Further Techniques

About This Tutorial

Creating A Stereo Attenuator Module

- Exported pads
- Exploring device structure via the module tree
- Locating objects in the module tree
- Deleting objects via the module tree
- Using context menus in the module tree
- Creating new modules from circuits
- Manually redrawing windows and tools
- The Pad List in the Project Explorer tool
- Renaming pads via the Pad List
- Exporting pads via the Pad List
- Unexporting ("deleting") exported pads
- Exporting pads via pad context menus
- Moving modules out of a module
- Folding portions of a circuit
- Moving modules inside an existing module
- Using drag and drop in the module tree
- Invisible connections – a first look
- Drag and drop in the module tree (continued)
- Module tree: drag-and-drop vs. cut-and-paste
- Cut-and-paste: module tree vs. Project Window

Creating The Device And Its Surface

- Preparing the device
- Drag & drop from the File Browser into the module tree
- Device surface states: flipped in, flipped out
- Adding a fader to the device surface
- Connecting the fader via cable
- The MinimizedView and the Minimize button
- Connecting the fader via the Pad List
- Connections via pad context menus
- Connections to controls via module context menus
- Tracing and deleting connections via the Routings List
- Adjusting the fader characteristic
- Modifying the surface label: the Text Editor tool
- Adding and duplicating labels
- Hiding, unhiding and reorienting module pads
- Tightening up the device surface layout
- Resizing the surface: a first look at Edit Mode
- Modifying surface appearance: the basics

The TinySynth Revisited

- A (very) quick-and-dirty TinySynth surface
- Final remarks – what to do next

Return To Main Table Of Contents

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. Arbritrarily 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