialog.

Export Texture as TGA: If you want to work on a texture in another program you can export it as a 32-bit TGA file. Click here and provide the directory and filename.

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the texture will be transferred to the next selected object. Use the multi-arrow to transfer it to several successive objects.

Texture Browser:

The Texture Browser is yet another File Browser. It deals only with textures. To edit a texture stored on hard disk, drag and drop it to the Texture dialog surface (but not the Preview window).



Dir Up: Moves to the next higher level in the directory hierarchy.

New Direcrory: Creates a new directory.

Cut: Cuts the selected texture (removes it to a copy buffer).

Copy: Copies a texture or directory to a copy buffer.

Paste: Pastes (inserts) a texture or directory which has been put in the copy buffer by Cut or Copy.

Delete: Removes the texture or directory.

View: Switches between large and small icons.

The Blitter Options Dialog

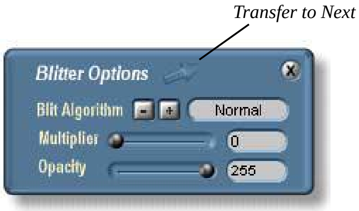
Normally, whenever an object is drawn on a display it is drawn over existing objects. The new object replaces those parts of the existing display it was drawn over. SCOPE's use of 32-bit TGA allows transparency, eliminating the requirement that objects be rectangular. SCOPE objects can take any shape. Only those parts of the shape permitted to be drawn by the 'alpha-channel' will be drawn over other objects. Further to this, the total or overall opacity of an object can also be adjusted with the Blitter Options Opacity control, independent of the alpha channel information. The Opacity control also operates independent of any transparency information contained in a gradient fill.

In addition to the transparency characteristics, SCOPE offers other options for how an object is drawn over its background. The following blit modes are available:

Normal: The standard mode. New pixels replace underlying pixels. Transparency exists when the new pixel is 'mixed' with the pixel it is replacing.

Multiply: The new pixel is multiplied with the underyling pixel. The underlying pixel is darkened depending on the brightness of the new pixel. The Intensity control has no effect if Multiply is selected.

Hardlight: New pixels lighten or darken exisiting pixels depending on their grey content. If grey content is greater than 50%, pixels are lightened. If less than 50%, they are darkened. To see this effect, the Intensity control must be greater than 0. Use this control when you want to make sure an object is always drawn with the 'correct' color - no matter what background it is drawn upon. Some of the supplied controls were created using Hardlight mode. A fader, for example, has a background to give the impression that it is indented into the surface. The Hardlight algorithm guarantess that this will work over any background.



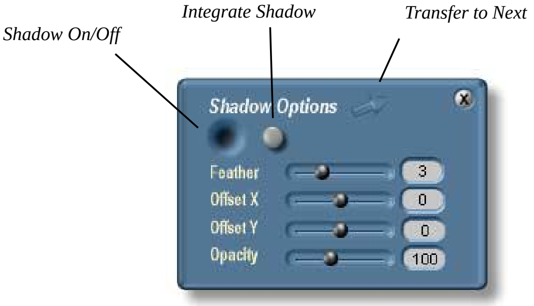
Blit Algorithm - + Normal
Multiplier 0
Opacity 255

The frames for text fields were also use Hardlight mode. Wherever you use a text field, the frame will adapt to the background. This is also useful for other indented or extruded object representations. You can change or modify the background fill at any time, and the Hardlight mode will automatically 'correct' the object to ensure it looks correctly indented or extruded.

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the blitter settings will be transferred to the next selected object. Use the multi-arrow to transfer them to several successive objects.

Shadow Options

In SCOPE you can assign a shadow to any graphical object. Shadows can help differentiate objects from their backgrounds - dialogs can appear to be ‘raised’, for example. The appearance of many objects can be enhanced with shadowing. To create a shadow, SCOPE uses the object as a mask to create another bitmap, which is filled with black, and made partly transparent.



Feather 3
Offset X 0
Offset Y 0
Opacity 100

Shadow On/Off: Click on this button to assign a shadow to this object. At first the shadow will be barely visible, since it lies directly under the object. Adjust the Feather value to see the shadow better.

Integrate Shadow: If you like the shadow effect you have created, and are sure you won’t want to change it, you can merge it with the original bitmap. When you do this, a new bitmap is created.

When a shadow is applied to a dynamic bitmap, the shadow is merged with the original bitmap, and then the Split Lines are reset so that the bitmap remains a dynamic bitmap. Note that the positions of the Split Lines reamains the same, although the bitmap itself has changed. This means you may have to reposition the Split Lines in order for the object to redraw itself correctly. There may be times when you cannot find useful new positions for the Split Lines - for example, when the bitmap is very small, or the shadow offset is very large. In this case you must enlarge the original bitmap, and then integrate it with a shadow.

You can also integrate shadows with graphical text objects. Integrating the shadow also reduces memory usage. You can still modify the text at any time, but the new text will not have a shadow, since the original text had no shadow either.

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the shadow settings will be transferred to the next selected object. Use the multi-arrow to transfer them to several successive objects.

Feather: This adjusts the softness or hardness of the shadow edge.

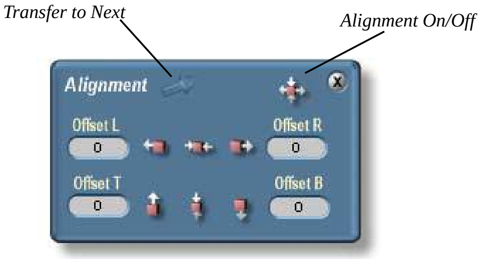
Offset X: Sets the horizontal offset of the shadow relative to the original object.

Offset Y: Sets the vertical offset of the shadow relative to the original object.

Opacity: Adjusts the intensity of the shadow. Higher values produce darker shadows.

The Alignment Dialog

Use the Alignment Dialog to set the positions or sizes of objects relative to each other. For example, a text background bitmap and the text object should lengthen if the text field’s frame is lengthened. Or you want t button to remain always in position relative to the edge of a dialog. The Alignment Dialog combines a variety of alignment options into an easy-to--use tool.



Offset L 0
Offset R 0
Offset T 0
Offset B 0

The alignment adjustments you make to an object are always made relative to its parent. You can work throught a module’s GO hierarchy to align several objects to their parents. For example, the text background child of a text frame can be aligned to the frame, and the frame aligned to its parent, et cetera, until finally everything is aligned to the default surface (the Parent of the text field).

Alignment On/Off: To enable Alignment for a GO, click on Alignment on/off button. Although enabled, no alignment will be in effect until one of the six options is selected. The offset of the center of a text field from its parent will be indicated, however.

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the alignment settings will be transferred to the next selected object. Use the multi-arrow to transfer them to several successive objects.

Offset L: Distance of the middle point of the object to the left edge of the parent.

Offset R: Distance of the middle point of the object to the right edge of the parent.

Offset T: Distance of the middle point of the object to the top edge of the parent.

Offset B: Distance of the middle point of the object to the bottom edge of the parent.

All values can be input numerically if you require a value that cannot be set with the mouse
.
Align to Left: Enable this option to align the object to the left edge of the parent at the distance defined in the Offset L field.

Align to Right: Enable this option to align the object to the left edge of the parent at the distance defined in the Offset R field.

Align to Left and to Right: Aligning to both the right and the left edges of an object would appear to be a paradox. But in SCOPE you have dynamic objects - dynamic bitmaps and graphical text. With these objects it is possible to align not the center point, but their edges of the respective edges of the parent. The object will then contract or expand when its parent is resized.

Align to Middle: Aligns the object horizontally in the center of the parent. The L/R offsets have no effect with this option.

Align to Top: Enable this option to align the object to the top edge of the parent at the distance defined in Offset T.

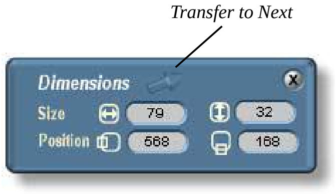
Align to Bottom: Enable this option to align the object to the top edge of the parent at the distance defined in Offset B.

Align to Top and to Bottom: Similar to the 'Align to Left and Right' above. However, again it is not the center, but the edges of a dynamic object that are aligned to the top and bottom edges of the parent.

Align to Center: Aligns the object vertically in the center of the parent. The T/B offsets have no effect with this option.

The Dimensions Dialog

Transfer to Next



Dimensions

Size 79 32

Position 568 168

This dialog displays different values depending on the operation mode. In Move mode, it shows the position and size of a module. In Edit mode it shows the position and size of an individual GO.

In the case of an object which is flipped out, the Position field refers to the upper left-hand pixel's position with respect to the display. With rounded objects, this pixel will be invisible.

When the object is in the Project Window (not flipped out) then the Position refers to the upper left pixel's position with respect to the Project Window.

You can input numeric values in the fields. Note that although you can change the position settings for any object, you can only change the size of dynamic bitmaps or graphical text.

Transfer to Next: Click once to turn the arrow red or twice to change it to a yellow multi-arrow. When the arrow is red the Dimensions settings will be transferred to the next selected object. Use the multi-arrow to transfer them to several successive objects.

Transfer Options

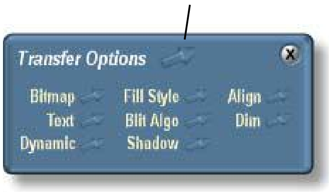
As you have seen several times, each Editor has a special three-state switch called ‘Transfer’ that lets you transfer all the settings for an object to an other object of the same type. It’s important to note that you can only transfer settings to another object that understands them. If you try to transfer text settings to a non-text object, the internal message will be ignored.

The transfer function is very convenient for doing things such as transferring the animation settings of a potentiometer to a number of other potentiometers already assigned and connected on a surface. Or if you are not happy with a color you used for a dialog, just change one dialog, and then transfer the settings to all the others. Simply click twice on the transfer arrow to enable it as a yellow multi-arrow and click on the other potentiometers or dialogs.

It might occur that you want to transfer one or more *selected* characteristics of one GO to another. You can do this using the Transfer Options dialog. The Transfer to Next buttons in all the Editors are subject to the settings in this dialog. By clicking on the Transfer All button, all the individual transfer buttons will appear in an enabled state to indicate which options are applicable to the current object. This makes it easy to transfer color, size or shadow characteristics from one dynamic bitmap to another.

The Transfer All button is a three-state button, just like the Transfer Next buttons in the Editors. Likewise all the individual transfer switches are three-state. If you want to transfer, for example, all the settings except Size to another go, simply switch off the Size option.

Transfer All



Control Design Examples

Control Design by Example

To design your own control, always start with an existing control. Load a control that contains the GOs that most closely match your requirements. Now all you need to do is edit the existing object. Of course, you can add additional GOs to the control if you wish.

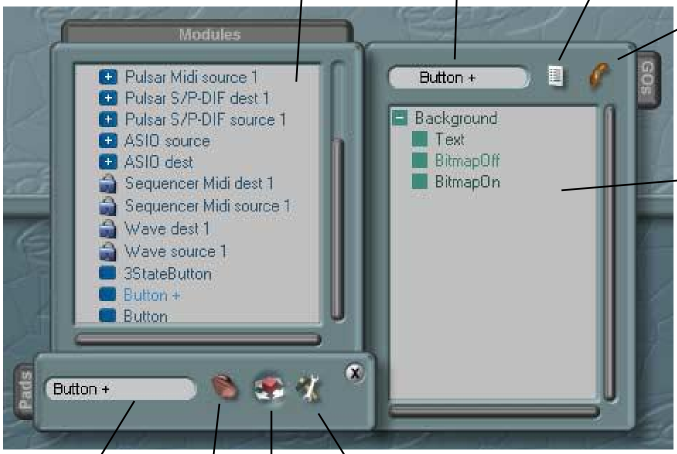
Certain GOs must not be renamed or deleted as they are required by the associated script. These are indicated in the Project Explorer’s Go Tree with a ‘lock’ icon.

We will use some common controls in the following examples.

Working with the Go Tree

Open the Project Window, and then its GO drawer as shown:

Selected Module *Properties* *Module Tree* *Add GOs*



GO Tree

Selected Module *Use Mode* *Move Mode* *Edit Mode*

If you select a control in the Module Tree, its associated GOs are indicated in the GO Tree on the right.

Selecting GOs
GOs can only be selected in Edit mode. In Move mode, all internal activity refers to the entire module. The quickest way to edit mode is by **<Ctrl><E>**. You can drag a wireframe around GOs, or use the mouse to select them in the Project window, or you can use the mouse in the Module Tree to select GOs. Once selected they will be highlighted in the Project Window. Sometimes it is easiest to selcet a GO in the GO Tree - especially when noe GO nearly covers another completely.

Moving Individual Control GOs
You can position GOs three ways:
1.) Move it directly with the mouse
2.) Move it with the arrow keys
3.) Type the coordinates and dimensions into a dialog

Copying GOs
Copy a GO by first selecting it in the Project window or the GO Tree, and then pressing **<Ctrl><C>**. The GO can now be inserted into the GO Tree at the desired level. If you now select a GO in the GO Tree, the copied GO will become its child. If you select a free area in the GO drawer, it will be place at the gighest level.

Inserting (Adding) GOs
In the **Add GO** drawer you will find a list of all GOs you have available to you. Use drag and drop to pull the desired object into the GO Tree. If you pull it to a free area it will be inserted at the top level. Dragging it over another GO will make it a child of that GO. Pressing the **<Shift>** key before releasing the mouse button inserts it as a sibling to the object (at the same level).

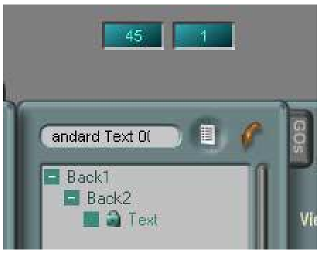
Changing the GO Hierarchy
As above, you can drag a GO already in the GO Tree to a new level or location within it. Dropping it on an existing GO makes it a child of that GO. Pressing the **<Shift>** key before releasing the mouse button inserts it as a sibling to the object (at the same level).

Deleting GOs
Select a GO and press the **<Del>.** If the GO has children, they too will be deleted.

Other GO Properties
Open the Properties drawer to view other GO properties. When you select GO properties such as whether or not it is selectable are displayed. If a text field is selectable, then you can edit it.

Textfields

In the **Controls\Text Controls** directory you will find a control called **Standard Text 001**. Drag this into the Project window, open the Project Explorer and its GO drawer.

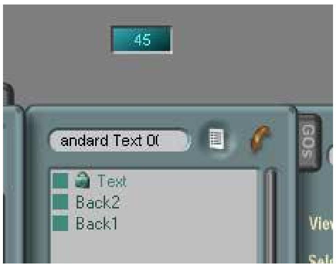


We see that this control has three elements:
The parent/child hierarchy determinens the *z-order*, or the drawing order, of the objects. Since a child is always drawn over its parent, in the above example, the **Text** is drawn over a colored background, **Back2**, which is in turn drawn over the integration background, **Back**.

Back1 is a dynamic bitmap designed to create the impression that the text field is indented into the surface. In the Blit algorithm **Hardlight** was used. Back1 is not simply drawn - it is modified by the background somewhat. Back2 is likewise a dynamic bitmap, but filled with a gradient to simulate a ‘display’.

The text itself is the last child in the tree, so it is drawn last. It is a **standard text** GO as it must change at run time.

If you want, you could arrange these at the same level to produce the same results. In this case the hierarchy would look like this:



Additional Note

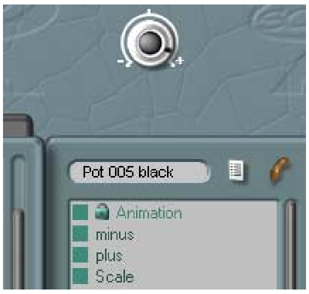
The **Text** and **Back1** were enabled for **alignment**, making it easy to change sizes. It is sufficient to change only the size of Back1 to resize the text as well. Click on Back1 in the Go Tree and adjust its size (with the arrow keys, for example).

In this case the z-order corresponds to their relative positions in the tree. All three objects must have the **Fixed Z-Order** set in Properties to prevent the order from being changed by selection.

Ultimately it is up to you how to use the tree structure. You may decide to use both representations - i.e. a simple list under a parent GO. The Fader example uses such a representation. One reason to use the hierarchical rather than the flat representation is the **Clip Children** property, which, when set, can greatly speed up drawing. This can only be used if the dimensions of the child object fall entirely within those of the parent, however.

A Potentiometer

In the **Controls\Potentiometer\Pot 005 modular (complete)** you will find the **Pot 005 black** module. Drag it into the Project window, open the Project Explorer, and the GO drawer.
This potentiometer has 4 elements, as shown below:.



All objects have their **Fixed Z-Order** property set. Note that with this control a ‘flat’ tree structure was used. A multi-level hierarchy makes little sense as the **Clip Children** option cannot be used.

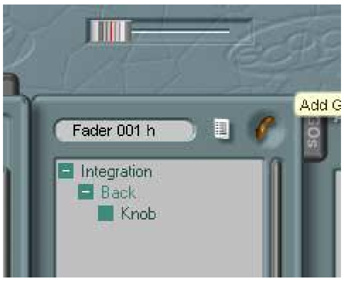
You can quickly adjust the appearance of this control by exchanging individual elements, or adding or removing them. You cannot rename or remove the animation, however.

The element of interest here is the GO animaion that contains the sequence of bitmaps required to make the knob appear to rotate. As you can see, the animation has a shadow. This was rendered in to the animation using **Integrate Shadow** to increase the frame rate.

The other items consist of graphical text for the ‘+’ and ‘-’ symbols, and a bitmap for the scale (min-max) indicator.

A Move Fader

In the directory **Controls\Fader\Move Fader** directory you will find a module called **Fader 001 h**. Pull this into the Project window, open the GO drawer in the Project Explorer and select the module.



Three elements make up this GO. Two of them are absolutely required for it to function - the **Knob** and **Back** - the rail on which it slides. Values are produced from the relative position of the knob on the rail.

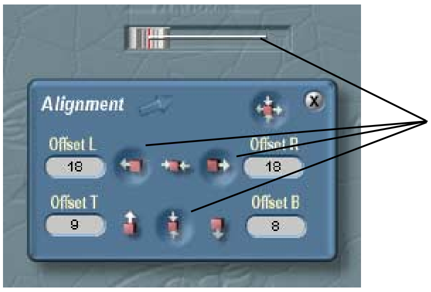
The **Back** (rail) image was implemented as a dynamic bitmap to make it easy to adjust the throw (length of travel) of the fader. Also, the **Hardlight** algorithm was enabled so the bitmab will adapt itself to different backgrounds. The aspect ratio of the fader could conceivably be adjusted to produce a vertical orientation, but the direction of movement of the fader will always be horizontal.

The **Integration** GO is purely a design element. This was designed as a dynamic bitmap to enable resizing. The z-order of all elements is determined by the parent-child relationships - the knob is drawn first, then the rail, and finally the Integration element. Adding text, and/or a scale bitmap may be desirable in this example.

Since none of the children extend outside their parents, the **Clip Children** property was set to speed drawing.

Additionally in this example, the **Alignment** options was used for easy resizing.

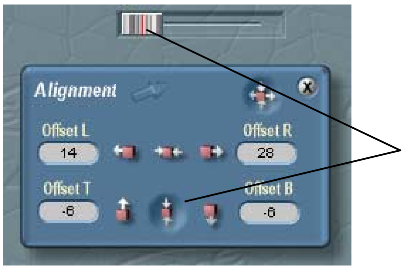
The **Back** bitmap is set to scale itself horizontally if the size of the **Integration** bitmap changes. However, **Back** does not scale itself if the vertical size is changed - it remains centered.



Offset L 18
Offset R 18
Offset T 9
Offset B 8

Horizontal Resize Alignment
Vertical Center Alignment

The same applies to the **Knob** - it will always remain centered on the rail.

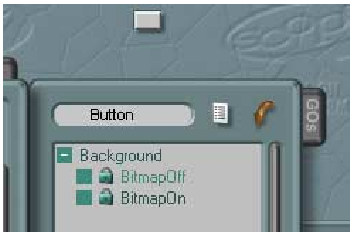


Offset L 14
Offset R 28
Offset T -6
Offset B -6

Vertical Center Alignemt

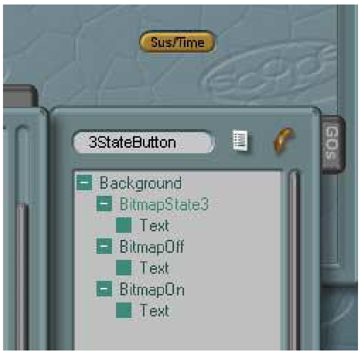
A Button

In the directory **Controls\Buttons** directory you will find a module called **2State Button 003**. Pull this into the Project window, open the GO drawer in the Project Explorer and select the module.



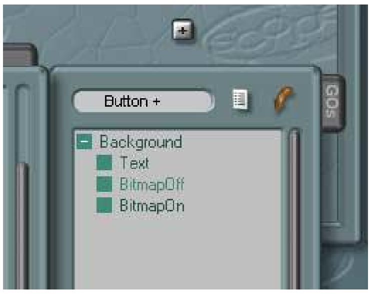
There are three GOs in this example. The **Background** GO is not strictly necessary, but the **BitmapOff** and **BitmapOn** GOs are required to represent the 2 possible states of the button - On and Off. If this were a 3-state button, you would require an additional bitmap for the third state.

The GO Tree shows all objects used in the control, but not all objects will be visible at the same time. When a button is off, the corresponding bitmap has it's **Visible** property disabled - it won't be drawn. When the state changes to On, the corresponding bitmap's **Visible** property is set.



This example shows a 3-state button. The buttons were created as dynamic bitmaps with alignment adjusted for resizing. Here, each button has additional text to indicate each of the three states. Further bitmaps could be added. Multi-state buttons compare to multi-state icons in their functionality.

In the example below, that uses child GOs, the text (a '+' sign) is to be displayed at all times. The text was inserted only once, but at the same level as the buttons. By enabling **Fixed Z-Order** it is guaranteed that the text will always be drawn over the buttons.



To further distinguish the 'on' and 'off' states of the button, shadows are applied - less shadow for 'on' (depressed) than 'off'. Finally, different fills were used so that the button appears to 'light up' when it is switched to the 'on' state.

Clip Children could not be used in this example, as the shadows extend beyond the dimensions of the **Background** parent.

To use such buttons at different sizes, create dynamic bitmaps to represent them, and use alignment so that merely adjusting the size of the background will reorient all its elements. Also, text or other elements (for example, bitmaps used when creating icons) should be made children, and alignment applied used so they remain centered.

Appendix

Return To Main Table Of Contents

Glossary

It's a fact of life that when somebody uses the word 'mouse' he or she may be referring to a small rodent, a computer peripheral, or someone with a less than manly deportment. For this reason we will identify and clarify (we hope) the terminology used throughout the manual ...

ADAT*
(Alesis Digital Audio Tape) Digital tape recording system developed by Alesis that puts 8-tracks of 16-bit, 44.1kHz or 48kHz digital audio on S-VHS tape.

ADAT ODI
ADAT Optical Digital Interface

AES/EBU interface
The serial transmission format standardized for professional digital audio signals (AES3-1992 AES Recommended Practice for Digital Audio Engineering - Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data). The interface specification calls for balanced line drivers to transmit two channels of digital audio over a single twisted-pair cable using 3-pin (XLR) connectors.

Algorithm
A structured set of instructions and operations tailored to accomplish a signal processing task. For example, a fast Fourier transform (FFT), or a finite impulse response (FIR) filter are common DSP algorithms.

Alignment
GOs (graphical) can be aligned to their parents (i.e. left, right...).

Alpha-Channel
In graphics work the alpha channel represents the transparency information of an image.

Asynchronous
In Scope, a tqpe of signal that is generated only when needed - for example, when a controller changes its value.

Atom
Basic DSP element

Attenuator
A module used to reduce the level of an audio signal.

Audio Connection Module
Leads a signal to to the outpus of a Scope Device.

*ADAT is a trademark of the Alesis corporation.

Bagpipes
A so-called musical instrument, also known as the dudelsack, sometimes used in times of war to frighten the enemy. Any attempt to virtualize this
instrument in Scope could result in severe damage to your computer, your ears, or the Scope hardware itself.

bandwidth
Abbr. BW The numerical difference between the upper and lower -3 dB points of a band of audio frequencies. Used to figure the Q, or quality factor, for a filter.

Bi-polar
One of the properties of a signal meaning its value is represented with both positive and negative numbers.

Bit
Abbreviation for binary unit or binary digit. The smallest amount of digital information. A bit can store or represent only two states, 0 and 1. (The orginal term binary unit was coined by John Tukey of Bell Laboratories to represent the basic unit of information as defined by Shannon as a message representing one of two states.)

Blit
In general, the copying of one area of memory to another, optimized for speed. The term is usually used with reference to display memory, as it is in Scope. Often some manipulation of the bits being transferred is also carried out during the blit. The etymology of the term is uncertain, but the commonly held belief that it derives from 'Bacon, Lettuce and Interactive Tomato' is known to be false.

Blitter options
Various options to control how GOs are drawn.

Byte
A group of eight bits.

Child
In an object hierarchy (i.e. Module tree, Graphical Object tree, etc.) a Child object is attached to, or, more accurately, pointed to by a Parent object. A Parent may have several Children, as in a multi-layered graphical surface design. The children of an object appear indented under it in an object tree.

Circuit
The modules and their interconnections in a project window.

Color selector
The dialog in which you select and edit colors for a GO.

Connection Module
Implements a *pad.*

Control
Various components like, faders, potentiometers, editable text fields etc.

Controller
MIDI controller - sends and can receive MIDI data.

Converter
A module that performs a conversion of one signal type into another, permitting the connection of otherwise incompatible pads.

DAC
(or D/A, digital-to-analog converter) The electronic component which converts digital words into analog signals that can then be amplified and used to drive loudspeakers, etc. The DAC is the last link in the digital chain of signal processing.

Decibel
Abbr. dB Equal to one-tenth of a bel. [After Alexander Graham Bell.]
1. A measuring system first used in telephony (Martin, W.H., "DeciBel -- the new name for the transmission unit. Bell System Tech. J. January, 1929), where signal loss is a logarithmic function of the cable length.
2. The preferred method and term for representing the ratio of different audio levels. It is a mathematical shorthand that uses logarithms (a shortcut using the powers of 10 to represent the actual number) to reduce the size of the number. For example, instead of saying the dynamic range is 32,000 to 1, we say it is 90 dB.

Device (*.dev)
A complete working design including a graphical surface, controls, circuits etc. Examples of devices are the various synths, mixers, and effects modules.

DSP
Digital Signal Processing (arithmetic) or Digital Signal Processor (hardware circuit optimized for digital signal processing)

Dynamic bitmap
A special, resizeable, bitmap.

Editing
Making any changes to modules, controls, surfaces, GOs etc.

Empty synth
A device containing the basic elements of a synth design. Used to begin a new synth project.

Folding
The act of taking a group of modules and creating a new module of which they become children.

Field
An area in which to insert standard text. Can be editable, scrollable, etc.

Flip- In
Puts a device 'inside' the Project Window where it can be edited.

OUT
Removes a device from the Project Window placing it in the general Scope environment.

Floating point
An encoding technique consisting of two parts: a mantissa representing a fractional value with magnitude less than one, and an exponent providing the position of the decimal point. Floating point arithmetic allows the representation of very large or very small numbers with fewer bits.

Gate
1. One of the signals available in Scope. A Gate signal is a control signal used to start an envelope generator, for example.
2. The noise gate effect module.

GO
Graphical **O**bject. All visible elements in Scope are comprised of GOs.

Gradient
The graduated fading of one color into another, used in surface
design.

Latency
The time delay between the initiation of an action and its result.

Level
Each module can contain one or more modules; one can therefore refer to the different "levels of a module, where these inner modules reside.

Mode, Edit-
In Edit mode a module and its elements or GOs can be arranged, sized, and otherwise adjusted.

Move-
Move mode allows modules to be moved around in the project window.

Use-
Use mode permitsa module's functional elements, such as switches and faders, to be operated.

Module (*.mdl)
Modules are the building-blocks of Scope circuits. There are normal modules (oscillators, filters, envelope generators etc.), special modules (Esyn-adder, DSP converter, PC converter etc.) and logical modules (adder, multiplier, divider etc.). Of course the various 'devices' incorporated into the modular synth could also be referred to as modules, but in Scope design the term refers to the former definition.

Modular
The modular synthesizer that comes with Scope. 'Modular' refers to the nature of this instrument as a collection of pre-designed circuits that must be patched together to create a functional device.

Single pole highpass filter
A highpass filter with a slope of 6dB per octave.

Object
Another way to refer to modules, GOs etc.

Object-oriented or object-based programming (Abbreviated **OOP)**
A software technique in which a system program is expressed completely in terms of predefined things (objects), consisting of a set of variables (properties) and operations which can be performed on them (methods), and the connections between objects.

Opacity
The degree to which a GO will allow the background to pass through.

Oscillators, free-run
Free-running, the wave is constantly being generated without a defined beginning or end.

Trigger
With the push of a button an Oscillator can also be restarted or "triggered".

Pads
Pads are the points at which signal connections are made . There are several types of pads that require specific signals (see 'Signals') such as audio, control, controller, frequency or MIDI.

Patch
Verb: to connect an output to an input (or vice versa). Noun: a 'patch' sometimes refers to an entire collection of modules and their connections to each other, as in "You should save that patch before you quit." In this sense a patch is like a preset (q.v.).

Pattern
A bitmap designed to be applied repeatedly to create a patterned surface design.

Preset
A 'snapshot' of a device's controller settings. Used to save and recall favorite 'patches' (q.v.) or settings. Scope devices can store and restore presets for all properties.

Project (*.pro)
The top level of the Scope working environment. A project can include
devices, modules and their connections (circuits), a device design in progress, or anything else you find in Scope.

Properties
Internally they are program variables that often appear as 'options' in a working environment.Modules, GOs etc. have properties, some of which can be edited, and others that serve to provide information about the object.

Q (upper-case)
Quality factor. Filters. The selectivity factor, defined to be the ratio of the center frequency f divided by the bandwidth.

Sampling (Nyquist) Theorem
A theorem stating that a bandlimited continuous waveform may be represented by a series of discrete samples if the sampling frequency is sat least twice the highest frequency contained in the waveform.

Sampleticks
The sample clock frequency. In Scope, all calculations are specified at 48 kHz. The environment adjusts them automatically for other sample rates.

Script
A text file containing programming code. Scope's controls, for example, are implemented as scripts.

Signals
Signals are the entities that carry digitasl audio, control, MIDI etc information from module to module. Signals have several properties:

asynchronous
All signals that are not updated with each Wordclock

Controller-Values
from socalled "Val"-Pads from the faders (asynchronous)
ESYN-
Signal provided by Envelopes toa MIDI Voice Controller required for correct voice handling.

f-
Bipolar frequency pad

synchronous
The value of a synchronous signal is updated with every sample clock. An audio signal, for example, must be synchronous

Gate-
trigger signal e.g. to start an envelope generator (asynchron)

Sync-
A synchronous signal used for sample-accurate triggering of FM operators.

S/PDIF
The Sony/Philips consumer digital audio interface specification. Similar to AES/EBU, but it uses unbalanced cables and there are some differences in the word format.

Surface
The graphicalrepresentation of a module or device.

Switch
Switches a signal.

Texture
A bitmap used to provide a textured, rather than a flat surface. Texture bitmaps are applied repeatedly to provide the effect.

Tooltip
A feature to provide instant fundamental information about something. Tool Tips appear when the mouse cursor remains over an object, such as an action icon, for a couple of seconds.

Two’s complement
A way to represent integers so that both positive and negative values can be used. Scope's bipolar signals, for example, use this representation.

Value
A pad used by controls such as faders and potentiometers.

Word -digital
Can be of different lengths. i.e. a 16-bit word. Longer words can represent a greater range of values.

-sample
Also can be of different lengths
-timecode
An 80-bit digital word that is formatted to carry timing information. Used for synchronization.

Word clock
All digital systems must be controlled by a single wordclock that provides the sample rate. Wordclock is integrated in AES/EBU and S/PDIF data transfer formats (self clocking).

Abbreviations

Only the full name of the abbreviation is given below. Use the SCOPE Help facility and check the manuals for detailed information.

A	Attack
Amt	Amount
ATSl	Attack Slope
At	Aftertouch
AtT	MIDI-Aftertouch Table
BP	Band Pass
Chan	MIDI-Channel
CF	CutOff-Frequency
CFA	Crossfade Amplification
CFD	CoarseFine Display-output
ChaD	Channel Display Connector
CoFM	Coarse-Value
CTu	Coarse-Tune
D	Delay or Decay
Det	Value range
DRSl	Decay, release Slope
DSP	Digital Signal Processing
DT	Delay-Time
E In	Envelope Generator In
ESYN	Envelope Sync
f	frequency
FiFM	Finetune-Value
F In	Fade-In of a LFO
FM	Frequency-Modulation
foff	Frequency Offset
F Out	Fade-Out of a LFO
FTu	Fine-Tune
FTuD	Fine-Tune Display Connector
G	Gain
HKey	Highest key of a tranposition area
HOut	Hard-Out with Clipping
Inv	Inverted Output
L1, L2	Level modulation inputs
LFO	Low Frequency Oscillator
LKey	Lowest key of a tranposition area
LVel	Level Velocity
M1, M2	Modulation Inputs
M In	MIDI Inputs
M Out	MIDI Outputs
Mul	Multiplication factor
MVC	MIDI-Voice-Control
MVIn	MIDI-Value-In
Note	Note Number data corresponding to a played MIDI note
Out	Audio Output
PGT	Portamento and Glissando Time
PoGl	Portamento and Glissando Mode
PW	Pulse Width
PWR	Pitch-Wheel-Range
Q	Quality
R	Release
RC	Ratio-Control for Operator Frequency
Res	Resonance
Rtrg	Trigger ON, OFF
Sel	Mode selection
SOut	Soft-Out
SP	Start-Phase
Sync	Synchronus Signal - input
TKF	Time Keyfollow
TOut	Table Out
Tune	Input to adjust tuning
TVel	Time Velocity

Vel	Word corresponding to to MIDI velocity
VelT	Velocity Table
Vol	Volume
Wht	White Noise

Support

Return To Main Table Of Contents

Version 1.2

Important Technical Information

Because of the high precision of the SCOPE hardware and the thorough tests that the product is going through you should encounter no problems related to our hardware. Experience tells us that most of the problems you can run into are related to the computer configuration or wrong Windows settings. We try to shed some light on this in the following chapters and provide you with some information which is useful for running SCOPE in the Windows environment. Please have a close look at these pages before you call our technical support team. Thanks in advance!

Graphics Cards

Certain Matrox PCI graphics cards have been observed to cause severe problems in conjunction with SCOPE if the following entry is not included in the system configuration file system.ini (located in the Windows directory):

```
[mga.drv]
PCIChipset=1
```

If you're using a Matrox PCI graphics card, please add this entry to the file system.ini (Start -> Run -> type "Sysedit<Enter>" -> select SYSTEM.INI, make the modification as shown above, SAVE the modified file!)

In general, to avoid problems, the acceleration functions of the graphics card should be disabled via the card's driver setup utility. In particular, any option with the name "PCI bus retry" or similar should be disabled. Also, you should always try to use the newest drivers available. These can be downloaded free of charge from the Matrox website at http://www.matrox.com/mgaweb.

If you're using a graphics card with an S3 chip, please add the parameter busthrottle=1 under the topic [display] to the file system.ini.

```
[display]
busthrottle=1
```

AGP graphics cards are preferable, since they make no use of the PCI bus.

PCI bus frequency

Many computer users attempt to obtain extra performance from their computers by raising the CPU frequency beyond stated specifications. However, this often also results in an increase of the PCI bus frequency, to perhaps 37.5 MHz or 41.5 MHz. Unfortunately, this is well in excess of the PCI specifications, which call for a maximum bus frequency of 33

MHz. As a result, the bus in such computers is no longer a valid PCI bus!

If you run into problems starting SCOPE, or see error messages such as:

```
adr 0xxxx reads 0x...
```

or:

```
data 0xyyy reads 0xzzz
```

you should check the bus clock frequency and set it back to 33MHz if it is set to a higher frequency. See the manual of your motherboard for details.

SCOPE may run with a higher PCI bus frequency. If it does, this is simply a matter of luck. We don't guarantee it. As the manufacturer of SCOPE we guarantee only compliance with PCI Rev. 2.1, which means smooth operation at 33MHz.

Error message: 'PCI-Master-Overflow'

In rare cases, you may encounter this message:

Should this occur, we recommend using a text editor to add a maxPCchannels entry under the [dsp] section in the file CSET.INI (located in the directory ..\SCOPE\App\Bin). The complete entry should appear as follows:

```
[dsp]
maxPCchannels=100
```

In a newly-installed copy of SCOPE, a higher value (256 or 512) is used. This is a program default value and does not appear explicitly in CSET.INI. The occurrence of this error message indicates that the effective PCI bus bandwidth is not sufficient to transfer the required number of channels via the PCI master interface. This situation can be caused by the graphics card, its driver or other PCI cards and its drivers. Experiment with various values smaller than 256 until the computer works flawlessly. Good "acid tests" include opening and closing SCOPE program windows and bulk-copying files in the background while audio is playing.

In "Display .. Properties", selection of the "High Color (16 bit)" setting will minimize the likelihood of this error occuring.

The BIOS setting *PCI latency timer* can be increased up to 64 PCI clock but do this only when you enable PCI 2.1 support at the same time!

Please also have a look for the latest drivers for your PCI cards from time to time!

Additional tip: If at all possible, use an AGP graphics card!

SCOPE is already running

You may encounter this message upon trying to start SCOPE directly following Windows startup. During its startup, Windows automatically loads a SCOPE background project which activates the card inputs and outputs (for more information about this, refer to the section about the Windows Standard Project in the Installation Guide). The SCOPE program cannot be started while this is going on. Simply wait a few moments until the SCOPE icon appears on the Windows Task Bar (next to the Windows clock, if you have it displayed), indicating that the background project has completely finished loading, and then start SCOPE. Bear in mind that the required startup time increases correspondingly with increasing complexity of your background project.

SCOPE sample rate settings

When using several digital audio devices together, it is essential to be aware of the fact that only one wordclock master can exist in a digital system. SCOPE operates as wordclock master by default, with a sample rate of 44.1kHz. For example, if you want to use Cubase VST with CreamWare's A16 Audio Interface (without starting the SCOPE program), you must set the A16 to operate as a wordclock slave (optical).

Whenever you make changes to the wordclock master/slave configuration, always be sure to define the new master device first, before changing the configuration!!! For example, set the A16 to Master mode before switching SCOPE into slave mode.

An improper master/slave configuration can result in distorted audio playback, including clicks, pops and crackling. If you experience such symptoms, the first thing to check is your master/slave settings. See the manual for instructions on how to do this, as well as for additional tips!

Installing additional audio hardware

If SCOPE is installed on your system and you install an additional (non-SCOPE) audio card, you may be confronted with a "blue screen error" when you attempt to install the drivers for the new card. If this occurs, first close the SCOPE application by right clicking on the little SCOPE icon next to the Windows clock in the task bar and click on 'close SCOPE'. Once you've done this, you should be able to install the new drivers without problems.

Windows installation fails

If, in addition to SCOPE and the "usual" assortment of cards (graphics adapter, SCSI controller), you have other cards in your computer – such as network cards or other sound cards – you may in rare cases experience computer freeze-ups during a new installation of Windows with a SCOPE card already in the computer.

Solution: Remove the SCOPE card from the computer before installing Windows. Once the Windows installation is complete, you can replace the SCOPE card in the computer and proceed to install it in the usual way, as described in the Installation Guide.

Noise when switching the computer on or off

Keep in mind that SCOPE transforms your PC into the equivalent of a professional digital recording studio. In no such studio would you switch your monitoring system on and turn up the volume before switching everything else on! The same applies when switching things off – the monitor volume should be kept all the way down, or the monitoring system switched off altogether, while the rest of the studio is being switched on or off. Please do the same with your SCOPE system, to avoid possible damage to your monitoring system resulting from powerful transient signals which can occur when the computer is switched on or off!

SCOPE card not recognized by Windows / "Timeout" error messages

In principle, the SCOPE card should work in any PCI slot. Should Windows 95/98 nevertheless exhibit difficulties in detecting the SCOPE card during the hardware recognition process, simply change the position of the card once. Note that the first PCI slot, particularly in older motherboards, is not necessarily bus-master capable. According to our experience, the "middle" slots – i.e., slots 2 and 3 – are the most problem-free positions.

Can't lock sample, not enough physical memory

This message may appear when loading Akai programs/ samples into one of the sample players. These samples are loaded into PC system memory. In some instances, the system simply does not make the required amount of memory available. The problem can sometimes be solved by simply repeating the load operation.
Should this fail to work, exit SCOPE, restart the computer, and then, as your first step following the reloading of your SCOPE project, load the Akai samples.

Optimization Tips

Virtual Memory

In general, the selection "Windows manages the settings for virtual memory" should be adequate in order to work effectively with SCOPE. When using this selection, make certain that you always have adequate free disk space available (at a minimum, an amount equal the amount of system RAM you have installed in your computer).

Insufficient swap space may be signalled by SCOPE via error messages during startup or while in operation ("runtime error").

An alternative is the use of a permanent swap file. The swap file must be larger than the amount of system RAM you have installed! We recommended setting the size of the swap file to twice the amount of installed RAM. You'll find this setting under *Start -> Settings -> Control Panels -> System -> Performance ->Virtual memory*.

The following settings are significant in connection with hard disk recording:

"Disable write-behind caching for all drives"

Check the "Disable write-behind caching for all drives" option. You can find this option under *Start -> Settings -> Control Panels -> System -> Performance -> File System -> Troubleshooting*. Write-behind caching must *not* be activated in a real time recording environment.

Read Ahead Cache

If you experience audio problems during recording or playback, change "Read Ahead Cache" from its default value of 64 kB to a smaller value such as 16 kB or 8 kB, or turn it off completely. This setting is found under *Start -> Settings -> Control Panels -> System -> Performance -> File System*. You can also change the *Typical role of this machine* to "Network server". This setting improves overall performance under Windows 95/98.

VCACHE parameter

Particularly in connection with hard disk recording, having this entry in your SYSTEM.INI file will improve performance. To determine the best value for your computer, divide your installed RAM amount in MB by 4 and multiply the result by 1024. (Example: 128 MB RAM -> 128/4 = 32 -> 32 x 1024 = 32768). Add this entry under the section [vcache] in SYSTEM.INI (*Start -> Run* -> type "Sysedit<Enter> -> select SYSTEM.INI, make the modification as shown below, SAVE the modified file!)

```
[vcache]
MinFileCache=32768
MaxFileCache=32768
```

Defragmentation

Use SCANDISK and DEFRAG regularly (*Start -> Programs -> Accessories -> System Tools -> Disk Defragmentation*).

BIOS Settings

Disable *Power management* in your computer's BIOS setup, along with all other time-activated options. If you aren't using any USB devices, disable *USB support* as well.
PCI 2.1 Support in the 'Chipset Features Setup' can be enabled. The *PnP OS installed* setting in the 'PnP and PCI (Configuration) Setup' should be activated (=Yes).

Also the *PCI latency timer* can be increased up to 64 PCI clock (especially when the 'PCI Master Overflow' message appears too often). Increase this value only together with enabling PCI 2.1 support!

Hardware Conflicts

If a device is marked with a yellow exclamation point or a red X in the Device Manager (*Start -> Settings -> Control Panels -> System -> Device Manager*), there is an IRQ or address conflict or a driver incompatibility which needs to be resolved. Often, this can be handled by installing a so-called "chip-set patch". You may find this on the CD that came with your motherboard, or you may be able to download it via the Internet from the Web site of the motherboard manufacturer. Only a system which is free of hardware conflicts can guarantee smooth operation of SCOPE!

You should also check whether the SCOPE card is sharing an IRQ with other PCI cards. This is generally permissible but should be avoided where possible in order to maximize the stability of the system. Ideally, the last PCI slot (4) should not be used for SCOPE, as this slot often uses the same IRQ as all other installed PCI cards.

Getting Help

The Hotline

As mentioned above, SCOPE works flawlessly with most computers if the issues in this document are properly attended to. In addition, via the support area of our website:

http://www.creamware.com

we will constantly publish solutions for new problems as they are discovered. If you have Internet access, please check the information which is posted there before contacting us directly. The latest information and trouble-shooting assistance will always appear there first.

If you have checked all of the information presented here and on the Web site and you are still unable to solve your problem, there are various ways to contact us directly for advice. Once again, however, we request that you recheck all of the information presented above *before* doing so! In any event, only *registered users* are entitled to direct technical support, so please register now!

If you write or email us, be sure to provide us with all required information about your system. You will find a form below to assist you with this. If you call us, please have this information ready to give to the technical support representative. It's best to first fill out the form completely and then either send it to us or have it ready at hand when you call us.

Be sure to provide us with *all* required information about your system ...

1. I have tried all suggestions given in this document: Yes

<u>2. PC CONFIGURATION</u>

Processor:
Main board (chip set!):
Hard drive(s):
Graphics card(s) (which slot / IRQ / driver version?):

RAM:
CD writer:
SCSI controller (which slot / IRQ?):
CD-ROM:
Sound card (which slot / IRQ?):
other internal devices/cards (which slot / IRQ?):
How old is the power supply (!):

<u>3. CreamWare products:</u>

triple1-Board ROM serial number (which IRQ / Port?):
Your software keys?

triple2-Board ROM serial number (which IRQ / Port?):
Your software keys?

TDAT16 ROM serial number
(which slot / IRQ / driver version / program version?):
Your software keys?

A8 / A16:

Pulsar ROM serial number:
(which slot / IRQ / driver version / program version?):

Pulsar SRB ROM serial number
(which slot / IRQ / driver version?):

SCOPE ROM serial number
(which slot / IRQ / driver version / program version?):

<u>4. Connected Devices:</u>

Mixers:
Synthesizers:
Synchronizers:
Samplers:
Recorders (DAT / ADAT etc..):
Other (Atari, MAC, MIDI patchbays etc..):

<u>5. Installed Software</u>

Operating system:

Sequencer software:
Audio applications:
Other:

6. Description of the problem

When and where does it appear?
Can it be made to recur via a specific series of actions? How?
Which parts of the program are involved (modules / devices)?
Which devices? *How* are they connected?

There are four ways to reach our support department:

In the USA and Canada:

email: support@creamware.com
Fax: (604) 435-9937
Phone: (604) 435-5158
Mail: CreamWare US Inc.
446 Harrsion St.,
Sumas, WA 98295, USA

All other countries:

email: support@creamware.de
Fax: (++49) 22 41 - 59 58 57
Phone: (++49) 2241 - 59 58-12
Mail: CreamWare Datentechnik GmbH
Support
Wilhelm-Ostwald-Strasse 0/K1
53721 Siegburg, Germany

But for now – enough hints about possible problems. As an experienced computer user, you are no doubt well aware that neither software nor hardware which is one-hundred percent perfect exists. We at CreamWare strive continually to improve our products, and we welcome your criticism and suggestions.

Having said that – we hope (and expect) that you won't encounter problems with SCOPE, and we wish you all the best in working creatively with SCOPE!!!

Sincerely,
Your CreamWare team!

Warranty Support

Before sending in your SCOPE card for warranty support, please call the Support office to obtain an RMA number for your card.

Shipping Address:
CreamWare US Inc.
855-C Conklin St.
Farmingdale, NY 11735
USA

Sales and Support
Office:
CreamWare US Inc.
6915 Russell Ave.
Burnaby, B.C.
V5J 4R8, Canada

Tel	604-435-0540
Fax	604-435-9937
Sales	800-899-1939
Support	604-435-5158

Warranty and Disclaimer

CREAMWARE GmbH ("CREAMWARE") warrants this product to be free of defects in materials and workmanship for a period of one (1) year for parts and for a period of ninety (90) days for labor from the date of original retail purchase. This warranty is enforceable only by the original retail purchaser.

To be protected by this warranty, the purchaser must complete and return the enclosed warranty card within fourteen (14) days of purchase.

During the warranty period CREAMWARE shall, at its sole and absolute option, either repair or replace free of charge any product that proves to be defective on inspection by CREAMWARE or an authorized service representative. In all cases disputes concerning the warranty shall be resolved as prescribed by law.

To obtain warranty service, the purchaser must first call or write CREAMWARE at the address and telephone number printed below to obtain a Return Authorization Number and instructions concerning where to return the unit for service. All inquiries must be accompanied by a description of the problem. All authorized returns must be sent to CREAMWARE or an authorized CREAMWARE repair facility postage prepaid, insured, and properly packaged. Proof of purchase must be provided in the form of a bill of sale, canceled cheque, or some other positive proof that the unit is within the warranty period. CREAMWARE reserves the right to update any unit returned for repair.

CREAMWARE reserves the right to change or improve the design of the product at any time without prior notice. This warranty does not cover claims for damage due to abuse, neglect, alteration or attempted repair by unauthorized personnel, and is limited to failures arising during normal use that are due to defects in material or workmanship in the product.

ANY IMPLIED WARRANTIES INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE LIMITED IN DURATION TO THE LENGTH OF THIS LIMITED WARRANTY. Some states do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you.

IN NO EVENT WILL CREAMWARE BE LIABLE FOR INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES RESULTING FROM THE BREACH OF ANY EXPRESS OR IMPLIED WARRANTY, INCLUDING, AMONG OTHER THINGS, DAMAGE TO PROPERTY, DAMAGE BASED ON INCONVENIENCE OR ON LOSS OF USE OF THE PRODUCT, AND, TO THE EXTENT PERMITTED BY LAW, DAMAGES FOR PERSONAL INJURY. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

This warranty only applies to products sold in the United States of America or Canada. The terms of this warranty and any obligations of CREAMWARE under this warranty shall apply only within the country of sale. Without limiting the foregoing, repairs under this warranty shall be made only by a duly authorized CREAMWARE service representative. For warranty information in other countries please refer to your local distributor.