Scope Development PlanetZ forum information

Help with the Control Ranger

For a test I created a trivial device. In an empty effect insert I put a mastergain circuit and controlled its level in the circuit with a fader. Then connected a 'potentiometer metal' to control it. Worked fine. Then I want to use the control ranger to set an audio taper for the control. I get an exception "PepNGo:GetViewItemOfGo: invalid value for the 1. argument ParentViewItem and the control ranger is a blank box. I tried to open in circuit and surface and at different levels both on the circuit fader and in the surface knob and tried in use, move and edit modes. When I do it on the surface know it messes up the knob positions separating some of the graphical components.

The problem was where I was when I tried to do this. You need to be in surface view and you need to be down inside the device (from circuit view) and you need to be in move mode(I think). Not obvious from the error I got. I was simply trying to change the fader to a log taper, easy to do if you are in the right view.

I have not noticed it matters what mode you are in. just make sure you dont use a "protected" pot from the default panels (well you can use them but you cant see the control ranger values).I just click on a control and change it. I haven't ever seen an error like that.

The Control Ranger can be buggy. I never use it. It's much better to delete the Panel & SurfaceInterface modules from inside the Mastergain module you're using (or any other module with it's own surface panel), then load a separate Mastergain module & copy/paste it's PadList control & range text values accross to the PadList control vals for the new controls on your own panel (in this case the panel of your empty insert effect). Then just delete the extra mastergain module (or whatever) that you loaded for extracting the values.

The PadList val for a log 'curve' is 1 (the default val is 0 which is linear)
The PadList val for an exp 'curve' is 2 ... and so on. - Shroomz

---

Help - SDK Help window too small

The help windows do not word wrap properly with in the box. There a no scroll bars so I cannot scroll to the right to see what it says. For example what I do help on the Variable Deszip bipolar the help window says: "Tau Scale Positiv Negativ This is an adjus". Is this related to screen resolution or some font size setting somewhere?

Yep. go into the dsp directory and read out the txt file there

I can't read the full module description and the sizing of the help window seems fixed? Is there any way around this?

you can read the text files in the DSP folder itself.

the help window in sdk doesn't text wrap so it's useless as I can't read everything, is this the way it is? any way for it to work?

You can directly read the help (nfo) files in the folder where the dsp atoms are stored.
scope sdk\app\dsp-> *.nfo
thanks, but I can't find what I wanted there, just some info the the insert text window, it's be great to have the help window word wrap

but the help window displays exactly the text from the nfo files ?! Nothing else.

No oscilloscope in Tools

The Quick start manual says to use the oscilloscope found in the tools directory. I don't have a tools directory. Has this been dropped?

The oscilloscope I released for the modular is just a 1 to 1 version of the one for DP, but Neutron's correct - it's not that great. That scope for DP (imo) was only created as a quick in house visual reference for wave shaping development work in DP so that it's easy to see the effects of experimentation. It has that use as the modular version as well imo, but for a serious (detailed, hi res) visual reference you should definitely use a standalone scope, be it software or an external hardware unit. - Shroomz

---

How to - PresetList building instructions - Shroomz

Here's some instructions on how to make sdk devices preset compatible. I think it's quite complete, but if anyone spots a mistake or something I've missed, please let me know & I'll add/remove or ammend where necessary.

Note:- These instructions assume that your devices' surface controls are connected to your circuit via circuit level Controller Pad modules which you'll have named appropriately according to their given parameter names in the Project Explorer.

1. You should be in 'circuit' mode at the level of the Controller Pads for your Surface controls & be in 'use' mode with a PadList window open.
2. Select your main device parent module in the Project Explorer.
3. Imediately open the 'Parameters' list & 'freeze' it.
4. Imediately Open the 'PresetParameterList' & 'freeze' it. (if you deselect the paraent module, make sure to reselect it)
5. One by one, click on each of the Controller Pads for your Surface Controls (pots, sliders etc) in your Project Window & drag the 'val' parameter for each of them from the PadLIst across to the 'Parameters' list.
6. Rename your val parameters in the parameter list you've just created (click on 'val' & press F2).
7. Check if the little pull-down list on the 'PresetParameterList' module displays 'PresetList'. if it doesn't, press the create button once & it will appear.
8. Drag all of the new parameters added to the 'Parameter' list accross to the 'PresetParameterList'.
9. Press the 'open' button on the 'PresetParameterList' window & a familiar preset window will appear. Close that new window.

Your preset list has now been created & will appear along with a new surfaceinterface module inside your parent module's tree in the project explorer.

10. Close the 'PresetParameterList' window.
11. Select the newly created PresetList in the Project Explorer.
12. Go to the PadList & drag the 2 parameters named 'PresetList1' & 'PresetList2' accross to the 'Parameter' list.
13. Making sure the presetlist is still selected, drag the LastRestoredPreset pad in the PadList accross to the 'Parameter' list.
14. Rename the 'LastRestoredPreset' parameter to 'cwLastRestoredPreset'.
15. Connect the presetlist button 'val' parameter on your device surface panel to the PresetList 'show' parameter.
16. Save your project, then delete all circuit gos or optimize 4 XTC, protect then save parent module as mdl or dev.
17. Close sdk not saving on way out & open Scope.
18. Load your new module & open it's preset list window.
19. Create a new bank with your default/factory pres inside it, then save the new bank as a .pre in your Scope presets folder.

The preset file you've just created can be distributed with the new module & should work fine if the instructions above are carried out properly.
You forgot to add the bit about connecting MIDI to the Preset List!

1) In the Project Explorer, select the PresetList
2) Select and store midiIn pad
3) Connect it to the Midi In on your device.
4) Also make sure midiChannel pad of the PresetList is connected to your device's MIDI channel control, such as the CSynth pad in the MIDI Channel Linker

Also, note that you can name your preset list using the Caption parameter of the PresetList. Highlight the default name value area where there is nothing (slow double-click or use F2), and rename the PresetList. It will change from the default of "U KNOW 007" to your new name.

You will also want to store other pads of the preset list that determine behavior in a Project. These are: Show, PosX, PosY, SizeX, and SizeY. These are not to be stored in any preset, but tell the Project about the state of the Preset List (showing or not), as well as the location and size of it on the screen. To set these kinds of parameters correctly, you need to go into the ModuleParameterList and change all the little check boxes (which normally default to things for Preset List storage) to only a few columns, as follows - RP, RS, the SP in the middle (next to RR), and SS.

While we are talking about the ModuleParameterList, I thought I would list what those abbreviations are supposed to mean, as was told to me by Matthias Klag long ago:

P - Public (means the parameter can be 'seen' by programs like Cubase which can list available parameters in a device)

RP - Restore in Project
RS - Restore in Screenset (you can have multiple Screensets by using Ctrl + any number key)
RP - second one, this time for Restore in Preset
RR - Restore Related (?) - having to do with modules that are children of parent modules
SP - Save in Project
SS - Save in Screenset
SP - second one, for Save in Preset
SR - Save Related

I think the check box under the title 'Default Value' was actually something for use in Noah, as the area to the right of that box is where the actual Default Value will show up. You have to type into the empty space to enter a value (and I only have used it for the Show parameters of PresetLists and Surfaces, set to 0 so they don't show up when you load the Device)

I had some problems with this. If I chose this procedure, right-clicking on the device would let me choose a nil-referenced "SurfaceInterface".

With a created PresetList, there should now be two SurfaceInterface entries directly under the device tree. One of them is for the surface panel and the other one is for the preset browser. You can easily find out which is which though the last one in the tree is usually the one you want.

Find the 'AddShow' value and store->connect it to 'val' of presetlist button on device surface panel.

Whenever you right-click device, it will now correctly show option for opening "Panel" or "PresetBrowser". - Voidar

---

Problems with presets

I am trying to figure out how to do the parameters presets. I found the HOW TO -- PresetList building instruction by Shroomz from a while back. But I am getting stuck on number 5. It says to drag the val across to the parameters list. When I try to do this cursor turns into an arrow with a cross through is so I cannot do this. The parameters window is transparent, I can see the routine window stuff in the middle. Not sure what I am missing. The docs don't discuss this window. What do the icons in the upper corner mean. I understand the freeze, then there is an icon that looks like 2 pieces of paper(clicking this does nothing obvious), one that is an arrow that is blue (runs red on first click, to 3 yellowish arrow on the next click, then back to blue), then a clipboard, then a >(when i put the mouse over this I get write uuid file, read uuid file, create from uuid file, assign new uuids) then the close

1.first you have to drag the controls (potis etc) to the Parameter list (point 5)
after that, you have to create a Preset List (point 7). point6 (renaming) is important too. all that is written in Shroomz' little but excellent tutorial.
2. (icons) this is for copy ("paper"), transfer (arrow), paste (clipboard), load/save.
I use this just for GOs. don't know if or what sense it makes with the Parameter window, but in any case you can live w/o it.
btw the docs ***do*** discuss this topic (Project window/ToolBar dialogs) - roy thinnes

Thanks for the reply. But I cannot do point 5 like you suggest. I am supposed to drag the pot value to the parameter list. This does not work. When I try to drag the muout pointer get an x over it not allowing the drag. Maybe I have the wrong window for 'parameters'.. I click window->parameter->parameters. Is this the correct window for point 5? When I open it I get a short window that has <No Parameters available> in the header and nothing it body. If I enlarge the window a box within the window appears that is transparent allowing me to see the routing window below it. I can even select routes and devices right thought this window. This is definitely not a normal window. Am I looking at the right thing? This parameters list seems to completely useless. I am getting very frustrated with this.

ok then, let's try this:
-create a new project (File/New)
-draw an empty synth into it (Circ.Design/Basics)- Circuit mode!
-double-click on that empty synth
-change to Surface mode
-draw a GO, say: a potentiometer onto synth's surface
-in ProjectExplorer, select EmptySynth
-open Parameters (Window/Param./Parameters): do you see some pre-configuered preset parameters? Freeze it
-select the pot on synth surface
-draw "Val" (PadList) to the Parameters of EmptySynth - window
ok?
if not, then there is something utterly wrong with your setup.

I reinstalled from scratch and now the parameter window is completely different. I can even drag stuff in to it now. OK, at this point I can save the device and all the values are restored when reloaded, but I cannot load the presets. I can save them (at least they appear to be saved) but when I try to load a preset nothing happens, no knobs move. The parameters list shows Preset File1, preset File 2 and cwLastRestoredPresets are shown as not assigned in the Module column. Is this normal? The parameter windows indicates all my controls are stored in preset and restored in preset. Is there something else required to make the presets load?

not at all! it's an hint that you did something wrong, maybe:
-you accidentally deleted/replaced the GO
-you didn't freeze the Parameters List
delete all "not assigned " entries and make a new try, precisely following Shroomz' instructions.
hey, this ain't difficult at all, believe me.

Agreed, is should not be difficult. However, there is so much going on under the covers, if you make a mistake it is not clear where you went wrong and what do do to fix it. I did follow the Shroomz directions, every step. I assume I

misunderstood one of the steps, or one of them failed and I did not know what ever.
To do it again can I simply delete the parameter list and preset parameter list and do all the steps again?

first of all, all these "not assigned" entries have to go - just select and delete them, one by one. shouldn't be difficult for one or two GOs resp. list entries.
-btw, you are making a test device, just to learn how to make presets and stuff, or? at least I woukd recommend that; if you're familiar with it, you could go for "real" devices.
Don't delete the Preset List in Explorer.
If you have the feeling that you irreversible did something wrong, just load a new EmptySynth device and try again. just one or two GOs, and then Shroomz' instructions. takes 1-2min max.
btw in these instructions, I've found one little incoherency, namely
p. 12: Preset **File** 1/2 instead of Preset List 1/2
in point 13, the LastRestoredPreset is meant, not LastRestoredPreset (unknown).

I had realized the preset list 1 and 2 were wrong. "LastRestoredPreset is meant, not LastRestoredPreset (unknown)" They look identical to me. I assume you were pointing out a spelling error(I don't have the instructions with me right now). It is pretty clear what he meant on this one since there are no values even close to this name.

Ok gurus, I cannot get this to work. The process seems easy, everyone says it works, the why can't I get it to work. I created a super simple project. I started with and empty effect. In the circuit I put a master gain 12db. I connected the inputs to effect input, the output to effect out. I then added a horizontal fader which I connected to the gain node on the fader. Then I went to surface and added a fader there as well. I selected the fader, clicked on val in the padlist, switched to circuit view, clicked on the fader, selected the val and clicked connect. Module done.

Then I went to surface view and opened the parameters and preset list windows. I selected empty effect in the project explorer and froze both windows.

Then I selected the fader in the surface and then dragged it to the paramaters list and renamed it fader. Then I clicked create in the preset list, opened the preset list and dragged fader to the preset list. Then I clicked preset in the project explorer and dragged Preset file 1 and 2 and LastRestoredPresets from the padlist to the parameters. I renamed LastRestoredPresets to cwLastRestoredPresets.

Then I went to the tope of circiut view, right clicked on empty effect, save as new effect. Closed sdk. Opened scope 4.5 and loaded the device. Opened the surface and preset list. Double clicked midi controller. Clicked preset then store and named it 1. Then I moved the fader to 100%. Clicked preset again and store, said no to overwrite and named it 2. Now I should be able double client 1, to reload the first preset. Nope, neither one opens.

If I save the project, then restore it, the device loads as it was saved. What did I miss????????

1. when you copy the MasterGain to the circuit, first delete its panel and surfaceInterface, you won't need it, it only disturbs/confuses you.
2. after copying (circuit mode), double-click the emptyEffect and you see the circuitry. change to Surface view and drag the fader fo the panel of the empty effect. (I don't know why you added 2 faders?)
3. then do the preset list & PPL thingies as you described
4. open Scope 4.5 and load the device
5. check out if the presets work by changing fader settings & saving + loading presets.

what do you mean with "Double clicked midi controller"?

That worked now. It seems like deleting the surfaceinterface and show panels for all the internal modules was the problem. I went back to the original project and removed them and now it works.

2-there were 2 because that is the way it was done in the first use docs.

5-double click midi controller. My mistake, should have created a bank the double clicked on it, not the midi controller bank.

Alignment tool / Dimension tool

How do you use the alignment tool? i just want to make a bunch of knobs the same height. i have never gotten it to work right.

The alignment tool can occasionally be out by a pixel depending on how you're using it, but it's *generally* spot on. What I normally do when aligning a bunch of pots (while in surface mode, which is a must) is get one pot into position on say the far left of the interface, then in the project explorer copy it & paste it into it's parent. At that point it will appear copied over the top of itself, but if you click on it on your panel surface, you'll have selected the newly copied version. You can now either right-cursor it to the next desired pot position in the horizontal plane or manually type a pixel value/position into the alignment tool (which always works perfectly in this case). The same procedure applies even if you're needing it to be in a different surface group (for a different page for example), except that after moving the new (copied) pot into position, you then need to manually move it from it's copied position in the project explorer hierarchy, to it's desired new position in it's destination surface group.

I know you'll most likely know this, but I'm going to say it anyway. NEVER try to use the protected pots from the panels of sdk modules on your own panel (by moving them or whatever), as it's very bad practice for various reasons. New (unprotected) pots from the SurfaceDesign sdk folder (whether they're Creamware's or your own) should always be used as a starting point. The 'trick' I've used is only valid when you have a completed interface design. It's basically a case of accurately placing your pots over their desired position in PS (or whatever) & creating a black & white reference/positioning skin which only has the control positions in white. After positioning all of your controls according to the white areas, you can then replace the skin with the actual skin. That process doesn't apply to interfaces that are made solely with the sdk tools & don't have a tga or bmp skin.

oops, I thought you were talking about the 'dimensions' alignment tool Neutron. (which is what I was referring to & use for pot & control alignment) - Shroomz

what i am talking about is a tool that looks like it should be able to take the height of a certain item, and then you could use the little multi arrow to "put" that height to the other items. Then you could do the same for side to side location. 3dsmax has a tool like that i use all the time at work, i just thought this was for that.

You can use the "dimension" tool to transfer the size of a resizable graphic module to another (text displays, inserts, fader "line"..)..
you make one of the size you want and transfer that size to the ones to be resized. works for x/y axis at once (with the multiple-transfer arrow).
But you have to target well: for ex, a text is made of 3 layers, you must target the background in order to resize the whole group at once...

Aligning various modules on the same horizontal or vertical line (like in synthmaker) is not done automatically in SDK (at least, to my knowledge) (That's what the Alignment Tool does. If you've got multiple GO's inside a surface group, you can align them to specified top, bottom, left and right offsets in relation to the surface group. I don't use it because it's really simple to just use the Dimension Tool - Shroomz) – SpaceF

ok i went back to looking at the dimensioning tool and nudging with the arrow keys. - Neutron

---

Mod Wheel

I seem to remember in my old scope DP there was an example synth with a mod wheel in it, that one with the flames - but now with SDK i do not see it. i just want to add a mod wheel amount to control some modulators, preferably without the actual animated modwheel graphic.
i searched for "wheel" in my scope SDK folder and don't see anything.

You could just connect a pot or slider that's assigned CC1 through a controller modifier & set the curve to linear. The output from the controller modifier would then be connected to the gain pad of a mix module for attenuating the LFO or modulators.

I will try that, with an invisible pot. Thanks. i would like to look in one of the example synths and see if it is done that way

blue synth has it... assign CC1 to LFO gain... that's it.....

That's a pretty simple mod wheel, but i get the picture

Inferno has one too

Number of voices of synth in livebar

i believe the top DynVoicesOfParent has to be connected to something, i cant remember what

Try unfolding it into an empty synth module, minus everything in the circuit that you'll find in the empty synth module.

Ok i tried the unfold in a empty synth trick it solved that problem, but created some new ones, such as now the preset list is the "panel" and the outputs can not be connected to anything without an error. What i need to do is connect the voices pad to @Paradox NoVoices like the empty synth is, but i do not know where to find that connection.

That first problem may have been caused by the control ranger being open? anyway now its fine except the preset list opens in scope instead of the main panel
(you can get it to open with "open panel" right click though - at least i can set the number of voices from the livebar now.
Ok i got it figured out -you have to delete your original surface interface and use the one from the emptysynth, reconnect it to the panel, and reconnect the close button and preset list button and make sure the controlranger is not open when you do any of that.

This unfolding into a new empty synth module trick can also somehow/sometimes fix the problem of midi controller assignments not being remembered upon saving & reloading a project. No idea why though.

try doing that with the control ranger open, that's what screwed me up.
there were little connection lollipops all over the ins and outs and things they should not be able to connect to.

have to say I don't like that Control ranger at all. Bug central, so never use here.

Sync an osc = retrigger

i just tried to check the text and sync an oscillator to start over i connected the gate to the sync Saw oscillator S but i can't make it to retrigger to it's initial value but i can't - doesn't it suppose to restart if i connect the gate to the sync on Saw oscillator S ?

The sync in is retriggered by a zero crossing of another wave, normally they would use a saw, dont tell anyone but you can actually use anything. but squares can give a bit of a "nasty" transient. The sync output on a sync master osc is nothing more than a sawtooth wave.

Thats not to confuse "sync" with an LFO, which does use the gate. it makes it start at the same point in the wave cycle when a note is pressed. if you want your oscs to do that you can use an LFO as an audio osc. and it will always start at the same place when a note is pressed. (and you can even adjust where with the phase control)

so if i just connect a slave to the gate it won't retrigger ? I have to connect it to another master that i can't see his syncing option it seems that i use this oscillator as a gate to sync then

sync m oscs are have an extra out for sync. Sync s ones are slaves.

But gate or x-crossing: if you monitor flexor's gate, you'll find gate is near 0 in like the 10th decimal place, so they're practically the same for monophonic operations.

Filter design

there is an option in sdk to fix a filter to one dsp, meaning you can build complex filters with polyphony, but while it works fine in sdk the feature is ignored in scope 4.5. So its doesn`t work.
I guess its called "per voice" in the on same dsp attribute options. (sdk)
Otherwise scope tries to load polyphony filters onto one dsp (yes option), which limits the possibilities a bit.
I hope this gets fixed with next scope / or scope sdk release....
so you can fix a structure (lets say a filter, which is a must normally) to a certain dsp with option set to YES.
now if you set voices to 3, the scope card tries to fit the 3 filter modules (3voices a 1 filter) to one of the dsps.

so the maximum dsp power you can use for a module (lets say a filter) with the option set to YES is 1/16 of ONE dsp if you want to reach 16xpolyphony
if your filter utilizises 1/8 of ONE dsp and option set to YES, you get 8xPOLY and so on.
so an dsp expensive filter lets say which uses 1/2 of an single dsp sharc can only played 2 x POLY.
that's the limitation.
With the PER VOICE option you can build dsp expensive filter designs and they are split over various dsp with multiple voices.

than a synth with lets say a filter, an envelope, an VCA and so on gets split over dsps in a different way.
1 voice is applied to one chip, while the others voices may be loaded to a different sharc.
by marking groups of atoms you can assign them to different sharc chips.....

try around a bit and you will learn the workflow, its more a technical aspect, but with more complex filters you need to understand this quite well or you won't succeed.

With On same dsp set to NO, the filters will break out due to latencies between the chips, because the atoms inside the filter may be split over various sharc chips, which will introduce a little latency (a few samples). Its written inside the sdk manual, i think i remind myself when reading it.

The per voice option is a good way to build complex sounding poly synths. I`m sure CW high end (minimax, prophet, prodyssey) synths have to use this option.

But as i said: its a bug inside scope or sdk. The setting works inside sdk and not in scope.

So its either that sdk does not properly save this setting or Scope 4.5 doesn't read it properly. Or its not a bug and CW deactivated it. have a look at the picture.

by the way, i talked with red_muze about this problem and he confirmed this and he also has not found a solution so far. Thats why adern filters have different poly count. They must be set all to yes – hifiboom

---

Neutron7 scope projects

The DSP files for the WPS_Protector_mono and Stereo mdl's seem to be missing. Are there any other missing files?. What was your original set up for your projects?. I get exceptions warnings regarding midi and ADAT when loading the project files

I had the same messages. I found I could load the projects anyway, so I saved the modules individually one by one. Now I can load the modules in a clean project without getting the messages.
i forgot to mention that the stecki atoms are in another zip in the original file. you need them in your DSP folder for some devices to work.

btw, you can fix the issues of checking out Neutron's projects in sdk 4.0 simply by loading them & re-saving them from the sdk. You might have to use the open project & 'all files' option when re-opening them though, but I can't remember now as it's been ages since we took a quick look at them. Anyway, it *is* possible to check them out without the crashes & other problems.

Schmidt trigger

Is there such a device in the SDK? I want to take the beat frequency between 2 oscillators (at low frequency), feed through a Schmidt trigger to trigger an envelope. Can it be done or is the Schmidt trigger redundant?

all that is is a gate which is triggered at a higher level than it releases, (it has hysteresis)
are you trying to have the beat frequency actually turn the envelope on and off? why would you do that?

Yes I want to turn the envelope on and off with the resulting square wave produced by the Schmidt trigger. The beat from the 2 oscs would pass through a low-pass filter then be squared up by the Schmidt. Does an envelope generator require fast leading/trailing edges or could you feed it the waveform from the low-pass filter and let it trigger when the level is high/low enough? (In an electronic circuit, this wouldn't be repeatable or predictable enough.) In emulation maybe you can't connect an audio source to a trigger destination.

you have to change the resulting signal to a "gate" i believe it would work without anything else, that stuff usually works on zero crossing or you can just use an envelope follower set to very fast attack and decay, on a low enough frequency it will just turn on and off and you can adjust the threshold.

Surface

Just wondering if there is a way with the sdk to create a device that works like the 2448 mixer where when you select a channel the display reflects the setting of a different channel. How is this done, or can't it be done with the sdk alone? I am working on a mixer type of device. I would like to have a view of information the way the 2448 mixer does. You see the compressor, eq graph and auxes for one channel at a time in the master section, but the channel display has limited controls. If you select a different channel, that master display changes to show the controls of the channel. In other words, I want to have controls on a single display that change based on selecting the channel from another display. Can something like this be done with the sdk? Another option would be a way to click a button on a channel to change the size of it, and hide or show different elements. (the drawer concept I have seen on some like the sonic timeworks devices to is another way. Can any of these things be done with the sdk?

Yes it's possible to do all of the things you've mentioned with the SDK. If you start off with a surface group and place all of the surface elements (controls, text, etc.) inside the surface group (in the project explorer) you'll then be able to use the surface group 'show' pad to show/hide the group. Then if you create more pages or in the case of your mixer example you might want to just copy and paste duplicates of the surface group for one channel, you can switch between them by creating a simple logic circuit - I usually use radio buttons and 'if' modules. Hope this helps - Sharc

I am not familiar with what you mean by surface group. Are there any examples of this in any of the modules supplied with the sdk, or do you have any very simple examples of how this is done you would be willing to share?

Check in the Surface Design\Surface Kit folder. I can't think of any example SDK modules, but I'll knock something together for you

Most of the SDK modules use Surface Groups to group controls together. In the Project Explorer they've been renamed to Controls Group. One of the difficulties with using Surface Groups is that when you drop an empty group into your project, you can't see it. I usually take a note of where I've dropped the group (on one of the background scope logos) then switch to edit mode and resize and position the group. I've made the following module - The GO type has been switched to a dynamic bitmap - which might be easier to use for some – Sharc

---

Modifying modules

I would like to have a simple parametric eq that all the controls can be changed by midi. For example, take the 4 band parametric eq and mid-ize. First, how do you get the surface to display. Do you have to create your own surface or can you use the one provided. Then how do you set the channel? How do you connect the midi to the controls. The documentation is quiet on this. It simply says add the midi controller linker and the midi channel linker, connect the CH to CCon and export the midi pad. Is this supposed to be all there is to it. How do you select midi channels and controller numbers?

You can and should create your own surface. Dig into the GO menu and tools.

why do they provide a surface that is not to be used? Knobs, controls, pads, ect that are not needed, resource waste isn't ti?

I guess it is for quick access while constructing the device before editing the surface .

Viewing waveforms

What is the best way to view waveforms within the SDK. I have a "real" CRO but not sure if I connect intermediate signals to Audio out is there any loss? Will I see supersonic components? Also want to view control signals etc. I understand the Oscilloscope that comes with it is unstable. I have spectrum analysis and Fourier tools (software), how would I send signals to them?

SDK has all I/Os so you can send the test signal to e.g. a wave dest and connect any native soft scope to the creamware wav outs in the OS. There is a modular module from shroomz around here that is quite usable within SDK.

i recommend Visual Analyzer - just connect the signal you wanna view to the wave dest output.

How to use panels of an SDK module in a project

I know I've asked about this before and that is being able to use the provided surface of the sdk modules. The answer I got before was for quick use but you must design your own surface.
The compressor module then presents a problem with that. One important need in using a compressor is the gain reduction meter so you can tell what it is doing. The compressor module does not have any outputs for connecting an external meter for gain reduction. So how can you use this module for anything if you cannot add the meter?
I tried to use the meter by moving it to the parent. I used an empty synth and when you switch to surface mode, the synth window and the meter show up. But you cannot place the meter on the synth surface. It appears to go behind the synth surface. I have tried messing with z-order, topmost, bottommost, etc but nothing works.
So my original question is restated, "how do you use the provided panels, or selected gos of an SDK module in a project?"

click on the VU in the panel + save it (proj explorer). you now can use this GO as any other GO (pot, fader), by dragging it to your synth's/fx' surface.
btw. the provided panels also have the function to give you hints how the various parameters should be handled - here eg what is the range for comp attack/release and so on.
btw2: some GOs like the compressor display require PackDevice/Optimize and Protect (rightClick in proj explorer), otherwise they cannot be seen.
this means that you have to save at least two versions of the device: a "normal"/unprotected for editing, and another for testing.
(after you protected a device you can't edit it any more – roy thinnes

"the provided panels also have the function to give you hints how the various parameters should be handled - here eg what is the range for comp attack/release and so on." How do you access these hints?

just by looking at the vars of the module.
say, you want to use a delay module, eg a 1kmDelay. without the provided panel you won't know the delayTime range.
so what you can do is just click on the panel's DelayTime pot, look at the various vars (a.o. min and max) and copy the settings to your panel's pot/range text.
(another way is to save & then load it into your own panel

Graphics elements

I have yet another question about graphics in SDK:
How do I create "-" and "+" buttons with the following range [0;6]. When you press the "+" button the value increases with one and the "-" decreases the value with one..
I choose 2 button.mdl modules with the following parameters :

Plus button :

Min: 0
Max: 6

Step: 1
Function: 1
Disable: 0
CtrlNr: -1 (?!?! - have no idea what this means)
That's for setting a continuous controller number on the fly for that particular control, in this case a button.

Minus button :
Min: 0
Max: 6
Step: 1
Function: 1
Disable: 0
CtrlNr: -1

I connect the both the val parameter and I get the behaviour Im looking for, except that the system wraps back to Minimum if you press the plus button one more time at Maximum value. Thats not the case the other way around - wraps to Maximum?!?!?!

I'm back in the studio today & figured out a very simple solution to this problem. Basically all you need to do is set the max value of the minus button to one higher than you need, so your two sets of button parameter values should look like this:-

Plus button :

Min: 0
Max: 6
Step: 1
Function: 1
Disable: 0
CtrlNr: -1

Minus button :
Min: 0
Max: 7
Step: -1
Function: 1
Disable: 0
CtrlNr: -1

Note that the Minus button's step value should be -1, not 1.

So, this will now work as you want it to & without using any unnecessary logic. A connection example (for anyone new to sdk) would be to simply connect both +&- buttons to a text fader val & also connect the text fader val to your destination such as a switch val selector, midi channel selector or whatever. - Shroomz

Solved that wrap around from max to min issue as well. Just connect the plus button val pad to a logic IF module IN. Set the IF value to the same as your plus button's max value, ValYes = 1, ValNo = 0, then connect the IF module OUT to the Disabled pad of your plus button. Now it will stop at the max value & it won't wrap around.

Thank you for the solution, but as tgstgs pointed out : the animation stops before going back to normal state and therefore the button stays ON. Tgstgs did come with a solution for that problem, but I chose another approach :

Still using the IF module in the example from Shroomz, but connected the OUT to the "Step" pad and changed the "Max" values to Max+1 on the plus button. Problem solved.

IF module :

ValIF : Max-1
ValYes : 0
ValNo : 1
In : (connected to the "Val" pad of the plus button)
Out: (connected to the "Step" pad of the plus button)
Condition : >

How to change the polyphony in the sdk

How do i change the polyphony in the sdk when i'm testing a synth?
i think i looked at all the relevant modules for a value to change

Use the DynVoicesOfParent module for Poly sections. In the module attributes for your device you can also set the number of voices.

Now i have found another strange problem. if i have polyphony of 3, then no problem but if i raise it to more than that then every nth note (n = number of polyphony) the sound is different. i think it might be to do with loading on a different DSP?
rightclick -> attributes -> OnsameDSP ->yes

I had tried that but the atom i thought was causing the problem was not. it was the mixer after it. and that seems to have done the trick.

Line Text

I wanted to put a editable text like and a go, so if you have several of the same device open you could tell them apart. For example, say you had 3 or 4 eqs patched in line to a mixer and you wanted the text one to say guitar, bass on another. I was able to do this with a line text (I think that's what it was). When you opened the device you could edit the field. However, I could not get that field to save with the project. I added the string valued of it to the parameter list, but when I load it always goes back to the default value. Am I trying to save the wrong parameter?

Use "standard text", and save <Str> in parameter and preset parameter list.

Sorry, I did not use "Single line text, I used "NewText" as found in ./surface Design/text. What is 'Standard Text" ?

CircuitDesign/CircuitControls.
newText is ok too, if you change the Disabled var to 0.
and it has the advantage that you can also use non-standard fonts, because the text is saved as a (editable) GO.

I can get newText to work ok but I cannot save it. Putting the str value in the parameters does not work.I tried using "standard text", but when I use the field and try type something into it, it displays only the first letter and follows it with a number. I can see the string data I entered in the attributes of the pad but it displays wrong. The format field is blank so I tried changing it to %20s, but then when I edit the field it says something to the effect of only allowing 1 argument. Any more ideas?

new text: did you change the "Disabled" var from 1 to 0?
standard text: format field is blank by default and shouldn’t be modified. sorry, no idea why it does

not display correctly.

The disable flag was 0 already. If I set it, to 1, then I could not change the string. This is what I would expect. There must be some other reason it cannot be restored. Is there an attribute somewhere that enables/disables restore? There is a restore level in the context menu of a pad setting 'restore in preset' or 'don't restore in preset' or 'don't restore after loading'. (from chapter 3). I am away from home now so I cannot check, but if the were set to 'don't restore after loading' would that stop it from restoring, or does this do something else?

forget about that 'restore in preset' flags, of course every var can be restored in presets by default. no, I think you made some other mistake, maybe forgetting to save the str in the ppl, whatever. see, whether you want a pot, a fader or a textmodule in the preset list, it's always the same procedure. and as you already got the fader working, I think you'll be successful with the standard text too.

Restoring of button state

I have a couple of buttons and check boxes used to switch the audio path. The simplest one (big button) on the surface surface is connected to a switch to turn the signal off (like a mute button). I have the value of the button in the parameters and in the preset list. All seems to work great, except....
If the button is on when I save the project, when I load it again, the button is still set but does not work. If press the button turning off, then again to turn it back on, it works. If I load the preset it also works, but not when loaded from the project.If I load the project via the sdk, I find that on load, the button val is some large random number like 3342523. But the button values limit is 0 and 1. Once I click the button it sets to 0 or 1 and all is fine.

The parameter list does have restore on project set so it seems like it should load the value. It acts like the displayed state of the button is saved and restored but not the val var on load. THe switches require 0 for off, 1 for on and anything else is off.A second issue (which is the same) is i use a check box to select either 1 or 5 from a 1-6 selector. The button is set to min 1. max 5, step 5 (I forget the exact values) and all works fine except when the project loads the val is random again so nothing is selected until I click the button.
Seems like this should be a simple problem but I have yet to figure out how to correct this problem.
Anyone know why this is happening?

Are the buttons referencing the circuit directly or are they connected to val-fields? I think it is good practice to use val-fields and store these in the preset list. Be careful with the val-range on these though. A thing I have noticed is that some things work different in SDK versus SFP.

I connect the button val to the switch control input (select the button on surface, click on val in pad list, click save, go to circuit view, click on the switch, select the switch control in pad list, then click connect) .
The vals are stored in parameters list and preset list.

You should create an identical non-surface (circuit button) which interfaces with the rest of the circuit (switch).
Then connect the surface button val to the circuit button val. The circuit button val should be stored in the preset list, not the other way around.

I think this is important because GUI is loaded last and you will thus have a broken circuit.
The circuit button could be a val-field too; though that is not so interactive when designing.

If you load eg the onOff button, default values are 0 and 1; no random values and such.
I think first thing to do should be to get things running the way it should be. meaning an onOff button should output values 0 for off, 1 for on (except you had defined different values).
after your circuit is running ok, you can take care of presets.

The button is set to min 1. max 5, step 5 (I forget the exact values)
step 4 in this case, otherwise it will not work.

Synth crash when loading preset list

My synth "Alchemy" (see in the device section) crashes when i load the preset list from the SFP window.
i'm having trouble with preset handling.
first, i got crashes when i was removing the synth from SFP. then (after searching the forum) i addressed the problem as being the existence duplicated entries in the parameter window: basically i think moved into the parameter window both the entries from the ENV module and the pot controlling it.
so i solved this issue but rebuilding the presetparameter without duplicated entries, and the crash when the synth was removed from SFP was solved.

but now i have this other preset issue: all the preset load/save are correclty working, the synth works, but when i save the preset file, remove the synth (or close SFP), reload the synth and load the preset list, SFP crashes and i even get a window blue screen. what can be the issue now!?!?!?!?

if only valid parameters are in the parameter list and in the presetlist only parameters of the parameter list
else if the presetlist is set up correct;
check what's checked in the parameters;
took me a lot of time to find out what the hell RP and so on means;
finally if right mouse click in the grey backgrounded top line you are able to sort by -> getting the full name;
maybe macros to record and playback mouse movements are no good idea;

after countless testing i think i have found the problem, which is related to the modular waldorf oscillators.
the waldorf pots were in the parameters windows and then in the preset parameters list.
the presence of those pots in the preset parameters list made the synth crash when i loaded the preset list!
now i have crated "dummy" pots linked to the waldorf ones and this seems to work

---

How to make a device with different panels

I tried several ways like using the close pad to close one panel and open the other in order to navigate from one panel to the other it doesn't look as good as other paged devices

Try using SurfaceGroups and connect your switching mechanism to the 'show' pad

After a few tries it worked, still need to master it though to get nicer results. I don't know how to open the windows separated ... they open up at once
How do i set the show var to another int ? beside 1 and 0 ?
and how to make the button var to do the same thing ?? routing is simple - now how can i make it work?

Use logic conditional IF modules is the easiest explanation. Say you have 3 pages & a 3-way button/pot (or whatever), then simply connect an IF module to each one. The first should send a 1 val when it receives a 0, the second will send a 1 val when it receives a 1, the third will send a 1 val when it receives a 2 & all three should send a 0 val when they receive anything other than their IF val.

Unloading Osc from synth

In order to save DSP power, is there a possibility to "unload" an OSC section from the synth? i.e. if a patch uses just 1 or 2 of the 3 osc, can i unload the remaining from the DSP?

DynVoicesOfParent is one option. Put one of these inside any module or circuit section & toggling it to '0' unloads it from the dsps (although not completely).

this works for effects?

sure, it works for any module or circuit of modules that's loaded on dsp. A '0' val = off dsp, a '1' val =1 voice etc etc. This is outlined in the 'Attributes' descriptions section of the documentation.

How can i arrange things in order to set the voices number of a synth from the SFP live bar???
for now i can just specify a number in dynVoicesofParents module (so maybe you can control it from the synth surface), but in now way i was able to "Link" the voice number to the live bar, like in stock synths...
How can i do this?

Connect the voice pads of all DynVoicesOfParent modules throughout your synth circuit to the voice pad of the main DVOP module inside the first circuit level of the parent module. Only do this for circuit sections which need to be polyphonic though.

i have just ONE DynVoicesOfParent module in my project, located in the uppermost module (the one which contain everythin)... is this correct?

Well, if you have your whole circuit located in the uppermost level of the main parent module then you probably don't need extra DVOP modules . They would probably only be useful in the majority of cases for loading/unloading modules or parts of the circuit from dsp.

i gave up the idea about unloading OSC, because they are located in the main surface...

This is pretty bad. You should ideally try to reconstruct the synth without using any modular modules *at all* & if you *really* need to use a modular module it should be used at circuit level, not surface level. The way you're building it will make it practically impossible for someone to build a new gui without having to reconstruct the synth circuit themselves, possibly from scratch.

the "problem" is that for the module i used i found no possibility to locate the module themselves at the circuit level, because certain text fields (wave selector for the waveosc, sample loading for the sampleOsc, filter type) are not linkable to anything. the M3 modules have a protected surface, they cannot be customized and for this reason they must be brought at the surface level

The atoms & modules you need are either in the Circuit Design folders or in some cases in the DSPModuleList. The tools are certainly there to recreate those modular modules.
You could set up some internal midi control to access the modular modules, but that's not an ideal way to do it for various reasons, so I wouldn't do it that way.

Polyphony causing crash

I built a synth but I can't remove it or shut down scope with it loaded if the polyphony is higher than 1, and for this synth it's a must. It causes the computer to reboot every time, any help

Are you getting any error messages? Does it behave the same in the SDK?
If it's only doing it when you raise the polyphony it could be a module in there that doesn't like being loaded more than once. Try deleting any modules you suspect of causing this and try it out then.

it was the stereo chorus module

If you want to lock the chorus module to one voice, you could put a DynVoicesOfParent module inside it with the voice count set to one (connect a range text to the Voices pad) and save the chorus module. Then if you load that into your circuit, it shouldn't give you any problems when changing the synth voice count.

right, that's a good tip for the future. It's really not needed for the chorus to load in more than once. Thanks for the info, I hadn't thought of that, it's strange with sdk, little things that I know but I don't apply to situations, or it doesn't occur to me to do that, I was thinking of that module (dynamic voices of parent) really along the lines of synth polyphony more so than with any device you don't want to load more than once. as it is I removed the chorus because it wasn't really necessary, I posted the synth already in the devices section, String Machine, if anybody wants a chorus it's easy enough to put one on the channel.

Async modulo arithmetic

Ok say i had 2 4 way switches, and a 16 position knob which would step through the switches like

11
12
13
14
21
22
23
24

31
32
33
34
41
42
43
44

i can probably do the first switch by (integer) knob value/4 + 1. but how can i get the remainder for the second switch. i would prefer not to make a 16 way switch !

You could use a pot & TableTextFader - Connect a pot to the sel count pad. Set the Count var to 16, then enter your desired values into the value array. The output would be the sel value pad. Fold it into a new module, export the pads & call it ArraySwitchEX or something. Also, you wouldn't need to 'make' a 16-way switch as such, as you already have the SwitchEX.
Here's an example module. Connect your pot up to the switch input. Val1 output will give you the 16 values that you mentioned. Val2 pad output is a string from the Val2 array that could obviously have various uses.
(ArraySwitch2)

That's much simpler. i forgot about how much you can do with the controls themselves.

Up until now I've been using RangeText & Switch modules to create switching parameter value arrays, but this way should be much better.

this worked very well for me. except the editing of the arrays is a bit of a pain. (can i edit in notepad?)
can these arrays be changed to have more columns? or should i add another array to the device, if i do will it work in tandem with the already existing value and text ones?

Haven't found a way to make the editing of arrays easier in the normal array edit window. We were working on a way to make editing arrays easier a while back, but it turned out to be quite unstable.
To add more arrays you could just duplicate the array switch module inside the Array Switch2 module, then connect their count & sel count pads.

basically thats what i was planning to do. i dont know how much these "pep" arrays use, but i doubt multiplying a switch like that would make a massive DSP hit.

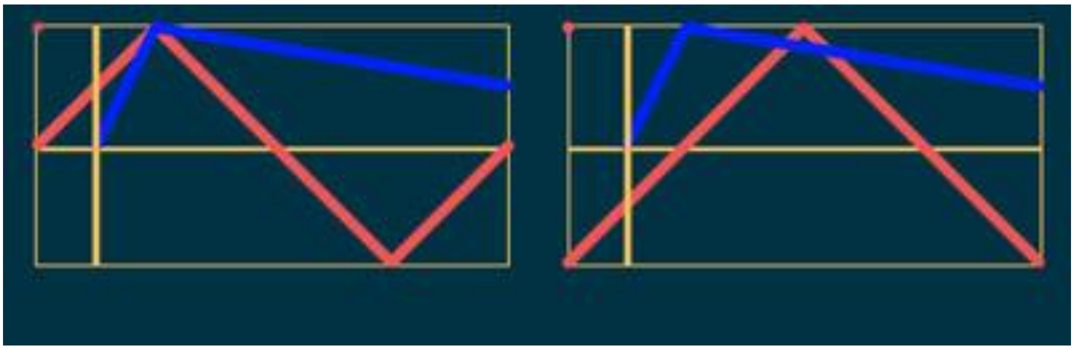
Duplicating those shouldn't have any DSP hit at all.

Envelope attack not consistent

I'm building a drum machine but I'm having a small problem. The envelope attack seems to be slightly fluctuating on it's own, so the kick will go from hearing the snap to not hearing it.
I have deleted the envelope panel so as to get 0 attack times, and the attack pad has only one control hooked to it which is the snap control of my gui. Any ideas would be appreciated.

it could be because at that low frequency the free running oscillators are at different positions in their cycle when the envelope attacks. For instance, the time period of 50hz is 20 ms so say a quarter of that for the rising of the wave from zero to max, is 5ms. if your attack is at 2ms you will still have a 5ms attack instead. worst case scenario with a narrow pulse oscillator would be almost the full 20ms late!
If you use an LFO as an oscillator you can make it start on the envelope trigger, and also you can change the start phase,

which might give more tweakability to your drum.
Heres a little pic. the blue is the envelope and the red is the wave(i used a triangle wave due to laziness)



Thanks, I was wondering if this was the problem and I even thought that it'd be good to have the osc trigger new at each gate but I didn't think of using an lfo, I'll try that. It was the only thing I could think of and the problem is really only very slightly noticeable at higher freq's, the higher is the better response.
Well, that didn't work, you can get a lot more snap though, too much really, by adjusting the phase. But even with the osc retriggering it's still fluctuating. Is this maybe just an issue with the envelopes?
I know that the EDS16i doesn't do this, is that maybe a custom atom they used?

rightclick -> attributes -> OnsameDSP ->yes

it seems better but it's still happening, any more ideas?

you just need to make sure the envelope and osc are inside that module. and both are onsame dsp.
if you are using one of the antialiased oscs, it will not work since the aliasing gives a delay between 0 and 1 samples
using the sine osc should be fine
and using the sync output of oscs is also fine.

It's working now, after putting the kick module to "on same dsp" then using the lfo osc instead of the sin it's working good. It must have been because of the low freq's, like Neutron was saying. Because the other modules are good once I put them "on same dsp". Right now it's working good, I just had to adjust the phase a bit on the lfo to get the attack.

Modulation muddles

I have a problem that I cant figure out based on my limited knowledge of the various modules and atoms and was wondering if someone could point me in the right direction?
I have a signal were the pitch is modulated by an LFO. I want both the LFO and the inverse of the LFO to be able to modulate the pitch - LFO modulation increases as bipolar pot increases the positive and the inverse LFO increases as the bipolar pot increases the negative. When the pot is 0 there is no modulation (hope that makes sense).
What module / combination of modules can do this? I have the LFO signal, the Inverse LFO signal, a bipolar pot and the mod input of a pitch modulator. What sits in the middle?

What do you mean by inverse LFO? Are you using the inverse module after the LFO - if that's the case - then you don't need that module at all.
You just need your LFO module and the "Mix 1" module - if you choose the [-2147483647 ; 2147483647] range for the Gain tap on the "Mix 1" you'll get what you want. If you use negative values for the gain parameter on the mix modules you'll get the inverse signal.

You were correct in your assumptions however, what you suggested wouldnt work for on OSC. Yes it would!
Basically, I want to be able to modulate pitch with a choice of 6 mod source:
Saw down, square, triangle, S&H, Noise and an Env (the bipolar mentioned in my initial post is for +/- mod depth).
In need the mod source to retrigger how can I get it to do this if the mod source is an OSC?

you should convert bipolar modulation signals to unipolar signals for sending to the mod input of a Pitch Modulator. To build a bipolar to unipolar convertor, connect a sync multiplier in series with a sync adder/subtractor. Connect a range text circuit control to the second input of each module & set the val of both range text modules to 1073741824.

Its way faster to use the Mix modules as mentioned above. You just need a single Mix2 module, like :

In 1 : Input
C1 : 1073741824
In 2 : 1073741824
C2 : 2147483647

The Mix modules acts like a multiply + adder/subtractor of X inputs in a single module (and with a minimum of DSP cycles)

I actually managed to figure something out shortly after posting that involved a few If modules and a rectifier (going into mix module gains). Haven't checked your suggestion Shroomz, about the sync combo but will at some point. I read in some other thread that If modules were a bit hungry? Is that so? Seems a basic function to me. My SDK doesn't show the DSP cycles within the attributes for that module.

IF modules are a necessary part of the toolkit & wouldn't be there if they weren't.

Problems viewing Insert Effect menus / Mod shells / Preset lists

if you're having problems viewing Insert Effect menus, using the Mod shells, using PresetLists etc within the sdk, then take a look at your Scope cset.ini file. For those who haven't looked at it before, It can be found under C:\Scope\App\Bin. You can view it with a text editor like Notepad. The bit we're interested in here is the StaticPath section. If you copy the relevant entries from this section into the cset.ini file in your sdk install, you'll then be able to do quite a few things that you previously couldn't.

Conditional logic for async

I'm looking for an AND function module that will output the input signal value if the AND condition is met rather than outputing high or low. Is there such a module?

From the stock logical booleans; And, Not, Or & XOr, you can easily make NAnd, NOr, XNOr & some Flip-Flops. Maybe that's not what you're after though..

Those are the logic operations I'm looking for but I was hoping that these modules would work with integer values. Not a problem though as I can divide values by themselves to get the logic 1.

I've done some different flip-flop latches this way. All async though. Make your DSP count

---

Any way to get Sin or other math function?

i just want to get a sin function with the simple math operators? or maybe with the FM operator.

You could use polynomial approximation, like taylor series :

sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - ....

You could optimize the coefficients with the Remez/Remes exchange algorithm - you will then get :

sin((pi/2)*x) = 1.5707963268 * x - 0.6459640619 * x^3 + 0.0796915849 * x^5 - 0.0046768800 * x^7 + 0.0001530302 * x^9

Remeber that x have the following range [-1 ; 1]

Or

Use the "fx LFO" module with the following parameters :
f = 0
Sel = 0
Gain = Max (2147483647)
WF = 1
RC = 3
RC = 3
P1 = Variable [-2147483647 ; 2147483647]
P2 = Same as the above

P1 = x, out1 = sin(x)

I used the above for the calculation of the width parameter in my reverbs.

That LFO method looks great! i will try that tonight

well that idea didn't work. i need to be able to input the signal i want to do the math on in realtime, and that only works with async.
yes a lookup table would be a great atom, as would "opamp", "resistor" and "capacitor" to go with "diode"
when im feeling a bit smarter maybe ill try that polynomial approximation. but i think it would be a mess

Well it updates for every 14 samples.
If you need sample based calculation then polynomial approximation is the right solution.
Here's an example of a polynomial approximation - (see Approximation Sin (x).mdl - Warp69)
I don't know what you need it for or the range of the parameters. Try connect the module to the Raw Tri osc and see the behaviour of the module.

This is great! The wave made by "approximation" looks a lot better than i thought it would after reading about that method. Tests so far indicate it will work for what i want to do!

Saving MIDI CC's in a project

I'm having problems getting midi cc's to save within a project. I always have to load the midi controller preset for any devices I have made, it's a small thing but I'd like it to save in the project.

If it saves in presets, then it should save in projects too ... no ?
There is an absolute rule i discovered though :
- don't connect several graphics together. It bugs midi cc functions.
Also try to see if connecting a controller pad between the pot/fader changes something (even though, if your CC work fine in midi-cc-presets, there should no need for this).

I've had feedback reports of this problem, but I think the reports must refer to the fact that my devices don't recall midi CC assignments automatically without using the midi-CC-presets (which is something that stock CW effects can do – you don't need to save a midi controller preset for them to recall the CC assignments). My devices DO seem to work fine here with midi-CC-presets though (whether inserted or standalone).
Another issue I noticed when comparing my insert FX to stock CW ones is that mine don't pass the midi CC data out of the inserted effect & on to it's host's midi output (whether I put midi I/Os on the insert effect or not).

By the way, have you checked the project checkbox in the "module's preset parameters" : i remember i ran into something weird once by changing some options, but it was a long time ago and don't remember quite well , it could be something else.

Yeah, I've got all those checkboxes checked apart from the Noah-related 'default' one.

try to go back to default by unchecking all RS and SS .
may be it doesn't change anything, just to try...

it's strange because when you reload the project and open the midi assignment it shows the controls assigned, but when you then open the controller settings it isn't there.
I tried connecting my controls directly to pads instead of to the other controls and that doesn't seem to help either. I unchecked and rechecked all the boxes too but it didn't make any difference.

Using samples in SDK

Does anybody know how to go about using samples in devices built with SDK?
I've looked at a load of modules and atoms but there doesn't seem to be anything that will allow the use of samples (as far as I can tell).

you can use modular's modules for that...
it's certainly not the most beautiful solution, but it works....
1) you can load modular II/III modules into SDK projects
2) in the project tree, if you move "modular view" into the "panel" subtree, you will see (and can use) the modular module surface into the panel once you open it!
this goes directly to:-
3) you can load the modular II/III modules: SAMPLEPOOL, SAMPLEOSCILLATOR and then, form the surface, you can load samples into the pool, and then move the sample you have loaded into the sampleoscillator module.
in this way you can manage mono waves, and you don't have access to multisampling, but it's better than nothing!
moreover, you can load only the SAMPLEOSCILLATOR module, assign a button to the surface to the "Load sample" button of the module itself, and load the sample from the synth surface without having the modular view on the surface itself...

I was aware that Modular modules could be used but wanted to avoid this if possible due to the fact that not everybody has either Modular 2 or 3.
If this is the only way that i'll be able to use samples in this device then so be it but I would prefer not to.
I've seen there are a couple of atoms that have an input called filename such as the playback DLL atom but there doesn't seem to be an atom that is the equivalent of the modular sample pool module so not quite sure what would go into the filename input.

Maybe you should have a look at the wav_oszi.pc dsp module

Original colours in graphics

So I made a nice front panel for my new device, exported it as 32 bit TGA, opens fine and everything but its only greyscale. I'm sure someone has asked before but i searched about and couldn't find anything. I remember from before just colouring them in scope, but is there a way to use the colours you designed?

i did it a second time, but removed the gradient that was there first, now it worked fine

To summarize you removed the gradient setting it as a plain color? So the color doesn't affect the loaded image while the gradient does?

I am not sure exactly why, the first time i tried it it came out semi transparent and greyscale I forget what order i did things in, after that i thought i had saved the file wrong, but that's not the case.
Second time i had a gradient on there and i changed it to a colour, it came out greyscale when I added the bitmap.
Third time i clicked the leftmost icon on the top of the gradient designer title bar, it removed the gradient and it left a neutral grey background, it accepted the bitmap fine then. it would probably work with colour designer as well.

Midi Drum voice control

Can anybody tell me how this module works?

It looks simple: Has the Midi Input and also a Midi Note input along with mute. I'm presuming the gate will only output when the Midi input receives the same note as the Midi note input?
What value should be put in the Midi note input for it to trigger? If for example, I want C3 to trigger then what is the value that needs to be input? Is there a way for working this out for the other notes?

the key text automatically shows the note, you just need a knob that goes from 0 to 127.
so when you input note 60 it will show c3
what i did so i wouldn't have to adjust each one was to use a int adder with the note in from the previous module on one input, a "1" on the other and the output going to the next higher one.
then delete the knob and everything except the key text. (it might not be called key text my scope machine is off right now, its one of the text controls that works with key names)
that way you just change the lowest note and all the others will follow along.

I had tried that that though - connecting a text field, setting it to C3 so it fed a value of 60 into the voice but still nothing.

well you have to use note text, or if you use a regular one just put 60 in

It was note text . Also, can anybody recommend any reading material about creating a BPM synced delay?
It appears I was right last night - I cant get the drum voice to work.
I've got the Midi plugged in and the note number set to 60. The drum voice has the gate out plugged into the AD Vintage EG which then goes into the exponential attenuator along with the sound source.
Nothing complex really, its all straightforward but it doesn't work!

did you plug the esync thing from the EG back in to the voice control? they never work without that.

It's the esync connection that's the issue.
2 options:-
1 - connect the EG ESYN output to the DVC ESYN input.
2 - connect the DVC gate out directly to the DVC ESYN input.
Where more than one EG is in use, you can use ESync adders or where preferable, use step 2 above.
There are other options involving the Gate ESync module.
Use whatever method works best for your particular situation.

Tried both those options with the DVC 1 E before and still couldn't get it working - just double checked - same result.
Also the DVC1 doesn't have esync and that doesn't work either so don't think its esync. Do I need to do anything with the mutes? Presuming its not muted by default upon loading?
The DVC is rcvg MIDI but doesnt seem to be outputting anything whether the specified note is input or not.

Try loading the MIDI Drum Voice Control module via your sdk FileBrowser window from the 'Circuit Design/Voice Controls' folder instead of loading the basic atom from the 'DSPModuleList' window.

What a fool for not loading it from there in the first place. Works straight off! Wonder why the atom doesnt work though? Never mind.

Device works well in SDK but in scope none of the controls are visible

I'm sure its something obvious i forgot. i copied the knobs from the panels of the individual atoms then deleted the panels.
if i drag a new knob from the file browser to the panel of my device it does show up in scope, i checked the Go tree properties and both the new and old knobs have the same properties, they are also on the same level of the tree on the project explorer.

Check out the 'dimensions' window.

Yes, everything looks like its where it ought to be.
Oh, it looks like it needs to be protected before it can work. now i just have one other thing
Pepremoverouting: pad not found cannot remove routing
something to do with the inserts i believe - nevermind i fixed it

had the same thing once. solve it by going back to a previous saved version. :)

Building a device using the Dynatube cabs

I'll restrict myself to four dedicated devices at first (mesa, marshall, vox, fender), so I need to do some IFs based on the project sample-rate to load the right module (44.1K or 48K) and then I guess I would need to use the dynamic voice of parent-thing.

Is project sample-rate a val easily accessible?

If I succeed in building this device, would I in anyway impact on copy rights if I would release it in the Device-section? I don't think so as you would need a card registered with DT for it to work in the first place, right? I seem to remember a similar discussion on use of the FleXor atoms.

Why would you want to access the project sample rate from your device instead of the project sample rate window? If you want to toggle your device between different sample rate modes all that's really needed is a val switching system for the divisor pad of any delay-based range text modules that you're using so that they read out the correct delay time values for the current sample rate. Maybe I'm missing the point of what you want to achieve?

Yes, you are missing the point a bit (but good tips!).
Dynatube comes with two versions of each amp and cabinet module, one for 44.1KHz and one for 48KHz. They both internally oversample and Dynatube won't run in 96KHz.
So the problem is: my device needs to know what the project sample rate is set to so it can load the correct module. running the 48KHz module @ 44.1KHz rate, or wise versa, has audible effects

So you're saying that Dynatube automatically loads the correct one depending on the currently set project sample rate?

Yes, that is what happens. Also, if you have sample rate set to above 48KHz, a box will appear soberly stating that DT won't load under these settings. So there is definitely some logic applied at loading.
If there is no real way to look up sample rate and ULLI, then maybe one would need to construct some sample-counter or so. That is if there is some real-world clock available.

There must be a way to access the current global/project settings.

I'm also looking for something like that. VDAT have the same functionality - can display the chosen sample rate.
I have a DSP module that can determine the sample rate with a loop, buts its a Sonic Timeworks DSP file and I can therefore not distribute that file. But I would much rather have a module that have access to project settings so it can determine the ULLI settings.

I've got a couple ideas at the moment. First thing I could try to build a digital counter using logical operands.
Another idea would be to compare two sines. One controlled by a frequency val and one controlled by a time val. If the time variables are anything to count on, the one controlled by the time val should change it's output based on the set sample rate.

Ok, I've succeeded in building a preliminary device which lets me utilize dynatube cabinet models as well as set the mic-position. Also, the correct model (44.1K or 48K) of a cabinet is used based on sample rate.
However, for reasons I don't understand, the device crashes Scope when I delete an instance from a project, but otherwise works efficiently.Any ideas?!
Anyway, I thought I would like to elaborate a bit on my sample rate detection. My current solution only knows how to detect 48K and "every-thing-else", as in not 48K.
It works and does suit my needs as of now for this particular device, but I am pretty sure I know how to create a detector that will be able to do 32K, 44.1K and 96K as well, and which will be far from as ugly as my current solution
In SDK, If you drive an OSC with a ranged text controller, let's say one that says "1 Hz", you can be sure the OSC stays at that pitch at any given sample rate.
Enter the async2sync atom.
If you where to drive the OSC with the same ranged text controller via the a2s, the OSC pitch will be scaled depending on set sample rate.
Knowing this, you now have a means to compare a static value and a value that responds to sampling rate - the relation should be able to tell you what it's set to.
It would also help to know that at 48K, both values share a 1:1 relation, as in they are in "sync".
Interestingly, this tells me that the hardware-interfacing is built around 48K operation as its default rate.

For those that won't bother constructing one themselves, I hereby share my device. Now detects 32K, 44.1K, 48K and 96K.
I ended up using a Ranged Text controller, async2sync and three Ifs, and Or and Nor.
As for ULLI I would have no idea :/. I think you would need some sort of loop-back for that. Not very transparent at all.

Some information :
Every time Scope starts it has a module loaded called WordClockPipe - the top module in the Project Explorer. The tap "MFreq" determines the Sample Rate :

Value 0 : 32KHz
Value 1 : 44.1KHz
Value 2 : 48KHz
Value 3 : 96KHz

You just copy that module inside your device.

I'm also able to read the ULLI setting, but unfortunately not in real-time - I'll have to reload the module if I change the ULLI setting - it then shows the correct value again. I'll have to look more into this.

That easy was it. That makes my module pretty excessive.
I think Scope needs to restart on every ULLI-change anyway, right?

As stated before - I'm quite impressed with your approach.
Well - here it is: SampleRate.mdl- shows both the sample rate and ULLI setting.

Very nice trick & observation!
I still fail to find a difference in values though, when changing SR
That makes my module pretty excessive.
Not at all. The small drawback of the wordclock module is it sets the project SR settings to the given setting, so it may change your current project setting (i.e. from slave to master like in my case).
Btw, here you can find a module, which reads out ulli only :
ulli-readout.zip (NOT AVAILABLE)
Still not realtime, but in DP/SDK you can right click on it to read out the actual value from cset.ini.

You mean my SR-detector isn't working for you?
It sometimes gave me a hard time getting it to work as it could be very inconsistent. But when it does, and you save it, it works here, at least the updated one. The old one would work in SDK, but not in SFP (for 44.1K) for some reason.
If you handle the wordclock module correctly I see no real danger really. Though seeing as controllers are bidirectional, I see how one could accidentally set something instead of reading.

To make the connection read only/unidirectional, just insert a diode.

The point with the wordclock module is -> it sets SR and master/slave without having any connections done. A Diode doesn't help here.
For a short test: save WC module, change SR setting, load WC module

Yes, I discovered the WC module acting like this, so I went back to my SR.
I wonder if having a controller pad hooked up to the main/project WC-module via store/connect will work?
I solved the 44.1K issue with some logical gates, so now 44.1K output only goes 1 when the others are 0.
All in all it is a very frustrating device to build, and I suspect it is because async values only update "when they have to".
Thus try hooking the sync output to an osc. Things should start to work. If not, replace the ifs that don't work.
When everything works fine, just delete the osc, move out of the sub circuit, check if it works and save it.

Mouse down pots

Does anyone know how I can make a mouse down pot which has a mouse down action similar to a momentary button/switch?
The purpose is that I want the pots on a device to send a momentary val of 1 when they are clicked (not moved) & back to a val of 0 with the release of the mouse button. Any ideas?

- load a button

- go to the var section and change function value to 4.
just try them out, its just 5 or 6 you can choose for various other tasks.
default is I think 2.
the same with the pots, where you can change the rotation handling.
up/down - rotate in cw, etc...... (Hifiboom)

But I already know how to get the momentary action on a button.
What I want to do is get the same momentary action on a potentiometer....
In fact it doesn't even need to be a momentary action necessarily... if I could even just get a pot to send a single val of 1 when it's selected with a mouse click, that would be ideal.

What's the solution here? I know it's possible to create a val when the pot is moved, but how do I do it when the pot is clicked? It must be possible because the sdk recognises a pot selection.

what is exactly your intention, you want a knob, that can control something by the knob movement but also does some kind of selection?
so when you click f.e. one of 3 knobs an lcd gets changed to a new parameter?

Hifiboom isn't too far off the mark, except I had more than 3 controls in mind (and no LCD). One of the main uses I can see for this type of control is for a parameter related paging system, where a page will update depending on the selected control. I've tried adding a button but then the button is selected instead of the pot. You can make the button smaller in the centre of the cap and select either pot or button, but this isn't as intuitive as getting both click and movement responses from one control.
I can also get the paging to update from pot movement, but then you need to alter a parameter to get the page to update. This would be OK for an LCD type parameter update, but it's not ideal for what I had in mind.

i thought about something like the lcd thing also some time ago.
where you have various knobs for different parameters and the one lcd display, that for example shows "size 30meteres" when you move the sizing knob on a reverb and "decay=20s" when you move decay time etc....
cleaning up the gui a bit.....

The orbitone syn chrome used to have that if i remember right, and it was one of the very first synths.

Delaying in samples

How can I arrange to delay a signal by samples like in the mixers? I am trying to build a parallel compression tool, but I need to delay the original signal to line up with the one going through the compressor.

Use the "200 Delay DSP" module.
Delete the surfaceinterface and panel from the module, so you have a clean module - then use the standard "range text" for the actual delay value in samples.

you got flexor? maybe a Sample Delay will do the trick...

What warp said does work, just like the channel delay, but it's not working for what I want to do.
I even tried it in scope, if I run the same signal into 2 channels and invert one the signal disappears, so I know there is perfect alignment, this works. However if I insert vinco into one of the paths I can not get the signal to cancel at all, from what I understand the compressor adds an amount of delay and once you compensate for that the signal should cancel again, but it's not working, has anybody had luck with this at all?

I never tried that, but I understand vinco like any other compressor as a device that 'colors' the signal.
This is probably not only a frequency change but also one in the phases per frequency. .
So how would you cancel the signal by only adding a delay, since it does not do the phase shifts like Vinco ?

Stardust is correct. Vinco is too coloured to be able to fully phase cancel it with a parallel delayed copy of the signal. (If you set ratio to 1:1 the signal content shouldn't differ to the original.) That said however ***most inserted*** effects in Scope have a common delay of 14 samples as I discovered when I built .

LA COMP, which is designed to measure inserted effect latency as well as ext' send/rtn latency. If you spend some time testing effects with that device, you'll quickly notice that send & return paths to external (standalone) Scope effects commonly deliver less than half the latency of the same effects when inserted in insert slots. Commonly 5 samples if I remember correctly, but I'm not in the studio, so can't double check that. Try it for yourself & you'll soon see the emerging patterns.

Wow shroomz, I have completely overlooked this gem provided by you.
Thanks a ton, this helps really when you microtune a effect in SDK.

here's something kind of strange, I just did a little test, I ran a sine wave through scope into 2 tracks in cubase, one through vinco first, the other direct. I didn't use vinco as an insert though, I ran straight from my adat in to asio out. In cubase when I select samples for the ruler and zoom in, there is no difference between the 2 starting points, they line up perfectly. Why/how would this be? There should be a slight delay of the track running through vinco, no?

No, not necessarily. As Shroomz said (a little bit hidden) the plugins behave different, if used as insert or directly in the project window. It can also change, when loading the project or optimizing dsp allocation or loading it on a different system or ..

Shroomz, tell me if I am using your device properly. I'm not using the insert at all. I connected the send out into vinco, and vinco out to return. Then ran a test signal and clicked auto, it ran some numbers, then locked at 1, this means there is just a one sample delay with vinco? In your release post you said that the signal needed to be able to pass through dry without bypass, I'm not sure if this is the case with vinco or not, does all this sound right to you?

Whenever I para-comp a drum-bus via Vinco, I find it much easier to phase-invert the dry signal before delaying it. When you hear the pumping/imploding effect, the tracks are aligned.
Beware that RMS-mode needs a longer delay than peak-mode does

---

How to fill up an array

Has anybody a tip for me how to fill up a large array to create my own wavetables?
To modify already existing wavetables does not work for me.
I tried to figure out any method with the Array Viewer but wasn't successful until yet.

I just can edit the values if I double click the values in the array. An array list appears and the entries can be edited each by each.

Do that 2048 times for example, that takes time

the only wavetable player is the waldorf one. you can open them in wavelab and edit them. they are labelled wt**.dsp in the dsp folder of your install.
i don't know what format they are but still fun to play with.

The wt**.dsp files are just modules with an integer array of 2048 entries. You can create them quite easy. Also you can copy all data from an wt** module to your own module by connecting both arrays.
But what exactly do you mean by edit them in wavelab?

if it is a plain format of 2048 values in a row maybe a renaming to raw and editing in an audio editor (like wavelab) is the way to go ?

if these were raw wavetables in one form or the other, then someone would have come up with an editor long time ago. To me they look as random as any other encrypted module

I am also interested in creating my own wavetables....
I would even write a small wave to wavetable.dsp software if I would know the format of the array storage
I also don't know how to fill up an array automatically step by step ....
doing 2048 entries by hand really sucks....

so you got a solution?

Only for creating wavetable arrays with WAV samples. Not for filling arrays with specific HEX values for midi or anything like that yet
Maybe an sdk user only tool/module would be the way to go with this? To go down that line would require the module to be as close to perfect as possible, but it could maybe even be open to a certain extent. Any thoughts?
Ok, well here's a preliminary tip .:- take a look at any WAV > array related device (or devices which contain them) & go from there.

now do you talk about the wav oscillator or a wav-2array converter.... that really converts a wav file into a .dsp file?

Well, if we had an sdk module/s that fills an array/s based on a 'dropped in WAV sample', then that would be a good starting point, wouldn't it?

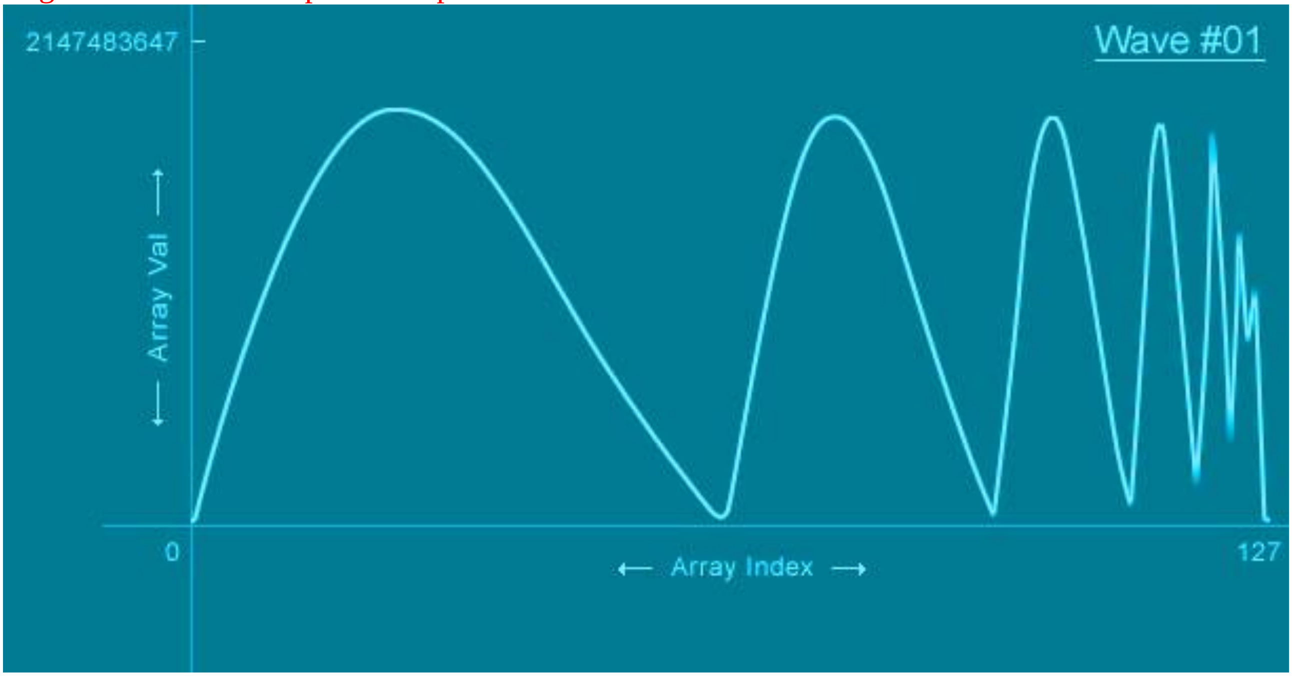
would say the problem. is not to fill an array with data whatever;
problem. is to fill it with data that makes sense in realtime!
for example:
if blowing the audioIn into an array;
where to start where to stop recording?
without an editor for manual selection its sort of random

the problem is to fill an array offline with wav file data in SDK. - and i guess shroomz talks about that possibilities
that would at least allow us to build something like a virus wavetable oscillator for mod3
the advantage are that the wavetables are inside the mdl file, so you don't have to distribute 128 wavefiles with your mdl file
and you get the waldorf like walkthrough different wavetables....
I would be completely happy if I can address the array(WT) offline.
BTW tgstgs a delay is more or less a "realtime array

We're building some new array tools & have made some good progress, but I know for sure that the Waldorf Osc is very unusual in the way that it reads data from the arrays. Our understanding of it isn't yet complete, but from what we understand so far, it will be a **lot** of work just to put together a *single* wavetable.
anyway, the WT arrays of 2048 values seem to be split into 16 blocks of 128, but there's a *lot* more to it than that...
The Waldorf Osc reading strategy appears to be 16 blocks of 128. The major problem is that each block of 128 contains (what looks like) 7 octaves of waveform data. Each of these Octaves is represented by a half-cycle symmetrical waveform. In terms of the array structure that works out at sizes of 64, 32, 16. etc up to 16*128. The real issue with generating compatible wavetables is that it seems like each block of 128 not only includes octave sections, but is also morphing to the next block, hence making wavetable creation for these Osc's a relatively difficult task without a piece of software which auto creates the desired morphing based tables. If there's a way to do it in sdk, we've still to find it.
So where does this leave us? Well, so far we've managed to generate a sine wave.
Based on what we've learned it should be possible to put other symmetrical waveforms in there which means that sample data can to some extent be embedded in devices, but I wouldn't expect full blown wavetables from us any time soon.

I know that it would lead to difficulties.
maybe we should simply put this somewhere into the SC device/sdk wishlist.
It`s def. on top of my personal wish list. a flexible oscillator, with very high quality pitching algo based on wavs like the wav-oscillator but with some sort of anti aliasing pre processing. and very high quality pitch shifting.
For example it sounds different if you take an analogue recorded osc waveform and pitch it in your wave editor with highest quality or if you do the same with the wav-osc in realtime.
and it should have a sync feature which the wav osc is missing at the moment.

here's a visual representation of the first block of 128 from the Waldorf Osc's 1st wavetable (Wave #01). Each wavetable has 16 of these strung together to make up an array of 2048 with subtle differences between them creating the morph. This first block creates the first 4 of 64 morphing stages in Wave #01. Hope this helps.

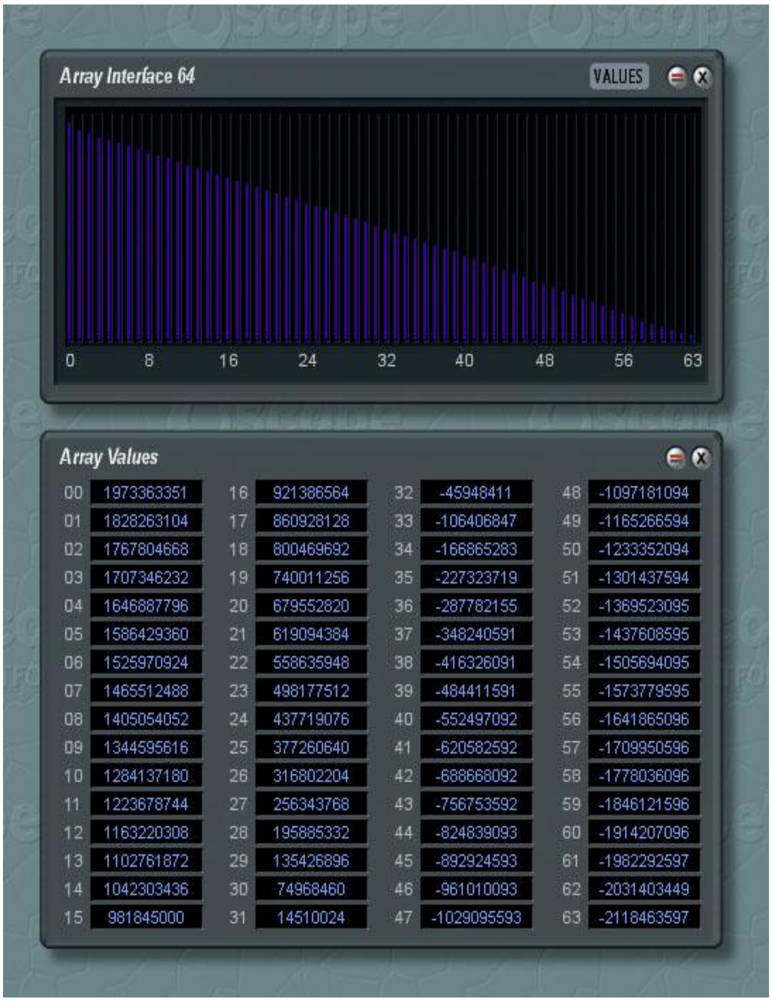


Shroomz

thanks for the info, shroomz,
don`t know exactly but it looks like the stages seem to represent some kind of pitching also
If you look at the image I posted above... well the Waldorf osc uses the first 64 samples of the 128 block for the bottom 4 of it's pitch octaves, which is interesting to say the least.
theoretically if i read through a wav file specification I could build some sort of wav to waldorf wavetable converting tool by programming a little tool.
but the question is : do we really want this?
Well, that's the point really. It was more of an exercise into understanding a bit more about arrays than anything because I can't see anyone here including myself making a better job of creating 64 wavetables for that osc than the ones Waldorf produced for it themselves. What's already there is everything you need for that osc & has that classic PPG sound.
the waldorf oscillator is some sort of digital sounding oscillator and due to the small amount of samples per waveset it has inferior quality also intentionally like the original stuff sound digital also.
btw how did you visualize the array in that nice way?
I`m also working on a phase distortion waveshaper that can change the sound of an basic oscillator (sin, saw, tri) in drastic ways, still keeping the quality high.
My major plan with all this wav shaping and wav based oscillator stuff is, that it should give new sound options with out affecting the sound quality.
The Flexor draw waveshaper is a good approach, but its still not yet fixed regarding the jump point

and so it delivers non-perfect results.
btw, regarding the wavetables we succeed to fill it based on integer values.
But what about converting the values into and float array?
As you can connect a float 128 array to the wshaper module and transfer basic waveform through the waveshaper quite nicely.
I did this by manually inputting different values into the 128 array slots. quite time consuming, but it can give some nice new waveshapes.
so we could build a basic wavetransformer around the wshaper.
I guess I understand how the waldorf oscillator works generally.
the first 64 samples are used for the lower octaves through the fact that downshifting doesn`t introduce aliasing
lets say the first 64 samples represent c3, then c2,c1 and c0 is done by downpitching the first 64 samples.
c4 is represented by the next 32 samples as pitching one octave higher should half the the wavecycle... and so on
c5 16 samples
c6 8 samples
c7 4 samples
c8 2 samples
that's just how i image the general workflow, I did not verify this.
the higher octaves representations may be antialiased to deliver better pitching quality.
that's why the look of the waves may change in the higher region.
so basically the first 64 samples represent the wave in its pure form, the higher regions the pitched versions.
just a quick try to explain what's going on, but its not unlikely that I am wrong on this.
That would also explain the inferior quality of the waldorf osc in the down region, where it has no "pressure" and "depth" through a weaker downpitching.
it would be awesome if SC would build an oscillator with 2048 samples in lowest pitch (c0) and the other pitches are represented with fractions of this base pitch : 1024 (c1) and so on, So you could use the high quality pitching of a wave editor to generate the octave pitch steps..and between the octaves another performance algo blend s through the pitchings in a satisfying way.
some sort of flexible and quality wave based oscillator.

Been trying out this array draw tool today. We've got it connected directly to the wavetable input of the Waldorf Osc, so it's now a wave draw Waldorf Osc. Seems a little unstable at times but it sounds completely insane



NOT AVAILABLE YET !!!!!

Wolf's Scope development tips

Order of connecting pads

The order of connecting pads is very important, when working with buffers or async->sync conversions. You can easily try this with midi modules. Connect pad A to pad B, and pad B will take over the value of pad A on device/project load. Now connect pad B to pad A and you'll see that pad A takes over the value of pad B on device/project load. If you are experiencing, that a device doesn't output what it should on loading it into a project, consider this fact.

Preset list sharing between devices

You can let share several devices (i.e. mono & stereo) their preset lists. Just copy the whole device - i.e. the stereo version - and remove all things not needed for the mono version.

Precondition is that you already thought about that in your "most complete" version and have only parameters of pads in the preset list, which are not considered to be removed in a cut-down version (i.e. the mono version).

This works for modules inside a device as well. Simply create a preset list for the module, copy it and you can use one preset list for all copies.
This can be taken to a 2nd level:
If your preset list just contains parameters of separate controller pad modules, which are connected again to the circuit, you can change the whole circuit without changing the preset list (and therefor keeping already done presets).
From another point of view you then can exchange presets even between a chorus and compressor (example makes no sense, but htink about it).

Preset list and inserts

If you want to have parameters of inserted plugins saved with the preset list, always put this preset list to the most uppest level of your device

DSP saving routines

Here is an old one :
Copy the module “Voices Of Parent” into the module you want to unload from dsp.
If you want to unload the parent module from the cards dsps, just change the “Voice” parameter to zero. Setting it back to one reloads the module’s circuit onto dsp.
This helps a lot to create dsp saving devices, when done intelligently.

Keep preset list, while changing circuit

Redoing preset lists is always a pain in the ass.
So here’s a tip to avoid that :
Always connect pads, you wan’t to add to a preset, to a separate controller pad and use this one for the preset list.
This way you can easily change the underlying circuit without having to rebuild a new preset list.

Keycommands for the pad list

In case this was not clear, you can move the pads of modules to any side:
Select the pad, you want to move and hit T(op), B(ottom), L(eft) or R(ight).
This helps cleaning up the design.
The first exported pad will be listed first, if more pads are in a row.
Some more keycommands for the pad list:
I(nput) sets pad to green arrow
O(utput) sets pad to red arrow
E(xport) exports the pad (rename pads before exporting them)
S(tore) stores the pad
C(onnect) connects the pad to the previous stored pad

Track pad connections

Being able to track pad connections, after you've forgotten the routing :

Export pads and other connections to graphic elements and connect them from the exported pad to the graphics elements. This way you'll see the connections in the connections window (direct connections from some pads aren't shown there).

Of course this is not a good idea for the topmost module.
If not needed for connections to other modules, you can hide the pads by selecting it and hitting "H"

“Integer division by zero"

Either while project load or while doing something this alert might pop up and is always frustrating. If it doesn't stop to pop up, there's usualy a dynamic signal like a LFO fed to it. If you have the project running as slave, switch off the master clock. If you have it running as master, switch to slave. Both ways have the goal to "disable" the dsp processing. This gets you rid of the reoccuring alerts and lets you remove the reason for it. Basically you should save the async division module with the value 2 for in2 and use that from now on.

Midifying devices with no midi in/out

•load i.e. TransientDesigner into SDK/DP.

1. •also drag the Midi Controller Linker & the Midi Channel Linker Module into the project
2. •connect "CCon" pad of the channel module to "Ch" pad of the contr. module
3. •connect a midi source to the "Midi" pad (green) of the channel module
4. •connect a midi destination to the "Mid1" pad (red) of the channel module
5. •select the Midi Controller Linker & the Midi Channel Linker Module
6. •click on "move into"
7. •click on the TransientDesigner module
8. •save it as new device with its new midi in/outs

Control elements into insert FX

okay I figured out how to build an insert fx....Now my question:
how can I add control stuff into the surface....(insert effect surface)?
when I use the default surface it works, but when I try to drop faders or knobs to the insert effect surface, I cannot drop control elements from the filebrowser.

you need to be in surface mode not circuit mode (toggle @ top RHS of your project window). Also, you should use the ones located in \Surface design\Controls\....... , not the ones in \Circuit design\Circuit controls\.....

Actually, I didn't articulate myself very well there. What I should really have said is that I'd recommend that you use the modules located in \Surface design\Controls\ ...... as these are there in that location specifically to use as surface controls as opposed to the ones you're trying to drop in, which are really there in the circuit design folder to use as internal trimmers etc in circuits (although they will work as surface controls as well). You also have a much wider range of controls to choose from in the surface design folder. Hope that explains it more clearly

enter inside the device (doubleclick) at the level where the surface module is visible in the circuit, then go in surface mode and drag your controls there.

Step Sequencer DSP

The module names give you a little hint as to their use. Just guessing here.............

SSQCON10.DSP ------ module related to building a control sequencer
SSQGAT10.DSP ------ module related to building a gate sequencer
SSQMNO10.DSP ------ module related to building a midi note sequencer
SSQPAT10.DSP ------ module related to building a pattern sequencer

---

MIDI – Byte2Extract Problem

Im pretty new to the SDK, and someone asked me about making a simple device to allow only specified CC's while blocking all others.
It seemed pretty simple, and I used a module Byte2Extract module to accomplish it.
However, It seems not all the values are getting picked up by the byte2extract module. Some get dropped, and it depends on how quickly the values are sent.
I guess it makes sense as the midi stuff is mostly asynchrnous, right?

I am just wondering if there is a way to improve the responsiveness of the module, or if it's just the cost of doing business in the asynchronous midi realm?

I think some midi values being dropped when sent quickly is to be expected from midi controller (CC) data in general. It's not a fast or high resolution spec, so it just can't cope properly with fast manipulations.

Fixing MIDI CC assignment with SDK

A few question about MIDI device programming with SDK:
1) Is there a way to fix midi CC assignment with the SDK ?
This would help me to link my home-made device to be controlled by an external MIDI controller.

1.I'm sure there's several ways of doing this. One way would be using logic with 'byte extract' & 'if' modules
Another option could be to assign a midi CC to a circuit level pot & connect your surface pot to that (the pad connection between them will be bi-directional). I can't get on the sdk right now to double check this train of thought, but off the top of my head, I think that should work.
I think the best option (currently) is simply to assign a midi CC to each surface control (pots etc) via the pull-down midi control assignment panel in the control ranger. It seems to work fine here.
Select the Vars section of the padlist, select the control you want to assign a midi cc to, then click on value field for the 'CtrlNr' Var parameter, press F2, enter a cc no. & hit enter.... done! If you want direct access to the var parameters to alter their values from your control surface, you need to right-click on them & select 'create pad'. Then when you switch the Vars page of the padlist off, you'll see your new pad waiting to be connect to whatever you want. If you've figured that out already, you're doing quite well

---

Empty module project required

I would like to build up my first very own module for modular but as far as I worked so far I could only achieve some synth or effect structured devices. Could anybody help me about that as this would help me to release my first module before I could provide a complete synth device...

You can open an existing modular 1 module and start from it to make new modular modules

---

Building a hypersaw oscillator

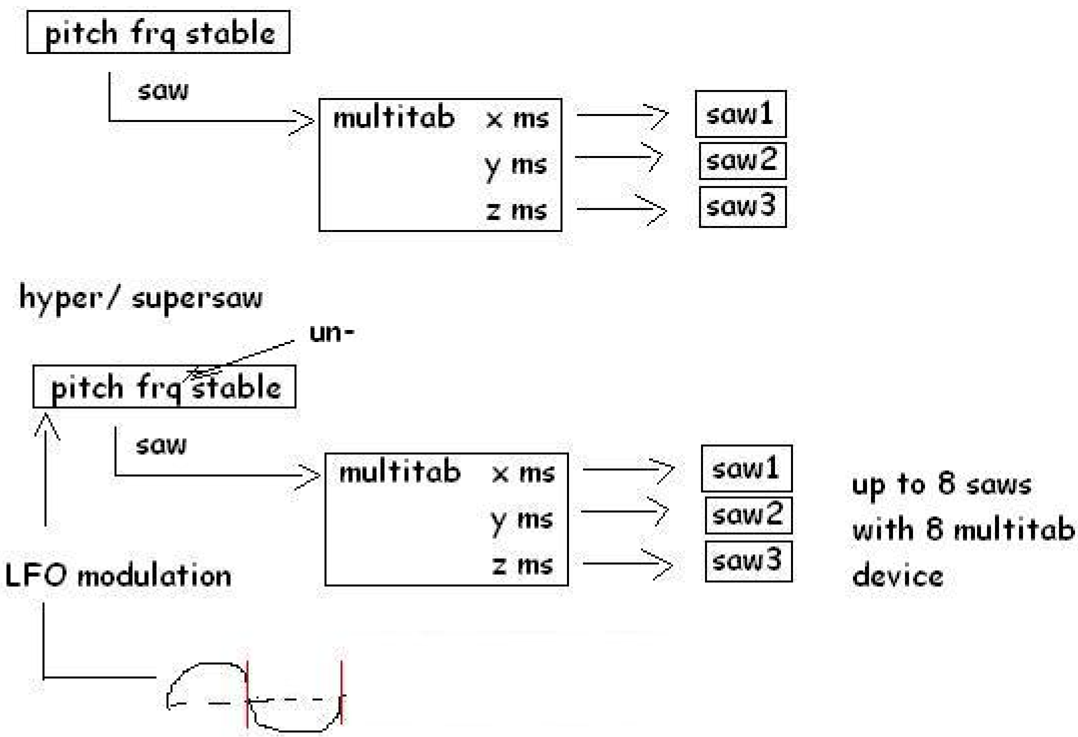
while building my synth I may have discovered a possibility to build a hypersaw oscillator out of just one saw oscillator.
first one raw saw wave, then I blend in the two hyper_saws resulting in one 3voice saw - at the dsp-cost of only a bit more than one oscillator.

for sure it should be possible with mod too -
the trick for the hypersaw is to use an lfo to pitch modulated the saw, depending on the delay time the saws are differently pitched.If you carefully setup the lfo modulation amount and the lfo speed, you get the supersaw.

the logic is like following,

when saw2 is stable at y ms,
saw1 and saw2 are unstable
when saw1 is stable at x ms,
saw2 and saw3 are unstable
and so on......

up to 8 voice supersaw is possible with one multitab delay.
i don`t know if an lfo takes the same amount of dsp as a saw oscillator, but at least with more than 2-voice hyper saw it makes sense dsp wise, if an lfo is less dsp-hungry than an oscillator.



Step sequencer

i'm building a synth device in SDK and now i like to integrate a (tight) step sequencer for triggering notes into the design. - i spend several days trying all the different sequencer dsp files but have no clue how to get em to work it's all in the logic. Google or wiki search for logic functions, logic counter or shift register & you'll come up with plenty of info. Some of the nord modular sites are interesting in this respect (advanced understanding of logic circuit implementations in dsp-land over there). Beyond that you'll need to actually think about it & experiment. It IS possible in the sdk, I assure you!!

back in the day, i made one for the "micro" synth
it is just a ramp wave which is compared to 16 equally spaced values. and when it is equal it outputs the next step - you can alter the odd numbered step values to get swing.
I cant remember if i had to use any custom atoms, but it was only DP back then.

How to make a synth

I have always been end user of the platform and I am curious how do people make their own synthesisers and the graphics for the synth?
is it as simple as to start with the modular II or III and build up something?

attaching the atoms to complete synths is basically similar to the modular II III stuff, as the whole scope platform is some sort of big modular construction kit
But you need to get in a bit deeper to get used to the sdk.
So basically I`d say, it gets more complex and bit more difficult but less limiting compared to modular.
If you are familar with modular and feel limited by the possibilities the modular is offering, the sdk may be worth a try.
Basically for what the sdk can do, its very easy to use. But "easy to use" here really depends on your skill and background, I think.
a good amount of basic mathematical understanding will help you alot, and some basic understanding in programming won`t hurt too.
Things you don`t need to know that indepth for building a modular patch.

Modifying existing devices

is it possible to import into SDK existing device (even creamware stock devices) and modify them or use them as "templates" for new devices?
for example, is it possible to make a single device using, for example, the standard gate, the comp and the EQ provided with SFP ? i imagine a window in which you "import" these three devices, chain them, make a new GUI and save your new device...
for sure you CANNOT open cw stock devices...
the basic manual is quite good for a rough understanding, the rest: "learning how to use the modules/atoms" is up to you.
There is no real explanations for the atoms, but most of the stuff you can find out for yourself
When you drag i.e. Compressor M into the workspace, you won't be able to see the internals, but the device surface will show up. You can rearrange the graphics. In that regard you can use standard devices in your own. No need to design the compressor from scratch.

Building GUI / Device problems

I have built a simple delay. How do I bring all the control surfaces I want to use to be able to build the gui? In the quick start guide they used just one control surface and dragged it up in the tree and then it would show up when going to build the gui. I want to incorporate the controls from many devices so how do I bring them all together into one gui? I hope that's clear.

basically there are two ways:
(1) move the GUI elements from the basic atoms to a global surface
as you described above, you do that by moving them in the tree
(2) or you drop new elements onto a new surface and connect them manually to the modules via store and connect.
Personally I prefer to use the dsp_module_list where the dsp modules come without any sub_gui, so you have to build it from scratch.
The mdls with small guis are specially useful when you are designing stuff where you need to try out many parameters quickly.

Also, how do I make it so this device can be midi controlled?
And when finishing up, to get a usable device, do I just fold it all and then save that as a new device or module?

don`t forget to protect.....if you dont want to share the device design with all of us.

I am not getting a surface to edit at all when I move the surfaces up in the tree, I can click on my device and open them, there are 7 surfaces I'm combining, but nothing shows when I go into surface mode. I tried moving just one surface but even then it doesn't show. What could be the problem here?
I will try to use the store and connect deal but I'm not sure exactly how that works, I'm sure it's the easiest thing, but it's all new to me. So I can open the default device from the menu and size it then add all the controls I want right? Connecting them to the proper place via the menu. Do I select the pad click store, then select the device I want to connect to, select the pad and click connect? Sorry but I haven't had a chance to check it out as yet.
How about the midi stuff? How do I put a midi in and out into the device and make sure all controls can be controlled by midi?
I guess I protect the device first then save it right? I see protect in the menu when I left click in the tree on the device, just below something about XTC, is that the right place? On a sub menu for something, can't remember what it is right now but it's the only place I've seen protect anywhere. Although anybody could figure out this device easily enough.

It sounds like you haven't built your device inside one of the basic empty modules provided. If you go to your Circuit Design/Basics sdk folder, you'll find various 'empty' modules. Put your device inside one of those like the 'empty insert effect' module, then inside the new module you'll have a 'panel' which will give you a blank canvas to build on. Alternatively, you'll need to fetch a desired panel module & put it inside your main parent module connected to a surfacegroup module as it should be. Either way would work. To physically change or move elements of a panel for a basic CW sdk module (if it has one) in surface mode, you need to be inside the module on the same circuit level as it's panel & surfacegroup before you jump to surface mode, otherwise you won't see it in surface mode. Hope that helps a little.
To better protect your device once it's finished, do as follows:- right-click on the main parent module of your device in the project explorer, select 'Delete Old Ciruit GOs', then right-click again & select 'Protect'. Now right-click on your parent module in the 'Project Window' & select 'Save as new module'.

Thanks, I have not built inside a template, I just started building from the new project, however I have the gui working fine, I just opened a default panel and went from there, it seems to work good inside SDK.
I really want to have this thing midi controlled though, I'm totally stuck on that one, what to do?

I recommend you opening the "emtpy synth module". Inside this you can seee some basics to conect midi. Depending on what you wanna do you have to load a mvc f.e for synths or another tool

like a midi time clock, ...and so on .... have a try..... and report back..
the sdk is learning by doing.
what do you want to control exactly with midi, tempo?

I just had to go back and do over my gui, as Shroomz said, I should have started with a module template, so I put what I had inside the empty effect and now I have everything working fine, the only thing is that when I bring the effect into scope all my text says "new text" not what I had typed, how to fix that? All controls work, they just aren't labeled properly.
I am building an effect, I just want a midi in and out on this thing so that all controls can be controled via mid, cut offs, delay time, feedback, everything.
It seems to be working now, I wasn't using graphical text, that was the text problem, and I got a message explaining the midi as well. The thing is I can't find the midi modules anywhere except inside the synth template, so I took them from there into my effect, is that the only place these modules are? I'm talking about the midi control changer and midi linker.

Frequency band separation techniques

I thought this subject might be an interesting one to discuss because there's more than one way to achieve band separation & it seems like the separation technique you choose would depend purely on the application (number of bands, device purpose etc). I've tried quite a few different methods now in the sdk which I'll detail if you like. I've had good results with all of them. I can see the benefits of certain methods for particular tasks, but thought it would be cool to discuss it & see what some of you guys think and have to offer on the subject. There's always the issue of cycle/memory cost as well, so that affects the method (module in the Scope paradigm) selection for particular tasks. What methods of band separation do you think are preferable & which do you prefer for given tasks? Ok, that's maybe a big question, but I think it's probably not & actually relatively simple for certain applications. Any thoughts?

I have no idea how to implement/build fir-filters... Is it possible with sdk?

I think there is even a fir atom in the sdk package....
fir: finite impulse response filter, basically done with the delayed signal added to the original signal.
iir: infinite impulse response filter, basically done with a delay feeded back & added to the original signal

a lowpass is a mathematical technique to smooth discrete values. quick discrete peaks are high freq. audio information.
fir uses a dedicated amount of delayed signals to calculate the new values.
iir uses the technique of recursive programming to calculate the new values.

By broad separation I take it that you mean creating crossovers using configurations of HP & LP filters? Or maybe in a HP-BP-BP-LP setup?
You could also use a network of LP filters (6-24dB), sync inverters & 2>1 adders to create a nice separation circuit up to a certain No' of bands. It works on the theory that if you feed 2 LP filters with the same input signal, set one to say 300Hz & the other to say 3kHz then subtract the 300Hz filter output from the 3kHz filter output, you create a bandpass filter with a pass band of 300Hz - 3kHz.
I haven't tried the FIR approach though. Would that not be a bit much tax on the old dsps?
I guess there's always the parametric eq network as well, but I haven't investigated that as far up as 31 bands or anything like GRAPH EQ.

When you using lp/hp/bp filters, remember that the filter frequency is the -3dB point. Therefore put to identically dialled filters in series for a good and easily usable -6dB point.

you can build your own filters or combine existing ones but don't invent the wheel for a second time;
simple lowpass filter:
good vibes + (good vibes - 1sample) = cutofffrequency samplerate/2

should be normalised to get the average value
goodvibes(t) / 2 + goodvibes(t-1) / 2 = cutofffrequency samplerate/2
so an add2N could be used in that case.

You could also connect the LP & HP outputs of a multimode filter to a x-fade module, getting a new filter type which has a controllable curve. In the 0 x-fade position the new filter will be a notch or bandreject filter.
Just to add to the comment above, the described method produces a notch filter which works well with zero or low resonance. To achieve a notch filter which works well at higher quality factors, try phase inverting the BP signal from your multimode filter & mixing it with the filter's input signal. This actually produces a notch filter which works well at both high & low quality factors

---

How to link controls

I am building a stereo effect and have separate controls for left and right side but I want the option to link them. What's the best way to do this? I can link the controls through a button but this just links them permanently is there an adjustment I should make for the button or what?

Ok, couple of tips which don't take into account all scenarios:-
1. Where possible, connect your surface level controls to 'controller pads' at circuit level.
2. You can try using switches to create uni-directional links. For example take a 2-1 switch connecting one control to input 2 & one to the output. When set to '0' (input 1) the switch will then break the link between the controls & link mode will be off. You can expand on this a lot.
3. You can make control connections uni-directional (so that they don't get feedback) by sending them through a module such as a switch (as above in 2.) modifier or even a logic module such as a diode (logic/conditionals).
btw, you would need to set up stereo links between all available parameters for the 2 controls, so if you have user adjustable control curves, min/max ranges etc, then those would need to be part of your stereo link circuit as well

fold your fx structure
build a mono block
export all pads to the upper fold
copy the the folded effect block
link all exported pads on both foldings via cables...
-> a no switch stereo device....
btw I recommend teaching yourself folding quite well if you are interested in building bigger projects. Otherwise you easily loose control over your circuit through the time. Complex circuits can easily blow up with 100s of atoms.

I want to be able to have the controls separate for each side or linked though, this sounds to me like they will all be linked permanently, or maybe I'm missing something.
I see already where the folding really helps things out

exactly, the description is for a fixed stereo device....
for sure you can add switches inside the mono block connections....

if you have 5 pads on both blocks, you could load 5 x 2in-1out-switches and connect block1 pads to switch-2ins and the outs to the block2 pads....
let switch-1ins open
then you connect all switch boolean inputs (0,1) together
Then you connect all block1 pads to controls (5)
and block2 pads to a second number of controls (5)
and a button control to one of the switch boolean ins (1,0)
now if you press the button (setting 1), all switches between to circuit blocks closes the connection and both blocks will act the same simultaneously.
if you click the button again, the connection is cutaway and you can control both arrays of controls independently.

I got it working.
I couldn't find any 2 x 1 switches so I used the 4x1 and deleted two inputs, are there supposed to be 2x1 switches in the kit?
I had to use asynchronous switch except for in 1 case where it wouldn't work, kinda strange.

Try the Signal Switch EX. It's a very versatile switch which I'd say is probably the most common one to use for either sync or async signal switching.

normally you should avoid to use bigger modules for smaller tasks,
for example if you mix 2 channels, you should take the mix2 and not a mix8.
If you don't, that easily can eat up the dsp power.
Especially on polyphonic synth designs where every voice eats up the power for a complete circuit. That multiplies over the voice amount.
f.e.
if you build a synth structure on overspecificated structures/atoms, you could end up using 3/4 of a dsp instead of 1/2 a dsp for ONE VOICE(monosynth).
Now if you make the synth 8-VOICE polyphonic the difference is 6 vs 4 used DSPs. So 2 full dsps difference and what a waste if you keep in mind that the dsp power isn`t cheap!
But with the async stuff like the switches and other async atoms its not that much important, as these are quite low on dsp load count.
Actually its unimportant if you use a 4-1switch instead of the 2-1 switch.
But my general recommendation for circuit building is: use the smallest module that is enough to do the job in your circuit.

But I don't see a 2-1 switch in my package, it's just not there, so I use a 4-1 and delete the pads and adjust the control to just 0 and 1, is there supposed to be a 2-1 switch?

Control knob problem

It seems that when I click on my control knobs they will quickly go to centre position then back as I keep moving, how to stop this? The faders work fine but the knobs will centre as soon as I start to move them.

Ok, here's what I have found, if I am using the movestyle where you click the knob and then move horizontally it seems to work good (movestyle 1 I think) but with any rotating style the knob seems to jump or want to always start from centre (or reference centre position first?). It'd be nice to fix this, help if you can.
simply set movestyle to different values for different controlling behaviour.
I think the range is 0-4.

I am doing this but at all settings the knob wants to jump to centre as soon as I start to move it, except for movestyle 1 but this doesn't allow you to rotate with your mouse, you move the mouse vertical for this, it's a small thing but I know that I should be able to rotate the mouse.

I should mention that everything works fine in SDK it's in Scope where the control jumps

---

LFO filter sweep

I want to set an lfo to sweep a filter between the values I set, so I want to set a high and low limit on the filter and sweep between that. The thing is that the lfo likes to go from positive to negative and overiding the limits I set so it pushes the filter past where it should go. How can I make it stay on the positive side sweeping within limits?

lfo_out * 0.5 + 0.5 = range [0;1]
You could use the Mix 2 module for that - remember that 0.5 is 1073741824
If you need another range than [0;1] then scale the values 0.5.

Regarding using finished devices and modules

i discovered that you can actually use the existing "stock" synths and FX from SFP and chain them to make a new device.
Is if it is possible to gain access to the control surface... not to edit its appearance but to actually USE it!! if i connect the stock gate and eq and save a new device out of these, it works in the common SFP window, but i'm not able to edit the settings!
i even tried to insert them into the SDK module "blank effect" which has a blank surface, or to drag the SFL FX into the surface itself... but still i cannot gain access to the parameters once the new device is in the SFP window...
is this possible or it is forbidden and i should spend no time in using existing SFP devices, since they cannot be actively used??
the same is for modular II and III modules: i can drag/drop them into SDk and use the connections, but how can i then edit the knob values once i have exported the device??
using modular in the surface would permit, for example, to use the "mod waveplayer OSC" to build a simple sampler...
i don't know if you made this, but i copied all the DSP files from the SFP directory into the SDK one: in this way i can gain access, for example, to dynatube (which i own) atoms . there are indeed plenty of atoms to choose from... moreover, i also noticed a "akai sample playback module", but how can one actually LOAD a STS or akai program into it ?
i noticed there are some ".pep" script files, which appears into the file browser (even in the filter). they have interesting names... but are they available into the SDK!?

you cannot use finished devices as basis of your creations.
You have to build your own devices based on the smaller atoms that are in your sdk folder or on your sdk cd-rom.

O.K. you dragged some modular modules, so far so good.
Now how do you build a surface? Just put knobs on the surface and connect whatever you want to it from the inside... You only have two mouse-buttons to choose from
Every inclusion of ready made modules by others should of course - at least I think so - be in

agreement with the involved parties. . You have enough toys to work completely without the modular stuff. It's better that way, I think.

Starting with SDK

Have a look at Neutron's examples and praise him.
This helped me more than reading the manual....at least in the beginning

my recommendation is to simply read the delivered manual, there is a first start project also.
Its pretty easy to understand what's going on and the manual is written quite well, but it misses some parts when you get a bit more experienced.
But if you simply want to learn the basics its quite good for that.
You can learn this quite easy, I can assure you that almost every other development environment that I know of has a much slower learning curve that the sc sdk.
my recommendation:
(1) read the manual once completely (you can do this on a half a day)
(2) read the manual again and try out the things that are explained carefully

---

Monitoring a fader level – set up text display

it seems very simple at the beginning, but i have problems...
the simple circuit design is:
test generator -> Noise gate (NGaterR4)
i have Vu-Meters connected both to the test generator and to the noise gate output

i have connected a standard fader to noise gate threshold and i want to monitor, with a standard "range text 0dB" the threshold level (so that i can see it on the control surface)
the vu meter says that the test generator level is -24 dB also at the gate out, so i have a check that the threshold level must be set to -24 dB.
now, moving the fader, i can open and close the gate, but the "range text 0dB" NEVER says that the gate is closed at -24 db... instead, i have values close to -8 db!
i think the problem is how to monitor the fader values or, which is the same, how to match the fader output to the -96 ... 0 db scale, which is the gate threshold input range...
how the hell should be the fader set up to achieve this simple task?

you have to maybe make the proper adjustments in the vars of the text display so that it will read correctly

---

Saving a stock device with MIDI assignment

i have just discovered that if you load a device, for example the MINIMAX, you can midi-assign all the controllers to remote control the synth with your external midi control surface, save the device with a new name (minimax MIDI.mld) and, voilà!, when you load this device the midi controller are ready to go!
you can set up a midi assignment and save it under the "MIDI" folder in the preset list.
later, you can recall the midi assignment by loading the appr. preset in the "MIDI" section of the preset list

---

LFO panning

i'm trying to add automatic LFO panning to my delay
the problem i encounter is this:
if i simply connect an LFO generator to the pan modulator, and i set up a saw or pulse LFO waveform, i get "clicks" when the wave go abruptly from a maximum to a minimum.
how can avoid this? i tried to use "dezip" but it doesn't work... basically the sound is panning anymore!
how can smoothly achieve panning LFO with saw or pulsar wave?!?

Use a lowpass filter after the LFO and before the pan module.

Pots movement and range

I'm playing with the graphic system in SDK right now and it has been years since last time I was playing around - but I seem to have some problems with the movement and parameter range of the pots.
The only range I can get is the standard one :
Range : [7 -> 8 -> 9 -> .... 4 -> 5]

Are you talking about values or graphics?

Movement in fact.I have looked at almost every device for the Scope platform and every single one use the standard movement/range for pots.
I will of course use the standard movement/range for the width pot, but what about the in/out pots?
The standard moverment will always start at 7 o'clock and end at 5 o'clock - no matter what graphics (or value range). And will jump from max to min/min to max through 6 o'clock. Atleast on my system.

Yes, but how do you get exact 6 o'clock if there is no image for that. As far as I remember the standard animations don't have 6 o'clock.
If you make new graphics & have lets say all positions of 360 degrees (What I meant was 12 hours)as an extra frame, the animation simply may start with the frame of your choice and end with the frame of your choice.

I'm quite sure that it's a graphical = pot-animation problem, not anything to adjust in sdk
you would need to render a knob with as many steps you want, starting with a keyframe of start position and end position. number of pictures should correspond to the control steps you intend to use, i.e. a standard knob would be 128 images with a knob rotation from 7 o'clock to 5 o'clock. that's the only way your graphics will interact precisely.
For your 12 to 9 animation you would need a key frame at 12 and one at 9, which represents 270 degrees. frames you render in between is dependant on how many control steps you are going to use. stupid example; if the AMS lfo dial was stepped, you would render 10 images. if you were to use the full CC range, 128 images should be rendered.
as for interaction with code i don't know if you are going to filter CC values or use the full range, so if you were to filter out 1/3 of the steps, you would need to filter out 1/3 of the frames in the knob animation as well (i mean render just 96 steps).

You can download the pots right here:
www.relab.dk/Width%20Pot.mdl
www.relab.dk/Input%20Pot.mdl
As you can see - the width pot does not have any problem because it already use the same movement as the standard one.
Where the input pot has problem because it should use another movement range - 12 -> 9.
How do you fix that?

After inverting the range (control ranger) it was fine too. Working beautifully as (I guessed) they should.
This works perfectly here. Really the movement is exactly from 12-9. I just had to invert it, because it was from 9-12 before. There must be some stupid error, something overseen. Exchange the project & I bet it'll work. Or I am totally stupid. It works exactly after your clock-demonstration 12-9 - wonderful. Nice knobs b.t.w.
To make it clear: The node (black marker) starts at 12 (on top of the pot) and moves around till it comes to 9 at the left side

first of all warp, your front end looks wonderful. congratulations.
I see your problem, but maybe this mousepointer relocation simply isn`t implemeneted in sdk?
maybe its a workaround for you to use the var setting "Movestyle" set to "1".
so the values change by moving mouse up and down. and not circulating. This seems most intuitive, for me at least.not perfect but maybe a solution you can live with.
Else i am afraid you have to contact SC directly to fix in a new movestyle type. Or some kind of GUI_Range_Min / Max settings.

yes it is the animation that must be redone. I use it often to make a wider range or even reverse control.
You need your frame 1 at 12 o'clock and a last frame at 9. adding a centre frame is a good idea (for all types of potis)
I used this a lot , and other frames techniques on the FP mixer potis, to centre 0, -12 or -6 db, they are all different animations.
In the FP 104 / 106 mixers, the Mono Master output has not only a different range, but has also a higher resolution around -6 dB.
It is fast at extremes, and get slower approaching -6 dB, so it is easy to select -6 db.
In a graphic program, that would be changing the timeline curve at the center (or putting more frames at center, or elsewhere).
Or I don't get it, are you talking about the Scope "round movement" or vertical/horizontal movement, and the fact that it is not visually syncable with the pots themselves... i don't think you can in scope right now.

Building devices using modules in DSP Modules directly

I found a window called DSP Modules, it seems that most of them are functional, so can we build devices using modules in DSP Modules directly? And how to connect controls in surface window to these "DSP modules"?

part1: yes you can use them,...if you know what they do.
part2: Building surfaces is either using existing ones or modifying existing with the GO editor.
Some are documented with txt files in the DSP folder.

How to create old style presets management

i need some help to create the old presets style for a device.
I need the old style, because i'd like to have the option of saving the device.

i've found a way using some old 1.0 module

Average of values

do any one can tell me how to calculate an average of values in SDK 4.0 ?
i mean for 1000 samples :
(sample1val + sample2val + ... + sample1000val)/1000 .

You would use normalised adders for that.

Pages within a modular module

not pop-ups, but pages within a module which function as the rest of the paged devices on the platform doI meant multiple pages accessible by pressing page 'tabs' or buttons. This would mean a relatively small module could be packed with extra controls without enlarging the size.
I'm thinking if it's possible it would need to be done in such a way that the patch connection sockets remain in place while a paged section above or to the side of the sockets allows skipping through pages of controls (pots/sliders)

Preliminary tests tonight showed that one can put page 'tabs' on a module surface, and have them work on objects.

Developing modular modules

I've got a couple of theoretical questions for the Scope developers here:-
A:- Would it be possible to build a single modular module which is triple the width of a normal one, but still snaps into position in the Mod Shell with LHS horizontal & top/bottom vertical alignment?

B:- Would it be possible to build a modular module specifically designed with internal modulation routing which allows on board patching within itself?

re: A - Not sure about these widths, I have not seen such 2U or 3U width modules yet.

re: B - check this mixing or switchingmatrix from j9k. The one with mixing is 8x8, and the switching one is huge. These matrices can route all modulation sources to destinations, without reconnecting stuff and their settings can be stored in presets...

i've made a try and it works.
For horizontal size it must be in multiples of 250 px, for vertical it must be multiple of 30 px.

it is possible to do 125 px surfaces, but if opened from standard modular shells it will be used as a 250 px surface, so, it will be not possible to place 2 of them in the same row.
It is possible to create a new modular shell, with different snap values, such as 125.
I'm working on it...

For Scope DP, you can't enter any values that are with decimal points, therefore, 62.5 pixels shows up as just 62 (and that's what the window tries to use).
It's easy to change the ModularWindow to 125 x 15 (half of current Window grid) - I just made one here, but haven't tested it yet.
Of course, you'd also need some smaller modules to try it.....

Changing polyphony with MIDI CC's

I just love Saturn - it has a good sound, and it's DSP usage is light, but you can call up presets with polyphony settings that vary from patch to patch. This is a great forgotten feature.

The only structure that allows this is the older style preset list. When the preset list object was re-designed, this feature to assign voices dynamically was removed (probably because the DSP re-allocation which needs to occur when you change voice count requires the DSP to be reloaded, which can take up a lot more time than would be desired between preset changes).
However, the older style list is available, and could be installed in a device....but there are other 'problems' with this list, mostly in that it stores every single parameter of every object inside, which simply means the preset list becomes quite large (as large as the original device, in my experience). Also, this list data is stored with the synth device, not as a separate object like the new style lists, so your synth ends up being quite 'bloated', and therefore also takes a lot longer to load into a project. Also, each individual preset calls up and changes many more parameters, so changing from one preset to the next just takes longer anyway, not including if you change polyphony for the preset.
I agree, it was one of the features I really liked about the old preset list, but the trade-offs of not using the new list for the old aren't really worth it, in my estimation.
Still, it's too bad the new list didn't include the possibility - you certainly don't have to use that function if you don't want delays from voice reassignment.

One version of SDX synth allowed this (voice in presets, new style presets) I removed it for several reasons, and because of user requests.
It can be annoying when browsing through presets, because polyphony changes make loading time longer. it can be insecure if you already have a fully loaded project. poly on preset might cause dsp overloads etc.
however, it can be an option on polyphonic device, having voice parameters to load or not. Having voice controllable by midi CC just need a control, not preset parameters. it is one of the options i intend to put in next revisions.

There is a simple way - you attach a parameter (knob or slider, etc.) to the voice count module inside. Since this would be the same as the registered cwVoice parameter, I wonder if a user changes this, if it gets updated in the Project display window? And if it doesn't is there any negative result?
This could be just another parameter like anything else, so could be stored in a preset. I'll have to try it to see if there's any negative issues with the system management software.

you can do a "bypass parameter" switch on the surface of the device: it remotes a switch with two parameters : the first one is not in preset (not even in the Tree), only the second one is stored with every preset. the switch button itself is not stored in presets.
by parameter, think "control" not module pad.
I used this on all my devs "tempo preset bypass" switch and in Echo35 "bypass pitch mod load on/off" parameters (go have a look at the manuals or at the demo to see how it works in real situations).
In one word, you need two knobs that are connected to a switch that goes to the relevant module pad. only one of each is in presets, and the other is not.
For midi cc, i guess it should be the same as for echo tempo parameters (ie, the displays are, or are not, registered in the tree).
You shouldn't need to put the voice in the list, just that single control that allows to change poly: it will recall voice only if the user has pressed the "yes i want to recall polyphony" switch.
The only problem i see, is that a knob can be sensitive and mismanipulation can have the user go from 1 to 16 voices in a millisecond without the possibility to think/understand what happened in case of crash/dsp overload. But that's the user's problem.
Myself I prefer typing a value: it can be dangerous if someone types a val > 16 or double clic in it (make poly 2,4 millions!) but at least it is nice in live too (you have to be fast with mouse and keyboard and have good eyes, but that's doable between 2 songs).
I hope i explain clearly. It is the only (and best) technique I know : i would use it on many parameters if it didn't require switches and surface buttons to load on/off presets.

it seems to work fine with no problems (I'm unable to test XTC mode at the moment).
Great - thanks, Mehdi! Sometimes the simplest things are overlooked (and also there's always more to learn!). Yes, I also used this technique in the RD Drum and Q-Wave (with double sets of parameter pads, some for preset or not), so I know it quite well, and the switch idea was just what I was thinking also!

all you need to do is to store the DynVoicesofParent pad in a preset or attach it to a knob or slider. It can then have a CC assigned and stored in a preset.
You could also include an on/off switch in the connection between the knob and the DynVoices module if you wanted to "shut off" the voices control with presets.

Yes, this is what Mehdi said about using the Bypass switch. I could even see adding an additional non-programmable switch for the Bypass switch (and make the Bypass programmable)! That way, you could program whether or not the Bypass is active (if you wanted to have the presets recall voice count sometimes and not others).
And then, of course, you could have a bypass for the bypass for the bypass switch, etc., et.

Circuit modelling in SDK

If I were to want to model an analogue or hybrid circuit component by component in the SDK, how would I go about it? Is it possible?

I've read that Flexor modules were built using basic building blocks which are available. Has anyone else got their teeth into this who is willing to share info on the concept?

http://www.musicdsp.org has plenty links. If you understand the maths used there, you can apply it to Scope. I'm not saying Adern uses these, but reading there will give you good insight on how filters and oscillators are made on a low level...

To be more specific, what I'd like to get my head round is how to achieve the realisation of a specific variant in a specific circuit in the SDK. The circuit I have in mind seemingly uses digital inverters instead of op-amps & this variant alone gives the circuit a very specific sound. Any advice on the implementation of such an idea in Scope would be fantastic.

nah, it's not that complicated...
The common source MOSFET is the basic circuit in Anderton's Tube Sound Fuzz. A similar circuit is used in Fender's Stage Lead, and EH "Hot Tubes" pedals. This circuit can produce very convincing tube-like distortion if it is carefully designed. The commonest way to do this circuit is with the CD4049 or CD4069 CMOS logic (yes, logic) IC. It can be misapplied by biasing it into its linear region and to function as an amplifier.

I was kind of hoping an SDK circuit modelling guru might drop us all a little building tutorial based on an example circuit implementation or something

no need to personify this hypothesis by looking for a blueprint in which you can simply replace some variables with your own items...and hope to get along without even having 'thought' about your project in detail......otherwise you would have noticed that circuit modelling is hardly applicable, as the respective part is operated beyond it's regular specs.
The acoustic result is a smooth transition into (close to perfect) symmetric clipping, so almost no even harmonics are generated - but a significant amount of noise.
It's not depending on variations of the circuits operation - it's either there or not.
A ladder filter (f.e) would be a different case, as it depends (mainly) on non-linearities of it's transistors.
Even then it's not the transistor itself that's modelled, but it's significant parameter(changes).
The trick is to find out which are the significant items and to which math rule they obey - that's all you need to know to start circuit modelling.
Of course a quality signal source plus a (non-SFP) scope and a multimeter will help a lot on more complex items that are not covered by a (single) manufacturer's doc.
yet programming is more about detecting operation patterns and rules - not so much about hacking examples….it's up to you to decide if you want to face the challenge or become one among the average.

It sure would be cool to have all the necessary components as modules called actual electronic component names with variable value settings for each. I think J9k has been thinking & looking at the SDK in this way to a certain extent, considering some of his modular module designs.

just pickup any analogue/digital electronics book, and start by building an ideal/practical mathematical model of the circuit you are trying to recreate. an ideal model will be simpler, so start with that, then you can start to add the practicalities and slowly build on this.
mathematical models in electronics are usually very precise, once you have something close it'll mostly be a matter of tweaking your model until it gets close enough (or sounds good enough compared) to the original. for a really precise model, you'll need pretty precise (and also pretty expensive) measurement equipment. some of it can be rented though, for the final tweaking.
also it might be a good idea to model the circuit in a circuit simulation program like spice or electronic workbench or some such first, so you can run simulations on the whole/parts of the

circuit, in order to better understand it's behavior and what is going on.
sadly it won't ever be as simple as just dropping a few blocks and getting a perfect recreation, it'll take some time, patience and hard work, but it's definitely do-able.

one of the things that will kill you right off the bat is x+y=(x+y)-1 above zero in the mix2 mdl. you have to correct for it if you want to do any sort of integration(like the trapezoid mdl). but this only comes into play in a few circumstances.
I think j9k means that in the total 2^32 numbers, there's 2^31 negative numbers, 0, and then "2^31 - 1" positive numbers.
There's no center position here in 2^32 or 32bit land, but the 0 is the first positive number. Try doing some proper maths with that
1+1=1
2+2=3
10+10=19
2147483647*1024=1023

http://graffiti.virgin.net/ljmayes.mal/circuittheory/Rcmathmodel.htm

---

Activate and assign MIDI controllers

how to enable assigning MIDI-controllers to surface elements...

you should add the values from the padlist to the parameters list.

Presets saving parameters of inserted effects

Now building own effects I experienced that my devices only save Parameters of inserted effects in Preset list, when they are used as inserts themselves. When using those presets outside inserts, which were made in an insert, they work!
Has this something to do with the insert-effect-shell? But the creamware multi-effect is an insert and saves values even outside inserts...

You need to save your device as a .dev file (in the Save As... box). If it's a .mdl file it won't save insert presets. I had to find that out the hard way. I also think you need to make sure the insert slot's "path" pad is saved in the parameter and preset parameter lists

---

How to set default pop-up x/y coordinates and link panels

I believe it's all set when you Save the device. So, open all popup surfaces, position them where you want, close them, and SAVE the device.

but something seems to be messed up concerning that point in my configuration. They always seem to start where they want + they have different positions inside "surface-mode", "circuit-mode" and inside Sfp, strange ... I will look into it again.
It was more important to find out, how to link them - this is solved now.

Did you register the X and Y coordinates for each surface in the Parameter List?

I will try that again... I remember playing around with registrating and kicking them out again, which worked better somehow... Maybe I forgot a value.
Btw. do you know, what could be the problem that (I didn't test too many) some 3rd developer plugs (including mine) don't remember midi-controller settings in projects)?
Should have something to do with parameter list too..

Beginners tip

Once you grasp the basics & general logic you'll be swimming in possibilities, although even then (as with synthedit for example) when you need to achieve certain things, you'll often have to go through a trial & error procedure. But doing so will only help you in the long run because often when you try to do something via 'trial & error' tests, you'll not only get to your end goal if it's even possible, but also discover other interesting things along the way !!
It's funny because even 12 months from now if you're starting to deeply understand SDK, someone will come along and say .."can we do 'this' because it'll improve your device" to which you'll reply, "well I don't know how to do that yet, but I'll have a try later" .... If you're hungry, then within 24 hours you'll probably have figured it out either by trial & error testing or by asking someone who already knows.
Lastly, I'll say that most of the guys you're asking for help are busy & if they do have time to answer questions or help you in any way, I'm quite sure that they'd feel more comfortable doing so privately rather than in public. That's not to say that they won't speak here, but rather that if you ask a few people some questions privately, i'm 99.9% sure that they'll get answered when you ask them.

Preset list building problems

'prest list does not match the device! - 'What is going wrong here???

WHAT I DO TO SET UP THE PRESET LIST AS OUTLINED IN THE (VEY VERY BASIC) INFO PAGE FOR SCOPE DP 3.12 PRE-RELEASE NOTES

1.First I click on the module in the project window and create a 'preset parameter list' in the 'presetparameters' list for the new device and click 'freeze' to keep the window visible...
2. next, i create an entry in the 'parameter tree' for each control i want to able to store as a preset.
3. I drag the completed list from the 'parameter tree' window over to the 'presetparameters'list and can see all of the devices parameters visible.
4. now if i look at the 'moduleparameters' list, i can see all of the options for each control/parameter of my device..
5.i then proceed with connecting the 'open preset list' buuton of my surface to the 'ADD/Show'pad of the 'Surfaceinterface' for the 'preset list'. I also click on the 'preset list' in the 'project explorer' and change the 'caption'

value to suit the exact name of my device.I also have a button on my surface of type '2' which i connect to the 'compare' function of the 'preset list' so that it can be a sort of 'A/B'switch directly on my device surface...cool hey...this seems to work ok and i can see in the 'connections' window that it has worked
6.when i save the module, the device still has the 'preset list' that i made but if i save and reload the project, the device i saved now has lost it's 'parameterslist' and i have to re-drag the 'list from the 'parametertree' back over to the 'parameterlist' and sometimes this works but if not When i load it into SFP3.1 or Scope4 i get a message 'preset list does not match the device!'...
7. once i successfully re-drag the 'parameter tree'list to the device 'parameterslist' and re-save the device, i then 'pack device=>protect' the device so it loads into Scope 4 or SFP3.1.
8. IF IF IF IF! the 'packed' device now loads successfully into Scope4/SFP often the 'Compare' function does not work anymore????

I don't think my problems have anything to do with me connecting an 'A/B' buton to the 'compare' function of the 'preset list' as I have tried all the same steps without doing this and it has made no difference to wether the 'compare' function works...plus i have one (1) perfectly working device where the A/B works perfectly and I don't get the 'preset list does not match device!' error...so it seems to be totally random...??

*********Also, and also maybe part of my problem is where is there a description of what all the tick boxes in the 'Moduleparameters' list do?????

The most important issue seems to be that Scope /Dp v3.13 does not correctly open saved projects with the MOST current device (.mdl). This means that the device you are working on in the project will not always be the latest device you right clicked on and "saved as" a new device before you saved your project.

Start every Scope DP project by loading the latest device/module you are working on into a blank project and always do this everytime you are about to make changes to it's circuit, graphics, change the preset list etc...Which brings me to the next point...

The order in which you create the preset list, add parameters to it and connect it to your surface etc...seems to make a big difference to the success you will have creating working preset lists...In particular the 'Compare' button in the preset list which sometimes works and sometimes does not.

I have found that refreshing the project by pressing 'F5' on your keyboard everytime you make any change or parameter changes in the preset list will make sure that what you are looking at in the project window is 'current' or really where things are at...especially when creating the preset list, connecting the open preset list button on the main surface etc.

Also, make sure you are in 'Use' mode before saving your optimised device...this can help.

try deleting your preset list module, and using a fresh one. it usually works for me. i cant manage to update old preset lists with new parameters. so i always delete the preset list and create a new one. also make sure you are freezing the parameters boxes when your working on them!

you can easily update and existing tree and preset list.
Just make sure you don't test in scope DP/SDK and that you always use a freshly loaded device in your test project in SFP scope.
Do not make presets in DP, it fucks up a lot of expected preset list behavior too.
and the latter above has no way backward (unless you made only one preset in SDK, but that's something to avoid) (well, there's a way backward but it is so complicated that noone wan't to be oblige to do it)

SDK problems

1) when I import a VU meter group (stolen from another device) into my new interface, I can easily link the meters to the VU meter source device, but I can't achieve the "peak hold" feature. Does is, the peak hold, another different device?
you should connect the "peak hold" value (in surface mode) to the "p1" pad of the device (in circuit mode)
2) the user manual and the (very) good tutorial from SDK forum lacks of "interface design" deepening: how to create/manage new animation (like i.e. vintage VUmeters) and hot to import/manage correctly the textures... I've realized a pair of beta devices with several fortuitos experiments that I sincerely cannot conscious repeat now!
under the gotree panel, you can see all the graphic components of a vu meter and the easiest way is to replace the existing parts of an interface.
If under the gotree panel you select the single parts of a vu meter, then you can replace it by importing your custom design (from the preview panel)
3) how I can export the module as *.dev instead *.mdl? The official manual (file management chapter, page 5 shows a menu about this... I can't found it!
when you save the device, choose "all file types" and name the device as .dev
4) does exist a list with short description of every circuit/device/surface present in the "packages"

folder into SDK program? I've found some similarities with modular modules, but some parameters & behaviours are for me different...
most of the devices are very similar to the modular ones, so it's not so hard to use the most of them.
5) same question but about "atoms". Some atom names seems to be very cryptic...
if you look inside the dsp folder there is a short description of some of the dsp modules.
6)about knobs curve response, how much curves are available? And the description? I can't find info about it into manuals...
the response curve of a knob is managed from the padlist(min, max, intensity, mod range and so on) and the best way for the first devices is to copy the values from the existing ones.

I know, in TD-1 the preset setting works... in LE-1 no! Very strange, I've tried it several times, maybe I should restart from an older level of backup.

In example (regarding point 4) I've found difference from adder and modular adder (A+B/2), in gain12db (I suspect some phase inversion with this dev), etc... in this way it's not so simply to port my old modular schemes into devices.
Maybe I should start to use the basic atom for some functions.

Pot animation

The question relates to an pot animation file format issue when loading rendered bitmaps into the GO of a pot module in SDK.Basically it is tha the tga files have to be sorted and enumerated ascending and they need to be 32bit with alphachannel. Photoshop is known to have an issue with handling transparency/alpha correctly for tga format.
No it doesn't. Photoshop is unbeatable for this type of work. Sorry to but in, but people shouldn't be fed misinformation so readily.
There's one thing that can happen which may throw a few people off track, but it's not inherant to photoshop. When sizing down animations which include an alpha channel, occasionally you may find the need to put an inner black stroke on your alpha channel mask layer if an imperfection appears around the mask edge.
Also Blender renders animation as single tga pictures in the correct format.
So do professional 3d apps like Max, Maya & Lightwave, so nothing new there.
In fact, they can also include an alpha channel transparency in the animation renders which means that those who don't know how to use their 2d application to mask & create an alpha channel, don't need to worry about it at all.
As sharc briefly explained in the gui elements thread, essentially you should only be using your 2d tool for croping & resizing your 3d rendered animation frames. The exception to that is elements with oddly shaped moving shadows.
Obviously level & colour adjustments are possible at the 2d app stage if needed (which they often are if you're colour-matching a control to the surface it sits on).

just to add an information about tga's. You don't need to crop or resize tga's after the 3d render, because scope sdk will calculate the alpha channel and just the visible part of the image will be used inside the surface and you'll see the animation as it is already cropped.

to get the best quality 3d animations for sdk use, the 3d rendering should be done at a larger size & higher resolution than the actual control (knob or whatever) that's required. Animation renders should be done at at least 320x240px, but preferably 640x480 to obtain high quality images. It's then necessary to crop the canvas of each frame to 480x480 (for example) & resize each frame to your desired final size. This is done quickly in PS by creating action scripts for the crop & resize

necessary for a single frame of your animation, then using the script/s in an automated batch process to crop & resize all frames of your animation.
If you do as suggested above & compare the end result to an animation rendered at actual size, you should notice a big difference in the image quality.
The same applies to 2d work. More often than not, it's best to create 2d imagery at a much larger size/higher resolution then size it down once complete. Doing this means that you get very high quality anti-aliasing & better quality results all round. Here's a working example of what I'm talking about. I'm working on a new vintage-style VU meter at the moment as the one on Martin's SATEQ was really just an initial test. This new one has a round shaped outer bezel which requires high quality anti-aliasing. To obtain the best quality, the working canvas size I'm using is 1600x1600px even although the final size will be less than 100x100px. If I just made the design on an actual-size canvas of less than 100x100px, the end result would be nowhere near as high quality. Even at 300dpi you'll get better quality results by working on a larger canvas & sizing down your finished result. You should specifically notice a difference in the edge pixel anti-aliasing on anything which isn't a straight line, such as circles, curves, angles etc. Working on a larger canvas than you need also gives you higher quality selection capabilities for any advanced 2d work.

Good to know about the automatic crop of the SDK.
But the animation files are substantially bigger (at least before loading) if you have overhead not used for visible projections.
Does the crop feature include also a storage in the module with the automatic-crop file size ?

'm using CS2 here and have no problems with Alpha channels in TGA format. Like Shroomz said, resizing your TGA can sometimes result in a slight feathering of the alpha channel at the edge of the canvas. a 1-pixel inner stroke on the alpha channel always sorts this out. Use the actions and batch automation in photoshop. It's very quick and easy and allows multiple edits to be applied to all animation frames non-destructively. If your TGA renders don't already have an alpha channel you can create one in CS2 with the 'apply image' function (this can be a little more time consuming). For testing alpha channel transparency you can always make a selection using the alpha channel and copy and paste your control with shadow or whatever onto the background of your choosing.

that is what I saw as well. There is always a need to fiddle with the layers. copy paste and thus create an alpha channel (layer). It may be correct if one accepts that adobe defines the way a tga (32bit with 4 channels = RGBA) has to be mapped. I always dislike if a free format like TGA is "loaded" with manual editing for no reason. I never bought a CS2 or sth. It is simply much too expensive for me. So my experience derives from a trial version which may be castrated regarding flow through handling of tga files without manual creation of the alpha layer for the A of RGBA TGA.I honestly doubt that it is easy to see for a newbie how to create proper 32bit alpha channel tgas with batch mode automation.

There isn't an automatic 'crop' function as such in the SDK. Micron basically meant that the sdk automatically applies whatever alpha channel mask info is contained in the TGA files you use & 'automatically' creates your desired transparency. It was a slight misunderstanding because when I spoke of 'cropping' I was talking about doing what your alpha channels should be doing (which I explained that I wasn't. I was talking about cropping from a rectangular to square canvas such as 640x480 to 480x480). As I said earlier, the alpha channels for your animation frames should be included in your render output from whatever 3d app you're using. This simplifies things considerably by eliminating the need to create alpha channels for each frame manually, although If a control is perfectly round in shape you only need to create one alpha channel for the entire animation anyway. Having alpha channels included in your 3d render output is pretty much a necessity for doing something oddly shaped like the pointer knobs or chickenheads I made. Otherwise I *would* have had to manually create an alpha channel mask for every frame (yawn). Btw, 150 frames is WAY too many for any normal medium to large knobs. 61 is plenty over a 300 degree rotation & if the knob is small you'll even get away with 31 (31 is better than 33 for rotation divisions & creation of surround graphics)
You'd only need 150 for something MASSIVE!!!

To make it visible and tangible to other SDK users I give you an example to understand me, if you are willing.

A)
open photoshop.

create a new picture with transparent background.
draw a circle and fill it.
save as tga 32bit, no rle compression
load this tga
you have no transparency but a white background and PS creates a alphachannel layer that is completely white.
SDK shows this tga with a white background, since the alpha part of the tga is not transparent.

B)
open for example GIMP.
create a new picture with transparent background.
draw a circle and fill it.
save as tga 32 bit, no rle compression.
load this tga (in GIMP or PS or SDK)
It has a transparent background

IMHO this is a flaw of PS. And flaw means taint, nothing more and nothing less.
I fully admit that there is no perfect way to deal with transparency (as sharc pointed it out)
From my perspective as SDK user and graphics amateur it is tedious to create an additional alpha channel manually in PS to be able to save a 32bit tga that really has an transparent background.
And thus it is my recommendation to the people who ask me that the straight forward, low effort and low budget version to produce 32bit TGA that work in SDK is GIMP.

Alpha channels are worked with separately & manually in photoshop to give you more flexibility. For example, you don't always want just a straight mask, you'll often want to include shadows, glows or feather the edges of it slightly to compensate for anti-aliasing on the edges of your image (or series of images). These things need to be done manually. Photoshop does not do the work for you & it is not supposed to. It assumes the user will want the flexibility to dig deeper, that's all. I perfectly understand your points 1 & 2 and personally don't see the problem as I know it's not a flaw in the software. As my brother touched on, the only real problem with alpha channels is actually inherant (no matter what app you're using) to working with images which have anti-aliasing or feathered edges (although working on a larger canvas often helps with this issue considerably). With such images, the edges of a normal alpha channel mask may often need to be tweeked to obtain a smooth blend with the image/s without unwanted artifacts. Such issues will need to be dealt with manually by working on your alpha channel no matter what software you're using, so from my perspective it's not an issue or flaw, but something you'll always need to deal with when creating certain graphics for use in the sdk.

BMP Animation

I still have problem with animation.
Think has done with single picture so far as described properly (format / title).
Mine action-wise with load is quite wrong.
Become on in English and in German post, so that me at least somebody understands.
From ground alpha channel problems I have aniationen as bmp rendered.
Mine action-wise is on the following to see speaking in images.

picture1: empty synth loading

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

picture2: Store enes make potentiometer

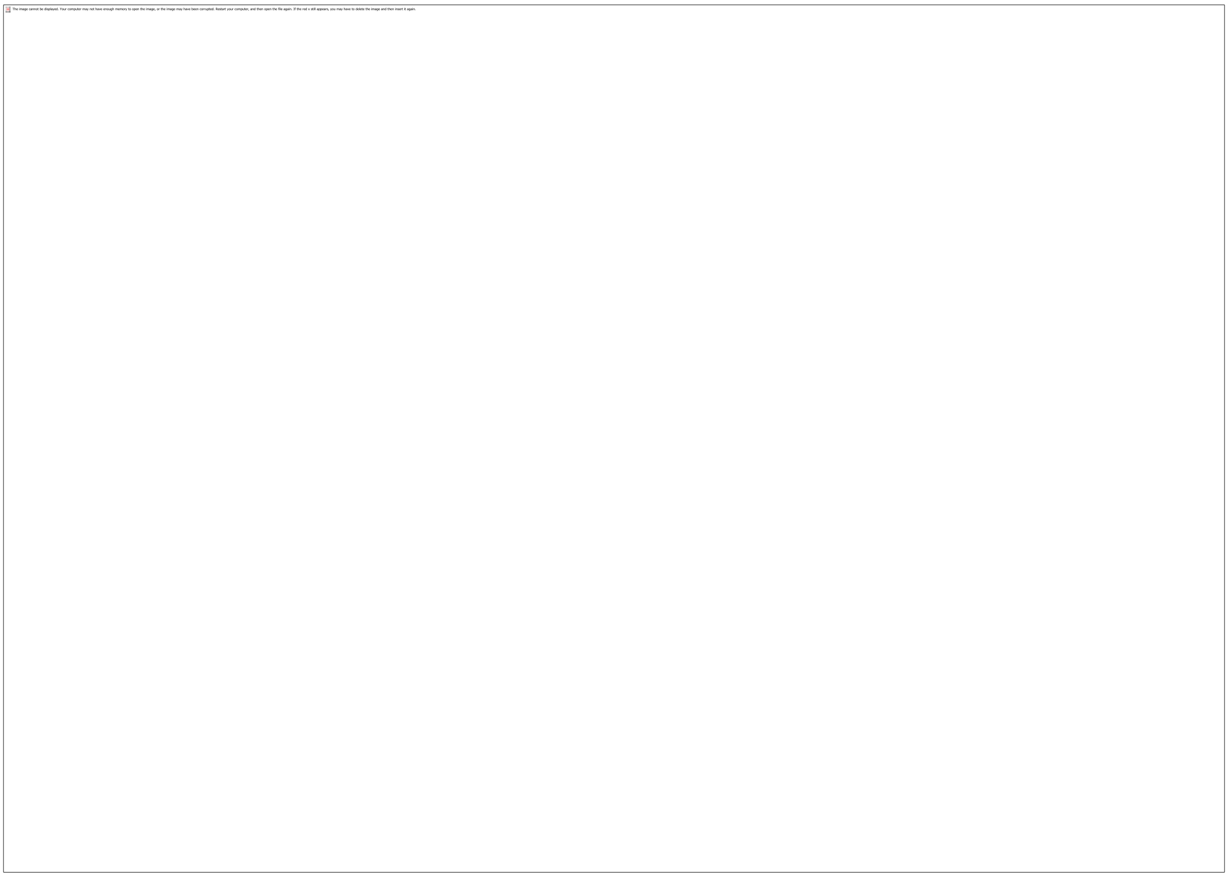
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

picture3: Pretend potentiometer

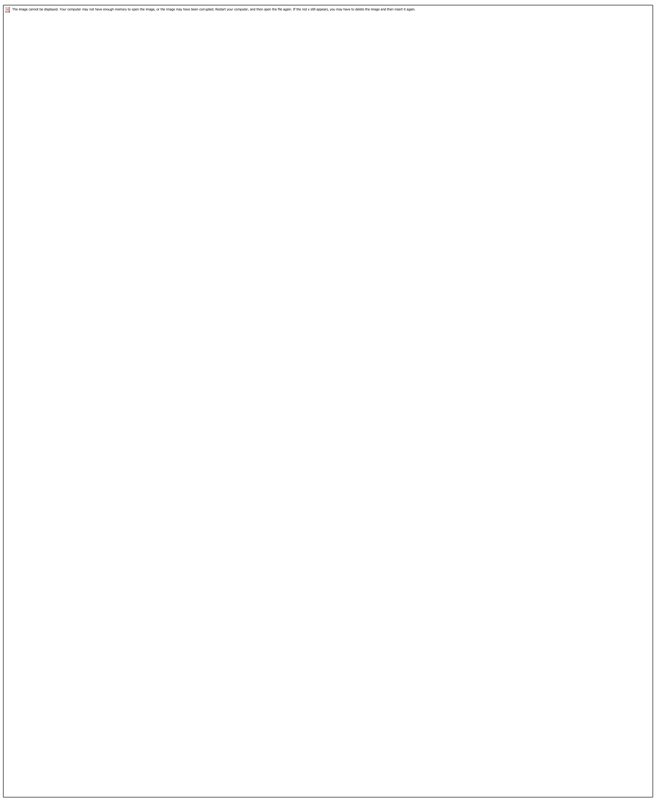
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

picture4: File search

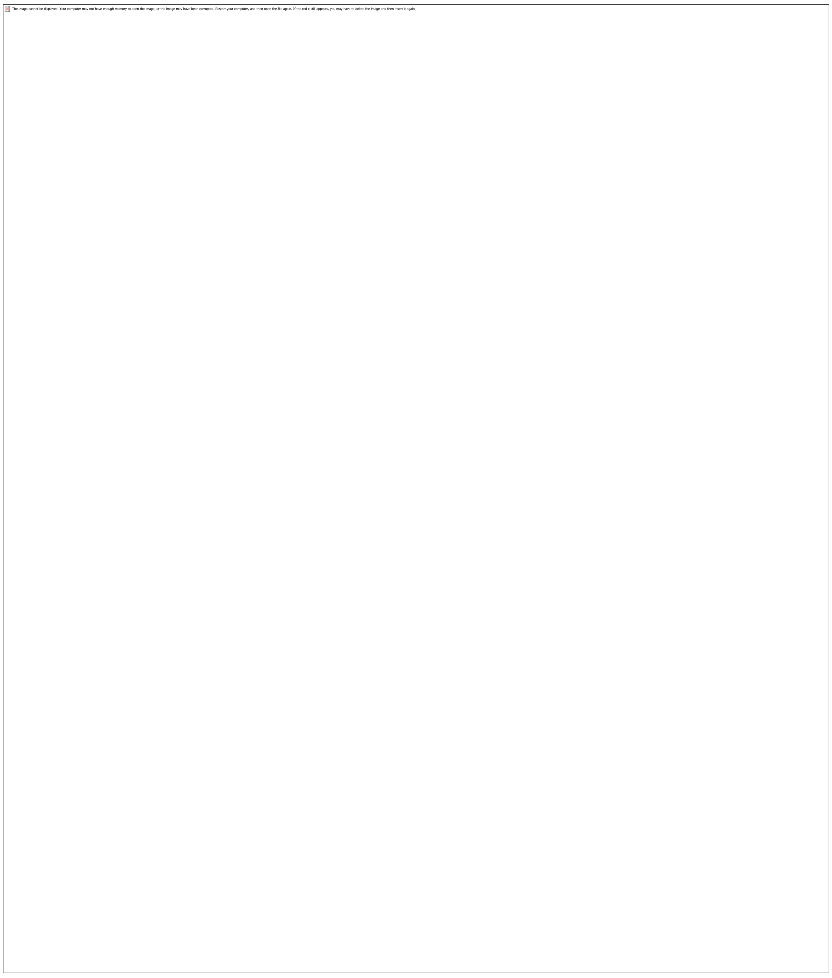
picture5: Choice start picture



picture6: Now picture is to be seen



picture7: Nevertheless, animation the old one remains!



Where lies here Mistake?
if i have understood,
first of all, if you need to use images with alpha channel, you must use tga's, because bitmaps are not recognized.
The second thing is, if you want to replace the design of a pot,
select the pot
go to the preview window
load a tga sequence by choosing "load bitmap"

in your example, you are replacing a pot with a fader, it should work, but just in case you want to replace the design of the existing faders, you will not need to use animations, but just the fader background and knob and you will be able to replace it by selecting single components on the gotree view and then loading bitmaps as above.

GUI Elements (Shroomz)

The concept I've been working on for a while is that we build a fresh & high quality collection of gui elements (control animations) which can include all sorts of controls that we'd like to make available to the SDK developer community. Anyone can contribute gui elements to the thread & they can be either custom, new designs or 'modelled' on existing & classic control designs of pots, knobs, sliders, dials, switches, toggles, leds, lcd animations, basically any relevant gui elements.

What are the specifications? (size, number of colours, etc...

they need to be 32-bit TGA (Targa) files with an alpha channel defining transparency. This allows for shadows to be included in the animation, although this is not always necessary.
Size doesn't matter
You can have any frame count, but it should preferably be an odd number such as 61, 65 etc. It's also best to keep the animation frame size to an odd pixel number like 55x55, 65x65 etc, as that seems to eliminate the common 'wobble' problem. Here's an example of how the black ribbed pot with the new lighter shadows would look when placed on a dark background skin with light graphics.

you (or anyone interested in) may consider to set the rendering params to more than one and to more diffuse light sources to avoid the artificial look - imagine 8 of these in a row
Even better if the angle of the light sources varies with control position on the surface, or the shadow
vanishes occasionally. Shouldn't be too difficult as it's a reusable setup (and just variations) anyway.
i have never used DP myself, but i can imagine the performance of the device would be sacrified if you were to render a pot for every position on the finished device. i ASSUME that using one animated knob instead of say, 20 different renders (approx minimax estimation from my memory) will save cycles. can anyone confirm or deny this?

the 'rendering' of the knob is done with an external (preferably 3D) program, resulting in an animation clip (a sequence of bitmaps). It doesn't matter how complex the design process was, in that context all bitmaps are indeed created equal. The audio performance cannot be affected as the GUI interaction is native CPU code. if the system is smart, it could use a single 'mastercontrol' per GUI element type and picture the corresponding frame of the animation by a copy of what's at the respective index position. If it uses an individual sequence for each single element then there's a load penalty (memory), but not much else to expect regarding performance.

loading say 20 times more animation frames into a device would only really alter it's loading time in SFP (loading 20x61 frames as opposed to 1x61 frames for example) This would also put a little dent in native CPU & Ram usage.

Note that the shadow direction is barely even visible when the pots are put on a dark background with light graphics like the one in the example above.

Extensive custom variations can be achieved, but they are better done at the request of an actual developer as customized versions of a basic design specifically for a device he's making. For example, modifications to shadow & reflection colours could be made to suit a particular colour of background skin (especially brightly coloured ones), which is a relatively simple process. The whole process of creating designs such as these from start to finish however, is a lengthy & very involved one. Multiple versions from different perspectives (for example) would be an extremely time consuming endeavour when you consider that a new custom or modelled pot animation takes 1-2 days minimum to create.

I assumed that the most time consuming process is the geometric model and it's texture - a day isn't too much indeed. The 'environment' with the virtual lightsources on the other hand remains rather constant (imho), after you found a setting that's visually appealing. Then you record one animation sequence of the rotating (or otherwise) moved control. Shift the control to a different location, record another sequence, shift etc. (according to my humble 3d experiences)

(Shroomz) I don't really want to get into the depths of tutorial stuff here, but here's a neat trick for anyone using these pots who wants to get them in a nice straight line on their devices... When you're at the stage of placing pots or other controls on your device in the SDK, simply make 2 versions of your background skin, the actual skin & a temporary version of it to use for alignment which has visible guides (lines) wherever you're putting a row of controls. Using the version with guides, you can accurately place your controls in nice straight lines then simply swap the background skin for the actual version without the guides. Et voila!

A little about aluminium knob caps. There are several factors which can cause a cap's light, shadows & reflections to change:-

1. When you turn a real knob which has an aluminium cap, your fingers create shadows & a reflection of your 2 fingers appears on the cap. This is not applicable here unless virtual fingers are included in the animation, which although funny would obviously be rather stupid.

2. If the pot shaft which a real knob is attached to has a 'wobble' (isn't straight) the knob will obviously wobble too, which causes a slight shimmer on an aluminium cap. This isn't applicable here unless we want a collection of wobbly knob animations. We can however provide such a customization if requested by a developer.

3. There are several external factors which could cause changes in a knob's aluminium cap, such as a bird flying past your studio window for example. Again this factor is not applicable here.

I've had people who might contribute here asking privately for some specifics about the lighting configuration used for these controls. Instead of going into too much detail here, I'll try to find the time to post a very basic gui element tutorial thread which outlines the methods & tricks used here.

In the meantime, I should explain that I've been using the same 3ds project for all of these controls. The project has several strategicaly placed omni lights. To retain their positions & settings & keep some consistancy throughout the collection I just load up the same project each time, delete the knob or whatever it was modelling & then build my new model in it's place (in the same central location)

Anyone not well versed in max however, should be aware that material editor attributes, settings & maps are very very important in terms of a surfaces' reaction to light, shadows & reflections etc (not to mention colour).

Other important things to note are:-

1. To cut down render times & get proper alpha channel inclusion in the animations, we're no longer including background shadows, so there should be NO background 'plane' or 'surface' beneath your model

2. Rotation motion should be done in 3d. The knobs here are rotating 300 degrees in total across 65 frames (150 deg' left & right of center) Straight key-in & key out tangents should be used to keep a consistant speed, otherwise (with default settings) your animation will speed up at the start & slow down at the end, which you don't want.

3. When rendering your final animation, the minimum size you can get away with is 320x240, but 640x480 is advisable if you want decent results. This way your animation frames will be relatively high resolution before you begin the process of editing & sizing down in 2d.

4. If you're using ph0t0sh0p, learn how the action scripts & batch processing work if you don't already know them, as they are essential for this type of work.

I used blender for the knobs I rendered. It works very well. There is no need for the 1000$ apps when there is an open source one. And for the creation of pictures there is no need for PS or something.
GIMP does the job

Re: Gimp and Blender, I've tried both of these out over the past few months. I've actually still got Gimp installed here on the Net machine. They're certainly both capable bits of software and for most people wanting to dabble with some graphics they'll usually do the job. That said, IMO there's really no comparison between Gimp and CS2. The way I see it, I've spent lots of money on music hardware and software for my hobby and as a professional designer I didn't mind doing the same with graphics software. Don't get me wrong though. If you're after the free alternative, you won't get much better than Blender and Gimp.

You can use whatever software suits your needs & pocket, but the saying 'you get what you pay for' *does* apply to design software. Having thoroughly checked them out I really don't like the apps in general or the workflow in either blender or gimp. They're 3rd rate to be honest & definately not even close to comparable with max or PS. Very few design studios in the world would use the free, open source tools, simply because they aren't good enough. I liked the 'idea' of both blender & gimp being free, thinking they could maybe get us away from expensive upgrades every 2/3 years. Unfortunately, having given them a shot out of interest, they just don't cut it. With a lot of time spent learning they're workings, I'm sure some half decent results are possible, but I've yet to see any decent gui elements made with blender & or gimp.

the slider caps are fantastic
what program do you use?
I know nothing......
can you help me

I am looking for a (step by step) site or tutorial where they explain how you make buttons, sliders and knobs
And how you get them working in a GUI

basically shroomz told us he is using Max
http://en.wikipedia.org/wiki/3ds_Max
There you create an environment with lights and camera and you 3D model your GUI element.
Finally you take a screenshot in the TGA format to link it to the SDK modules.
In case of pots you additionally animate the 2D screenshots
But of course shroomz and sharc are in all that little nifty tricks to make it more than just a 2D copy of a 3D world.

The problem with creating a tutorial is that it would take a considerable amount of time to produce and in the end some key areas of the tutorial would be very program specific. For learning how to model and animate your own GUI elements, I would recommend that you use the tutorials supplied for whatever piece of 3d software you're using.
Then it's just a case of rendering the animated scene with each animation frame set to be saved as a TGA. If you're 3d software allows for it, set each render to produce an alpha channel for transparency, otherwise you'll have the tedious task of masking individual frames in a 2d app.
You then load the TGA files into you're 2d app (photoshop, gimp, etc) and crop and resize them to their finished size

---

Error message in SDK

When creating new devices in scope sdk and check the performance of the device(before creating the surface), i often get the message:
"15.. view tree groups ,5...non custom size groups - it is recommended to use view tree groups and customize groups very carefully"
What does that mean and how to avoid this ?

It always says that no matter how much stuff you have or don't have in your device. It's just to tell you how big your device is.

How do I make a GUI?

I have made a module with three submodules. Each of the submodules can be controlled with only one knob, and I have exported the pad, so all three pads are available in the first level of the mother module. But I can not figure out how to make a GUI. I do not need anything special - just a box with three knobs and three corresponding text boxes. I can make those three knobs and text boxes in the first level, and they work fine. But that is just "on the table". How do I make it fit in a box?

Ok, I found out ... except, that it does really work :/ (meaning I did *something*, but not the right thing)
In surface mode, I had to be in edit mode too. Then I could draw pots and texts from the file browser to the default panel.
Then I connected them to the controls in the children panels by using "store -> connect to". It works now, **but** I can not open the surface from "surface -> open surfaceinterface". I get an error:
Nil reference on left side of "." as a result of expression "o".
When I do a "tools -> check performance", I get:
10 ViewTree Groups
15 Non Customize Groups

Did you drag in the "SurfaceInterface" module and connect the "Show" pad to the "Show" pad on "DefaultPanel"?

Yes - those are connected in the top layer of the module.

Firstly when dragging graphic objects in the graphic window onto your surface, the last mode you need is edit...Use M mode (press m) and drag them onto the surface. Edit mode will cause nightmares. Make sure that the new graphic object you dragged is listed under the 'Panel' sub folder in the project explorer. This does not always seem to be the case. If it is in the main list, first open the 'dimensions' dialog, click on the new object you have dragged and take note of it's vertical and horizontal position and WRITE IT DOWN!

now just drag this graphic object into the 'panel' folder in the project explorer list. If it disappears, that means it is still there but it's position on screen has moved and you will have to use the 'dimension' dialog box again, click once more on the object in the project explorer list and type in a new value for it's horizontal and vertical position into the 'dimensions' dialog. Phewww!

If you have problems when you try to open a surface of your device, it usually means you have more than one 'surface interface' module in your 'project explorer' list. There should be just one at this stage but at the end of the device creating process, you will have another one for the preset list dialog. That makes two in total. If you have 2 already, just look at the 'connections' dialog to see which one is actually connected to something and delete the one that is not...

Starting out in the SDK

I'm totally and utterly lost in the Scope SDK!

You guys have read the manual, right? Then it's a simple operation of dragging modules in and connecting pads together. When you want to connect surface elements to circuits, connect two pads together via Store --> Connect in the Pad list. It's not too bad.

Two other windows you'll need to access fairly often is the GO Tree (sometimes you need this to access certain settings in a surface element) and the parameter window, to drag in the surface parameters if you want to make presets. But that comes later. For now, start with "Empty Effect" or "Empty Synth" and drag some modules in. There's also the DSP atom list, but those are harder to figure out.

Actually, I figured out how the SDK worked by looking at CW's modules and seeing how they hooked up each module's surface elements to its internal circuits. Then I figured out how to do the same for my own device. Just do a lot of experimenting.

well if this helps, the best advice i could give is !!!!start really basically!!!! but just start!!!!!...
Drag an 'empty effect or 'empty synth' module into the window.
!!!Note that standard 'empty synth' modules from Creamware will NOT work in insert slots!
double click to go inside it's circuit and Just connect something like a gain atom to a distortion atom and connect the inputs and outputs to the ins and outs of the empty synth module then add some standard old graphic knobs to the blank surface in the 'graphic window' and experiment connecting them to the effects functions...whatever you want. At least get the ball rolling expanding your circuit as you go.

Get to know things like the 'control ranger' and what the different values for the knobs do so you can change the amount and type of data they send...very important in getting the most out of the atoms. For example, setting the 'invert' value of a knob to '1' will reverse the knobs direction...simple but not so obvious when confronted by the huge amount of dialog boxes in Scope.

I created filTheR on my first night with scope with no advanced programming and just a bit of electronics knowledge just by experimenting in this way and at the end of the night, i had all the basic tools and modules worked out (well, i had tried them all at least...) and even better had a new device!

Expect Scope to crash, it's ok. It rarely effects your operating system. Just make sure you have regularly saved your device and scope projects separately before the crash(most important!)and reload Scope. In fact, if Scope does not crash while building a device, you are not being adventurous enough

Fold useful/successful parts of your circuits together and then save that new 'atom' for future use so you don't need to remake the basic parts of every device all over again every time.

Is there a tool to create 127 images horizontal for Dial-knob

I'm trying to create an animated knob. I want to generate 126 images from this one image that rotates 2.1259843 deg each time. And images have to be next to one another horizontally. Is there a tool that does this for us?

for automatically rendering an animation you can use pro 3d softwares (can be hard too use), or even a gif anim thing (or flash or ?) and convert to bmp afterward (image by image) boring, but once it is done, it is done..

Well, the solution to my problem is Adobe "action sets" . This way you can record your steps.
I've recorded this action 127 times:
copy selection , select doc , paste selection , and this 127 times. Know i can just repeat it by pressing play.

when i used to do this i would create the knob in 3d studio max, do an animation of it turning for (however many frames) and set it to output separate TGA files instead of an AVI or whatever. and then import them to scope. I seem to remember you can even have max number the files exactly as scope needs them to import.
the 3d software makes the shadows on the knob look better anyway. just make sure you have your light shining from the upper left (well that's the "standard" anyway) or so it will match how you are doing your interface.
I uploaded an example knob MAX file, which is one i used on some of my devices. it was made in max4 so should work in 4 or any later version. there are no texture files, it just uses the material editor.

http://www.neutron7.com/files/neutron_k ... le_max.rar

They might even work in GMAX which is a totally free version of 3d studio max but with limitations on rendering. it is made for people who mod and develop game levels.

http://www4.discreet.com/gmax/

and don't worry about having 128 positions, because you don't have enough pixels on your screen to see such a small movement. however do use an odd number so the top position will be dead centred and if you have something with markers for say 0 to 10 (11 marks) then make it so the marks will be dead on with say, 5 or 7 positions between them and if you want it really accurate you need a text display for the real value, because even a 128 position knob is nowhere near the resolution for most scope controls you are likely to make.

How to create an on/off toggle to link parameters

No matter what I try and use I just cant seem to find the correct way to link 2 matching controls, I have tried on/off switches, buttons, and cutters and nothing does the trick as I would expect it too. Any one care to give me a tip on how to accomplish this task?

Try cut in the switches folder. Put it between the two faders or pots, I have just tried it and it works, linking and unlinking the two controls.

SDK newbie problems

is there any1 in here that could provide a simple synth device as a template ? I tried already to build the Midi to osc pitch but did not succeed. There is an error message popping up.

The SDK documentation tells nothing about that and the chapter about building algorithms is missing for the moment. Any hints here are appreciated.

When using the Midi voice control connected to the Midi source, and then trying to connect the f in pad of the Oscillator to the f out pad of the MVC i get the following error message:
Problem: Cannot add notification to sync output Midi Voice Control.freq

Finally I figured it out. This error appears when you have connected a surface control to the f in and then want to connect in addition the f out from the MVC. So use only one or the other. Sounds simple, does it ?

on request now the solution in detail:
The oscillator modules in SDK come with a default surface object.
This is visible in the project window by the antenna symbol.
In Project explorer there are the surface objects as childs of the 'panel' object.
I stumbled into the issue, that the f in pad is connected to the default surfaces already.
If you then try to connect the MVC f out pad the error appears because you try to connect the surface control 8pot) and the internal connection in parallel.

->> The way out is:
delete the frequency control group connection from the oscs default surface.
This is clear since you intend to control the oscs pitch by the midi generated pitch signal.
the most convenient way is to select in project window and then find and delete in project explorer.
If you still need a surface control you need to set up a adder/modulator thing that brings midi pitch and controller pitch correctly together.

How to make synth polyphonic

The MVC modules seem to be set to "single load", I can't change that to make it polyphonic. What should I use instead?

To make your synth polyphonic put a "polymix 16" mdl in the patch and a "dynvoicesofparent" mdl. the "dynvoicesofparent" mdl will alow you to set the number of voices. when this module is set to 0 it will unload whatever mdl it's inside of from the dsp.

That worked! I also found that I was able to insert two PolyMix's so I could have a stereo output (as I wanted polyphonic filter routings).

Is the SDK missing a lot of the graphics elements?

is there a very limited amount of graphic data included with the SDK? I look at lots of devices and at least in my folders non of these buttons or pots and such are available, any idea?
IS this a major difference between DP and SDK or am I missing something?

The graphics is something you can 'easily' do yourself.
But what may be missing are the underlying scripts that link the graphic elements together.
here there seem to be differences between DP and SDK.
Just remember to create your own bitmaps (bmp and TGA format) and replace the existing ones reusing the scripts.

The DP 3.13b does not have anything special compared to the SDK. No graphic or modules. And there's no manual. Pre DP 3 versions did have a lot of graphics and additional modules (fft, vdat etc) - I have no idea why they're not included in the DP 3.13b or how to get them.

MIDI CC standard map for Scope (SpaceF)

Advantages of a common standard for Scope :
- It includes default midi assignment (GM?) such as 74 for LP Cutoff, 71 for resonance, 5 for portamento etc. Of course, sometimes it doesn't make sense to assign "10" to pan, so I use it on all devices to control the static LowCut/Bass Cut (Static/FInal Hipass) of all my synths.

- it may allow users to make a preset for their controllers, and this preset will be valid on all devices. This is the case in particular for the ADSR group of Amp& Filters Envelopes. This one should be the same on all devices. Same for the Pitch Env and additional parameters (A/D slopes etc).
- Some controls are "buttons". Through all devices, they will correspond to buttons too (a button in one device is a button in another device).
- DXD is big and includes 4 oscillators, 5 filters (the fifth is dual) and 3 envelopes (AMP, Filter, Pitch), an oscillator mix section etc etc. So the exact same assignment can be re-used to fit a synth with less controls.
- Assignment by groups : you can see colours, or look in the "list by groups" and focus on the FM section. It includes 20 controls for that oscillator : now, if you re-assign parameters of that group to control, for ex, noise oscillator, and save a preset, it means that one memory/preset of your hardware controller can be re-assigned to other parts of the synth, simply by changing a preset of the Midi Controller Bank of the device. It means 100 % hardware control is possible.

Always unassigned : CC 32 (Noah bank Select) and 64 (Sustain pedal) are never assigned in any of my devices. They can be used by users as unassigned midi controllers if needed. I also propose that those 2 controllers be never assigned in any synth as a default rule.
With a smaller synth, it simply makes more unassigned (or undefined) controllers.

You can read more by having a look to this page. I will also post the AN-Osc assignment map very soon, so it will be more obvious for you to see what i mean by comparing the two lists.
For the moment it is spacef only, but i wanted to share it with you, because I think it can only benefit everybody, developers and especially users.
I think it also saves a lot of presets/memory of the hardware controllers.
I will try to make a standard map, but for the moment I am finishing a few other devices on the same "map".
Have a look here for the map (an-osc map will follow): (NOT AVAILABLE NOW - strav100)
(ps the "common midi controller group is not shown here" because i didn't finish that html page yet. It will also include the map of "remapped CC" from the DXD Midi Controller presets)
I may use CC 118 for Noise resonance so you can add this one to the list).

Pitch modulation question

I just found Pitch Modulator C in Modular -- this allows me to do linear pitch modulation, so any oscillator can be an FM Operator! I'm not sure which module allows me to do that in the SDK. Does anyone have a clue?

Oh, I just got it...I simply add the output of one oscillator to the pitch input of another oscillator.

That would make an AC frequency control... What you may want to do is keeping the original (DC) frequency, and modulating that one with the other Osc. Maybe that's what you meant, but I couldn't make it up from what you wrote.
Another remark: do all, or some, CW oscillators take sync signals for input? In modular, for example, that's usually not possible - even though you can amp-modulate the frequency by an osc, that doesn't necessarily mean the 2nd osc will react as fast as the modulator oscillates.

Can't save as .dev only as .mdl

Im certainly following the instructions but no matter what area I try and save from I only have the module option, I must be over looking something..

Just rename them to *.dev. There is no difference regarding the file format, just the extension is different.

Anything you can think of as to why none of the options allows me the correct manner to save new devices as .dev? even the surace selection only save save new module.

You have to simply rename the file as .dev. However, there is ONE difference: if you place any insert slots in your device, saving your device as .mdl will NOT remember any of the insert effect's parameters. Saving the parent device as .dev will remember the insert effect's parameters. I had to find that out the hard way!

So if I create a device with insert slots and I save it as a .mdl which seems to be the only option available to me for some reason then it will not remember any effect parameters, if I change that .mdl to a .dev will it then work? Or will I need to save as a .dev directly from within SDK, because this seems to be my main issue, no matter where I try and save from I am unable to have the option to save as .dev from any point within the SDK.

If you use "Empty Synth" in "Device Design" the SDK save it with .dev extension.

rethink Wolf's answer: *There is no difference regarding the file format, just the extension is different.*
the effect Shyne mentions refers to the runtime environment. You always generate one single version of a device, whatever extension is used. When the thing finally runs, another part of the system decides how to deal with inserts and uses the extension to distinguish. There are better ways (you obviously anticipated something more smart...), but the quick and dirty approach is not uncommon at all

At 1st I couldn't rename the file and then realized since I have Reaktor on this PC it causes the .mdl files to show as Reaktor files and does not allow extension changes, I just rebooted into my internet boot and all was fine I could ghetto change the extension.

No need to reboot: in the explorer, under folder options,view, disable the "hide extensions for know file types". This will allow changing the extensions of registered filetypes.

How to hook up an array in SDK

Can someone tell me how to hook up an array.
e.g. Arrayint[10]
More specific question. How do you create a I/O "handle" to the array?

Use Async2Array

Midi Value Table Lookup

How to link my SDK device to screensets?

Can someone tell me how to link my SDK device to screensets. SFP isn't saving the position when creating a screenset.

In the Module Parameter List (where you have columns of little 'check' boxes), you have to make sure the surface position X and Y have both the RS and SS boxes checked.

the surface position X and Y?They are not in my param list..
I got:
SurfaceInterface [Showpanel], SurfaceInterface [ShowPresets]
But I can't find:
Surface[X], Surface[Y]
in the paramlist. Do you know what module contains those params? So I can choose "Create parameter in.." to fix this.

Then I think you have to create pads for these two parameters.
They are called, 'PosX' and 'PosY'. Click on your surface object and they should be there. If not, click on the VARS button to display variables, then right click on the PosX and PosY to 'create pad'.
Once you have them as pads, drag the pads into your ModuleParameterTree, then go to the ModuleParameterList and set up the check boxes. You don't want these to be stored in anything but the ScreenSet data (SS is Save in screenset, RS is Restore in Screenset).

Aah, there they are
Steps I had to take
Click on panel module.Then I spotted PosX PosY
-Right clicked on both of them and selected create pad.
-Then created parameter in list and set the toggleboxes to RS on and SS on. Rest is off.

Missing modules in SDK

I seem to be missing a bunch of modules (for example, MIDI Control to Number, Midibyte Extract, etc). Can't find anything in the manuals. I've also searched the forum here but found nothing (except Shayne mentioned at one point that he was missing a "Tools" folder -- which I don't have, either). Below I've listed what the Packages folder looks like on my install. Any idea what I'm missing? Or how to go about getting the missing modules?

There is also a dsp module list that can be opened from a dropdown menu (don't remember exactly which, if view or what), that has all the stuff (without g.u.i.).

Module in the SDK that acts like a %(modulo) does in c++

is there a module in the SDK that acts like a %(modulo) does in c++?
I can create it by using if statements and conditional sends , but there must be module that does this.
(I think i'm overlooking the dsp-modlist
Any help would be great!

Divider, Multiplier,Subtracter then? Think that's the only way.

I would be very interested in your findings. I suspect we're after the same end -- getting 7 "true or false" values out of a MIDI signal (0-127) -- but if I'm mistaken, my fears will be completely irrelevant to you. What I'm afraid of: I can't route from MIDI outputs to to int divider, multiplier, subtracter modules. Any tips?

In SDK midi is a different type of signal.
add , subt , multipliers are either sync or async. It is possible to connect sync to async. But not to connect midi to one of them.
To know exactly what midivaluus are coming in. Use the Midibyte Extract module.
Out of this module comes async signals.
Statusbyte , Data1byte, Data2byte.
Then the only way to get the signals back to midi is thrue the midicontrol2value module.
But that needs some changing the bits before it outputs the right message.
I'm looking for a way to do make a 16channel midi in transform(harmonize) thingy.

I got my MIDI cc bit extractor working this morning with the MIDI Val Extractor (under Pacakages/Circuit Design/MIDI). But I haven't come across any of the modules you mentioned... Where should I look for these?

In scope there is a window called module list. Look in this list for the modules I mentioned. The modules you found in those directories are with GUI. Thw ones in modulelist are without GUI.

Much appreciated. The output of "MIDI Control 2 Number" seems to need division by 16,777,021. What a strange number...! (2^24 = 16,777,216.) Maybe I'm using it wrong?

mm Divide ? I subtract it like this:
note off status byte 128 --> -12288 + note number
note on status byte 144 --> -8192 + note number
Additional channel add.
for note off 128 -127 is channel 1
for note on 144 -143 is channel 1
add it to the above outcome.
BUT! I've not managed to get it working correctly though.
When changing the velocity value it outputs a note , and when changing the note value it outputs a note. So guess what happens when updating velocity and note number? Right , it outputs two notes , instead of one! Now I wish it were possible to add the velocity to the already converted status byte+note number+channel so you'd only have to use the right inlet of the module. Anyway I hope you'd get further with this and you'd tell what solution you've come up with. I badly want it to get working.

Yeah state is really tricky with the SDK. I wish there were some kind of "synchronize" module that waits for all inputs to change before "flushing" the output. I think maybe the conditional value send (Circuit Design / Logic / Conditional) could somehow be used to do something like that... But I haven't wrapped my head around how to do it yet.

I'm afraid I have no idea what to suggest. However I have done some MIDI state machine type stuff. Not with notes or velocities, mind you, but... Even though I can't see how it would help you offhand, maybe if you take a look at it yourself you'll get ideas?!?
So I've put all the modules I've made so far online. None of them have GUIs! And most of them are just helper modules.
Beware, some of them might make your CWA cards melt or something. I reloaded a project in SCOPE last night that uses the "Brass Tusk Mammoth Buses" module, and I started getting continuous C++ exceptions and error messages like "no such board ID 0" etc. I have no idea what went wrong, but no matter how many times I reloaded the project, I couldn't get it to come up, until I finally deleted my module, and then the error messages went away. I've seen a lot of SCOPE error messages, but these ones were by far the scariest I've ever seen.
Anyway with that warning in mind, feel free to check these out -- and if you feel the inclination, feedback would be very very welcome! I need to figure out what my bad habits are... Better to learn from a mentor than by cruel experience.

I've looked at the "Midi State Machine" Controller and Program.
Unfortunately there isn't much to learn from these that could help us in our quest.
What I want to achieve first is a correct working circuit. This circuit gets midiinput , then has to split all databytes and assemble a new midimessage for output.
This is what I came up with:
http://www2.hku.nl/~casper0/CreamwareDe ... 20Midi.mdl
It basically uses the midicontrol2value module.
This module can do more than just control messages. Just by adding a certain number to the notenumber will let us create that midinoteon message and noteoff messages.
This module is partly working ok , and definitely the way to do this.
If we could make this circuit failproof , then we have everything we need for a harmonizer or channelcycler , or w.e.
I found a module that can get a set value and than 'bang' that value out its outlet.
"TrCol2e" & "TrCol4e" & "TrigSW"
Not tried anything jet though.
Well ,I hope someone could perfect my circuit and ->share-< his/her findings here on Z.
p.s. somewhere on Z I posted something like this too, but now this module is doing a little better.
Also I mentioned some vars that weren't correct. But just look at the controller pads if you want to what values to add for on/off message.
Beware of different midiimplementations!
My roland D10:
noteon -> Status byte 144-160
noteoff -> also Status byte 144-160 only velo 0
+ an additional message 176 123 0 wich is a sort of special control message.
Cubase:
noteon -> Status byte 144-160
noteoff -> Status byte 128-144
You can see why things would go wrong if you'd create something for roland d10 only and than try using it with cubase.So beware....

What are the MIDI Voice control modules supposed to do? I can't get them to do anything?

The MVC is the module that supports voltage controlled signals in conjunction with midi.
You'd have to connect an oscillator to the freq.
An ADSR to the gate and esync.

Also need a lin/exp VCA to control amplitude. Hook up the ADSR to the VCA controller input , and the oscil to it's other input. If you'd want polyphonic voices , then connect the polyvoicemodule after the VCA. Now hook up the midi to the MVC and play your midikeybord.
If you'd want to have two ADSR that use esync you'd have to add the esync's from the two ADSR and connect that to the MVC-esync.
If you've done all, you basically created a 1 osc synth device.
Have you ever done a modular patch ? Same thing.

S-H lev-enb module

"I wish there were some kind of "synchronize" module that waits for all inputs to change before "flushing" the output"
Found this module!
it's the "as S-H lev-enb" in the modulelist.
It let's you input a stream of numbers and then pick just one number out of the stream to triggering.

this module "as S-H trig-enb" I wanted to mention. Almost the same...

Latency compensation

OK I've been searching all night and can't find any delay modules that show the delay in terms of samples... I'm looking for a delay module that can delay by 1 sample, 2 sample, ... up to 127 samples. (In my case, 0 to 2.8798185941043083900226757369615 ms.) Anyone have any suggestions? There should be something, since some of the CW mixers have latency compensation delays in terms of samples...

pre-calculate the 'perfect' values (with whatever applies) and store them in a 1 dimensional array. Then use numbers_of_samples as an index to pick the proper value.

.44.1 samples is one millisecond (if you're running at 44.1 khz). In the effects/delay folder of the SDK, you'll see a module called "200 Delay DSP". The field readout on the module is in milliseconds, but as you adjust the knob, you'll see in the PadList that the value is in integer numbers...those are samples. Change the field on the delay's surface to the following pad settings:
Divisor: 1
Format: %1.0f smp
Max: 2147483647
You'll have a sample delay now!

as a mr. lind told me "the feedback mdl will give you a one sample delay" no more no less i might add......

Making a compressor in SDK

Do you have flexor? You can try patching it in modular, it's very easy to make a basic compressor:
input > ringmod in1 > output
> hyper follower > half invertor> ringmod in2
That's for a most basic compressor. Instead of ringmod, you could have use dynamic shaper set to max, and hyper follower to dynamic shaper's negative modulation for a more characteristic/coloured sound. Apply band split filter and send both to negative mod of the shaper and you have sort of a 2band compressor.
Just stating some simple methods to patch your own compressor, have a look at alfonso's compressor patch for more idea's.
It's quite easy to make a decent compressor in modular or SDK. But why wait for someone to make a device, if you can patch your own?

Can you build modular modules in the SDK?

can you? can sdk run in SFP 3.1c?
Does sdk installation changes anything in your current setup? (everything works perfect at the moment)

I've finally found time to spend with the (free) SDK ... and I can't figure out how to make a modular III module. For sure it's not in the documentation that came with the SDK. I think that somehow I need to substitute the modular style for the default style module ... but it's not obvious (to me) how to do it.
If anyone can shine a little light on this topic I'd appreciate a hint or two.
I've got several not overly ambitious modular ideas I'd like to try out

I used an old Modular I file to create my modules; I don't know about Adern. If your SDK doesn't include the 'Modular type' file, then you are not able to make Modular modules.

We do not have this modular file, but I'm sure there is a way to build some modular modules.
We just have to find out how (well, that's not the easier part...) (You can - strav100)

Blue screen and error message using Win XP

I have been developing using Win 98SE and have developed without any blue screen problems. My recent project Strings MK1 produced blue screens when the module Chorus S was used in Win XP by users.
Have any developers had problems with Win XP and blue screens with this or other modules?. Are there certain ways modules cannot be used?

nope. Causes could be multiple. Could be some connections on wrong pads, due to unnoticed mismanipulation, or something else.
Advice :
- redo this part of your dev from scratch, test it well (in SFP) then import it in your synth and redo connections carefully.
- in bug chasing, it is important to determine the cause of the problem and make various things that you will test to eliminate the cause of the problem.
- get a buddy to test your beta final version to minimize possibilities of bugs.

I changed the Chorus S for one from DSP module list and the device now works in Win XP as well as Win 98SE.

Modulating/ Warping Audio Signals with envelopes, lfo's etc.

can any developer give me an idea of how an audio input is synced with envelopes , lfo's?
i would like to be able to modulate/warp
the input signal - but i am not familiar with the routing and circuits required to get the effect
any help is very appreciated.

Just as the oscillators in a modular synth are manipulated, any extern audio signal can be processed inside a patch... I love to send a long recorded and stretched Flexor Ramp on its own ASIO channel to an External Effect module which sits in Insert Fx in my patches. I use Harm to multiply that ramp usually as a first step, then add a couple of other shapers that will make the actual audible modulation.
That's it for the tight modulation sync, which can also be sent over ADAT from for example Fruity on an RME machine instead as from ASIO... Each time we get together with a couple of machines we're amazed how tight sync can be! I get the same grin on my face as when I played Minimax for the first time.
If you don't have access to this technology, an LFO from Mod1 can also be retriggered with a MIDI note to control some filters etc, but I think Flexor is the way to go if I understand what you mean by 'warp'.
Flexor contains a whole bunch of shapers that can this time be used to add harmonics, filters to remove the excessive harmonics, and some granular stuff that can make some nice real-time pitch/scratch/reorder effects.

do you mean you're using the ramp to modulate controls on the effect, or generating audio from the ramp signal ?

I think he talks about getting modulation signals...
BTW never let a slow ramp reach your speakers, as it is a DC monster that can destroy them without mercy...
All the FleXor stuff is designed to be modulated by audio rate signals, and also the FleXor sequencers are moved by ramps.
This is a much tighter solution than using midi-gated data. With the shapers you transform the ramp and the modulation obtained, and instead of triggering ramps with a keystroke via midi, You can directly put these ramps on audio tracks, so your

modulation is "sample accurate".
As I said, never let such a ramp reach your speakers though!

GUI Design

I am interested in doing some GUI design work for up and coming scope synths, I wish I could get my hands on the SDK, but since it seems to have vanished some how I figure the next best thing would be to get involved doing some GUI design. After reading the Drum Box thread I started messing around with some ideas, here is a shot of just some ideas thrown into the mix to see how things lay out, its obviously far from complete but a start.

Anyway, I do not know exactly how the creation of Scope GUI's work but I am guessing its standard graphic formats. If any one is developing a device and is looking for some GUI's let me know as I would like to try this out soon. Here is the sample I started (picture not available – strav100)

Hi there, can I be critical, may be it will help ? that's why I will allow myself a few comments.
I think this gui is simple and efficient.
I like the background texture, very good. also the green LCD display (not sure it would look as good for DP, unless with the right font).
I would say, the gui is OK . It looks like a gui, but that's it. but that's because you did it like a test with no particular device in mind. Here, I see too much device defects to be able to judge the look favourably.
For example, there is no insert on each individual instruments : now to put this, the whole look would have to change a lot. That way, your current GUI wouldn't stand. Another one would have to be made, and it would have a different look.
It is not a matter of the individual buttons, the textures etc... no, no, no. It is more global and I am totally unable to judge the above picture because I am biased to judge GUI in terms of functionality rather than only the look.
I understand how you made it though, but please go on reading, it might be useful.
My philosophy is that it is only when a device is finished that it is time to think for a gui (but I think other do it just the reverse way).
My advices :
1/ don't think that, if no developer contacted you it is because of the GUI. It is more because of practical problems and financially, it is not very interesting for anybody. Plus some of us have particular tastes or skills that they want to practise.
2/ if you have the possibility/will, just make another version with a more "thought about" GUI. just imagine what your drumbox should DO (functions) and HOW (user

friendliness) .... the LOOK itself comes after (it0) counts a lot, but not more than the concept of the device itself - i mean what is the gui of a dev with a super look but no user friendliness or easy access to all functions ? .
Just my two cents, and I know this is not easy (it's just theory the hardest part is... screen size...
And now I have a qwerty keyboard and have hard time getting used to it (sorry for q=a and w=z never realised it could be so hard to type)

Actually you are correct, there is no actual device that goes along with this graphic, its just like a tes run to see if I could create a GUI, in fact this was GUI #1 hehe, I actually understand fully where your points are directed and it makes perfect sense, if I were to develop a GUI for a particular device clearly I would want to know exactly what capabilities the device contains, all information on where the placement of the rotary/faders/buttons are located, actual size of device, what important aspects of the GUI need to be pronounced and which need to be less upfront, does the device use paging or would all info need to be on the front screen etc etc.

I think I can add that is is quite difficult to work on a surface with a 'remote developer". When Laurent did the controls of Synthetic, there were several times when he had to redo a lot of things just because we added or removed a function, or moved some parts a few millimetres.... This, and add the fact that I am picky or had special requests for the general layout of the synths parts, etc... it is actually work and is not always fun.

Building a basic mixer with stereo aux sends

I"m going to try to build a basic mixer with stereo (!) aux sends. It probably shouldn't be too difficult, would it?

shouldn't be too hard. you will want to spend most of your time building the first channel and then duplicating it for the rest of the mixer. also put all of your controls in a "surface group" mdl on a dummy panel inside of the channel so all of your connections get duplicated also. then you can move the surface groups to the main panel later.

get used to using audio and controller pad mdl's when you want to export a pad on a mdl. it will save a lot of time if you make any changes to your circuit.

Close surface not working

I'm working with a module that I built from scratch, and I tried to stick a Close Surface on it...but the surface won't close! The "Always on Top" button works, but not the Close Surface. I even went into "Empty Effect" and saved the close button from there, but when I imported it into my own module, nothing happened. Any ideas?
My first device I'm working on with the new SDK is a simple panning device with MIDI support (which I got working, surprisingly enough). It's an amazing amount of work!

connect the close button with the show parameter of the panel module.

i have a problem with creating my plugin.
In the Develop env my pluginsurface seems to be ok. But when I open my module in sfp 4.0 a part of my skin is missing? The part that shows were knobs and faders I took out of the control dirs. But the other part, the whole limiter interface, I duplicated and put it on my surface. What do I have to do to really integrate these parts in my surface ?

before testing in sfp optimize and protect the device and save it. also for a quick save you can ctrl-drag the device into the file browser.
i have two computers running one with dp and the other with the sfp both networked. it saves a lot of time to not have to close dp

---

SDK Module question

I just got the SDK a couple of days ago, and I'm slowly but surely learning how it works...however, there seem to be some basic functions I need that I can't find. I found the unipolar to bipolar module in the huge DSP list, but I can't find a bipolar to unipolar module. And what I need right now most of all is a module like Modular's Diode -- basically it filters out all negative values. Does anyone know if there is a module like Diode, and if not, how I would go about making one? Nothing I found seemed to suggest anything like filtering out negative values.

the easiest way would be to download the one from http://www.steckenleiter.de/ but if you want to make one there are two ways.
1. run your signal into a bitchop set to 1 then into both inputs of a mix2. this will create a square wave that changes when the waveform crosses the zero. now use a sync multiplier with the original signal in one input and the square wave in the other. this is a full wave rectifier.
2.run your original signal into a mix2 and mix with -2147483647 run that through another mix2 and mix with 2147483647. this is a half wave rectifier.

You can use the 'If' (async) module or the 'compare' (sync) modules.

J9k, I tried your suggestion but I don't think the bitchop was working properly -- it didn't output anything. (I set N to 1 -- was that what you meant?) And then, adding all the other modules, the result was that I had a very narrow value range (I guess I should have said I'm working with value numbers, not audio). The rectifier module I downloaded didn't seem to do anything either. I KNOW ALMOST ABSOLUTLY NOTHING ABOUT ANYTHING THAT WASN'T IN MODULAR.

Warp69, I put in the "If" modules and fiddled around with parameters, but I couldn't get it to do what I wanted. I guess I don't really know how it works.

Make the No/Yes value taps visible. Set the 'If' value to zero and the condition to '>'. If the value is above zero the signal hats connected to the Yes value tap will pass through the 'If' module and if the value is below zero then its the No Value that will pass through the module.
Ex. : Connect the 'In' and 'Yes Value' taps and also connect the 'In' tap to a negative module and back into the 'No Value'.
That will rectify (full) any signal.
If you need a half rectified signal then the 'No value' should be constant zero.
Remember that the 'If' module is async.

Thank you so much! After much tweaking with the "If" module I got it to do what I wanted.

Circuit Design surfaces disappearing

Whenever I use a module out of the C:/Scope SDK/Circuit Design dir , the surfaces show ok. (example Limiter) But when I export the plug as module , the surface of the Limiter won't show. All manually inserted surface objects do show properly. Is it possible to use these surfaces out of F:/Scope SDK/Circuit Design ? Or are they somehow protected only to use them without surfaces ?

Save a "protected" version of the device to load outside of SDK. Obviously you still keep the unprotected one for modifications.

How to get VU meter to show

I'm trying to build a device that has VU meters. I found the VU meter 2 circuit, which is great, but there was no corresponding representation in Surface Design. So I saved out the visual meter that was in VU Meter's surface. I loaded that into my own device, made all the connections, it worked great. I loaded my device into Scope Platform 4.0, and there was no meter!!
I can not remotely figure out why it will display in the SDK and not in the normal Scope software. Does anybody have any ideas on how to resolve this?

Hey, I had the same issue. What I did wrong is that I didn't protect my device.
If you don't protect, every item that was dragged from the circuit Design dir disappears in the sfp4.0!
So now I save 2 copies of the same project. One unprotected for editing, and one protected for testing in SFP4.0.

Thanks -- that worked! It's annoying but it's OK. Unfortunately I had deleted all my VU meters in disgust, and now I'll have to put them back in! Oh, well....By the way, I've learned something...if you have too many surface groups the GUI slows down to a crawl. I moved some surface children to parents and it got faster. So if you ever have performance issues, you can try that.

DSP Management question

I'm working on a 16-channel mixer with stereo bus sends...I originally didn't think it would take up much DSP, but it now is. I've been putting DynVoicesOfParent modules set to 0 on every channel and every bus send, but it hasn't helped much -- with only a mute button, volume, pan, and a send to main mix button, each channel is taking up a large amount of DSP power. In fact, when I put DynVoicesOfParent ON THE ENTIRE DEVICE and set it 0, this mixer was STILL taking up 3 notches of my 22 DSPs. Why is this happening??
Does anyone have any tips? I'm getting pretty frustrated. I tried "Optimize - XTC compatible" but XTC on a mixer doesn't work, and it threw up lots of errors. I also tried deleting unused panels attached to the internal circuts, but all those run on ASync so it didn't make much difference. I have no idea why everything is taking up so much DSP power.

Its pretty hard to help you with that one, since you don't tell which modules you use.
But DynVoicesOfParent don't have any impact on modules which uses a lot of DSP ram.

Most of the device is made up of gain controls, panners, 1/4 switches, 4/1 adders, and insert slots. I don't think the insert slots take up any DSP when they're not loaded, and the rest of the modules aren't fully turning off.

Are you using the softclip module? You could replace some of the modules, like the 24dB gain module with Add02L (that will give you 6dB gain). I don't have access to the SDK so I don't know which modules you have - you have probably more than me.

I meant gain control as in Mixer Master Gain -- i.e., volume fader. J9k graciously offered to check out my device, but if you want a copy of it too, I'd be happy to give it to you. It's 10MB, though (when I initially made the device I inserted pan modules and adding modules that all had their own surfaces, and now I'm having to go through the device and delete all those surfaces -- it's a lot of work and hasn't been finished yet, so it's bigger than it should be; doesn't affect DSP load, though).

Insert slot question

I'm getting near to the final stages of building my new mixer. However, I have run into a problem...all parameters save with the project *except* for the parameters of insert effects!! The insert slot loads the effect but doesn't save any of its parameters, and I'm not sure how to do that. Does anyone know?

You have to register the insert module's 'Path' as a preset parameter, that's all.

I dragged "Path" into the Parameter list, both on the parameter list of Effectloader and the parent device, and dragged "Path" into the Preset Parameter list, and the settings still don't restore in the project (I made sure restore and store in project were set in the parameter list). What else do I have to do? Am I doing something wrong?

not sure but i think it is the insert itself that has to be parametrized in addition to the slot. or it will always load with its default values (+ before it is active you save and reload the dev).... (one or the other solution i think).

Nothing I'm trying is working. What EXACTLY must be parametrized in which modules and should I do vars or pads?

Ah, sorry, yes...you must parametrise the EffectLoader module's Path, not the insert slot itself.

I did that, and it didn't work.

OK, I want to go over this STEP BY STEP to see if I'm doing anything wrong. I really want to get this to work so I can release my new mixer with STEREO bus sends.
1. I open up the Parameter dialog box and freeze it on the parent module. I'm using Empty Effect with an insert slot to test, and there are four entries...Keep on Top, Close Panel, etc. There's a tree on the left with the same items. I've also tried using the parameter list for EffectLoader.
2. I open up the Parameter Preset dialog box. I've already created a new preset list for the parent device.
3. I navigate to EffectLoader. I drag the pad "Path" into the Parameter box. All the save options are checked except for the screenset options and "Load Default."
4. I drag the Path parameter into the parameter tree on the left of the box.
5. I drag the Path parameter into the Parameter Preset box, which is showing "NewPreset" or something like that.
6. I save the device.
7. I load up Scope 4.0. The presets do not affect the effects in my device. I tried parametrizing the "Load" button for the insert effect, and that worked. But the effect presets are not working.
I also tried dragging "ParameterWrap" into the Parameter box, but that didn't work either.
Please tell me what I'm missing here!! I'm really anxious to show you all what I've been working on!

OK -- I figured it out!! I have to save the device as .dev and not as .mdl. If it's .mdl it won't work, if it's .dev it'll work.

How to install SDK

Install is straight forward.
Execute the setup.
Then unpack the zips in the sdk folder.
Use the unpack with subfolder structure option if this is not your default anyway.
These packages contain all the SDK modules you need to build devices.
If you already have a normal scope 4 installation you don't need a driver update.
You have already the newest drivers.
Don't worry about installation, Scope and SDK coexist without any problem.

Finally regarding the Manual I am afraid that you are right.
It is very sparse, not to say minimalistic.
The links between the PDFs worked when unpacking in the same folder on the same level i.e. without subdirectory.
And now..Welcome as a SDK developer

Configure logic control

I´m attempt to config Logic control, I understand all, but not this algorithm.

Header: F0 00 00 66 10
Host send: <Hdr> 00 (F7) Device Query
Logic Control send: <Hdr> 01 ss ss ss ss ss ss ss ll ll ll ll (F7) Host connection Query

Host must calculate four variables.

Host send : <Hdr> 02 ss ss ss ss ss ss ss rr rr rr rr (F7) Host Connection Reply

ss = Serial number (7 bytes)
ll = Challenge code (4 bytes) Random
rr = Response code (4 bytes)

Algorithm:

l1 to l4 = challenge code bytes 1 to 4
r1 to r4 = response code bytes 1 to 4

r1 = 0x7F & (l1+(l2^0xA)-l4);
r2 = 0x7F & ((l3>>4)^(l1+l4));
r3 = 0x7F & (l4-(l3<<2)^(l1|l2));
r4 = 0x7F & (l2-l3+(0xF0^(l4<<4)));

My doubt;
& = and?

yes, bit-wise and operator e.g. 10001001 & 10000001 = 10000001

>> = ? I think = bit´s to Right
example l3 = 0x23 '100011' >>4 = 000010 ???
yes - right-shift bits e.g. 11110000 >> 4 is 00001111

<< = ? I think = bit´s to left
example l3 = 0x23 '100011' <<2 = 001100 ???
yes
edit: one usually deals with 8-bit bytes, so your example might be better expressed as:
00100011 << 2 = 100001100

| = or?

bit-wise or & bit-wise exclusive or
or: 11110000 | 00001111 = 11111111

xor: 11110000 ^ 11000011 = 11000011

This operators in Windows calculator?
no, not all

& = key 'And' It´s correct?
^ = key 'x^y' it´s correct?
| = key 'or' it´s correct?
<< and >> = key '?'

Can you calculate a example for me?
l1 = 77, l2 = 3B, l3 = 23, l4 = 0C

I'll do one line for you:
r3 = 0x7F & (l4-(l3<<2)^(l1|l2));

(btw, it'd make reading MUCH easier if you use a capital 'L', not the lower-case one!
Compare: L1 & l1 - Are all the values you're giving hex? It'd also help to make that clear: write 0x77, not 77

L1 | L2 = 0x77 | 0x3b = 0x3f
L3 << 2 = 0x23 << 2 = 0x8c

At this point, one needs to know whether the exclusive-or takes place before the subtraction - is this dependant upon operator precedence, do you know, or is it left-to-right processing?
i.e. is the sum

r3 = 0x7F & ((l4-(l3<<2))^(l1|l2));
or
r3 = 0x7F & (l4-((l3<<2)^(l1|l2))); ??

IF we're strictly using operator precedence, then in fact '-' has a higher precedence than '^', so the next stage is:
L4 - (L3 << 2) = 0x0c - 0x8c = 0x80 (IF we're using byte-sized arithmetic!)

edit: i.e. 0x80 represents -128 (sorry, got the sum wrong first time!)
The next stage is to X-or the two together:

0x80 ^ 0x3f = 0xbf

I think you need to know what the order of calculation in the equations is before you can calculate them properly.
edit: actually, both operator-precedence and left-to-right processing would dictate that you perform the subtraction before the exclusive-or.

Bit wise in SDK

Hello, do you know a dsp module for make bit displacement left and right?.
DP developers! can you help me?
I think in a module with four pads
1: Input data
2: Number of bit displacement
3: Direction of bit displacement
4: Output data

You could use multiply modules for that.

I remember now... I must multply or divide by two...

FM Sine Operator

I am trying to use the FM sine operator (in the SDK oscillators section) and have not yet figured out how to use it correctly. The FM sine operator does not seem output a sine wave on its output,like say a saw oscillator does and no audio output is detected for a changing input frequency. Does anybody now how to connect the FM sine operator's correctly so I can use the frequency control and the frequency ratios and so construct a synth?

Well, I know that in the Modular you need to hook an envelope generator directly to the operator before anything will work.

Suggestions for first SDK project

I'm looking for ideas for my first SDK project. Something simple but useful that would be a good way to get my feet wet. Any suggestions?
What are you guys working on? Does anyone feel like sharing their projects with the rest of us newbies; something we can use as a tutorial?

j9k suggested a small mixer. I would add: anything simple that you have missed

Thanks for the suggestions...keep them coming! I built a simple guitar distortion stomp-box this morning. Nothing I couldn't have built in Modular, but it was good practice. I only built the circuit, so maybe I'll work on a GUI for it tomorrow.

that's the proper way of doing it
start with something you're familiar with in real world - analyze it's function and control chain - find the elements in SDK that could represent these items - arrange and check if it's behaving as expected.
Modify the flaws (which are likely to show up) and possibly look for alternatives that may replace existing blocks by new arrangements of simpler elements.
Don't bother with the user interface too much (in the beginning) - it's always the most sophisticated par
pretty obvious blurb, but it's often neglected by wanting too much.
The more systematically you study the simple blocks, the faster revelation shows up and things almost flow by themselves.
It has been mentioned here that music and programming aren't that different.
another suggestion from the midi world which would be nice to have and doesn't exist yet:
an omni-note changer - whatever midi note arrives, it outputs the same predefined note.
In case you wonder: all SFP drum modules (for example) rely on a certain note set for the instrument (voice).
If you have existing multi-channel patterns, you'd have to modify each stored note to the appropriate value for the desired instrument.
With such a device whatever is stored on a channel automatically triggers the right drum.

Sorry if I have to say it again : for those who have Reaktor there are plenty of ideas / paths already explored, sure you can't do (modules equivalents + ...) and possibly are not willing to do 1:1 ports but there's quite some brain spilled there since some years !
It's the *only* reason, I'm hanging on my Reaktor license, for later ...
Also Synthedit creations to study and others ... (insert names - ideas)

Midi questions

I have to develop a MIDI device for convert and filter midi msg.
For example: Incoming note on, I want convert this in controller and the assigned MIDI channel.
How I can to fill a array for send complex MIDI msgs? And how I can to fill a array from MIDI msgs?

I've tried playing with FIFOs and Arrays, only managed to get SDK to crash though. I don't think it's a good idea to mess with Sysex on this platform, even if you manage to figure out arrays and build something, it's going to eat a terrible amount of DSP.
I got a simpler MIDI message transformation thing in the works, almost done, just debugging stuff and hooking up the interface. Kind of a clone of Logic's MIDI Transformer object. MIDI decoding is trivial. For re-encoding, the module you want to look at is the MIDI Control 2 Number module.

Missing folders in SDK

I installed in e:/SDK I missed the three folders, which I then manually copied from the packages.zip on the CD. I still miss the tools folder though, so I can not find the oscilloscope from the quick start guide. Are other people missing this too? Do you have any idea how to get it?

What quick start guide? I never got a quick start guide.

It is in the manual folder. Just search a step deeper, then you find it.

Nope. No quick start guide. The SDK CD has a DPManual.zip file with only six items:
Circut Design Tutorial.pdf
SCOPE SDK Manual.pdf
Circut Design.pdf
File Management.pdf
SCOPE Paradigm.pdf
User Interface.pdf

According to the items listed in the Scope SDK Manual, I'm missing:

ReadMe.rtf
Installation Guide.pdf
QuickStart.pdf
Surface Design.pdf
Device Design.pdf
Appendix.pdf
Edit nfo Tutorial.pdf
DSP Programming.pdf (I don't care about that!)

I miss the same files! Same files are there , and same are missing. I got the SDK two weeks ago. Different versions perhaps ??

I don't think it's a problem to give me the files because the PDF I have says I'm supposed to have them. I also got the SDK a week or so ago, I think they just forgot to include the files.

I received my SDK CD on tues this week. I have the PDFs but i am also missing the tools folder-shame really cos i could really do with an oscillloSCOPE! Did you manage to get the missing tools folder?

As far as I know, the "missing" folders are from an earlier version of DP. I have this old DP demo that doesn't have any audio enabled, and it includes the mentionned documents. That said, some future version of SDK/DP might include them, I don't know. Before anyone asks, they're not terribly more useful than whatever comes with SDK. It's not even the same software exactly, so it's more confusing than anything.

SDK Development tip

I'd suggest to get a little distance from typical programming details in the context of SDK and first try to get an overall viewpoint about the system's basic strategy and capabilities.
Even if it's called a SDK it's in no way related to traditional programming languages.
It certainly cannot do all and everything, but some limits at first sight can be worked around - more or less smartly, once you get more familiar with it.
You may be surprised, but there's a rich source in Alfonso's modular devices regarding the strategy you need to apply to achieve a certain processing in SFP.

Problem with DynVoice of Parents

I get and issue with the polyphony. My Synth was working correctly and suddently with SFP4 it only works in mono.
As soon as I add more voices the synth drop midi info and plays only every 2 notes (for a 2 voices setting) 3 notes (for a 3 voices setting) and so on.
If I load the Empty Synth and create a small synth with just a 'Voice Control 16' + OSC I have the same issue.
Does the DynVoiceofParent atoms has changed in the last SDK version? Does anyone else has the same issue.
Note : this seems also to be the case with the Neutron Projects...

You're using the PolyOut16 (or something like that) module at the end of the polyphonic chain, right?

SDK Project window bug

When I load a new project or one of the user projects I get 2 windows - one where I can add modules and wire them together (but if I move them about the pins/red blobs on the side of the modules get left behind/ drop off the module) and another one with a grey border with the old scope logo in. If I click on this the grey border suddenly says 'no project' on the top and the SDK shuts down and brings me back to the desktop. This seems a really buggy and shitty environment to work in - or is my install corrupted?????

I don't get the grey window with a new project. Simple solution: don't touch the grey window =P Might have gotten it with the Neutron stuff, which comes from a different version of the software (sounds like it's your problem also, given the "old-style" logo.). And it's not really "shitty", more like "software for internal use." I think it's great that it was released, if you don't like it, you can probably get an AD/TI DSP dev kit and code your own =P.

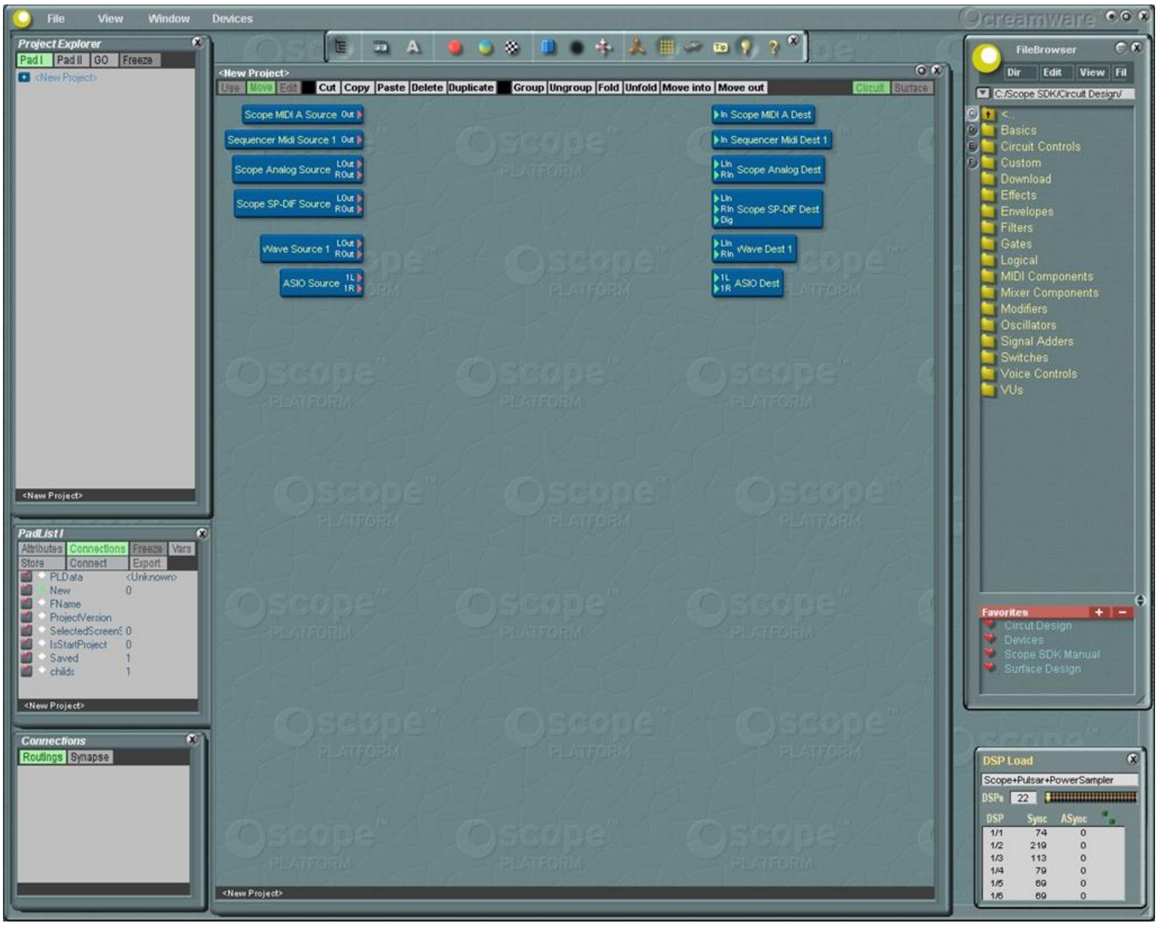
the bug where the pads float off the mdl has been there forever. i don't know about the rest of it.

If you load the Neutron or old DP projects, remove first all your Modules/drivers from the current project, then import it from the file browser, highlight the device/circuit you're interested in, save it and you have it ready to be loaded in a normal sdk pro. This is what I came up with to avoid unwanted phenomena.

I had the same issue when loading one of the Neutron projects.
just delete all the doubled or non existing I/Os save it as device not as project.
Then you're rid of the project window ghost.

Appendix: First Time Use by Shayne White

This tutorial's purpose is to introduce you to the SCOPE SDK software. It will cover the basics of using the software in addition to providing you with the fundamentals of assembling a device.
To start, run the SDK by double-clicking (or single-clicking, depending on your Windows customization) the *Scope SDK* icon on your Windows desktop. Please note that all other Scope applications must be closed before running the software!
The SCOPE SDK will now appear. It will look similar to the picture displayed here.



You will see several windows: the main SCOPE SDK window, which currently houses the Project Window, the File Browser, and the Toolbar. Other windows will be opened and used later.
The Toolbar gives you quick access to many of the components of the SDK. Most of the items on the Toolbar are graphics options for designing device surfaces which we will discuss later in this tutorial.
The File Browser allows you to navigate your hard drive and load modules and other items into the SDK through a drag-and-drop method. (For more information on the File Browser, see *Software Interface Guide*.)
The Project Window is where you will be doing most of your work. As you can see, it is blank except for a number of blue-colored boxes called *modules*. Modules are very important and will comprise the underlying infrastructure of your device. There are hundreds of modules available that can be loaded into the Project Window. Modules can perform a vast number of functions, and they can be connected together in various ways via *pads* and virtual "cables." The *Circuit Design Tutorial Part 1* contains more details on how modules work and interact with each other. Devices will be made up of multiple modules.
A *project* saves the current configuration of the modules and devices, including how they are arranged on the screen. Projects also contain certain settings related to the SDK program itself. It

must be noted that a project can save modules inside of itself, whereas it must *reference* devices directly off the hard drive. In the case that devices get moved to another directory, the SDK can search for them and load them automatically. Later in this tutorial you will be building a device inside of a project.

As you can see in the default project, there are several modules loaded. The default modules represent your SCOPE software and hardware I/Os, building the basis of your project. If you have multiple hardware boards, the project will display the basic I/Os of the first board. It is likely you will want to customize this default project to your liking.

Initial Setup

Before you begin your work, you will probably want to customize the SDK to appear exactly how you want it to look. Some changes will be global: i.e., the software will look and act a certain way no matter which projects are loaded. Other changes will be project-based.

At the top left of the main SDK window, you will see several menus: File, View, Window, and Devices. The File menu is for loading and saving projects. Once you have customized the Project Window to create a default hardware/software I/O configuration (one that you will want to load automatically every time you load the software), you will want to save your project via the File menu. The File menu contains the following items: New, Open, Save, Save As…, Open Recent (via a sub-menu), and Exit Scope.

The View menu is where you will customize global and project-based settings. The menu contains the following items: SampleRate, Settings, DSP Load, Registry, MIDI Controller, Control Ranger, and About.

SampleRate

SampleRate changes a number of hardware settings. The SCOPE hardware is always operating at a certain rate (number of samples per second), which can be one of four rates: 32KHz, 44.1KHz, 48KHz, and 96KHz. For best compatibility with other hardware and software, leave at 44.1KHz. Changing the sample rate affects the sound quality of your devices (higher sample rate = higher quality), so you may want to change this setting sometimes to see what effect it has.

Most of the time you will want to have your SCOPE board(s) generate the sample rate (clock) signal. If, however, you have your SCOPE boards connected to other external hardware, you might want SCOPE to sync with the other hardware's clock signal. To do this, determine which SCOPE input the other hardware is connected to (must be a digital connection, not analog), then select that input in the list. Click on the "Slave" button. *It is advisable to have the speaker volume turned down in case of transients!* In a moment SCOPE will readjust itself to the other hardware's clock signal. To return to a SCOPE master signal, click on the "Master" button. If the "Slave" button is activated but no other hardware is present, SCOPE will not operate at all until a signal is detected. Typical digital inputs will be *S/PDIF* (Coaxial), *ADAT A/B*, *S/MUX*, and *S/PDIF Optical*. The various inputs on your unique hardware boards will all be displayed at once, allowing you to select one input for clock detection. There is an additional setting as well: allowing you to select between S/PDIF Optical and ADAT, as these two protocols go over the same I/Os. This option acts slightly differently depending on which I/O daughterboard you have on your hardware (*Scope Project* and *Scope Professional* only).

If you have a Classic or Plus board, and the ADAT option is selected, an S/PDIF signal is permitted using the S/PDIF modules, as well as a 16-channel ADAT signal via the ADAT I/Os. If the S/PDIF optical option is selected, the ADAT A port switches to 2-channel S/PDIF optical. Output is transmitted via both the optical and coaxial S/PDIF ports, and coaxial S/PDIF input is disabled. Use the normal S/PDIF modules to access the S/PDIF optical I/Os. ADAT B continues functioning in 8-channel ADAT mode.

If you have a Z-Link board, there is only one ADAT I/O. In this case, the S/PDIF coaxial is unaffected by either setting. If the ADAT option is selected, the ADAT port functions in 8-channel ADAT mode. If the S/PDIF optical option is selected, the ADAT port switches to 2-channel S/PDIF optical. Use this protocol via the special S/PDIF optical modules.

The SampleRate settings are project-based.

Settings

This box is divided into several pages. By click on one of the page buttons, you can switch to that page. The pages are: Global, Config, ULLI, and Directories.

Global:
Enable Control Shortcuts: Don't know what that does
Pot Movement: When making a device surface it is likely you will use rotary knobs and linear sliders, called *potentiometers*, to control parameters. There are three methods of adjusting the potentiometers (pot for short). Round 1 uses a "brute-force" method: when you click on the potentiometer, the pot will automatically "jump" to the place where your cursor is located, creating a large shift in the sound. In some cases this may be rather dangerous, and it is not advisable to use this method. Round 2 is the same as Round 1 except it does not "jump," and it only moves when your cursor moves in a relative fashion. In both Round movements you must move the cursor (holding down the mouse button) *around* the pot to adjust it (linearly for sliders), and the further the cursor moves away from the pot, the finer the range of control becomes.
Vertical is an easy method: both knobs and sliders get moved linearly—up and down for vertical sliders and knobs, left and right for horizontal sliders. The range of control does not change when the cursor moves away from knobs, but it does change when moved away from sliders.
Screen Mode: In earlier versions of SCOPE all components and windows were contained within one giant window (or desktop) which was usually confined to one monitor (dual-monitor setup was difficult and cumbersome). This is now known as "Classic" mode. In version 4.0, every window and dialog box displayed is independent of the main SCOPE desktop, which allows for easy movement to a second screen. It is advisable to ensure this setting is in the new "Standard" mode.
Standard Cursor: SCOPE has its own unique set of mouse cursors that assist in assembling devices. It is possible to use the standard Windows cursors instead by selecting this option, but it is best left deselected.
Enable Tooltips: When hovering the cursor over certain items, a Windows-style tooltip will appear with an explanation of the function of that item. Usually this setting should be selected unless the tooltips become cumbersome in some way.
Save Recovery Files: Don't know what that does.

Config:
Don't know what any of that does.

ULLI:
The adjusts the *latency* of the Windows wave and Steinberg ASIO audio drivers. As the computer has to process audio in certain segments (buffers), not in a continuous stream, there will be a slight delay (latency) moving the audio from the host computer to SCOPE. A lower buffer, or latency, will result in greater performance, but increased CPU usage; a higher latency will increase the delay, but utilizes fewer CPU cycles. Slower computers will have to leave the latency high, whereas faster computers can decrease the latency. This setting does not affect the internal processing of SCOPE, only the wave and ASIO drivers.

Directories: Not sure what this does either.

All options in Settings are global.

DSP Load

The SCOPE boards can only process so much information at one time. The more DSPs you have on your hardware, the more devices you can load. The DSP Load box can tell you how much power you are using on your DSP boards. It also tells you how many boards you have (up to 3) and how DSP chips are on each board.
You will also see two columns: *Sync* and *ASync*. These terms *must* be familiar to you when you build a device. See *SCOPE Concept Guide* for more details.

Registry

Certain devices from CreamWare and third parties are copy-protected: they contain a *key* that is tied to one of your DSP boards. This dialog box tells you how many devices are registered with your unique boards, as well as the serial numbers for your boards and the key numbers for your devices. These numbers have very likely been imported from the Scope Platform 4.0 software or from CreamWare's Online Shop.

MIDI Controller

If you are working on a device with MIDI capability, and you have a surface with potentiometers, you can assign a MIDI controller number to the pot by clicking on the pot, waiting for a MIDI controller signal, and clicking the + button when one appears in the "New" section. To remove the controller number, click the – button.

Control Ranger

This is a part of MIDI Controller, but at this time it is not functional.

About

This brings up the programming credits for the SCOPE SDK.

Congratulations! You have now set up all "under-the-hood" settings. Next we will adjust the display of the SDK windows to fit your needs, and then we can begin building a device!

The next menu on the menu bar is Window. These items can open windows and components of the SDK which will assist you in building a device. The items on the Window menu are: Toolbar, ProjectExplorer, Project Window, File Browser, DSPModuleList, Parameter: Parameters and PresetParameters, GOTree, Alignment, Blit, Color, Dimensions, Gradient, Preview, Shadow, Text, and Texture.
You should see checkmarks next to Toolbar, Project Window, and File Browser; that's because they're currently open. Click on ProjectExplorer.
The Project Explorer is a companion to the Project Window. It shows you everything that is loaded in your project as well as its subsequent devices and modules. It is organized in a *tree* structure, with *parent* and *child* modules. We will discuss the uses of the Project Explorer below.
The four windows you now have open you will need to access frequently, so it is best to leave them open permanently. Two other windows should also be open permanently:
In the Project Explorer, click on *Pad I*. This brings up *PadList I*. You will see an option to open

Connections; open that as well.
The Project Explorer, the PadList I, and Connections will all be used together, so they should be located very near each other on your screen. Note how the SDK is arranged in the following example.
We are now ready to build a device! It is advisable to read the *SCOPE Concept Guide* and *Software Interface Guide* before continuing, so that you can get an idea of the underlying structure of device creation and functions. If, however, you are too anxious to wait any longer, you can just try experimenting along with this tutorial.

Building a Device

Our first device we will build will be a simple lowpass filter controlled by an LFO (Low Frequency Oscillator), complete with volume and pan modulation. To start, navigate the File Browser (for more information on the File Browser, see *Software Interface Guide*) to Scope SDK → Circuit Design → Basics. Click on Empty Insert FX and drag it into the Project Window.
You will see a new module in the Project Window as well as a module surface. Close the surface for now. (*To open a module's surface again, right-click on the module and navigate to Surfaces → Open Panel.*) Connect the input pads of the module to the desired audio source by clicking on the pad *InL* and immediately afterwards clicking on the pad *LOut* of *Wave Source 1* (or whatever audio input you wish to use). Then press the letter *n* on your keyboard to hook up the right channel. Then connect the pad *OutL* of Empty Insert FX to the pad *LIn* of your board's *Analog Dest* (or whatever output you desire). Double-click on the module.
You are now looking "inside" of the module. Further modules can be dragged into this space, which in turn can viewed inside. These modules are known as *child* modules inside of a *parent* module. You will notice that the Empty Insert FX is now displayed inside the Project Explorer. You will also see that all the modules contained inside of Empty Insert FX are displayed in both the Project Window and the Project Explorer. It must be noted that while the Project Window can only view the insides, or *level*, of one module at a time, the Project Explorer can "see" everything at once.
The uses for most of the modules contained in Empty Insert FX do not need to be explained right now. You will see two modules with labels next to them: Effect Input and Effect Output. You need only pay attention to those modules for now.

Creating Circuits

Main Effect Circuit

In the File Browser, navigate to Circuit Design → Filters → Sound Design. Drag the *LP 24db Vintage* module into the Project Window TWICE, one below the other. Connect the pad labeled *in* on the first filter to the pad labeled *outL* of Effect Input, and connect the right output of Effect Input to the *in* of the second filter. Then connect the pad labeled *out* on the first filter to the pad labeled *inL* on Effect Output, and connect the right input of Effect Output to the *out* of the second filter. You have now created a *circuit*! (See *SCOPE Concept Guide 1*.) When you run audio through your new device, you will hear the effect of the lowpass filters. However, at the moment you can only control the filter of the left or right channel at one time. To control them both at the same time, further modules must be inserted.
Double-click on one of the filter modules. This will take you inside the filter module. Right-click on the two modules you see, SurfaceInterface and Panel, and delete them by selecting *Remove Module*. Return to the previous level by double-clicking on a blank space of the Project Window. Do this same step for the other filter.
Now drag in a surface control by navigating the File Browser to Surface Design → Controls → Potentiometers. Drag *Pot Dark* into the Project Window.

You will see a pad labeled *Val* next to the knob. Connect it to the inputs labeled *f* on the two filter modules. These cables are colored green because they are an *ASync* connection (see [*SCOPE Concept Guide*]).
For the moment you are done with the circuit. You can arrange the position of the modules on the screen by clicking and dragging them; good organization and low clutter is the key to building an efficient and effective device.
You can now control the lowpass filters' filter control by clicking on the *Use* button in the top left corner of the Project Window. This allows you to manipulate the knob's output value instead of changing its position on the screen. By adjusting the knob you should be able to hear it filter out the high frequencies of the audio as you turn it to the left.
The response curve of the knob may seem strange as you adjust it: there will be a huge sweep in the audio during the low values of the knob's output, and at about ten o'clock the sweep will have nearly reached the top. We will now change the pot's internal response to fix this.
As was mentioned before in the window arranging section, the windows Project Explorer, PadList I, and Connections should all be open. The Project Explorer can open PadList I by clicking on the *Pad I* button, and PadList I can open Connections by clicking on the *Connections* button. Arrange these windows near each other on your screen.
Having clicked on the pot, go to PadList I and click on the button labeled *Vars*. This will show you the available internal settings for the pot. You will see that *Curve* has a value of 0, and *Intensity* has a value of 10. Change the value of Curve to 1 and the value of Intensity to 100. Now the curve response of the knob should be satisfactory.
If you adjust the knob quickly, you may hear a "zipping" noise. This is because the resolution of the knob is coarser than the resolution of the filters. An interpolate module must therefore be inserted between the pot and the filters.
Disconnect the cables between the pot and the filters by clicking on the individual cables and pressing the Delete button on your keyboard.
Click on the Window menu and select DSPModuleList. This brings up a list of the DSP modules available. These modules are different than the modules you have been dragging in previously: they are the lowest-level modules available, also known as *atoms*, containing only executable code and no further modules inside. They have no surfaces for this reason.
Scroll down until you see the module *Interpolate*. Drag this into the Project Window. Connect the *Val* output of the knob to the *In* of Interpolate, and connect the *Out* of Interpolate to the *f* inputs on the filters. You can adjust the position of the modules by clicking on the *Move* button next to Use on the top left corner of the Project Window. In Move mode you can adjust the position of modules on the screen as well as enter into a module's level. Switch back to Use mode to adjust the knob's output value.
The filter sweep of the knob should be much smoother now. It is very fun to play with this sort of thing!
As you can see, there is also a resonance control on the filters. This boosts the signal right at the point of the filter cutoff. You can control the resonance in a similar manner to the filter cutoff: switch back to Move mode, drag in another Pot Dark, and connect the output to the Res. inputs on the filters. You probably won't need to insert an interpolate modules to control the resonance. Be careful that you don't turn the resonance too high!

Modulation Circuit

So far, you have only been able to control the filter with mouse-based controls. How about some internal controls, such as an LFO (Low-Frequency Oscillator)?
To begin this next phase, disconnect the cables between the Interpolate module and the filters. Bringing up the DSP Module List again, find *Pitch Modulator 2 R5* and drag this into the Project Window. This will allow an LFO to modulate the signal output from the pot. Connect the output of Interpolate to the green *f* input on Pitch Modulator, and connect the red *f* output to the F inputs on

the filters.
Next, in the File Browser navigate to Circuit Design → Oscillators → LFOs and load Tiny Sine LFO. Keep the LFO's surface interface open where you can get to it easily.
Connect the output of the LFO to *m1* on Pitch Modulator. The frequency of the LFO is currently set to 40hz; change that to 1hz. You can adjust the gain (volume) of the LFO as well as the hard output of the filter pot to get a wide range of modulation coverage.

Mixer Circuit

Now let's add some volume and pan modulation—basic mixing tools.
Switch to Move mode and disconnect the outputs of the filters from the Effect Output module.
Navigate to Circuit Design → Signal Adders → Mix and drag in *Mix 1* twice, one below the other. Position them between the filters and Effect Output. Connect the output of the first filter to the input of the first Mix 1 and connect the output of the second filter to the input of the second Mix 1. Close both surfaces of the two Mix 1s for now.
Navigate to Circuit Design → Mixer Components, and drag in *Script Pan*. Connect the output of the first Mix 1 to the left input of Script Pan, and connect the output of the second Mix 1 to the right input of Script Pan. Then connect the left output of Script Pan to the left input of Effect Output, and then press the *n* key on your keyboard.
Change the left and right panning of the signal by adjusting the pan knob or entering in an amount in the Script Pan surface. Double-click the knob to return to Center and close the interface.
You now have a pretty complete signal path for this type of effect. You have a stereo lowpass filter with LFO modulation, along with volume and pan modulation. Of course there are loads of other goodies you can add to this effect, but for now we'll keep it fairly simple.

Circuit Organization

The circuit is beginning to look a little cluttered. For such a simple device there seems to be a lot of modules! How can you clean this up? You can clean it up by *folding* modules.
Folding modules means taking two or more modules and combining them into one module. For example, we can take the two lowpass filters and fold them into one module while preserving the I/O connections. When you double-click on the folded module, you will be able to look inside the module and see the two filters you folded together.
Click on a blank space in the Project Window to deselect all modules. Then select the two filters by shift-clicking them (clicking on the modules while holding down the shift button on your keyboard). Fold them together by pressing *<Ctrl>-<F>* on your keyboard.
The two filters will disappear, and you will see a new module titled: *Folded_LP 24db Vintage*. The pads are still connected, and when you double-click on the module, you will see the two filters, which are now child modules of the folded module.
What are those little arrows you see around the pads of the filters? Those signify that the pads are *exported*. Exporting means sending a module's pad to the parent module. As you can see, every pad on these filters is exported. When you double-click on the Project Window to bring you up to the previous level, you will see that all those pads are likewise on the parent module, *Folded_LP 24db Vintage*. Any pad on any module you see can be exported, and we will manually export our own pads later in this tutorial. Look at those Switch modules at the top of the Project Window; you can see some pads are exported. Those are the audio I/Os that you connected when you first loaded up Empty Insert FX.
Let's fold some other modules now.
The two knobs plus the modules Tiny Sine LFO, Pitch Modulator, and Interpolate all serve one purpose: to modulate the filters. We'll fold those five modules together.
Again, click on a blank space in the Project Window to deselect all modules. Then shift-click the five desired modules (be careful when clicking on the two pots: don't click on the connection pad)

and press <*Ctrl*>-<*F*>.
It doesn't appear that these modules folded together correctly. The output of Interpolate is connected to Pitch Modulator *outside* the new folded module. This is because of the Sync/ASync connection. This can, however, be fixed. Double-click on the new folded module to bring you inside of it. You will see arrows on Interpolate *Out* and Pitch Modulator *f* signifying that they are exported. How shall we fix this? *Unexport* the pads.
Right-click on the pad Out on Interpolate and select *Delete Exported Pad*. Do the same procedure on the f pad of Pitch Modulator. Now connect the two pads together.
Return to the previous level, which is the main view of Empty Insert FX.
We can fold the Mix modules and Script Pan together to form a *Mix* section. Select the two Mix modules and Script Pan and press <*Ctrl*>-<*F*>.
This is a much cleaner setup now, isn't it? If you had many more circuits in this effect, you could fold together the three folded modules to clean it up even further. However, the way it is now is sufficient for this effect.

Naming Modules

We have a couple more tasks to perform before we begin creating a device surface.
Now that the circuits have been completed, it's a good idea to save your new device. Double-click on the Project Window to move up one level. Right-click on Empty Insert FX and select *Save as new Module*. At the moment it only allows you to save the device with an extension of .mdl, which should only be used for individual modules, not a full device. In the drop-down list labeled *Save as Type:*, select *All Files*. Navigate the save dialog box to the directory of your choice, and save the device as: *Tutorial Device 1.dev* .
Once it has been saved, double-click on the device again.
The module names *Folded_This* and *Folded_That* are not very pleasing. Also, some of those modules you have been working with, such as Interpolate and ASync Add, would quickly get lost if you had a lot of modules of the same type in the circuit. How would you differentiate between them? You can rename the modules!
Every module can be renamed to your choice. This is done in the Project Explorer.
Click on Folded_LP 24db Vintage. You will notice that that module has been selected in the Project Explorer—its name has changed color to a light blue. Click on the name, and a couple of seconds later click on it again. This will allow you to type in a new name. Call it *Filter Section*.
The name will now change in both the Project Explorer and the Project Window. If the cables display out of alignment with the pads, just move the module a bit.
Select Folded_Mix1 (or Folded_Script Pan) and rename it in the Project Explorer to *Mix Section*.
You can also select modules directly in the Project Explorer as well; do this for Folded_Interpolate (or whatever the modulation section happens to be named). Rename it to *Modulation Section*.
Some of the child modules should be renamed as well. Double-click on Filter Section. Select the first filter and rename it to *Left Filter*. Select the other filter and call it *Right Filter*. Return to the previous level in the Project Window.
You can see in the Project Explorer that the names of the two filters are offset to the right from Filter Section. This shows that they are *children* of Filter Section.
Double-click on Mix Section and rename the first Mix 1 to *Left Volume*. Rename the other Mix 1 to *Right Volume*. You don't need to rename Script Pan unless you choose to do so. Return to the previous level.
As you can see on Filter Section, there are two pads labeled Res. and two pads labeled f. Which pads are which? Which modules are they connected to? You must rename the pads to remind yourself this in the future.
Click on Filter Section.
In Pad List I, you will see a list of eight pads. Two pads labeled *in*, two pads labeled *out*, two pads labeled *Res.*, and two pads labeled *f*. Click on one of the Res. pads.

In the Connections box, you will see two items: *Pot dark **Val*** and either *Left Filter **Res.*** or *Right Filter **Res.*** Click on the Res. pad again, and a couple of seconds later click on it yet another time. Rename the pad to either LRes or RRes, depending on whether the pad is connected to the right or left filter. You can have a maximum of four characters in a pad label.
Change the f pads in a similar manner, by checking its connection in Connections, to LFrq and RFrq. Change the In pads to LIn and RIn, and change the Out pads to LOut and ROut.
You should rename the pads of Mix Section as well. Click on the Mix Section module and rename the two In pads to LIn and RIn, checking the Connections box to confirm which pad you've selected.
As one last step, rename Empty Insert FX in the Project Explorer to *Tutorial Device 1* or *LFO Filter*. This will rename the device. You will see the new name in the top level of the Project Window. Save the device once more, overwriting the file you saved previously. You will need to switch to Save as Type: All Files again to see the device with the extension .dev.
We're now done with the circuits! Congratulations! There may be a couple of extra things to do in Circuit View while we're making the device surface interface, but for now we've completed the circuit design. Everything is ready for building the surface.

Building a Surface

To start building the surface, click on the button labeled *Surface* at the top right corner of the Project Window. Make sure you are first looking at the level of Empty Insert FX in Circuit View, otherwise you will not see anything. If you need to return to Circuit View, click the *Circuit* button next to Surface.
In Surface View, you will now see a little surface box labeled *Empty Insert FX*. The box is rather small, but it is big enough for the moment.

Onscreen Controls and Connections

You can begin dragging objects onto the interface. Navigate the File Browser to Surface Design → Controls → Potentiometers and drag in Pot Dark. Place it on the interface, not on the blank Project Window. *You must be in Move mode for this to work!*
Drag another Pot Dark onto the interface, placing it just to the right of the first one. We'll make the one on the left the filter control, and the one on the right the resonance control.
Click on the first knob. You'll see its name highlighted in the Project Explorer: Pot Dark. Rename the first knob to *Filter Knob*, and rename the second knob to *Resonance Knob*.
So far, you have been connecting all modules together via cables. How shall you connect these knobs to the circuits? By using the following method.
Return to Circuit View and go into Modulation Section. See the two potentiometers you have been using previously? While you could connect the surface knobs to any pad available, for now we will be connecting them to these knobs here, for ease of use.
Return to the previous level, switch to Surface View again, and click on the first knob. In Pad List I, you'll see a pad labeled *Val*. Click on the name of the pad once, and then click the *Store* button directly above it. This "stores" the pad in memory so you can connect it to something else later. *Remember: you can only keep one pad in memory at a time!*
Return to Circuit View. Double-click on Modulation Section and click on the knob connected to Interpolate. Then click on the Val pad in Pad List I and select the *Connect* button directly above it, right next to the Store button. You'll now see that a new connection has been made in the Connections box: Filter Knob **Val**. While you're here, you may as well rename the two circuit knobs. Change the knob you have selected to *Filter Knob Circuit*, and change the other knob to *Resonance Knob Circuit*.
Doucle-click to move up a level, change back to Surface View, and switch to Use mode. The left knob on the panel can now change the filter response! Of course, the new knob is in its default

response curve, which means you must change it to match the circuit knob. In Pad List I, switch to Vars view, and change Curve to 1 and Intensity to 100. Switch back to Pad view and manipulate the filter knob again.
Now let's connect the second knob. Click on the second knob and store its Val. In the Project Explorer, locate Modulation Section → Resonance Knob Circuit and click on the name. Click on Val and Connect.

Labeling Objects

We should label these two knobs on the device surface so that we know what they are! Change to Move mode and move the two knobs towards the top left of the blank frame in the device surface, leaving a little room at the top and left. Navigate the File Browser to: Surface Design → Text, and drag NewText into the panel. Move the new text to just above the filter knob. Drag in another NewText and place above the resonance knob. Move the items to the desired locations. As you move the items, you will have a rather coarse placement grid, and something may not be able to line up. To move objects one pixel at a time, you can either use the arrow keys on your keyboard or hold down *Shift* as you move the item. *(Shift also selects and moves multiple objects, so use the Shift method with caution.)*
We'll want to change the text to say something else, as well as change the font and color. Select the NewText above the filter knob.
It is now highlighted in the Project Explorer. Rename it to *Filter Text*.
To change what the text says on the surface, you'll see a pad in the Pad List labeled: *Str* (for String). Change the value of the String to *Filter*.
It just changed the text on the surface, now longer lined up with the knob. Move the text to the desired location.
Use this same procedure with the resonance text: click on the text, rename it to *Res. Text*, and change the String to *Res.* With shorter labels now, you can move the knobs closer together.

Color & Font

Let's change the color and font of the text.
Click on the Filter text and open up the Color Selector by clicking the Color button on the Toolbar. *(Will have picture in the final manual.)* This brings up the Color Selector.
It is desirable to ensure that your monitor is set to 32-bit True Color, as this will allow you to see the full range of available colors. You see there is an outer ring of colors as well as an inner box. The outer ring allows you to select the color's hue, and the inner box allows you to select the color's brightness. In addition, you can also enter in manually the desired color through the red, green, and blue control fields.
To change the color, click anywhere in the hue ring or the brightness box. The color point will automatically jump to the place you have selected. For fun, you can drag the points around in the ring and the box and see the color change in real-time.
It is not likely you will find a permanent color for the text, as you will be changing the color of the main surface frame later. So, for now, leave it at black (any hue).
Close the Color Selector and open the Font Editor, the "A" button just next to the Color Selector button. Here you will see a great deal of options, some rather similar to word processors with which you may be familiar.
If you have tooltips enabled, hover the mouse cursor over each object in the Font Editor. This will tell you what the various functions do.
There are two pages in the Font Editor: *Main* and *Add*. Main gives you such options as Bold, Italics, Justification, etc., and Add gives you text scaling options. Experiment with these options to arrive at the desired look. You can change the font size by dragging up or down the font size number or by entering in a number with your keyboard. Also, you can change the font face by

clicking on the arrow next to the font name. This displays a drop-down menu of the available fonts on your system. Try to use standard cross-platform fonts (search for a list on Google).
When you have reached the desired font, close the Font Editor.
Click on the frame of the device surface and bring up the Color Selector again. By changing the color you can change the main device color. You can choose any color you wish, but keep in mind that certain colors are not pleasing to the eye!
Perhaps we should change the device's name now. Normally you would change text with the method we have used before, but the title of Empty Insert FX is set to be non-selectable. You must select it by clicking on the module *Text "Panel"* located in *Panel* under *Tutorial Device 1* in the Project Explorer. Now bring up the Font Editor. Change Empty Insert FX to Tutorial Device 1.
Now that you have selected a color for your device, you can click on the Fitler and Res. text to change the text colors to a desired color.

Control Fields

What if you change the filter lowpass position and then wish to change it back? You have no way of knowing what the former position was other than guessing. For this reason, we must add a *control field*. This will show us the numeric value of the filter position.
Navigate the File Browser to Surface Design → Controls → ControlFields. Drag the module *Range Text* to just below the filter knob.
Store the pad *Val* in the Pad List. Then click on the filter knob, select Val, and Connect. Switch to Use mode and try moving the knob.
You see a huge amount of numbers that don't mean anything! This is because the control field is displaying the full scale of values, from 0 to 2147483647. There is a way to divide that number down to a reasonable scale.
Click on the control field—it doesn't matter if it wants you to type in something. The first three entries in the Pad List are: Val, Curve, and Intensity. Below that are *Min* and *Max*. Change Max to 127, giving you a total of 128 values. This is the standard MIDI scale, and you will be using MIDI later.
If you adjust the filter knob, you will see there are still decimal points. Click on the control field again and scroll down the Pad List until you find *Format*. It will read *%1.4f*. What does this mean? It means the control field is displaying the value in a format as follows: integer.four decimal values. If you change this to %1.0f, it will display only the integer values.
The scale doesn't seem quite right! When the knob is pointing straight up, the control field reads 12! That is because the filter knob is using an exponential scale; remember we changed its scale to 1? To compensate for this, we must change the scale of the control field in the opposite direction. Click on the control field. Change its Curve to 2, Intensity 105. Now it should display correctly.
The control field itself looks a little big; it's taking up far too much room. We must resize it.
Having clicked on the control field, navigate to the Window menu and select *GOTree* (Graphical Object Tree). This will show you a list of graphic objects in a surface module and will allow you to adjust parameters.
You will now see a new window with different options. There are two parts to the GOTree: the tree structure, which allows you to select components of a graphic module, and the parameter selection area. If you can't see one or the other, move the divider line to the middle of the window. *(Will have picture in final manual.)*
In the tree structure, you should see several items. Click on *Back1*.
Now bring up the Dimensions adjuster from the Toolbar. *(Will have picture.)* You will see four different parameters. The top right one is the horizontal width. Type in 30, and you will see the filter control field change size! You can now move it to directly below the filter knob, fine-tuning if necessary.
One last step: select the control field and rename it in the Project Explorer to *Filter Field*.
You should now be able to know how to add a control field for the resonance. Make sure you're in

Move mode, then drag in another instance of Range Text (placing it below Resonance Knob), connect it to Resonance Knob, change the max display value to 127, and change the format to %1.0f (you don't need to change the curve). Bring up the GOTree and Dimensions, change Back1 to 30, and rename the module in the Project Explorer to *Resonance Field*.

Adding Other Objects

Now we should add controls for the other components of the device before the surface design is complete. We need the LFO frequency control, the LFO gain control, the volume control, and the panning control. You will be using the same methods as before.
Drag in another Pot Dark onto the surface, placing it a little to the right of the resonance knob. Rename the pot to *LFO Freq Knob*. Drag in another instance of NewText, rename the module to *LFO Freq Text*, and change the String to LFO Freq. For a control field, we can use the module *Range Text Hz*. Drag it onto the surface and rename it in the Project Explorer to *LFO Freq Field.*
Click on the LFO Freq Knob and store its Val. Navigate the Project Explorer to Tutorial Device 1 → Modulation Section → Tiny Sine LFO. Click on the name of Tiny Sine LFO, and you should see a pad in the Pad List named *F*. Click on the name of F and click on Connect.
Are we sure the range and curve response of the LFO Freq Knob matches that of the LFO itself? We can check by bringing up the LFO's surface.
Click on the LFO Freq Knob again so that its pads are displayed in the Pad List I and switch to Vars view. In the Pad List I, there is a button labeled *Freeze*. Click on that button: it "freezes" the pad list so that no matter what other object you select, the Pad List only displays the pads of the object that it showed when it was frozen. How do you see other modules? There is a second pad list.
In the Project Explorer, click on the little "+" icon next to Tiny Sine LFO. This expands the listing of Tiny Sine LFO to show all child modules. One of the child modules will be a module called *Panel*. Right-click on the panel name and navigate to Panel Menu → Show. This will show the LFO's surface.
Open the second pad list by clicking on the button *Pad II* in the Project Explorer. This will bring up *Pad List II* which is identical to the first. Normally it would show exactly what is displayed in Pad List I, but because Pad List I list is frozen, it will show another module. Place it next to Pad List I.
Now manipulate the LFO frequency knob on the LFO's surface. As you can see, the value of the knob is updated on the Val pad on Pad List I. When you look at the control field next to the knob, you'll see it goes all the way to 150 Hz. It probably shouldn't go that far, because beyond 20 Hz, in combination with the lowpass filter, it won't sound very good. So type *20* into the control field. This changes Val in Pad List II to a value of 1789569. Type that number in the Max field in Pad List I.
Other settings that seem different between the two knobs are Curve, which should be set to 1, and Intensity, which should be set to 7.56 (of course, you can change the intensity to anything you like).
Now, close Pad List II and deselect Freeze on Pad List I. Connect LFO Freq Knob to LFO Freq Field. The only setting of the field that needs to be changed is *strMin*: remove the *inf.* value.
Change the color and font of LFO Freq Text to match the other labels on the device surface. The LFO Freq objects should be placed a bit apart from the filter objects to keep the sections separate.
Before we add any other objects, perhaps we should enlarge the device surface. Click on a blank space on the device surface and open up the Dimensions dialog box (the one you opened to change the size of the control fields). In the top right field, type in 450. You will need to switch to Circuit View and back again to Surface View to update the screen properly. Now you have more room to work with!
We'll now add the controls for the LFO gain. You should know the routine by now: drag in another Pot Dark and another NewText, but this time we'll add *Range Text 0db*. Rename the pot to *LFO Gain Knob*, the text to *LFO Gain Text*, the text string to *LFO Gain,* and the control field to *LFO Gain Field*.
Connect the field to the knob, and connect the knob to the *Gain* input of Tiny Sine LFO.

The response curve isn’t right, as you’ll see if you still have the surface of Tiny Sine LFO open. Click on the gain knob on the surface of Tiny Sine LFO: its curve and intensity are 1 and 221. Click on the LFO Gain Knob of Tutorial Device 1 and change its curve and intensity to match.
Now we will add the volume and pan controls for the surface, which are the last items.
Ensuring that you are in Move mode, drag yet another Pot Dark onto the device surface. Name it *Volume Knob*. Drag in another NewText, and rename the object to *Volume Text* and the string to *Volume*. Change the text face and color to match the others. Drag in another Range Text 0db and rename it to *Volume Field*.
As the volume mix is using the same db scale as the LFO Gain, the Volume Knob must also be set to a curve of 1 and an intensity of 221. Then connect the Volume Field to the Volume Knob. Store the Val of the Volume Knob.
Scroll the Project Explorer until you see Mix Section. Expand it. The child modules will be: Left Volume, Right Volume, and Script Pan.
Click on Left Volume. In Pad List I there will be a pad labeled *Gain.* Click on Gain and connect. Do the same for the Gain of Right Volume.
The volume is now complete. The last control is the L/R panning. The panning control will be a little different than what we have been doing, as there will be several settings to adjust.
As usual, drag in a Pot Dark and a NewText. Name the knob to *Panning Knob*, the text to *Panning Text*, and the text string to *Panning*. Change the text face and color to match the others.
Now drag in a standard Range Text. Name it *Panning Field* and connect it to the panning knob. Store the Val of Panning Knob and locate Script Pan in Mix Section. There will be a pad called *Pan*: click on it and Connect.
Switch to Use mode and adjust the panning knob. Of course the field is displaying nonsense numbers. Click on the field and adjust the Format setting to %1.0f.
Now, as this is a left to right panning control, it makes the most sense to make the center display 0, the hard left display -64, and the hard right display 64. Change the Max setting to 64 and the Min setting to -64.
But is that very obvious? How would you know at first glance what those numbers mean? There is an additional nifty tool that you can use. Continue scrolling down the Pad List. Just below Format are two settings labeled *strMin* and *strMax*. These tell the field to show something when the minimum or maximum value is reached. Type *Left* into the blank value of strMin and *Right* into the blank value of strMax. One additional item: you’ll also see *strMid*. Type *Center* into its value. Now adjust the panning knob from left to right: you’ll see those items you typed in!

The surface is now complete! You may need to adjust objects around to line everything up: remember to use the arrow keys to fine-tune!

MIDI

If you want to add MIDI control to adjust these knobs from a remote controller, do the following: Return to Circuit View and double-click on a blank area in the Project Window to return to the top level.
Navigate the File Browser to Circuit Design → Basics. Drag *Empty Synth* into the Project Window; you will need to swipe a couple modules out of it. Double-click on Empty Synth.
Two modules you need will be: *MIDI Controller Linker* and *MIDI Channel Linker*. Right-click on one of them and select *Save as New Module*. Save it in Circuit Design → MIDI Components. Do the same for the other module.
Now return to Tutorial Device 1. Load the two modules from the File Browser and place them near each other. Connect the pad *Ch* on MIDI Controller Linker to the pad *CCon* on MIDI Channel Linker. Right-click on the green pad *MIDI* on MIDI Controller Linker and select *Export Pad*. Move up one level in the Project Window. You now have a new input on Tutorial Device 1: *MIDI*. This input you can use to control the knobs on the device surface!

It is probably best to have the MIDI input pad on the left side of the module instead of the bottom. To change it, select Tutorial Device 1. In Pad List I, there will be a pad labeled MIDI. Right-click on it and navigate to Orientation → Left.

Parameters

There is one final task to do before the device will be ready: creating parameters. (Also see *SCOPE Concept Guide 2*). Parameters basically store settings that you specify that are read by the SCOPE software upon loading. Parameters are most useful for creating presets (a file that contains a certain state of the parameters), but they can also serve other functions as well.

To begin, open the *Parameters* window located in the Window menu → Parameter. There will probably be no parameters listed at the moment. If you are looking at the top level of the Project Window, click *once* on Tutorial Device 1. You should now see four parameters listed. Since every module can have its own parameters, and this view will change if you select anything else, click on the *Freeze* button on the Parameters title bar. This will lock in the view of Tutorial Device 1.

As you use this device, you will want to save the current state of the knob controls on the device surface to make *presets*. To do this, right-click on the Tutorial Device 1 module and navigate to Surfaces → Open Panel. Click once on the Filter Knob. You will see the familiar Val pad in Pad List 1. Select the name and *drag* it into the Parameters window, placing it on the gray area.

You will now see a new entry. It will tell you the *module* name is Filter Knob and the *name* of the pad is Val. You can rename that Val entry in the Parameters window without affecting its integrity.

Now, select every other knob on that surface and do the same thing, dragging the Val pad onto the gray area. As you will see a lot of "Vals," you can rename the name entries to match the module names.

These parameters will ensure that the adjustments you make will be stored when you save the project. However, you can not yet make presets.

Close the device surface, but keep the Parameters window open. Navigate to the Window menu → Parameter → PresetParameterList. In this window you can create a preset list and drag parameters into the Preset Parameter List, which enables a preset to store those parameters in itself. We do not yet have a preset list, though.

Click once on the Tutorial Device 1 module. As on the other window, select the Freeze button on the Preset Parameter List. Then press *Create*, which creates a preset list and adds new modules to your device automatically.

You can now drag every parameter ending in *Knob* from the Parameters window onto the gray area of the Preset Parameter List. When you have finished, you will have six parameters listed. If you did not rename those parameters, every parameter in the Preset Parameter List will read *Val*.

Now you can close the Preset Parameter List; every parameter you need for a preset has been added.

There are, however, other settings to add to the Parameters window that are not used in a preset.

In the Project Explorer, find MIDI Channel Linker and click on it. You will see a pad labeled *CSynth*. Drag this into the Parameter window and rename its *name* to *cwMidiChannel*.

Now, there should be two new modules located in Tutorial Device 1, but you will not see them until you close and reload the device. Don't forget to save first!

Once you have reloaded the device, you will see two new modules: *PresetList* and *SurfaceInterface*. Select PresetList.

You will see many pads in the Pad List. Scroll down to *PresetFile1* and *PresetFile2*. Drag both of those into the Parameters window (if it is not displaying the parameters of Tutorial Device, then unfreeze, click on the module, and freeze again). Then scroll down to *LastRestoredPreset*. Drag the pad into the Parameters window and rename it to *cwLastRestoredPreset*.