Basic Techniques

About This Tutorial

The goal: a quick (and easy) start
A friendly but IMPORTANT WARNING!

Basic Exercises

New Project: starting clean
Where to find modules and other stuff
Move Mode: the basic SCOPE working mode
Adding modules to a project
Making a connection between two modules
Quick sound check
Repositioning a module onscreen
Deleting a connection
Deleting a module via the keyboard
Deleting a module via its context menu
Selecting / moving / deleting multiple modules

From Simple Oscillator To Synth

Sine oscillator with pitch control
Use Mode: operating the pitch control
Adding a level control
Copying modules in the Project Window
Adding an envelope and a MIDI input connection
MIDI sound check
Opening and closing control surfaces
Saving the project
Reloading the project
Devices: the next level

Creating The TinySynth Device

Introducing the Empty Synth
Going inside the Empty Synth ....
.... and getting back out
Empty Synth internal components
Constructing a synth using the Empty Synth
Building the TinySynth
Saving the TinySynth as a device
Using the TinySynth
Opening the device surface
Adjusting the polyphony of a synth
Creating and recalling presets
Renaming the device and device surface
Saving modifications to the device

A Closer Look At The TinySynth

An inside look at the Sine Synth module
Variable polyphony and the Poly Mix 16 module
About the voice control and envelope modules
The voice control and envelope module control surfaces

Return To Main Table Of Contents

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.