

Tutorials

Overview

Tutorial 1: Synth Tutorial

- Building your own synthesizer
- First steps
- Extending your synthesizer

Tutorial 2: FX Tutorial

- Creating Custom Effects
- Getting Started
- Extending the Effect Patch

Tutorial 3: Hihat/Drum Tutorial

- Creating a custom hi-hat section
- Getting Started
- Extending the hi-hat patch

Tutorial 4: Sequencer Tutorial

- The First Gater
- First Steps
- Extending the Patch



Tutorials

Overview

The tutorials in this manual are intended to provide you with a basic understanding of how to work with a modular synthesizer. You will work your way step-by-step through the construction of a synthesizer, study examples of the appropriate use of monophonic and polyphonic circuits, and finally, build a small drum kit. If you're already an experienced Scope Fusion Platform user, you can skip Tutorial 1 and go straight into one of the other tutorials. In any case, we recommend beginning each exercise with a new project in order to ensure that you have the full available DSP capacity at your disposal and can work without hindrances. The tutorials are broken down into individual sections. For each section there is a .mdl file that contains the corresponding patch. Thus, if you're impatient to get to the end of a section, you can simply load this file. However, it is certainly worth your while to work your way through the following exercises yourself.

Tutorial 1: Synth Tutorial

Building your own synthesizer

In this tutorial you will build your own synthesizer. We'll begin with a very simple model consisting only of one oscillator, one envelope, a VCA and a few I/O modules. This synth will then be gradually extended and you will become acquainted with additional modules and functions. And now – in order not to further prolong the agony, let's begin the tutorial.

First steps

Begin by creating a new project and loading the **Empty Modular** device from the Modular2 folder. Connect the device to the MIDI source you typically work with and also to the audio destination you use for monitoring (use the Out1 and Out2 outputs of the Modular device for the audio connection).

Now open the Modular window. The easiest way to do this is by to double-click on the Modular device you just loaded. The as-yet empty Modular workspace appears. This workspace is divided into two sections. The larger of these is the **Module Container** into which the modules get placed. In a row directly above the Module Container are the light-blue **Module Folders**, which can be opened via simple mouse-click. Do this now with the **MIDI** Folder. A menu with several module names appears. Move the mouse over the entry **MVC A** at the bottom this menu. When the entry becomes highlighted, click it – then, while still holding the mouse button down, drag the mouse over the Module Container. When you release the mouse button, the MVC module will fall into the Module Container in the position where the mouse cursor was when you let go.

You've just loaded your first module. The MIDI Voice Control (MVC) module should now be visible in the Modular window. If you wish, you can reposition it by simply clicking on it, holding the mouse button down, dragging the module to the approximate position where you want it, and "dropping" it there.

Before proceeding to connect the MVC module, we'll load a few other modules. These should be placed near the MVC module. All of these modules are to be loaded into the Module Container using the same technique as was used to load the MVC module. If you're unsure whether you've fully grasped this technique, repeat the previous exercise a few times.

Let's continue with the additional modules. Click the **OSC** Folder to open the Oscillators menu. Select the **Multi OSC** module and, holding the mouse button down, drag it into the Module Container. Release the mouse button.

Next, load the **ADSR Vintage** module from the **ENV** Folder and the **Linear VCA** and

Poly Out 1 modules from the **Mix and Gain** Folder. For each module, proceed as described previously. When you've successfully loaded these modules, you can begin connecting them together with virtual cables.

In order for your synthesizer to be playable via MIDI, you must connect the **MIDI Out** jack of the **Modular MIDI Source** to the **MIDI In** jack of the **MVC A** module. As a rule, module outputs are found toward the right side of each module, inputs toward the left. Click once on the MIDI Out jack of the Modular MIDI Source, then position the mouse over the MIDI In jack of the MVC A module and click once again. You should now see a yellow cable connecting the two jacks.

The connections which follow originate at the MVC A module. Using the technique just described, connect the **Freq Out** jack of the **MVC A** module to the **Freq In** jack of the **Multi OSC**. This connection is needed in order to let the oscillator know the frequency at which it should play when you play a note on your keyboard, and is made with a blue cable.

Now connect the **Gate** and **Esync** jacks on the **MVC A** module with the same-named jacks on the **ADSR Vintage**. Gate sends commands for "starting" and "stopping" the envelope, while Esync (envelope synchronization) sends information regarding envelope status back to the MVC A module. These two connections are absolutely necessary for correct envelope operation.

The **ADSR Vintage** envelope will be used to modulate the amplitude (level) of the sound. Therefore, its **Out** jack must be connected to the **Mod** input of the **Linear VCA** module. The amplifier (VCA) is thus controlled by the envelope.

Now you need just a couple of additional connections in order to finally be able to hear your synth. Connect the **Out** jack of the **Multi OSC** to the **In** jack of the **Linear VCA**, and the **Out** jack of the **Linear VCA** to the **In** jack of the **Poly Out 1** module. This latter module is a sort of mixer for all of the voices of your synthesizer – only when it is part of your patch you will be able to play your patch polyphonically. As the final step, connect the **Out** jack of the **Poly Out 1** module to the **In1** and **In2** jacks of the Modular Audio Destination.

At this point, you should hear something when you play your synth. If not, recheck your connections at both the Modular patch level and the project level. Possible sources of trouble include the absence of a MIDI connection to the Modular device, or the use of outputs other than Out1 and Out2. If you can't find any mistakes in your circuit, check to make sure your monitor system is functioning correctly.

If you can already hear something – congratulations, you've successfully completed the first part of the Synth Tutorial!

Most likely, the polyphony of the patch is still fixed at 1. Open the Preset dialog and set the voice count to a value higher than 1, so that you can play multiple notes at one time with your synthesizer. Now, take some time to play around a little with the modules. For example, select different waveforms in the oscillator, or vary the envelope times. If you wish, you can also load the already complete patch for this part of the tutorial (**./Modular2/Tutorials/Tutorial1/Part1.mdl**). This patch also contains a couple of presets which demonstrate a few different combinations of module settings. Bear in mind that this is a very simple synthesizer – don't expect a sonic marvel!

Extending your synthesizer

In the second part of the synthesizer tutorial you will build additional modules into your patch, to expand your synthesizer's possibilities and liven up its sound.

Start by loading the following modules into your patch: **Pitch Modifier B**, **Multi-mode Filter B** and **ADSR B**. The pitch modifier module can be found in the Modifier folder, the filter in the Filter folder and the envelope in the Env folder. To keep the layout clean, place the pitch modifier module close to the MVC module, and place the filter and the modulation envelope near the oscillator.

Now it's time to connect the modules you just loaded. First, you must remove an existing connection – namely, the connection from **Freq Out** on the **MVC A** module to **Freq In** on the **Multi OSC**. Deletion of a connection can be done using the exact same technique as you used to create the connection: Click once on **Freq Out** on the **MVC** module, and then once more on **Freq In** on the **Multi OSC**. The connection should disappear.

Next, connect the **Pitch Modifier B** module between the MVC A module and the oscillator module – that is: connect **Freq Out** on the **MVC A** module to **Freq In** on the **Pitch Modifier** module, and **Freq Out** on the **Pitch Modifier** module to **Freq In** on the **Multi OSC** module.

Now remove the connection between **Out** on the **Multi OSC** and **In** on the **Linear VCA** using the method described above. Next, route the signal from the **Multi OSC** module through the **Multimode Filter B** module to the **Linear VCA** module. In other words: connect **Out** on the **Multi OSC** to **In** on the **Multimode Filter**, and connect **LP Out** on the **Multimode Filter** to **In** on the **Linear VCA**.

At this point, you should be able to play your synthesizer and verify that it functions with the newly-added connections. If you don't hear anything, repeat the above-described sequence steps and inspect your circuit for wiring errors.

Continue by connecting **Gate** on the **ADSR B** module to **Gate** on the **MVC** module. No Esync connection is required (or possible) here – modulation envelopes don't require envelope synchronization and therefore don't have an Esync jack. Now connect **Out** on the **ADSR B** module to the modulation input **CFm1** on the **Multimode Filter B** module.

Test your patch once again and play around with a few of its controls. For example, adjust the filter cutoff, change the modulation envelope times, and vary the cutoff modulation via the CFm1 control on the Modulation Env B module. As you'll no doubt quickly notice, the additional modules enable you to create much livelier sounds.

As a last step, we'll add an LFO module, which we'll use to modulate the pitch of the synthesizer. Open the LFO folder and load the **MW Triangle LFO** module into the Module Container. This module includes a built-in graphical modwheel and is thus quite easy to use. Initially, we'll use this module to create a simple modulation. Therefore, the module should be set for mono operation. The **Single option** is found in the module's context menu – the menu which you can open for any module by simply right-clicking on it ('Ctrl' + mouse button in Mac version). "Mono" means that, internally, only a single instance of the module will actually be loaded, and that it will modulate any and all voices in your patch in common. When you select this option, the module should change its color to green as a visible reminder that it is operating in mono mode.

Connect **Out** on the **MW Triangle LFO** module with **PMod1** on the **Pitch Modifier B** module. This permits the pitch of the oscillator to be modulated. In order

to produce an audible result from this modulation, you must turn the **Pmod1** control of the **Pitch Modifier** module up slightly and move the modwheel of the LFO module while playing your keyboard.

You've now completed the second and final part of the synthesizer tutorial. Experiment with the individual switches and controls on the modules, and perhaps try out a few other variations on the circuit. If you're interested, you can load **./Modular2/Tutorials/Tutorial1/Part2.mdl**, the already-complete patch with accompanying presets.

Following are a few ideas as to how you can further extend your synthesizer:

- Add another oscillator to the patch and detune the two oscillators slightly with respect to one another. The output signals from the two oscillators must be mixed together via a mixer module before they are sent to the filter.

- Use the existing LFO (or add another LFO) to modulate the pulse width of the Multi OSC pulse waveform.

- Exchange various modules in the patch with other ones from the module folders and investigate the functions these new modules offer.

- Load an effects module, connect it up following the Poly Out 1 module, and use it to process your sounds.

Tutorial 2: FX Tutorial

Creating Custom Effects

In this tutorial you will learn how to create your own Pulsar/SCOPE Modular effects. We will take an Auto-Wah effect as our example. A wah-wah effect consists of a filter and an LFO which modulates it. The name of the effect is itself a verbal imitation of the characteristic sound of signals which have been processed via this effect. The "auto" designation indicates that in this case, the effect is controlled by an envelope.

If you've already had some experience in working with the Modular, you need not have already read the foregoing tutorial in order to get started here. If not, we recommend working through Tutorial 1 before doing this one, since this tutorial omits some of the details which we assume you are already familiar with.

Getting Started

It's best to begin this tutorial with an empty project and the **Empty Modular** device. Since the effect is to be controlled via MIDI, you'll need to connect the device with the MIDI source you normally work with. Audio outputs 1 and 2 should be connected to the audio destination you use for monitoring. Later, when your effect is ready, we'll make connections to the audio inputs of the device.

To begin loading modules, open the Modular Window by double-clicking on the Modular device. You'll need the following modules for a simple wah-wah device: **Multi LFO A** from the **LFO** folder and **Multimode Filter B** from the **Filter** folder. The **Module** folders are located in a single row above the **Module Container**.

Connect the **Out** jack on **Multi LFO A** to the **CFm1** jack on **Multimode Filter B**. This is done by simply clicking once on one of these two jacks, and then clicking once on the other.

For testing the wah-wah effect, you'll need a sound source. The simplest way to take care of this is to load an oscillator module, such as **Saw Down OSC**, from the **OSC** folder. The oscillator requires a frequency input in order to produce an audible signal. This would normally be provided by the MVC module, but for test purposes, the **Constant Freq** module suffices. Load this module from the **Modifier** folder.

Now it's time to make a couple of adjustments. Set the **Frequency** value of the **Constant Freq** module to **110 Hz**, the **Rate** of the **Multi LFO A** module to approximately **2.00 Hz** and set the **Cut-off** and **CFm1** controls of the **Multimode Filter B** to their middle positions.

With just a few more connections, you'll be able to hear your effect. First, connect **Freq Out** on the **Constant Freq** module to **Freq In** on the **Saw Down OSC**, then **Out** on the **Saw Down OSC** to **In** on **Multimode Filter B**, and finally, the **BPF** output on **Multimode Filter B** to the **Modular Out 1 und 2** jacks. If you've correctly connected the Modular device in the Project Window and precisely followed all of the other steps, you should now hear the wah-wah effect.

You've now completed part 1 of this tutorial. Play around a little with the controls and try to find a sound that you like.

If you like, you can also load **./Modular2/Tutorials/Tutorial2/Part1.mdl**, the pre-constructed version of the device for this part of the tutorial, which contains some presets.

Extending the Effect Patch

In the second part of this tutorial, the patch will be extended so that the speed of the wah-wah effect is controlled by an envelope, which in turn will be triggered by MIDI messages from a sustain pedal.

To do this, load the following modules: **AD Vintage** from the **Env** folder and **Constant Val** from the **Modifier** folder. Since the envelope is to control the rate of the LFO, connect the **Out** jack on **AD Vintage** to **RMod1** on **Multi LFO A**. Set the **Attack** and **Release** faders on **AD Vintage** to mid position and the **RMod1** control on **Multi LFO A** to quarter-position.

Finally, connect **Out** on the **Constant Val** module to **Gate** on **AD Vintage**. We want to have Constant Val start the envelope in response to Sustain Pedal MIDI messages. Therefore we need to assign **MIDI Controller #64** to the **fader** on the **Constant Val** module. To do this, right click ('Ctrl' + mouse button in Mac version) on the fader to open the Controller Settings dialog. When the assignment has been made, you should

see the fader jump from minimum to maximum whenever you step on your sustain pedal. Whenever the output from Constant Val changes from non-zero to zero value, the envelope will be triggered.

Now fiddle around a bit with the effect controls to get it sounding the way you want it to. When you're done, you can delete the **Saw Down OSC** and **Constant Freq** modules, which were only there for testing purposes. In their place you will now use an external audio signal as a sound source.

In order to be able to process stereo signals through the monaural wah-wah effect, the left and right channels of your incoming signal must be mixed together before reaching the effect. To handle this, load **Mix 2** from the **Mix and Gain** folder and connect **In1** and **In2** of the **Modular In** module to **In1** and **In2** on **Mix2**. Then connect **Out** on **Mix 2** to **In** on **Multimode Filter B**.

Now, in the Project Window, you can connect any desired audio signal to **In1** and **In2** of the **Modular Device** and process it through the wah-wah effect.

As a final step, you can now set all of the modules used in this effect to single mode, in order to make certain that the effect patch will not be loaded as a polyphonic circuit (which would merely waste DSP processing capacity, since the effect processes only a single signal). Do this by right-clicking ('Ctrl' + click in Mac version) on each module in turn to open its context menu, where you can select the **Single** option.

The second tutorial is now complete. Here, once again, you have the option of loading the already completed patch **./Modular2/Tutorials/Tutorial2/Part2.mdl** which corresponds to this part of the tutorial and trying out the presets it contains. As with the patch you constructed yourself, you'll need to hook up an external sound source to this patch.

Before going on, here are a few ideas for further extensions to the Auto Wah effect:

- Hang a chorus or flanger module following the filter to fatten up the sound.
- Leave the LFO module in your circuit set to polyphonic mode, increase the voice count setting of the circuit, and investigate what happens.

– If you wish to process the left and right channels of a stereo signal separately, you can build a second, independent wah-wah effect circuit into the same Modular window alongside the first one and then rewire the source signals so that each circuit processes only one of the channels.

Tutorial 3: Hihat/Drum Tutorial

Creating a custom hi-hat section

In the third tutorial, you will load an existing patch and investigate it. The tutorial illustrates how a small hi-hat (HH) section can be built using the drum modules, how HH mute groups can be realized and how the patch can be expanded into a small drum set.

The first part of this tutorial requires no previous Modular experience. Before working through the second part, however, you should have already completed the preceding Modular tutorials or be an experienced Pulsar/SCOPE user.

Getting Started

Begin by loading the patch **./Modular2/Tutorials/Tutorial3/Part1.mdl**. Proceed as follows: Click on this file in the File Browser and, holding the mouse button down, move the mouse cursor over the Project Window. Then, release the mouse button. The Modular device – which already contains the desired patch – now appears in this window at the position of the mouse cursor. In order to play this patch, you must connect the MIDI input on the Modular device to a MIDI source in the Project Window and its audio outputs 1 and 2 to an audio destination in this window. This is done in the same way as with any other device. You should use the same Project Window inputs and outputs which you normally use with other devices. Test the keys F#1, G#1 and A#1 – you should already hear hi-hat sounds on these keys (which are the standard hi-hat keys in General MIDI drum sets). If you do, then you've done everything correctly.

You can now begin investigating the patch. Double-click on the Modular device to open the associated Modular window with its pre-constructed patch. At the top of the **Module Container** you'll find the standard **I/O modules** of the Modular device. Below these modules, at left, you'll find the **HH Source** module – the first module we'll be looking at here. This module belongs to the **Drum Modules** group and produces sounds whose frequency spectrum is well-suited to the generation of simulated hi-hat and cymbal sounds. It incorporates several sections:

- an OSC section for the generation of a "metallic"-sounding spectrum,
- a high-pass filter for removing selected portions of the OSC frequency spectrum,
- a bandpass-filtered noise generator which can be mixed with the OSC signal, and
- level controls for the oscillator and noise generator.

The patch uses the mixed oscillator and noise generator signals. Thus, if you adjust one of the **Tune** controls in the oscillator section, for example, or tweak the **Noise BPF**, you change the tone color of all hi-hats in your patch, since they all use the same signal.

This can be very convenient, since it means that you can modify the sound of all of the hi-hats in your patch in parallel by means of a single adjustment. Take a few minutes to experiment with the controls on the HH Source module and get acquainted with the way it functions.

In order to produce three different hi-hat sounds, the HH Source signal is fed to three identical circuits which have differing settings for envelope times, filter cutoff, etc. Each of these three circuits consists of a **DVC** (Drum Voice Control), a **Mute Adder 2**, an **AD Envelope**, a **Linear VCA** and a **6dB Highpass-Filter**. This circuit configuration is typical for generation of hi-hat sounds. The function of each of these modules as well as the way in which they are connected is explained below. You should take some time to experiment with each of these modules after reading the explanations.

The important thing is to understand what each of the modules is used for and how you could use them to build your own hi-hat circuits and groups.

The **DVC** is connected directly to the MIDI In module and responds to notes whose note number matches the setting in the **Note** field. The DVC is connected via **Gate** and **Esync** (Envelope Synchronization) to the **AD Envelope**. Actually, any other type of envelope module could just as well be connected here. However, an AD envelope is best suited for drum sound use, since the envelopes it produces always play through to the "end". **Gate** "starts" and "stops" the envelope, while **Esync** receives status feedback from the connected envelope. The DVC module also provides connections for **mutes**, which, along with the Mute Adder module, are explained in the next section. For now, you should make no changes to the settings on the DVC module.

The **Mute Adder 2** module, in conjunction with the DVC and its mute connections, plays a crucial role in the creation of **hi-hat groups**. These groups facilitate cutting-off of an open hi-hat when the hi-hat is closed – or, as is typical in drum machines, by the playing of a "closed" hi-hat sound – thus permitting the sound of a realistic hi-hat playing style to be simulated.

In the case of the Modular, this effect is achieved by means of cross-connection of the **Mute Outs** and **Mute Ins** of two DVCs or drum modules. The Mute Adder is needed in order to combine several such modules into a unified mute group. To facilitate a closer inspection of the mute cabling in this patch, use the colored buttons above the Module Folders at the top of the Modular window to hide all cables except the blue ones (cables of a particular color are hidden when the pale border around the button of the same color is not visible). The mute connections here ensure that the hi-hats play "exclusively" – only one of the three can play at any one time. The left hi-hat is cut off by the middle and right hi-hats,

the middle one is cut off by the left and right ones, and the right one by the middle and left ones. Inspection of the cabling will make it apparent that this is the case: the Mute In of each DVC is driven by the Mute Outs of the other two DVCs, after these are combined via a Mute Adder. This is the basic technique for creating hi-hat groups: each of the DVCs in the group is muted by all of the *other* DVCs in the group, whose Mute Outs are combined via one or more Mute Adders as necessary.

Once you've understood how the mutes work, you can unhide the rest of the cabling and go on to the next part of this tutorial. If you're still a bit unclear about what's going on here, reread this section and experiment with the effect upon hi-hat operation produced by deletion of one or more mute connections in order to achieve a direct understanding of the way mutes work.

The other modules are quite easy to understand. The **AD Envelope** controls a **Linear VCA** (Voltage Controlled Amplifier), which is followed by a **Highpass Filter**. Since the hi-hat signal is sent through the VCA, the **Attack** and **Decay** controls of the envelope control the level variation of the sound over time. Since **Lmod** (Level Modulation) on the envelope is connected to **Velocity Out** on the DVC, the individual hi-hats are also velocity-sensitive. The tone color of the hi-hats can be varied by means of the **Cutoff** control on the **Highpass Filter**.

Be sure to try out for yourself all of the parameters just described to get a feel for the effect each one has. Last but not least, we come to the **Micro Mixer**, which mixes the individual hi-hats, allowing you to set the level and pan position for each one before the combined signal is passed to the audio outputs of the Modular device.

Extending the hi-hat patch

The second part of this tutorial adds a bass drum and snare drum to the hi-hat patch, so that you end up with a small drum set. This is relatively easily accomplished as the drum modules group includes special drum oscillators for bass and snare drum sounds. Begin by loading two copies of the **Drum OSC** module from the **Drum** folder, as well as a **Mix 2** module from the **Mix & Gain** folder. Place these next to one another below the hi-hats.

The MIDI inputs of these oscillators must be connected to the MIDI In module, so that these oscillators can be played from your MIDI keyboard.

A few settings are necessary in order to have the Drum OSCs play the desired sounds, and in response to the desired MIDI notes. The first of the two oscillators will play the **Bass Drum** sound. Set the **Note** field of this oscillator to **MIDI note number 36**, so that the bass drum will play on the key corresponding to the standard General MIDI assignment for bass drum. The other oscillator is for the **Snare Drum**. Here, **MIDI note number 38** is called for.

To be able to hear these two oscillators, you must connect their (audio) **Outs** via the **Mix 2** module to the remaining **In** on the **Micro Mixer**. Thus, connect the oscillator outputs to the inputs of the Mix 2 module, and the Mix 2 output in turn to In 4 on the Micro Mixer.

Now, test carefully (caution – the sounds could be quite loud!) to determine whether the newly-added sounds are audible. If they're too loud, you can lower their levels via the controls on the Mix 2 or Micro Mixer modules.

Most likely, when you play the key intended for snare drum, you'll hear a bass drum sound, because the Drum OSC module is loaded with a bass drum sound by default. However, the drum modules are outfitted with their own built-in preset lists. Click on the P button on the Drum OSC module designated as the snare drum to open its list, then choose a snare drum preset from the list and load it by double-clicking on the name in the list. Naturally, you can also select a different sound for the bass drum Drum OSC using the same method.

If, for whatever reason, you were not as successful with the second part of this tutorial as with the first, you can load the corresponding pre-constructed patch **./Modular2/Tutorials/Tutorial3/Part2.mdl**.

If you're interested in going further with this patch, here are a few ideas:

- Wire in an effect module following the Micro Mixer – for example, the decimator – to warp the drum sounds.
- Use outputs 3 and 4 of the Modular device as individual outputs for bass and snare drum sounds.
- Replace the Drum OSCs with Drum Synths and investigate their possibilities.
- Add further drum oscillators and mix modules to fill out your drum set with toms and other percussion sounds

Tutorial 4: Sequencer Tutorial

The First Gater

In this tutorial you will become familiar with working with the Step Sequencer. We'll start by loading an existing patch, and then introducing the step sequencer to produce a 'Gater' effect. The Gater 'chops up' the sound of the synthesizer patch providing simple chords with a complex rhythmic structure.

You should complete the previous tutorials before advancing to this one, or at least review them to see what you need to know here.

First Steps

As with the other tutorials, you should start with an empty project. Now load the following patch: **/Modular2/Tutorials/Tutorial4/Part1.mdl** and connect it at the system patch level. You'll need a MIDI connection to your MIDI keyboard, and the Outs 1/2 should be connected so that you can hear the modular patches over your monitoring system.

Now open the patch. This patch is a very simple synthesizer comprising the following components: an MVC to receive MIDI data, a sawtooth oscillator, a 24db/octave lowpass filter, and an envelope generator with a VCA to control the volume. So that the synthesizer will be polyphonic, the Outs of the patch are routed to PolyOut1.

If you play this patch you won't hear anything very exciting - just a sawtooth wave slightly filtered by a lowpass filter. This is about to change!

Extending the Patch

First you need a MIDI Clock and a Clock Divider. Load these successively into the modular window and place them between the patch' In abd Out modules, in the upper center part of the window. Connect these two modules' Clock connections together.

So that the sequencer will play at the correct tempo, adjust the text fields of the Clock Divider's Out 1 and 2 to a relationship of 24/6. Adjust these values by clicking and holding the mouse button while moving the mouse up or down, or by simply (left = PC version) clicking on the field and then using the left/right arrow keys on the computer keyboard to change the values.

This relationship must be set because the sequencer will later be clocked to 16th notes. By way of explanation: The MIDI clock signal consists of 24 pulses per beat. When you divide 24 by 6, each note gets only 4 pulses, so there are 4 notes per beat - i.e. 16th notes.

We'll now load Gate Seq1 and Ctrl Seq - both required for the sequencer. Place these also in the center area, under the Clock modules. Connect the Out of the Clock Divider to the Clock of Gate Seq1. The run lights of the sequencer should move with the clock. Connect the Link Out of the Gate Seq1 to the Clock input of the Ctrl sequencer, which should also start running.

The advantage of connecting the Link output of the Gate Seq1 to the Clock on the Ctrl sequencer is that you can use the OnGate function. The Swing factor as adjusted in the gate sequencer will then be reflected in the control sequencer. The OnGate function works such that the control sequencer will only send a value if the respective step is also activated in the gate sequencer.

So that the two sequencers will run synchronously, one final thing must be done: connect the Reset lines of each sequencer together, and click on the Reset button in one of the sequencers. The two sequencers should now start together at the first step, and run in synchrony.

To finish the Gater we need to add two final modules: a vintage AD envelope generator and a linear VCA. Load the modules and locate them on the right side of the modular window under the patch Out module. These modules can be single to Single mode (monophonic) to save processing overhead.

Set Single mode by right ('Ctrl' + mouse button in Mac version) clicking on the module and selecting the respective entry in the context menu. The AD envelope generator and the VCA should now be colored green like the sequencer. Connect the GateOut of Gate Seq1 to the Gate of the envelope generator, and route the Esync signal of the envelope generator back to the gate sequencer. Connect the modulation output of the envelope to the VCA, Delete the connections to PolyOut1 (the lines coming from the patch outs) and connect PolyOut1 to the In of the VCA. Then connect the Out of the VCA to patch Out1 and Out2. All that remains now is to connect the Out from the Ctrl Seq to the modulation input CFm1 of the 24db lowpass filter.

If everything has proceeded according to plan, all the modules required to create the Gater should now be correctly connected. However, you will not hear anything yet when you play the keyboard, but don't worry. If the connections are correct all that's missing is to make the correct adjustments for the modules. Suggestions and ideas are found in under Tips, below. Don't get too impatient if you don't succeed immediately. If you don't get the sound or effect you expect, you can load a finished patch. This is found in **Modular2/Tutorials/Tutorial4/Part2.mdl**.

Tips: First you must set a few gates in the gate sequencer so that the AD envelope generator will trigger. Just invent any example, for now. The envelope times are quite small, so increase the Decay of the envelope generator somewhat. Now, if you play a chord, it will be broken up. Adjust the settings until the rhythm turns into something you either like, or intend.

To add some more life to the sound, use the filter controlled by Ctrl Seq1. First reduce the cutoff frequency somewhat, and then raise the value of the CFm1 modulation input.

When you adjust the faders of the control sequencer - some set relatively high, others low - you will hear that the cutoff frequency is modulated and that the modulation amount varies with each step. Again, adjust them to suit yourself. Make sure that the OnGate value Ctrl Seq1 is set properly so that steps are only triggered in which a gate in the gate sequencer is also set. Disable OnGate if you want all 16 steps of the control sequencer always to be sent. Otherwise, the value of the previous step is held through steps without a gate.

If you didn't load the finished patch mentioned earlier (**/Modular2/Tutorials/Tutorial4/Part2.mdl**), maybe you want to do so now to compare your result with ours. There's also another patch with an additional tempo delay to serve as a further suggestion. Find this in **/Modular2/Tutorials/Tutorial4/Part3.mdl**.

Now, get to it and have fun!