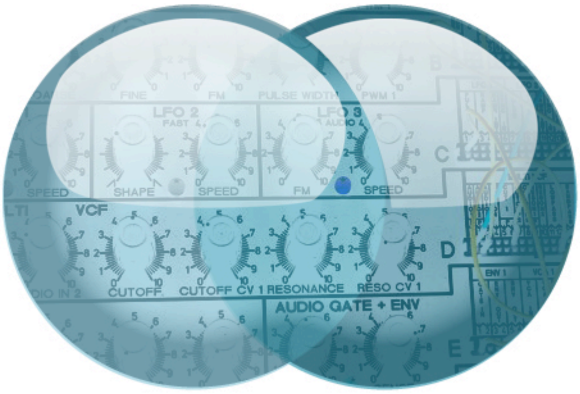
# TINYSIZER®




## Analog Modular Synthesizer

ANYWARE INSTRUMENTS

Congratulations and thank you for purchasing the Tinysizer modular synthesizer!
This little synth has a lot of power inside. The basic idea behind it is to have a powerful, versatile and great sounding analogue modular synthesizer that takes as little physical space as possible. For a start, the heart of the synthesizer is based on one of the best sounding synthesizers in history, the Oberheim® S.E.M., but expanded with the addition of a 350-connector patch bay system that makes it an extremely powerful modular synth. You´ll also find options not available on the original S.E.M. module, like linear FM, Ring Modulation, Portamento, Sub-Oscillators, Sample and Hold, etc. Not content with that, we added a powerful MIDI interface for a better integration within a modern studio, a system of AUX inputs/ outputs for complement other standard modular synthesizers and a digital FX section, very useful for live situations, among other things.
We hope you enjoy this analogue synthesizer as much as we do and find it inspiring for making better music.

*“It is a great pleasure for me to write a short forward for this wonderful instrument.*

*I have been fascinated by modular synthesisers for more than 40 years (originally inspired by Wendy Carlos) and what little I know I have learned from great masters: John Foxx, Daniel Miller, Vince Clarke.*

*It is 10 years now since I sold my Roland 100m system, and after a long love affair with some wonderful soft synths, I have been missing the creativity of the hardware modular synthesiser: no presets, no sound (until it is patched together), no saving, the sheer beauty of the hands-on physicality of the thing.*

*So I have been on a quest, a search for something that suits me. And (thanks again to a tip from Daniel Miller) I found what I was looking for in the Tinysizer. It was love at first sight (and first listen). The Tinysizer is richly endowed with great features; the complex patching capabilities, the comprehensive i/o, the tiny size, the overall design and the price. It ticked all my boxes and after a wonderful afternoon spent playing around with Daniel's machine I bought one immediately.*

*Since then I have spent many happy hours exploring the fears and loving the sound of this little beast. I will continue in a long and productive voyage of exploration with the Tinysizer  I am sure - full of surprises and happy accidents.*

*Tiny size. Big fun. It's genius. It might not suit everyone, but it suits me.”*

***Gareth Jones***
July, 2011
Producer / Recording and mixing engineer / Remixer
Depeche Mode, Erasure, John Foxx, Nick Cave, Anything Box, Gus Gus, etc.

# Index

## Introduction.

Main Features . . . . . . . . . . 06

## Features.

Oscillator 1 . . . . . . . . . . 13
Waveform Processor . . . . . . . . . . 16
Oscillator 2 . . . . . . . . . . 17
Sub-Oscillator . . . . . . . . . . 20
Noise Generator . . . . . . . . . . 20
Filter . . . . . . . . . . 21
Amplifiers . . . . . . . . . . 24
Mixer . . . . . . . . . . 26
LFOs . . . . . . . . . . 27
Envelope generators . . . . . . . . . . 29
FX Section . . . . . . . . . . 31
Audio Gate + Envelope Follower . . . . . . . . . . 32
Ring Modulator . . . . . . . . . . 33
Sample & Hold . . . . . . . . . . 33
Multiples . . . . . . . . . . 34
CV Source . . . . . . . . . . 34
Auxiliary . . . . . . . . . . 35
MIDI to CV Interface . . . . . . . . . . 36

## Programming Examples.

Basic Synth Patch . . . . . . . . . . 44
Using the Sub-Oscillator . . . . . . . . . . 48
Hard Sync . . . . . . . . . . 49
Waveshaper . . . . . . . . . . 50
Auto Wah . . . . . . . . . . 51

# REFERENCE

## Introduction

Tinysizer is an analogue modular synthesizer.

**Its main features are:**

2 X voltage controlled oscillator VCO - Oberheim SEM style

1 X control voltage CV - waveformprocessor - waveshaper

3 X suboscillator

3 X low frequency oscillator LFO

1 X voltage controlled filter VCF - State variable, Oberheim SEM style

2 X ADSR envelope generator

1 X envelope follower & audio gate & microphone preamp

1 X DSP effect, reverb & delay

4 X voltage controlled amplifier VCA linear

2 X mixer & inverter & overload

1 X control voltage CV source

4 X CV, 6 X gate, Midi interface

1 X white noise & pink noise

1 X sample & hold - s+h

1 X ringmodulator

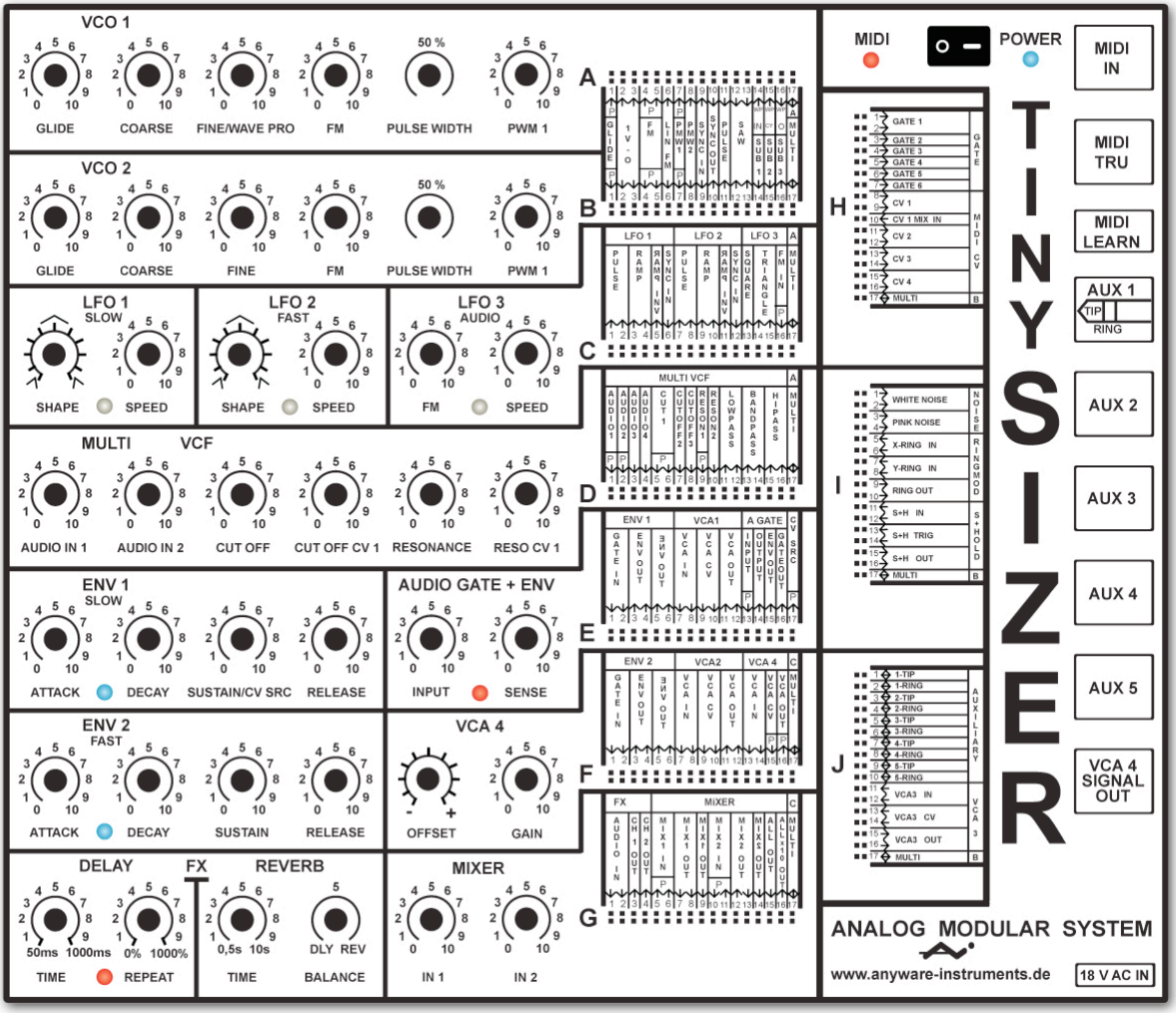
11 X patchble auxiliary in & outputs, bi - directional

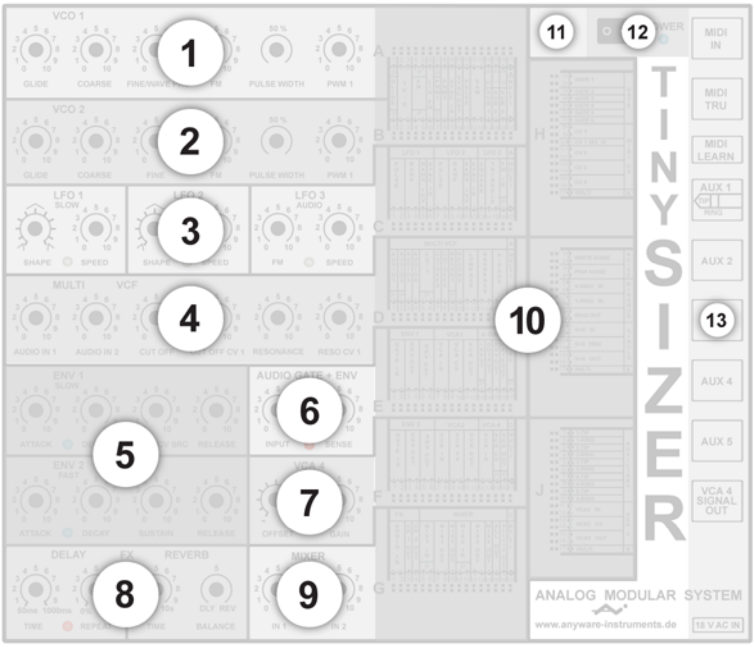
1 X powersupply

3 X multiples

## The Front Panel

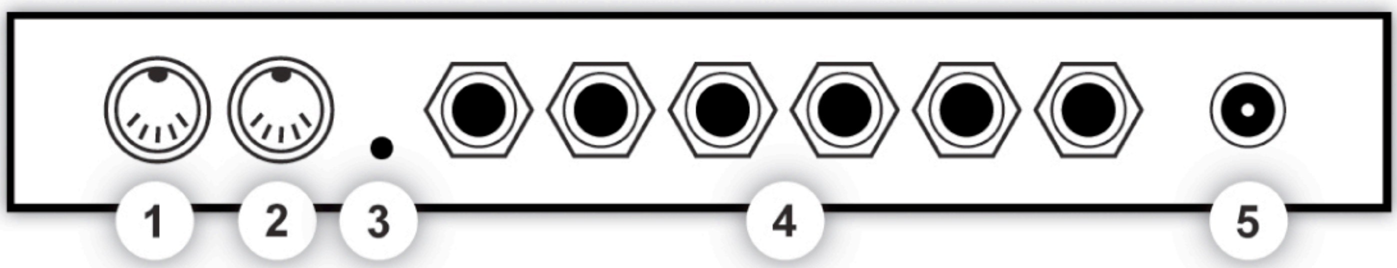
The front panel is covered with knobs. Every parameter has its knob, name and range.





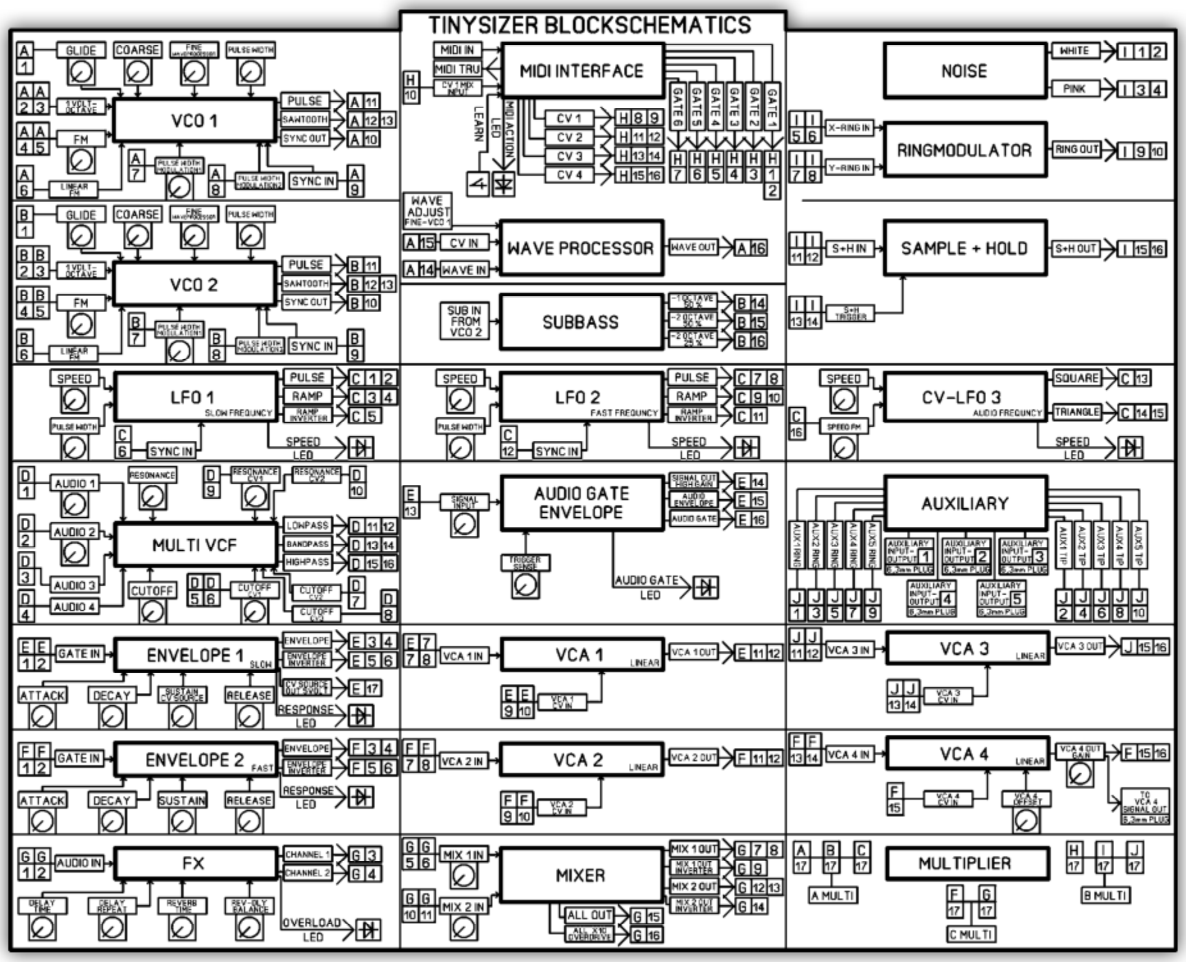

1- VCO 1 section.
2- VCO 2 section.
3- LFO section. LFO 1, 2 & 3.
4- VCF section.
5- Envelope section. Envelope 1 & 2.
6- Audio Gate/Envelope Follower section.
7- VCA 4 section.
8- FX section.
9- Mixer section.
10- Patch Bay section.
11- MIDI in led.
12- Power switch.
13- IN / OUT Section

**IN / OUT Section:**



1) MIDI In
2) MIDI Out
3) MIDI learn button.
4) AUX connectors.
5) Main voltage input.

**Signal Flow:**




**The Concept.**

The concept behind Tinysizer is to have a complete modular synthesizer in a small physical space. Tinysizer has a diverse range of modules, configured in a very smart and optimized way. Every module has its parameters assigned to knobs on the front panel (when applicable), but their connections are placed on the patch bay system at the right of the panel. There are almost no pre-made connections, so for one module to “talk” to another, they have to be connected via the patch bay with the provided special cables. Being a modular synth, the Tinysizer lets you connect anything to anything, but there are certain “rules” that will help you make the desired sound more quickly (in the case when you have a sound in mind, that is).

To understand how Tinysizer, or any other analogue synthesizer for that matter, works it is important to have the notion that there are three types of signals:

**Audio:** Audio signals are basically the ones that you actually hear and involve sound generation and sound processing modules. That includes oscillators, filters, amplifiers, etc.

**Control:** These are the ones that give shape to the sound by controlling different aspects of the sound generating and processing modules. They can also control other control signals, or even trigger some "gate" modules. The modules involved in this category are the Envelope Generators, LFOs, the MIDI interfaces CV outs, etc.

**Gate:** Gate signals are used to trigger certain modules. Envelopes, LFOs, Sample & Hold, etc can be triggered. The MIDI interface is the main generator of Gate signals.

Sometimes in a modular synth those signal types are blurred, as you can use, for example, an audio souce as a control signal, or trigger an Envelope with an LFO, etc. Some modules, like mixers and multiples work for audio and/or Control and Gate signals.

But in general most classic synths work as follows:

**Audio sources**: (oscillators, noise generator, external audio) are mixed and passed to sound shaping modules like filters. Waveshaper, ring modulator, sub-oscillator etc may be combined before or after mixing and filtering. Of course, being a modular synth, Tinysizer allows a lot more flexible connections than that, but the basic idea is still valid as an example.
A final volume-shaping module (VCA 4) is pre-wired as the final module before the audio leaves the synth.

**In the meantime**: Control generators modulate certain parameters of the sound generator and processing modules to give dynamic movement to the sound. Gate signals generally coming from the MIDI to CV interface trigger the Envelope generators that will shape volume and timbre. LFOs will provide cyclic modulation like vibrato or tremolo, and they don't need to be triggered, though they could be for certain effects.

This synthesizer is based on the 1 volt per octave (1v/oct) standard for control signals that most modular and standard monophonic analogue synthesizers use, which means that, for example, an oscillator's or filter's frequency is doubled by each additional volt the module receives by a modulator. There are Aux connectors on the right side of the synth that are basically standard TRS connector internally wired to the patch bay's sockets so Tinysizer can talk to other 1Volt/oct synths. That means that there can be bi-directional communication in which an oscillator from another modular synth can be used inside Tinysizer and mixed with the other two internal oscillators, and at the same time Tinysizer's LFO 3 can modulate that external oscillator for vibrato.

Included with the synth is a very complete MIDI to CV interface. This module converts MIDI signals like NOTE ON/OFF, Controllers, Velocity, etc to Control Voltage and Gate signals. This allows you to not only trigger notes, but to control various aspects of the sound in real-time from a MIDI controller, and to automate those movements. The converter features 4 CV and no less than 6 Gate outputs, and the function of each output is programmed from a PC application.

Another welcome digital addition is the FX section, something common on modern workstations but very rare in an analogue modular synthesizer context.

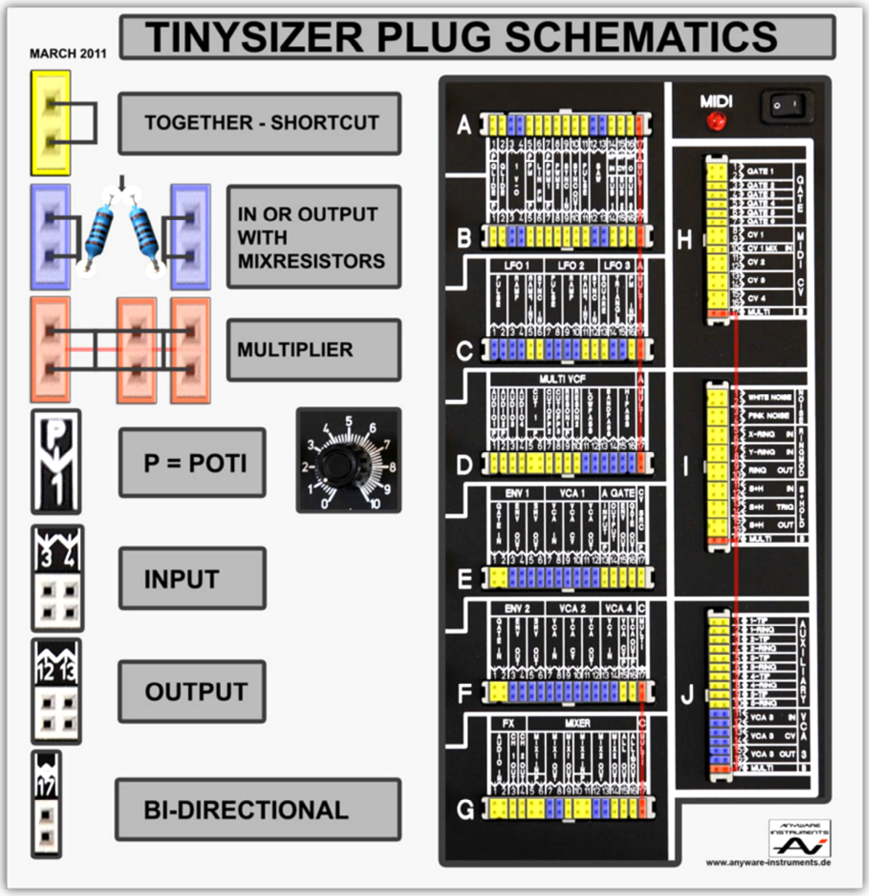
A note should be made about distortion. This synth can sound really clean and nice, but also very dirty and rough, thanks to its various distortion processors across the audio chain. You can distort the signals on the following stages:

- The Waveform Processor
- The Filter mixer
- The Mixer (all x 10 output)
- The Mic Preamp
- Sample & Hold as sample rate reduction effect (see S&H section)
- Plus, various feedback loops can be made to generate distortion.

All these have a different sound and characteristics, so don't hesitate to experiment!

## THE PATCH BAY SYSTEM

The patch bay system is the heart of the Tinysizer. Without some connections it's unable to make any noise. It uses a very exclusive and innovative socket format, that allows hundreds of patch connections to exist in a very tiny space. To take advantage of it, first you need to fully understand it. To help you, here's a nice graph:



At right you have an overview of the patch bay system. The first thing that may call your attention is the division by letters (A, B, C, D, E, F, G, H, I and J). The logic behind it is that each letter represents one synthesizer section, and each section has all the connections of the modules they represent. To make things clearer, let's define the sections:

**A**: VCO 1, Wave Processor and Multiple A
**B**: VCO 2, Sub oscillators and Multiple A
**C**: LFO 1, 2 & 3, and Multiple A
**D**: Mutli VCF and Multiple A
**E**: ENV 1, VCA 1 and Audio Gate + Env
**F**: ENV 2, VCA 2, VCA 4 and Multiple C
**G**: FX MIXER and Multiple C
**H**: MIDI to CV Interface and Multiple B
**I**: White and Pink Noise, Ring Modulator, Sample & Hold and Multiple B
**J**: Auxiliary, VCA 3 and Multiple B

Some modules only exist in the patch bay, and they don't have any control on the control panel for obvious reasons: most of those modules have nothing to be controlled by a knob, like Ring Modulator or Sample & Hold. Some modules DO have extra knobs to control the attenuators on some of their inputs, and they're labeled with a "P" on the patch bay. "P" means "Poti" or Potentiometer, and you know there's an attenuator on there.
The arrows that go upwards mean the patch point is an OUTPUT, and the ones that go downward means it's an INPUT. If it has both arrows, it means it's bi-directional, and you'll only find them on Multiples or Auxiliary modules.

The colours don't represent input/output, but the different types con connections possible in the patch bay. As you can see, each section (A, B, C, etc) has a double row of connections, so each module connection in/out has a minimum of two patch points. Some, like "1 V/O" oscillator input has four patch points. So how do they work? Do they mix signals? Do they act as multiples?
The answers are on the graph!

Here you'll find a BIG advantage of the Tinysizer over most modular synthesizers. You'll rarely need multiples, as most modules outputs have multiple patch points (2 or 4). For example, from LFO 1 you could connect four "Ramp" and four "Pulse" outputs to different destinations. That's eight outputs without a multiple!
In some cases you can do the opposite - you can use the various inputs as mixers. That's what the blue patch points are for. That's valid for control as well as for audio signals, but watch the levels - you'll probably overdrive the inputs with two or more signals without attenuation.
Inputs in yellow don't allow you to mix signals.
The different Multiples are divided across sections, and that's useful to send one signal from one point to another avoiding long cables.

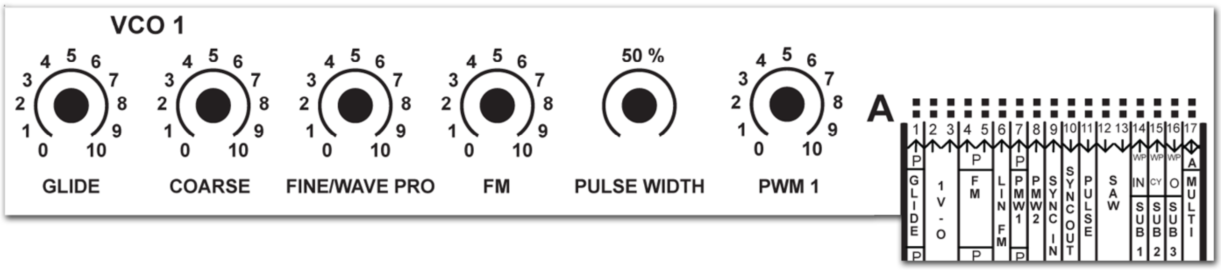
From now on, Ô symbol will represent a knob on the control panel, and [] will represent a patch point in the patch bay. [] symbol will be accompanied with a coordinate to locate the patch point. For example “[] A6” means, “section A (oscillator 1), patch point 6 (Lin FM)”.

## REFERENCE

### Oscillator Section

Oscillators are the primary sound source of a synthesizer; they provide the basic pitch and tone colour to the final sound. In the case of the Tinysizer, they generate two different basic waveforms, Sawtooth and Pulse (each has a different tone colour). A lot more timbre variation is possible by doing FM (linear and exponential), PWM, hard sync, ring modulating them, etc. By passing one Sawtooth though the Waveshaper module you can get an additional triangle waveshape.

### OSCILLATOR 1:




*GLIDE:
Glide, also known as portamento, is a voltage processor that, in the case of the Tinysizer, is inside the oscillator module. It’s basically a low-pass filter for control signals; what it does is round any sudden voltage change, like the ones that happen from note to note when controlled from a keyboard, making a smooth transition between notes. The time it takes to go from one voltage level to the next is determined by the Glide Knob.

Ô **GLIDE**: Time range: from 1 ms to 12 sec
Glide Curve Type: Log
[] A1 **GLIDE** Input: This input is a CV input to control oscillator 1 pitch that is first processed by the glide function.

---

### Coarse / Fine:

Coarse defines the base pitch of the oscillator, and it has a wide frequency range. For better tuning precision make additional adjustments using the “Fine” Knob. These two knobs will set the pitch of the oscillator if no modulation is applied. All pitch modulation will sum or subtract from this base frequency.

Ô **Coarse**: Range: 31Hz to 3.3kHz
Ô **Fine**: +- 150 cents

**Exponential Frequency Modulation Inputs:**

These are two 1 volt per octave modulation inputs. They're similar to FM inputs (A4, A5), but without an attenuation knob on the front panel. Connect a keyboard output here, as no attenuation is needed for chromatic keyboard tracking.

[] A2, A3 **1 V/O**: 1V/Oct inputs.
Range: 0 to 10 volts

---

**FM:**

FM means Frequency Modulation.
This is a shared control knob (attenuator) of the two CV inputs placed on the patch bay (A4 & 5).
Use this to control for e.g. vibrato depth if connected to an LFO.

Ô FM: Zero to no attenuation
[] A4/A5 **FM** Input: 1V/Oct inputs.
Range: 10 Volts

---

**Linear Frequency Modulation Input**:

Contrary to exponential FM, in which frequency responds by doubling with each additional volt it receives (hence "one volt per octave"), Linear FM responds by doubling the volts needed to raise each octave (Hz per volt). As Tinysizer oscillators are designed to work with the 1volt/oct standard, Linear FM is used as a timbre modifier by feeding audio rate signals. The benefits of using Linear FM this way is that the pitch of the oscillator being modulated will be much more stable on its central pitch. As Linear FM responds by increasing and decreasing the same amount of Hertz by the same amount of positive and negative voltage (e.g. form a triangle wave), central average pitch will theoretically not be modified, and so a "musical", pitched note will be possible. Any DC offset on the modulating source will shift the central pitch, so symmetrical waves are recommended for classic FM sounds.

[] A6 **Lin FM**.

---

**Pulse Width:**

This knob only works with the Pulse waveform. By turning it you'll change the base width of the Pulse wave, thus changing its timbre. Any modulation applied to the Pulse wave (i.e. PWM, A7 & A8) will sum or subtract to this base position. The 50% position (without modulation) is a Square wave, fully left position is 5% and fully right is 95%. If further modulation is applied, 0% and 100% pulse width can be obtained, which means silence.

Ô **PW**: from 5% to 95%

---

**PWM1:**

This knob is an attenuator to patch bay PWM1 (A7) input. That is, it controls the modulation depth of whatever is connected in A7 socket.

Ô **PWM1**: from 0 to 10 (full modulation).
[] A7 **PWM1**: Range -5 volt (0%) to +5 volts (100%).

---

**PWM2:**

PWM2 is the same as PWM1 (A7) but without an attenuator on the front panel.

[] A8 **PWM2**: Range: -5 volts (0%) to +5 volts (100%)

---

**Waveform:**

This is not a selector; each waveform has an independent output on the patch bay. By "independent" we mean: both Sawtooth and Pulse will have the same pitch and pitch modulation, and their phase relation will always be the same as the oscillator generates the Sawtooth wave and then a waveshaping circuit is used to modify that wave to convert it to Pulse with its own output. Pulse has one output and Sawtooth has two.

[] A11 **PULSE**: Pulse waveform.
Output Amplitude: -5 to +5 volts
[] A12, A13 **SAW**: Sawtooth waveform.
Output Amplitude: -5 to +5 volts

---

**Sync in/out:**

Hard Sync was originally conceived as a way to tune analogue (i.e., unstable) oscillators in perfect unison or octaves without any beating problems. Then it turned out to be a fantastic way to create new waveforms by applying different tunings and pitch sweeps to the slave oscillator.

Hard Sync works as follows: Two (or more) oscillators are synchronized by taking the Sync output of the "Master" oscillator and connecting it to the Sync input of the "Slave" oscillator(s). Each time the Master oscillator completes its cycle, (that is, when the Sawtooth wave goes instantly from +5 to -5 volts), the slave oscillator restarts as well; no matter where in its phase it is at that particular moment. In fact, the Sync output doesn't send a Sawtooth wave, but a short pulse each time the Sawtooth wave is reset. If both oscillators are tuned in unison, then they'll be perfectly in tune and there wont be any beating. The same will happen if the slave is tuned to any position within the harmonic´s scale, (that is x2, x3, x4, x5, etc). Octaves will be x2, x4, x8, etc.

[] A9 **Sync Out**: Pulse derived from Oscillator 1 used to retrigger another oscillator.
[] A10 **Sync In**: If used as a slave oscillator, this input will retrigger oscillator 1 each time it receives the pulse signal from the Master oscillator's Sync Out.

---

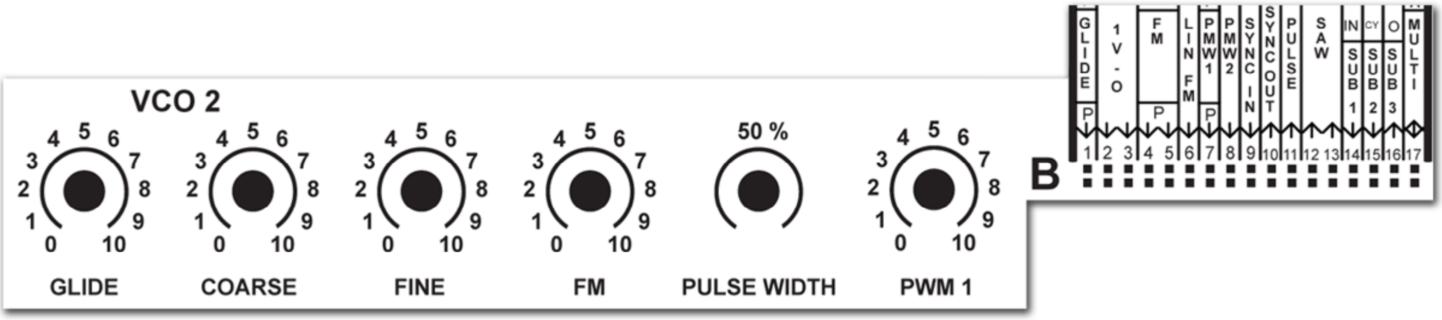
**Voltage Controlled Waveform Processor:**

The Waveform Processor is a waveshaping circuit that can be used for various purposes. Its depth can be controlled from the "Fine/Wave Pro" knob or externally by a control voltage.
A waveshaper is a non-linear distortion module mainly used to modify a waveform from an oscillator. If a Sawtooth wave from an oscillator or LFO is fed to its input, the output will be that same waveform if the knob position is 0, a Triangle wave at knob position 5 and an inverted Sawtooth wave at knob position 10 (exact center position may differ depending on calibration). Other input material will have more unpredictable results, most of the time resulting in audible distortion. Other uses could be to process the output of an envelope generator to modify its curves, or external audio signals for heavy distortion effects.

Ô **Wave Pro** (Shared knob with oscillator 1's "Fine"): Controls the waveshaping depth.

[] A14 **WP IN**: Input to the Wave Processor module
[] A15 **WP CV**: Voltage control input of waveshaping depth.
[] A16 **WP O**: This is the Wave Processor output.

**OSCILLATOR 2:**




**GLIDE:**
Glide, also known as portamento, is a voltage processor that, in the case of the Tinysizer, is inside the oscillator module. It's basically a low-pass filter for control signals; what it does is round any sudden voltage change, like the ones that happen from note to note when controlled from a keyboard, making a smooth transition between notes. The time it takes to go from one voltage level to the next is determined by the Glide Knob.

Ô **GLIDE**: Time range: from 1 ms to 12 sec
Glide Curve Type: Log
Ô B1 **GLIDE** Input: This input is a CV input to control oscillator 1 pitch that is first processed by the glide function.

---

**Coarse / Fine:**

Coarse defines the base pitch of the oscillator, and it has a wide frequency range. For finer tuning precision make additional adjustments using the “Fine” Knob. These two knobs will set the pitch of the oscillator if no modulation is applied. All pitch modulation will add or subtract from this base frequency.

Ô **Coarse**: Range: 31Hz to 3.3kHz
Ô **Fine**: +- 150 cents

---

**Exponential Frequency Modulation Inputs:**

These are two 1 volt per octave modulation inputs. They're similar to FM inputs (B4, B5), but without an attenuation knob on the front panel. Connect a keyboard output here, as no attenuation is needed for chromatic keyboard tracking.

[] B2, B3 **1 V/O**: 1V/Oct inputs.
Range: 0 to 10 volts

---

**FM:**

FM means Frequency Modulation.
This is a shared control knob (attenuator) of the two CV inputs placed on the patch bay (B4 & B5).
Use this to control for e.g. vibrato depth if connected to an LFO.

Ô FM: Zero to no attenuation
[] B4/B5 **FM** Input: 1V/Oct inputs.
Range: 10 Volts

---

**Linear Frequency Modulation Input:**

Contrary to exponential FM, in which frequency responds by doubling with each additional volt it receives (hence "one volt per octave"), Linear FM responds by doubling the volts needed to raise each octave (Hz per volt). As Tinysizer oscillators are designed to work with the 1volt/oct standard, Linear FM is used as a timbre modifier by feeding audio rate signals. The benefit of using Linear FM this way is that the pitch of the oscillator being modulated will be much more stable on its central pitch. As Linear FM responds by increasing and decreasing the same amount of Hertz by the same amount of positive and negative voltage (e.g. form a triangle wave), central average pitch will theoretically not be modified, and so a "musical", pitched note will be possible. Any DC offset on the modulating source will shift the central pitch, so symmetrical waves are recommended for classic FM sounds.

[] B6 **Lin FM**

---

**Pulse Width:**

This knob only works with the Pulse waveform. By turning it you'll change the base width of the Pulse wave, thus changing its timbre. Any modulation applied to the Pulse wave (i.e. PWM, B7 & B8) will sum or subtract to this base position. The 50% position (without modulation) is a Square wave, fully left position is 5% and fully right is 95%. If further modulation is applied, 0% and 100% pulse width can be obtained, which means silence.

Ô **PW**: from 5% to 95%

---

**PWM1:**

This knob is an attenuator to patch bay PWM1 (B7) input. That is, it controls the modulation depth of whatever is connected in B7 socket.

Ô **PWM1**: from 0 to 10 (full modulation).
[] B7 **PWM1**: Range -5 volt (0%) to +5 volts (100%).

---

**PWM2:**

PWM2 is the same as PWM1(B7) but without an attenuator on the front panel.

[] B8 **PWM2**: Range: -5 volts (0%) to +5 volts (100%)

---

**Waveform:**

This is not a selector; each waveform has an independent output on the patch bay. By "independent" we mean: both Sawtooth and Pulse will have the same pitch and pitch modulation, and their phase relation will always be the same as the oscillator generates the Sawtooth wave and then a waveshaping circuit is used to modify that wave to convert it to Pulse with its own output. Pulse has one output and Sawtooth has two.

[] B11 **PULSE**: Pulse waveform.
Output Amplitude: -5 to +5 volts
[] B12, B13 **SAW**: Sawtooth waveform.
Output Amplitude: -5 to +5 volts

---

**Sync in/out:**

Hard Sync was originally conceived as a way to tune analogue (i.e., unstable) oscillators in perfect unison or octaves without any beating problems. Then it turned out to be a fantastic way to create new waveforms by applying different tunings and pitch sweeps to the slave oscillator. Hard Sync works as follows: Two (or more) oscillators are synchronized by taking the Sync output of the "Master" oscillator and connecting it to the Sync input of the "Slave" oscillator(s). Each time the Master oscillator completes its cycle, (that is, the Sawtooth wave goes instantly from +5 to -5 volts), the slave oscillator restarts as well; regardless of where in its phase it is at that particular moment. In fact the Sync output doesn't send a Sawtooth wave, but a short pulse each time the Sawtooth wave is reset. If both oscillators are tuned in unison, then they'll be perfectly in tune and there wont be any beating. The same will happen if the slave is tuned in any position within the harmonic´s scale, that is (x2, x3, x4, x5, etc). Octaves will be x2, x4, x8, etc.

[] B9 **Sync Out**: Pulse derived from Oscillator 1 used to retrigger another oscillator.

[] B10 **Sync In**: If used as a slave oscillator, this input will retrigger oscillator 1 each time it receives the pulse signal from the Master oscillator's Sync Out.

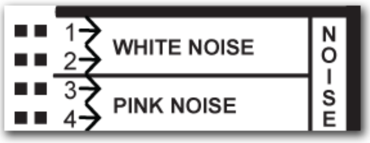
---

**Sub-Oscillator:**

A Sub-Oscillator is a frequency divider: its input is connected to Oscillator 2 Sawtooth, and it generates a square wave one and two octaves below it. It also generates a 25% Pulse wave two octaves below Oscillator 2. All three waveforms are available simultaneously on the patch bay (B14, B15 & B16).
To generate the first waveform (Square, one octave below), the Sub-Oscillator generates half a cycle for every cycle of Oscillator 2. Similarly, the other two waveforms are generated in this way, though every 2 cycles.
A Sub-Oscillator waveform will be in phase with, and will respond to, any pitch modulation of Oscillator 2. Use it to reinforce bass sounds, or as an additional sound source.
If you Hard Sync Oscillator 1 an octave above Oscillator 2, and mix Oscillator 1 & 2 Square waves with Sub 1 & 2 you'll have 4 square waves perfectly in tune and phase tuned 1 octave apart each. By careful mixing some interesting new waveforms can be created. Try it!

[] B14 **SUB 1**: Square wave one octave below Oscillator 2
[] B15 **SUB 2**: Square wave two octaves below Oscillator 2
[] B16 **SUB 3**: 25% Pulse wave two octaves below Oscillator 2

**NOISE GENERATOR**



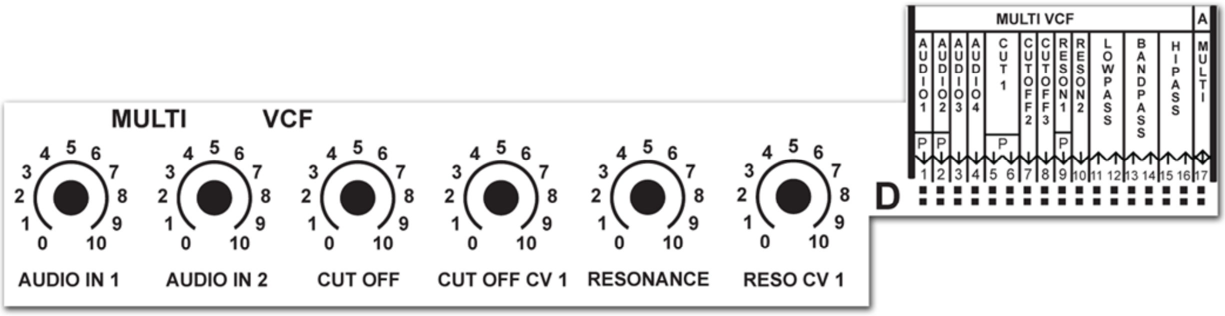

Our last internal sound source is the Noise Generator, which is a circuit dedicated to generating random electrical variations that can be very useful as an audio source or control. There are two so called noise "colours", white and pink, and may be used for different purposes.
White noise has equal energy across the frequency spectrum. It is used as an audio source for sound effects, percussion sounds, etc. It can also be used as a modulation source (slightly modulating an oscillator frequency for a grittier sound, for example). An obvious use is as a Sample & Hold source for the generation of random values (see Sample & Hold).
Pink Noise is a filtered version of white noise, which has a lower, more bassy sound. Its amplitude distribution across the spectrum is different; it's inversely proportional to frequency. Pink Noise is not as widely used as white noise, but may be used in cases when lower, darker noise is needed.
Noise Generators don't have any control parameter.

[] I1, I2 **WHITE NOISE**: White Noise audio output

[] I3, I4 **PINK NOISE**: Pink Noise audio output

---

**THE FILTER SECTION**




This voltage controlled filter is a flexible multimode, state variable, great sounding filter based on the classic Oberheim S.E.M. design. It has several desirable features, like CV resonance, a 4 input audio mixer, 3 simultaneous filter type outputs, various modulation inputs, input overdrive, etc.
This multimode filter has three simultaneous filter types available: lowpass, hipass and bandpass. If lowpass and hipass filters are used at the same time and mixed at equal levels, a notch filter is obtained.
Tinysizer's filter has a slope of 12 dB/octave. This means that if a lowpass filter is used, any signal at the double of the cutoff frequency will be lowered by 12 dB.

---

**Audio In 1 & 2:**

This is part of the 4 input mixer included in the filter module. Audio In 1 & 2 have knobs to control their volume, Audio 3 & 4 have no attenuators.
This 4 input mixer is very handy, as it's here where you'll normally mix all the sound sources to be filtered, like Oscillator, Noise generator, Sub-Oscillator signals and it saves the use of other mixer modules for this purpose. High signal levels will make the filter distort. This is a desired effect sometimes (it was designed that way intentionally), but if you want to avoid it just lower the input signals.

Ô **Audio In 1**: Control for volume of D1 patch bay socket. From 0 to no attenuation.
Ô **Audio In 2**: Control for volume of D2 patch bay socket. From 0 to no attenuation.
[] D1 **Audio 1**: Audio input 1.
[] D2 **Audio 2**: Audio input 2.

---

**Audio In 3 & 4:**

Same as above, but without attenuation knobs.

[] D3 **Audio 3**: Audio input 3.
[] D4 **Audio 4**: Audio input 4.

---

**Cutoff Frequency:**

Perhaps the most important parameter of a filter, Cutoff sets the frequency at which the filter begins to work.

Ô **Cutoff**: Filter Cutoff frequency. Range: 18 Hz to 16 kHz

---

**Cutoff CV 1:**

This is an attenuation knob to control Cutoff CV 1 inputs on the patch bay. D4 and D5 inputs are mixed and then controlled by this knob.

Ô **CUTOFF CV 1**: Cutoff frequency CV 1 attenuation knob
[] D5 **CUT 1**: Cutoff frequency CV 1 input
[] D6 **CUT 1**: Cutoff frequency CV 1 input

---

**Cutoff CV 2 & 3:**

These two modulation inputs have no attenuation knob; they're mixed to the output of the attenuated CV1 inputs. Attenuation from the Mixer module's Mix 1 and Mix 2 outputs is recommended in most cases.

[] D7 **CUTOFF 2**: Cutoff frequency CV 2 input
[] D8 **CUTOFF 3**: Cutoff frequency CV 3 input

---

**Resonance:**

This is a feedback loop within the filter that emphasizes the signal at the cutoff frequency proportional to the knob position. 0 Resonance means there'll be no feedback, while 10 means total feedback to the point of self-oscillation. The audible result of Resonance is the typical electronic synth sound, but it's also very useful to simulate acoustic characteristics like formants.

Ô **RESONANCE**: no resonance (0) to full resonance (10) knob.

---

**Resonance CV Input 1:**

Having CV control of Resonance is handy and not very common, even for a modular synth. This knob is an attenuator of Patch Bay input D9.

Ô **RESO CV 1**: Attenuation knob of Patch Bay input D9
[] D9 **RESON 1**: CV input of Filter Resonance.

---

**Resonace CV Input 2:**

Same as above but without attenuation knob.

[] D10 **RESON 2**: CV input of Filter Resonance

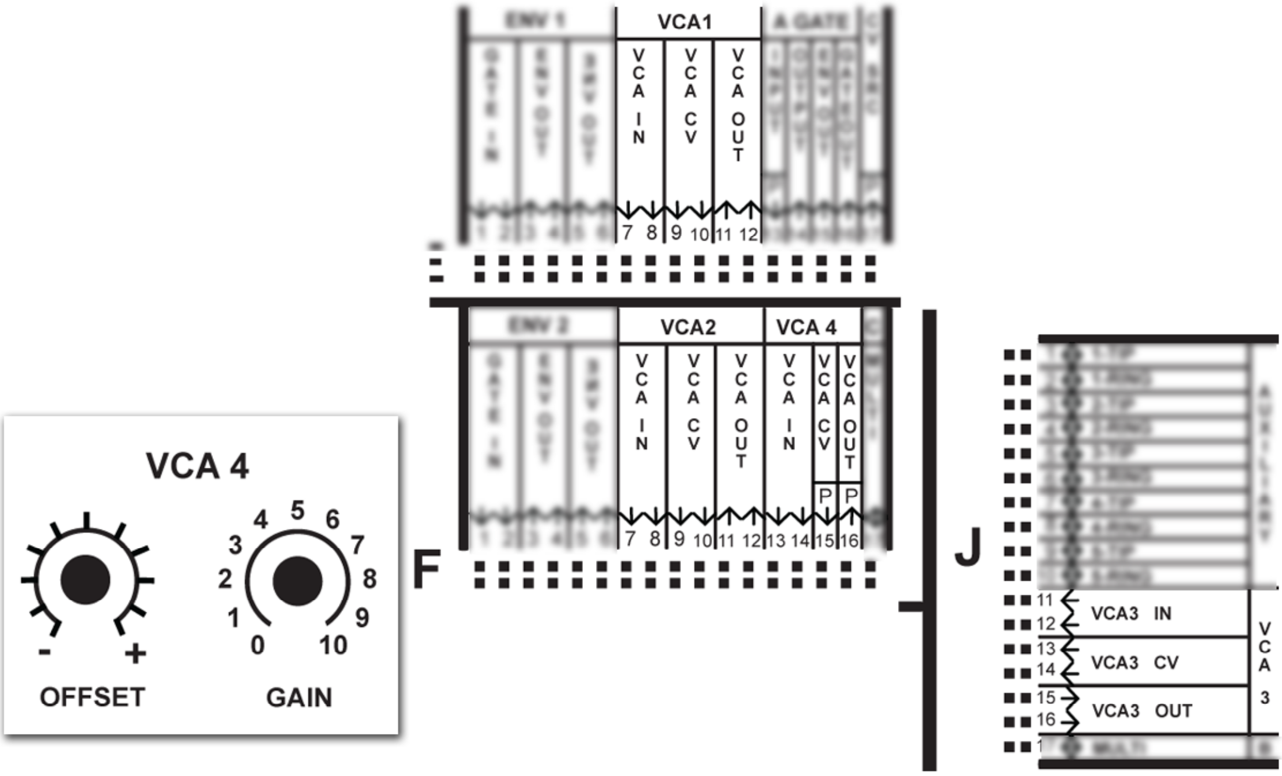
---

**Filter Type:**

These are the filter’s simultaneous outputs, D11 & D12 for Lowpass, D13 & D14 for Bandpass and D15 & D16 for Hipass outputs. By combining Lowpass and Hipass in parallel (mixing both outputs) a Notch filter is obtained.
The Lowpass filter will attenuate signals above the cutoff frequency at a rate of 12dB per octave. This is the most common filter type, as almost every analogue synth has one.
The Hipass filter will do the same, but with anything below the cutoff frequency. This filter is especially good for sounds without low frequencies, like harpsichords, hihats, etc.
Bandpass filters are a combination of Low and Hipass filters in series, so only the band around the cutoff frequency passes.

[] D11 **LOWPASS**: Output of Lowpass filter
[] D12 **LOWPASS**: Output of Lowpass filter
[] D13 **BANDPASS**: Output of Bandpass filter
[] D14 **BANDPASS**: Output of Bandpass filter
[] D15 **HIPASS**: Output of Hipass filter
[] D16 **HIPASS**: Output of Hipass filter

## AMPLIFIER SECTION




A voltage controlled Amplifier is a circuit that has two inputs and one output, and basically multiplies one input with the other. Consider it a voltage controlled attenuator, in which one input signal level is controlled by the other, and the result of such modulation goes to the output.

---

An amplifier module is an essential part of a synthesizer. The most basic use is to control the final volume of a patch, normally at the end of the audio chain, by an envelope generator. In that case the output of the filter (or whatever the source is) goes to the “VCA IN”, the envelope signal goes to the “VCA CV” input, and the shaped sound goes to the “VCA OUT”.
But that’s only a simple example, as it’s also very useful to control modulation signal levels with other modulation signals. By controlling an LFO output with an envelope generator with a slow attack and maximum sustain, you can create “delayed vibrato”, that is, if connected to an oscillator there will be no vibrato at first and then an increasing vibrato until the envelope reaches its sustain stage.

If we get extremely experimental we could take that VCA output and route it to another VCA’s input, and then connect one output from the MIDI interface which will be set to convert Velocity information to CV, and connect that to the second VCA’s “VCA CV” input. The final result will be a delayed vibrato with a depth controlled by how hard you play a key.

Another interesting use is to modulate an audio signal by another audio signal for Ring Modulation type effects.
Tinysizer has 4 VCAs, VCA 1 to 3 are identical with two inputs and one output on the patch bay. VCA 4 has those connections as well, but it also has two knobs on the front panel and its output is normalized to one of the standard 1/4" jacks on the side on the synth.

**Voltage Controlled Amplifier 1 to 3:**

These are simple VCAs as explained above.

**VCA1:**

[] E7, E8 **VCA IN**: Input for the signal to be modulated
[] E9, E10 **VCA CV**: Input for the modulation signal
[] E11, E12 **VCA OUT**: Output of the resulting signal

**VCA2:**

[] F7, F8 **VCA IN**: Input for the signal to be modulated
[] F9, F10 **VCA CV**: Input for the modulation signal
[] F11, F12 **VCA OUT**: Output of the resulting signal

**VCA3:**

[] J11, J12 **VCA IN**: Input for the signal to be modulated
[] J13, J14 **VCA CV**: Input for the modulation signal
[] J15, J16 **VCA OUT**: Output of the resulting signal

*Voltage Controlled Amplifier 4:

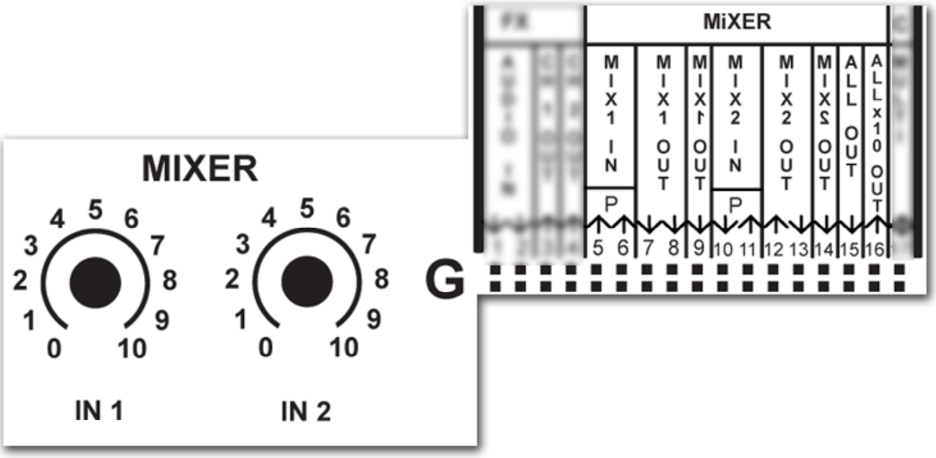
VCA4 is a little different than the other three because it has two additional controls on the front panel. Offset is a constant voltage mixed to the "VCA CV" input, and controlled by the "Offset" knob, and lets you do things like "open" the VCA without the need of a modulation input.
The gain knob controls the final volume of the VCA by attenuating the mix of its CV inputs ("VCA CV" and "Offset". VCA 4 output is hardwired to "VCA 4 SIGNAL OUT" connection on the right side of synth, and it will be used most often as the final synth output.

**VCA4:**

Ô **OFFSET**: 0 is the default setting, goes from -XX to XX.
Ô **GAIN**: Gain level, 0 to 10.

[] F13, F14 **VCA IN**: Input for the signal to be modulated
[] F15 **VCA CV**: Input for the modulation signal
[] F16 **VCA OUT**: Output of the resulting signal

**MIXER SECTION**




The mixer section on the Tinysizer is much more than a simple mixer, it's more like a multi purpose module. For starters - it's a mixer! It's a two-channel mixer with a potentiometer for each channel, and it can mix audio and CV signals as well. The sum of both channels goes to the "All Out" output.

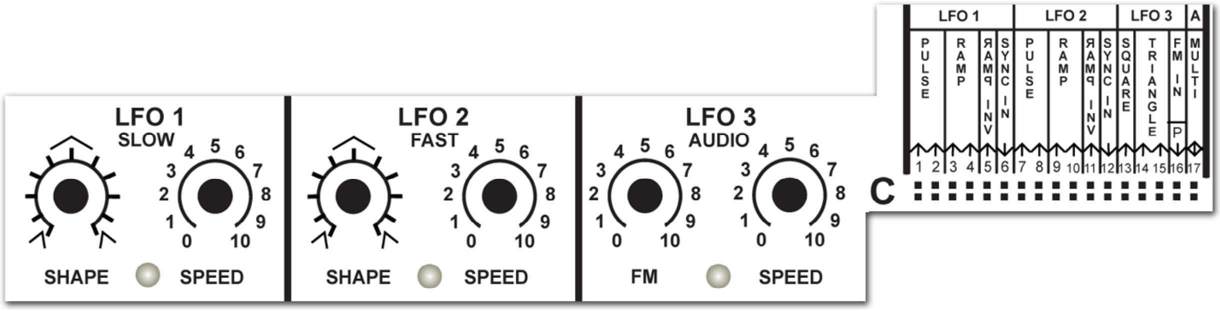
There's another output that is called "All x 10 Out", and as its name describes, it's the same signal as "All Out", but multiplied by 10. That multiplication will probably generate distortion. So here's another use of the Mixer section; as a distortion processor.

If you pay attention, each of the two inputs ("Mix 1 In" and "Mix 2 In") has an output ("Mix 1 Out" and "Mix 2 Out"). That means that the signal that enters a channel gets attenuated by the knob and goes to the mixer bus and also to an independent output. So... here's yet another use of the Mixer module: as two knob controlled attenuators! Also, each output has its inverted version, so you also get two phase inverters.

Ô **IN 1**: Attenuator knob for input one, 0 (full attenuation) to 10 (no attenuation).
Ô **IN 2**: Attenuator knob for input two, 0 (full attenuation) to 10 (no attenuation).

[] G5, G6 **MIX 1 IN**: Signal input for channel one.
[] G7, G8 **MIX 1 OUT**: Attenuated signal output for input one.
[] G9 **MIX 1 OUT** (inverted): Inverted and attenuated signal for input one.
[] G10, G11 **MIX 2 IN**: Signal input for channel two.
[] G12, G13 **MIX 2 OUT**: Attenuated signal output for input two.
[] G14 **MIX 2 OUT** (inverted): Inverted and attenuated signal for input one.
[] G15 **ALL OUT**: This output is the sum of both Mix 1 and Mix 2 inputs.
[] G16 **ALL x 10 OUT**: This output is the sum of both Mix 1 and Mix 2 inputs, but multiplied by 10, that is: distorted.

## LFO SECTION




An LFO is basically an oscillator whose frequency range is usually below the audible spectrum. Its used as a modulation source because it produces cyclic voltage fluctuations. By applying an LFO to an audio oscillator's FM input you get something like a sweep, siren type of sound if the LFO's speed is very slow and the FM amount is relatively strong; a vibrato if modulation depth is moderate and the speed is medium, and timbre change if speed is faster than 10/20 Hz. Some of the basic audio oscillator functions apply here, like waveform, speed, PWM, Sync, FM, etc. LFOs 1 & 2 are identical in functionality, only the speed range is different, and LFO 3 has other features and a different speed range.

---

### LFO 1:

LFO 1 has two simultaneous and waveshape-variable waveforms. Each has its own independent output on the patch bay. PULSE is a pulse wave whose width is variable from the front panel's "SHAPE" knob. RAMP waveform is similar, but the waveform begins with a rising Saw (also called Ramp) when the SHAPE knob is at the left, a Triangle wave when the knob is at the center, and a falling Saw when the knob position is at the right. RAMP INV on the patch bay outputs an inverted version of RAMP.
The Speed range of this LFO is optimized for slow rates.
A Sync input is provided for retriggering purposes. Every positive zero-crossing of an input will retrigger the LFO.

Ô **SHAPE**: PULSE: from 0% to 100% Pulse width.
RAMP: From Ramp (0) to Triangle (5), to Saw (10).
Ô **SPEED**: From 30 sec per cycle to 5 Hz.
[] C1, C2 **PULSE**: Pulse waveform output. Output level: 0V to +5V
[] C3, C4 **RAMP**: Ramp waveform output. Output level: -5V to +5V
[] C5 **RAMP INV**: This is the phase-inverted output of RAMP.
[] C6 **SYNC IN**: A reset input

---

**LFO 2:**

Identical to LFO 1, except for its speed range. This LFO is optimized for fast speed operation.

Ô **SHAPE**: PULSE: from 0% to 100% Pulse width.
RAMP: From Ramp (0) to Triangle (5), to Saw (10).
Ô **SPEED**: From 2Hz to 200 Hz.
[] C7, C8 **PULSE**: Pulse waveform output. Output level: 0V to +5V
[] C9, C10 **RAMP**: Ramp waveform output. Output level: -5V to +5V
[] C11 **RAMP INV**: This is the phase inverted output of RAMP.
[] C12 **SYNC IN**: A reset input

---

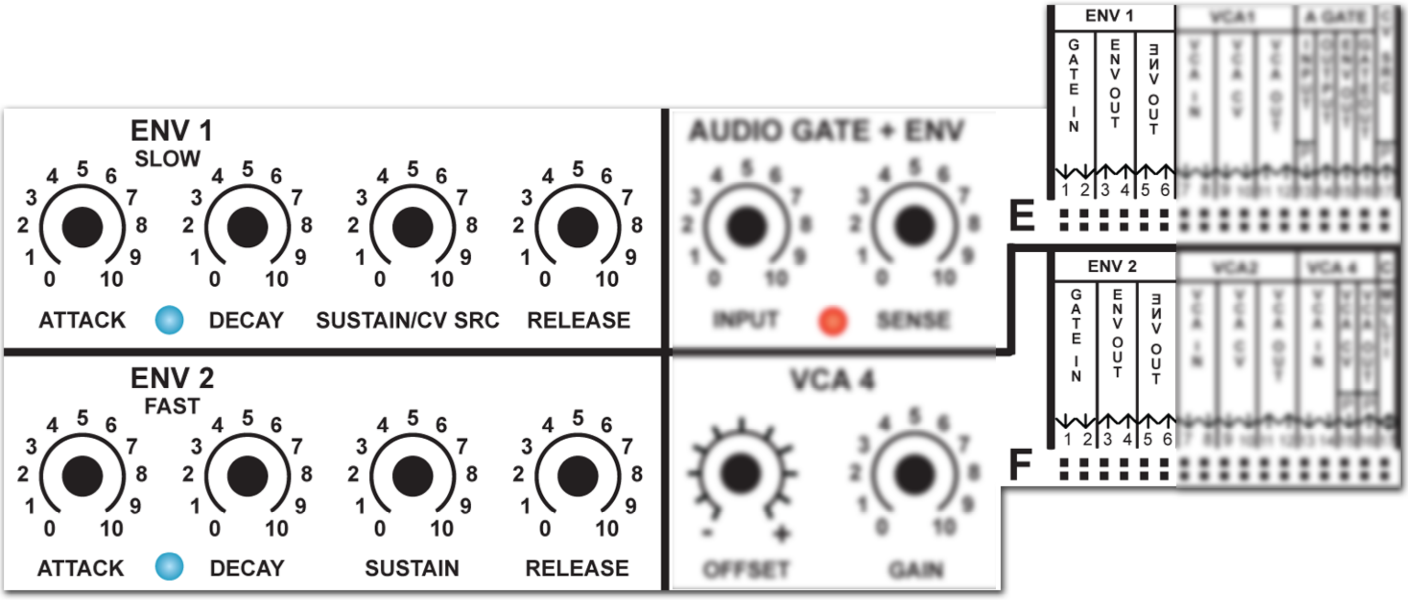
**LFO 3:**

LFO 3 is different from the other two; it's nice to have variety in functionality. It has two waveforms (Square and Triangle), but their shape can't be modified. The big difference lies in that it has an FM input for its speed to be modulated, and its frequency range goes well into the audio range, so in a way it can also be used as an audio oscillator (not ideal).

Ô **FM**: Attenuator for modulation input on C16
Ô **SPEED**: From 0.5 Hz to 310 Hz
[] C13 **SQUARE**: Square waveform output. Output level: 0V to +5V
[] C14, C15 **TRIANGLE**: Triangle waveform output. Output level: -5V to +5V
[] C16 **FM IN**: This is a modulation input for the LFO's speed

---

**ENVELOPE GENERATORS SECTION**




An envelope generator is a module that generates an evolving voltage at the moment it receives a gate signal at its input. The shape of that evolving voltage depends on the parameters attack, decay, sustain and release. Its used as a way to control parameters in a predictable way across the evolution of a note.

Envelope generators in theory have one input (Gate) and one output (Voltage), but Tinysizer has more. There are two Gate inputs (GATE IN), two Voltage outputs (ENV OUT), and two inverted Voltage outs (3NV OUT). To trigger an Envelope Generator you have to connect a Gate signal to one of its "GATE IN" patch bay sockets. Once the envelope is triggered, it will evolve through its various stages as long as the gate signal is still "ON". If, at any point the incoming Gate signal goes "OFF", the envelope will jump to its Release stage. Now let's take a look at the parameters themselves.

Attack: Is the TIME it takes to go from 0 (when the Envelope Generator begins) to maximum level.

Decay: After the Attack stage ends, the decay stage begins, which in fact is the TIME it takes to go from the Attack to Sustain stages.

Sustain: This is the next stage, and is the LEVEL at which the sound stays, after the decay, as long as the key is still held. If sustain level is 10, then decay stage is cancelled.

Release: This last stage defines the TIME it takes the Envelope Generator to go from whatever stage it is in at the moment we release a key to zero.

This Envelope Generator type is called ADSR for obvious reasons. There's a blue LED indicating when the envelopes are being triggered.

There are 2 envelope generators within Tinysizer. They're identical in their connections and parameters, but they have different time ranges.

---

**Envelope Generator 1:**

This Envelope Generator is used for slower time requirements; for evolving sounds like strings, drones, pads, etc. Sustain knob also functions as a "CV Source" (see page **XX**).

Ô **ATTACK**: Defines the Attack time, from 3 ms to 20 secs.
Ô **DECAY**: Defines the Decay time, from 3 ms to 25 secs.
Ô **SUSTAIN**: Defines the Sustain level, from 0 to 5 volts.
Ô **RELEASE**:Defines the release time, from 3 ms to 35 secs.
[] E1, E2 **GATE IN**: Gate inputs of Envelope Generator 1
[] E3, E4 **ENV OUT**: Gate outputs of Envelope Generator 1
[] E5, E6 **3NV OUT**: inverted Gate outputs of Envelope Generator 1

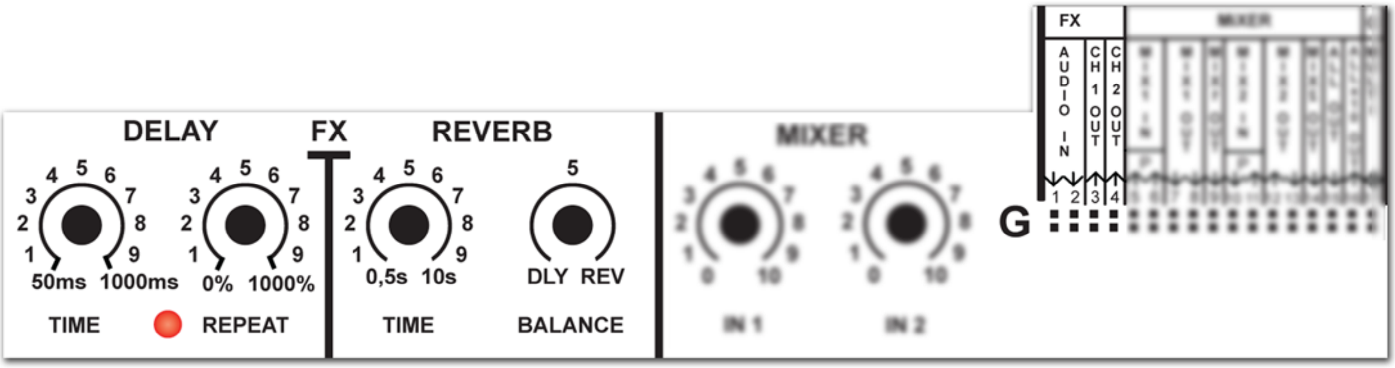
---

**Envelope Generator 2:**

Env2 is similar to Env1, but its times are shorter, which makes it better for very percussive sounds. Sometimes, one of the aspects that define if a synth is really good is the ultra fast speed of its envelopes times. Env2 minimum times are under one millisecond, which is extremely fast.

Ô **ATTACK**: Defines the Attack time, from 0,8 ms to 2 secs.
Ô **DECAY**: Defines the Decay time, from 0,8 ms to 4 secs.
Ô **SUSTAIN**: Defines the Sustain level, from 0 to 5 volts.
Ô **RELEASE**: Defines the release time, from 0,8 ms to 8 secs.
[] F1, F2 **GATE IN**: Gate inputs of Envelope Generator 1
[] F3, F4 **ENV OUT**: Gate outputs of Envelope Generator 1
[] F5, F6 **3NV OUT**: inverted Gate outputs of Envelope Generator 1

**FX SECTION**




Included with the Tinysizer is a stereo digital effects section, consisting of separate Reverb and Delay processors. This is very interesting and rare in an analogue modular synthesizer. It really comes in really handy for live situations, of course, as it saves the effort of carrying a separate effects processor for delay and reverb. But from an experimentalist point of view, it is really great to have those effects included and ready to be inserted anywhere in the signal chain. Things like filtered delay lines, or oscillators passed through a reverb before the filter, are just a couple of examples. What about a delay distorted with the waveform processor?

Both effects run in parallel, and there's a "Balance" knob for adjusting the amount of each, as they have a common pair of inputs and outputs.
These are not insert effects; they must be run in parallel with the original signal. Reverb is a pseudo stereo effect; that's why there are two outputs.

**Delay FX:**

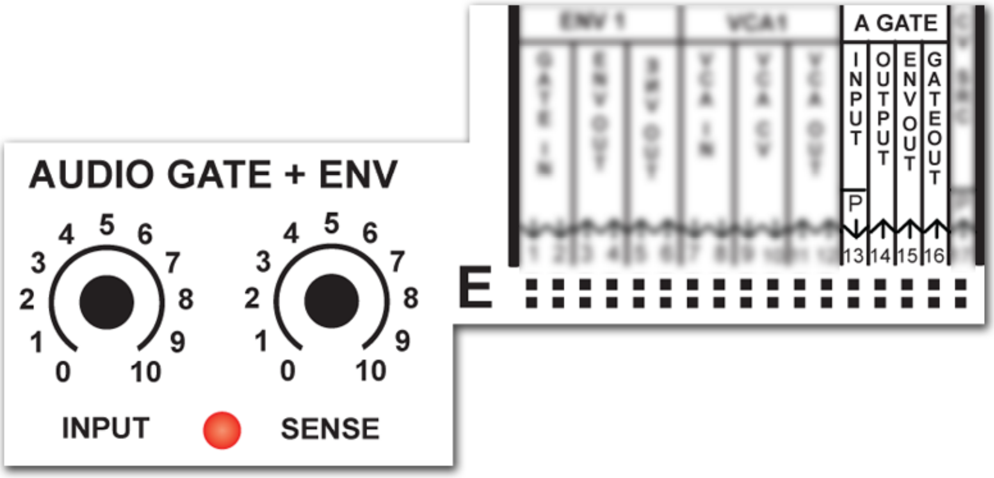
Ô **TIME**: Delay time, from 50ms to 1 second. It defines the time between "echos".
Ô **REPEAT**: This knob sets the "feedback" amount, hence the number of repetitions. Range 0 to 90%.

**Reverb FX:**

Ô **TIME**: Reverb decay time, from 0.5 to10 seconds.
Ô **BALANCE**: With this knob the balance between Delay and Reverb can be adjusted. Full left knob position will give you only Delay effect; full right position will give you only reverb. Anything in between will give a mixture of both effects.

[] G1, G2 **AUDIO IN**: Audio inputs to the effects processor.
[] G3 **CH 1 OUT**: Audio output 1 of the effects processor.
[] G4 **CH 2 OUT**: Audio output 2 of the effects processor.

**AUDIO GATE + ENVELOPE FOLLOWER SECTION**

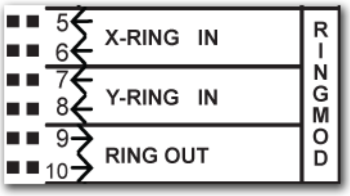



This is another very interesting section, especially for integration with the outside world. This is great for loop processing for example. What you get here is a mic preamp, an envelope follower and a gate detector. All this has to be used in conjunction with the AUX ins/outs.
First in the chain you have a hi-gain mic preamplifier (Preamp) for adjusting the external signal's level. There's an audio signal input on the patchbay that goes directly to the Preamp. There's also an audio output with the amplified signal. This can also be used as an attenuator or saturator if abused (another distortion stage, cool!). Once the mic/guitar/sampler is correctly leveled, the Envelope Follower (EF) can be used. An Envelope Follower is a circuit that generates a voltage equivalent to the amplitude of the incoming signal. The EF is already connected to the Preamp, so no connection needs to be made. If, for example, a guitar is connected and routed to a Band Pass Filter, and the EF is routed to the filter's cutoff, you get what it's commonly known as an "auto-wah", where the brilliance is controlled by the intensity of the guitarist's playing. (See the Tips section for a diagram on page 47).
Also included is a Gate Detector that senses the peaks, according to the input level and "Sense" knob, and generates a gate signal to trigger the envelopes. This is useful for triggering drum sounds from a mic, for example.

Ô **INPUT**: Input knob sets the preamp gain on the input level.
Ô **SENSE**: This knob controls the sensitivity of the Gate Detector. A higher level gives more sensitivity.

[] E13 **INPUT**: This is the audio input of the Preamp
[] E14 **OUTPUT**: This is the audio output of the Preamp
[] E15 **ENV OUT**: Output of the Envelope Follower. Range 0 to 5v
[] E16 **GATE OUT**: Gate output generated by the Gate Detector. Range 0 to 5v

**RING MODULATOR**




A Ring Modulator is a really cool audio effect commonly used for processing external audio or for bell-like tones. It consists of a special type of VCA, which modulates one of its two inputs with the other, generating new harmonics based on the sum and difference of each of the input's frequencies.

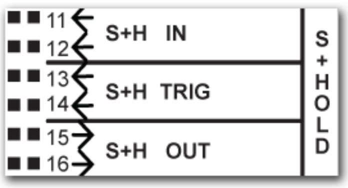
The only real difference between a Ring Modulator and a normal VCA is that the former allows bipolar signals on its modulation input. This generates a phase inversion on the main input every time the modulator goes bellow 0V.

This module has two inputs, called X and Y, and one output. In the case of the RM, there's no difference between the two inputs, as they both accept bipolar signals. The output sends the result of the multiplication, not any of the original sources.

[] I5, I6 **X-RING IN**: Input for signal X
[] I7, I8 **Y-RING IN**: Input for signal Y
[] I9, I10 **RING OUT**: Output of the multiplication between X and Y

---

**SAMPLE & HOLD**




Sample & Hold is a multi-purpose module that can be used for control or audio processing. It basically retains the value of the input voltage on the "S&H Input" every time a pulse is received on the "S&H Trig" input. If a noise generator is connected to the "S&H input", and an LFO to the "S&H Trig" input, typical robot/vintage computer effects can be created. But lots of other more interesting things can be created as well. As S&H circuits are used on A/D converters, sample rate reduction effects can be made by connecting an audio signal to the "S&H input", and an audio rate oscillator to the “S&H Trig” input. Varying the frequency of the oscillator would be equivalent to changing the sample rate of a converter. Try it!

[] I11, I12 **S&H IN**: Input for the signal to be “sampled”
[] I13, I14 **TRIG IN**: Input for the gate signal
[] I15, I16 **S&H OUT**: Output for the resulting signal

---

**MULTIPLES**

Multiples are really handy modules, and are used mainly to multiply one signal into various outputs. Connect the output of one module to one of the Multiple patch points and the signal will be multiplied on the others. Fortunately the Tinysizer’s various modules have multiple outputs/inputs, so Multiples are not as necessary as on other modular synthesizers.
In fact, Multiples may be used for other purposes. As they are distributed across the Patch Bay, you can use them as a way to route one signal from one side to the other without the use of long cables. For example, for triggering the envelopes, you could connect “GATE 1” (H1) to MULTI B (H17), and then MULTI B (I17) to ENV1 GATE IN (E1) and MULTI B (J17) to ENV2 GATE IN (F1), all with small patch leads.
Another use of a Multiple is as a mixer; connect the output of two modules to two Multiple inputs and that sum will be output on the rest of the Multiple patch connectors.

Anyway, three multiples are included, Multiples A, B and C. Multiple A has 8 patch points, B has 6 and C 4. Any kind of signal can be used, audio, CV or gate.

MULTI A
[] A17, B17, C17, D17 **MULTI A**: x2 because each one has two patch points

MULTI B
[] H17, I17, J17 **MULTI B**: x2 because each one has two patch points

MULTI C
[] F17, G17 **MULTI C**: x2 because each one has two patch points

---

**CV SOURCE**

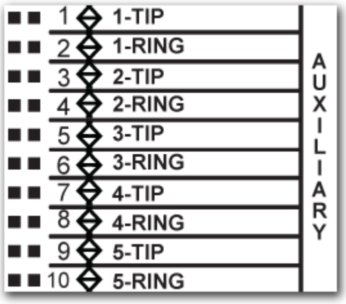
The ENV1 sustain stage also acts as a CV Source. That means that that knob has an independent voltage output that can control any parameter that has a mod input.
Tip: If you feel that you need more knobs or that you need ENV1, the MIDI interface can route MIDI CC messages to CV, so you can use any MIDI controller with sliders as multiple CV Sources!

Ô **SUSTAIN/CV SRC**: Knob that determines the CV level

[] E17 **CV SRC**: Voltage output of the CV source knob. Range 0 to 5v

---

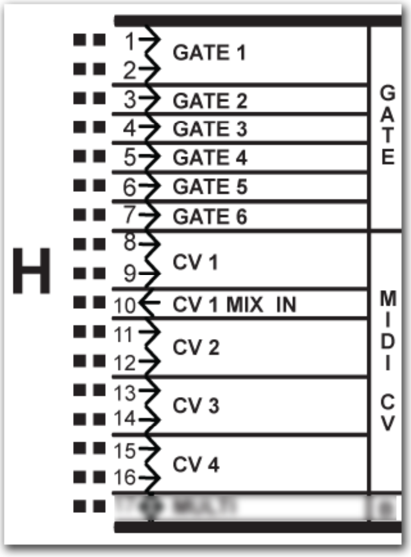
**AUXILIARY**




“Auxiliary” is the key to connect the Tinysizer to the rest of the world. The idea is simple, you get five stereo ¼” jacks - that means ten independent channels with their respective connectors on the Patch Bay. So you can use each channel as an independent input or output for audio, CV or gate. You may want to buy a stereo to (2) mono adaptor.
Then you can connect a guitar to AUX 1 (TIP) and route that from the AUX patch point (J1) to the Preamp input (E13), for further processing.
Or you can control external analogue synthesizers with the MIDI to CV interface. Route CV to one AUX and Gate to another and that’s it!

[] J1 **1-TIP**: Patch point of AUX1 TIP
[] J2 **1-RING**: Patch point of AUX1 RING
[] J3 **2-TIP**: Patch point of AUX2 TIP
[] J4 **2-RING**: Patch point of AUX2 RING
[] J5 **3-TIP**: Patch point of AUX3 TIP
[] J6 **3-RING**: Patch point of AUX3 RING
[] J7 **4-TIP**: Patch point of AUX4 TIP
[] J8 **4-RING**: Patch point of AUX4 RING
[] J9 **5-TIP**: Patch point of AUX5 TIP
[] J10 **5-RING**: Patch point of AUX5 RING

**MIDI TO CV INTERFACE**




Included with the Tinysizer is a 6 Gate/4 CV MIDI to CV converter. This provides a very comprehensive integration with a modern studio environment. Not only are MIDI notes recognized, but also Continuous Controllers (CC), Pitch Bend, Velocity, Midi clock and Star/Stop messages.
The interface can be used on a stand-alone basis or can be programmed with an application for PC. All settings are stored on flash ROM memory, so they're not lost when powered off. Thanks to the AUXs you can even control external analogue synthesizers!
The connections are very simple, but the function of each may depend on the configuration set by the PC software. There are six gate patch points; the last two may be used for MIDI clock conversion and Start/Stop messages, and 4 CV outputs for controlling elements with note on/off number, CC, etc.

[] H1 **Gate 1**: Patch point of Gate 1. Range 0 or 5v
[] H2 **Gate 1**: Patch point of Gate 1. Range 0 or 5v
[] H3 **Gate 2**: Patch point of Gate 2. Range 0 or 5v
[] H4 **Gate 3**: Patch point of Gate 3. Range 0 or 5v
[] H5 **Gate 4**: Patch point of Gate 4. Range 0 or 5v
[] H6 **Gate 5**: Patch point of Gate 5. Range 0 or 5v
[] H7 **Gate 6**: Patch point of Gate 6. Range 0 or 5v

[] H8 **CV 1**: Patch point of CV 1. Range 0 to 10v
[] H9 **CV 1**: Patch point of CV 1. Range 0 to 10v

[] H10 **CV 1 MIX**: Patch point of CV 1 mix. This is an input instead of an output. Anything you connect here (LFO, ENV, etc) will be summed to the CV1 outs (H8 y H9)
[] H11 **CV 2**: Patch point of CV 2. Range 0 to 10v
[] H12 **CV 2**: Patch point of CV 2. Range 0 to 10v
[] H13 **CV 3**: Patch point of CV 3. Range 0 to 10v
[] H14 **CV 3**: Patch point of CV 3. Range 0 to 10v
[] H15 **CV 4**: Patch point of CV 4. Range 0 to 10v
[] H16 **CV 4**: Patch point of CV 4. Range 0 to 10v

**Stand-alone operation:**

First of all, connect the MIDI output of your controller/sequencer to the MIDI in of the Tinysizer.

To program the MIDI to CV interface from the synth itself the MIDI LEARN button at the right side has to be used.
This is how it works:

**_To set the MIDI receive channel:**

1) Push and hold the MIDI LEARN button.
2) Send any “Channel Voice Message” from your controller - that is any message that includes MIDI channel information (note on/off, CC, pitch bend, etc)

**_To set the base note/transpose :**

(Base note is the lowest note the interface will play = 0v)
1) Push and hold the MIDI LEARN button.
2) If the MIDI Receive Channel of the Tinysizer is the same as the note you’ll define as the base note, just play the desired key on your controller. If not, the first note you play will define the MIDI Receive Channel first (as explained earlier), and THEN you’ll have to play the base note key.

**_To learn a MIDI CC message:**

1) Push and hold the MIDI LEARN button.
2) If the MIDI Receive Channel of the Tinysizer is the same as the CC message you want use, just send the desired CC message from your controller (turn a knob or move a slider). If not, the CC message you send will define the MIDI Receive Channel first (as explained earlier), and THEN you’ll have to send the CC message again.
3) The learned CC message will be converted on CV3.

So, to sum up: The first midi message will define the MIDI Receive Channel, next, if the following received message is a note on, it will define the base note and if its a CC message it will be learned an be assigned to CV3.

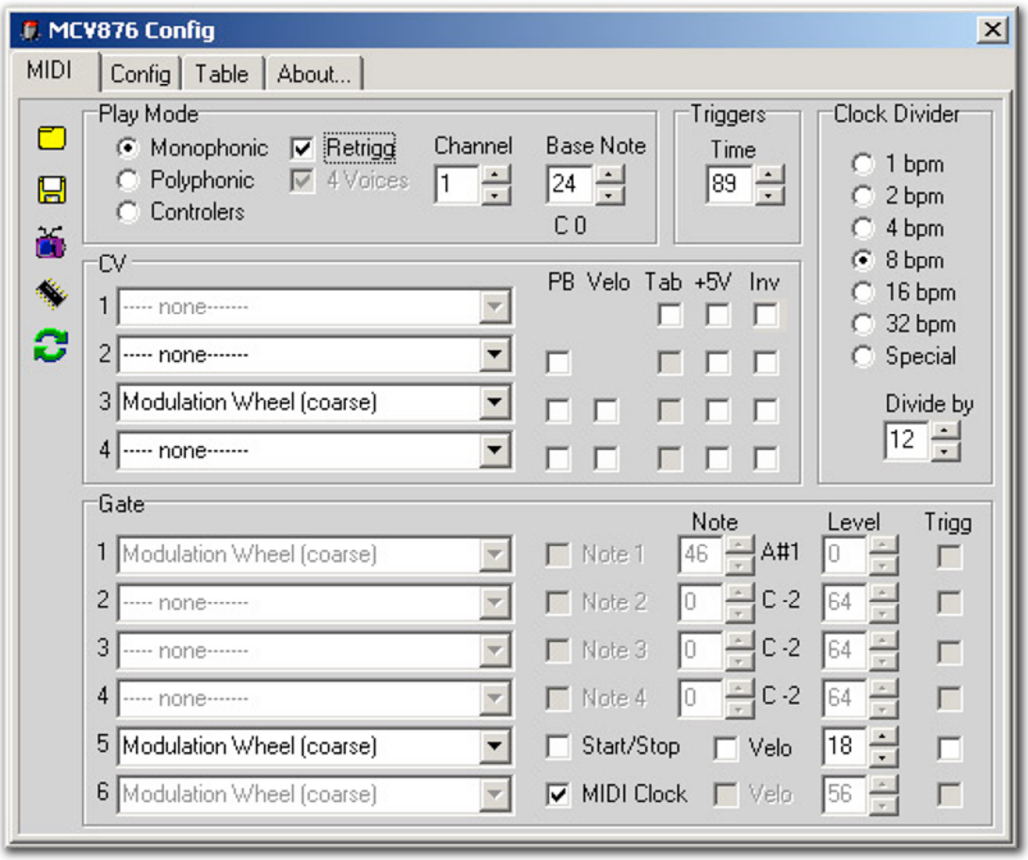
If the received note or CC message is in the same MIDI channel as the MIDI receive channel, then it's not necessary to set the MIDI receive channel each time, so the first step may be skipped.

**Remote operation (via PC software):**

The software application, called "MCV876 Config", runs on any PC with a MIDI interface. To run it just double-click on the .exe file.
First of all you need to set the MIDI interface.
1) Go to the "Config" Tab
2) Select your MIDI interface on the "Device" section

Then connect your computer's MIDI interface output to the Tinysizer's MIDI input.

**MIDI Tab:**




**PLAY MODE**

**Channel:** Defines the MIDI Channel.
**Base Note:** Defines which will be the lowest key on the keyboard, that is: which note will send 0 volts.
**Monophonic:** This should be the normal operating mode. “Gate 1” sends the gate signal, “CV 1” sends the note. “Retrigg” is an option that allows you to retrigger the voice each time a new note is played. By un-clicking it, you enter a mode commonly called “legato” in which a new note isn’t triggered if the first note is not released before.
**Polyphonic:** In this mode you can use the MIDI to CV converter to manage “voices”. There are two modes: Poly2 and Poly4. If “4 voices” is not selected, then you’re in Poly2 mode, in which Gate 1 & 2 and CV 1 & 2 control two different voices. The rest of the CV and Gate outputs can be freely assigned to controllers, etc. If “4 voices” is selected, then all 4 CV output and the first 4 Gate outputs are used to control 4 discrete voices.
**Controlers** (sic): In this mode all outputs (CV and Gate) become CV outputs from assignable controllers. CV 1 to 4 will be continuous, while Gate outs will have only two values (on and off) determined by certain conditions. (See next).

**CV** (1 to 4):

**Controller Menu:** This menu allows you to select the desired CC for the chosen output. The list includes all CC messages.
**PB:** Pitch bend Message. It works in monophonic mode only. It has a range of 0v to 10v, with a center of 5v. Not available on CV out 1. Its output has to be added to CV1 in the analogue domain for typical pitch bend effect.
**Velo:** Allows Velocity messages to be converted to the selected output.
**Tab:** Click here if you want that output to be controlled by the table. The table, on the “table” tab, lets you make different tuning scales.
**+5V:** Sets the maximum output level to +5V, which means it reduces the output level by half.
**Inv:** Inverts the output phase of the chosen message.

**GATE** (1 to 6)
**Gate Menu:** These options depend on whether you’re in Monophonic/Polyphonic or in “Controllers” mode. In Monophonic/Polyphonic mode you can only chose controllers for the last two Gate outputs (5 & 6). In the case of Monophonic, Gate 1 sends the Gate signal, and Gates 2 to 4 do nothing. Same thing with Poly2; Gates 1 and 2 send Gate signals, Gates 3 and 4 don’t. In Poly4, all four Gate outputs generate Gate signals.
Gates 5 & 6 allow controllers to be used to generate Gate signals. The CC list includes all available controllers. See next sections for more options for these outputs.
“Controllers” mode: This mode allows you to generate a gate signal from controllers or specific notes.
If the “Note” option is not clicked, then the Controller list is available. Once the desired controller is chosen, you have to set a “level”, that is a threshold at which the controller begins to generate the gate signal. The “Trigg” option lets you generate a Trigger signal - a short pulse whose duration is set by the “Triggers Time” at the top of the application’s window. A trigger is useful to reset an LFO for example.

If the “Note” option is clicked, then you can select any note from the keyboard to generate a Gate signal. This is useful in making drum kits, etc., in which each sound is assigned to a certain note.
The note in question is chosen through the “Note” menu.
Gates 5 & 6 are a little different. They can be used as Controllers anytime, with its threshold “Level”. They can also be generated by velocity instead of a controller in which case the “Level” sets the velocity threshold. Beyond that, Gate 5 can generate a Gate signal with Start/Stop MIDI Messages. In that case, your sequencer has to be configured to send those messages each time it starts or stops.
Gate 6 can generate MIDI clock signals also derived from a MIDI sequencer or drum machine. The “MIDI Clock” option has to be checked. MIDI Clock signals generate 24 pulses per quarter note, so the “Clock Divider” at the top right of the window takes action. There you choose the Clock division you prefer... if you’re working in a 4/4 time, “1” means one pulse per bar, “2” means 2 pulses per bar, etc. Obviously, for a 16th note you’d have to choose “16”. “Special” allows you to choose other note lenghts, like triplets and doted notes, just do the maths... 😉
For a dotted 8th, you have to think of an 8th (12 divisions) and add a 50% (6), so you need to divide it by “18”.

**OPTIONS**

At the left you have extra options:

**Open Conf:** Here you can open MCV Config files and MCV Note table files.

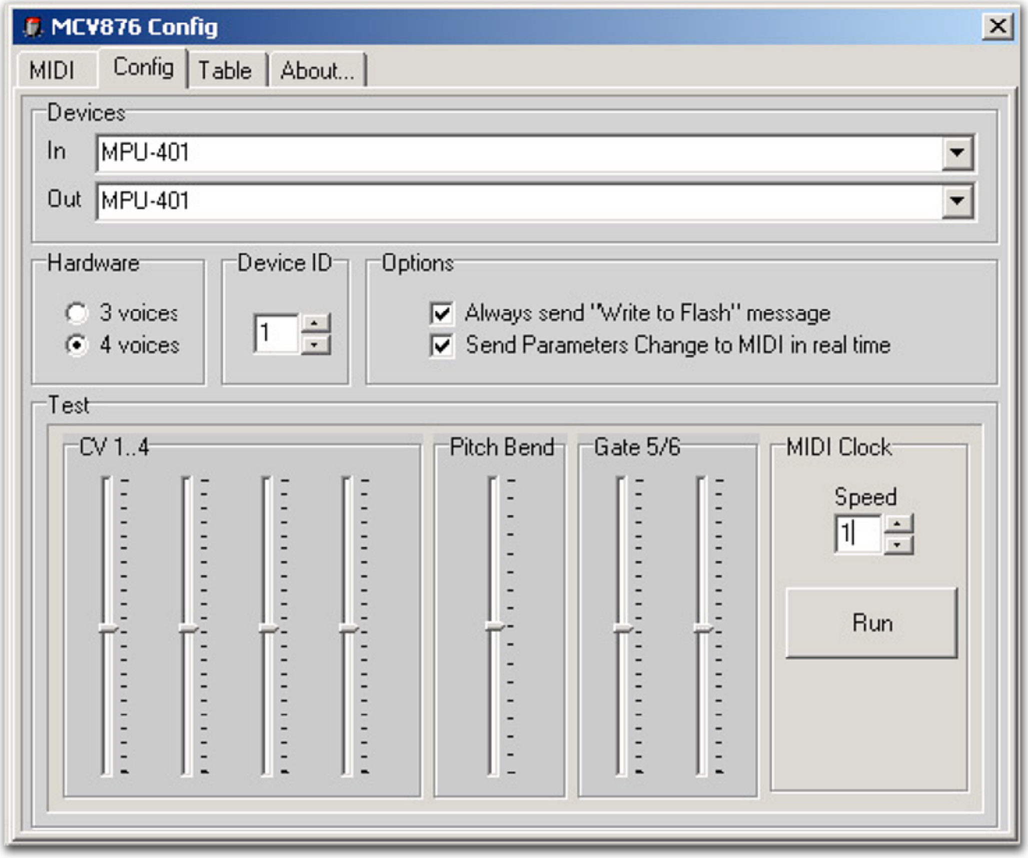
**Save Conf:** This is to save different configurations or tuning tables.

**Save Conf as Sysex:** This allows you to save the config as a sysex file.

**Send “Write to Flash”:** This option allows you to save the configuration made on the MCV software on the Tinysizer’s non-volatile memory.

**Send Configuration to MIDI:** This sends all the available parameters on the applications to the Tinysizer. Use this function if you load a Config file. This is also useful if the MIDI to CV interface begins to work erratically while you’re editing, (perhaps because it didn’t update some parameter), and this option will reload all of them.

**Config Tab**




**Devices:** Here you select your MIDI interface to be used with the Tinysizer.

**Options:**
"Always send "write to flash" message": This allows you to write to flash every time you make a change in the program. You don't need to push the "write to flash" button as previously explained.
"Send parameters change to MIDI in realtime": This also allows you to send the changes in realtime without the need to push the "send configuration to MIDI" to update the synth.

**Test:**
These sliders are for checking the CV outputs, as well as Gates 5 & 6 and Pitch Bend without the need to have a MIDI controller at hand.

**Table Tab**


This is a programmable table that allows you to program your own tuning scales.
There's a range of 48 notes (0 to 47), with an 8-bit resolution, that is 256 levels (0 to 255).
On the graph, the X-axis represents the note, and the Y-axis represents the level (tuning).

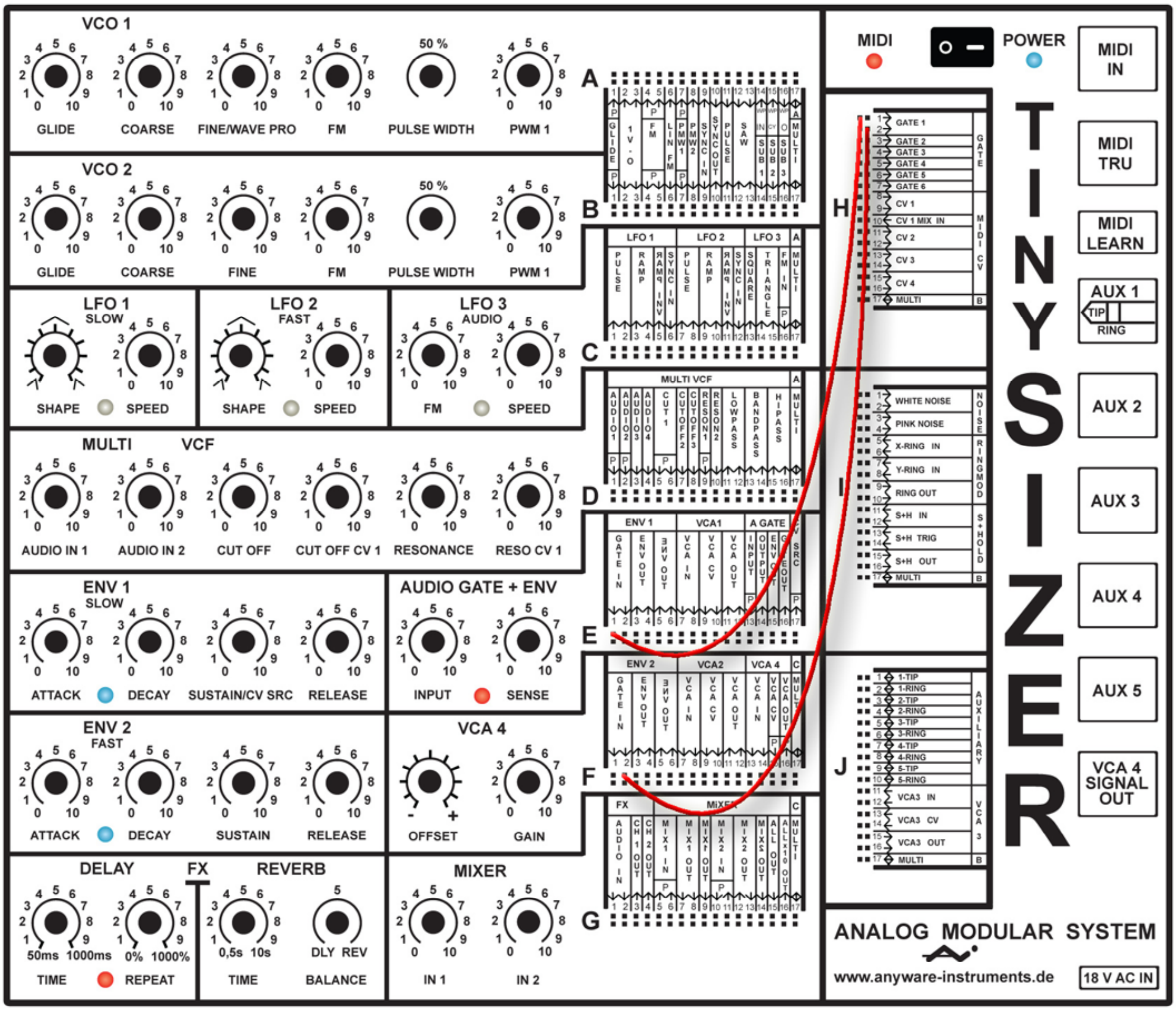
**New Table:** Reset current Table.
**Open Table:** This lets you load Note Table files.
**Save Table:** Save your new tables with this option.
**Generator:** This is a Table Generator for automatic creation of tables.
**3D/2D:** The Table can be seen as a 3D or 2D graphic.

# PROGRAMMING EXAMPLES

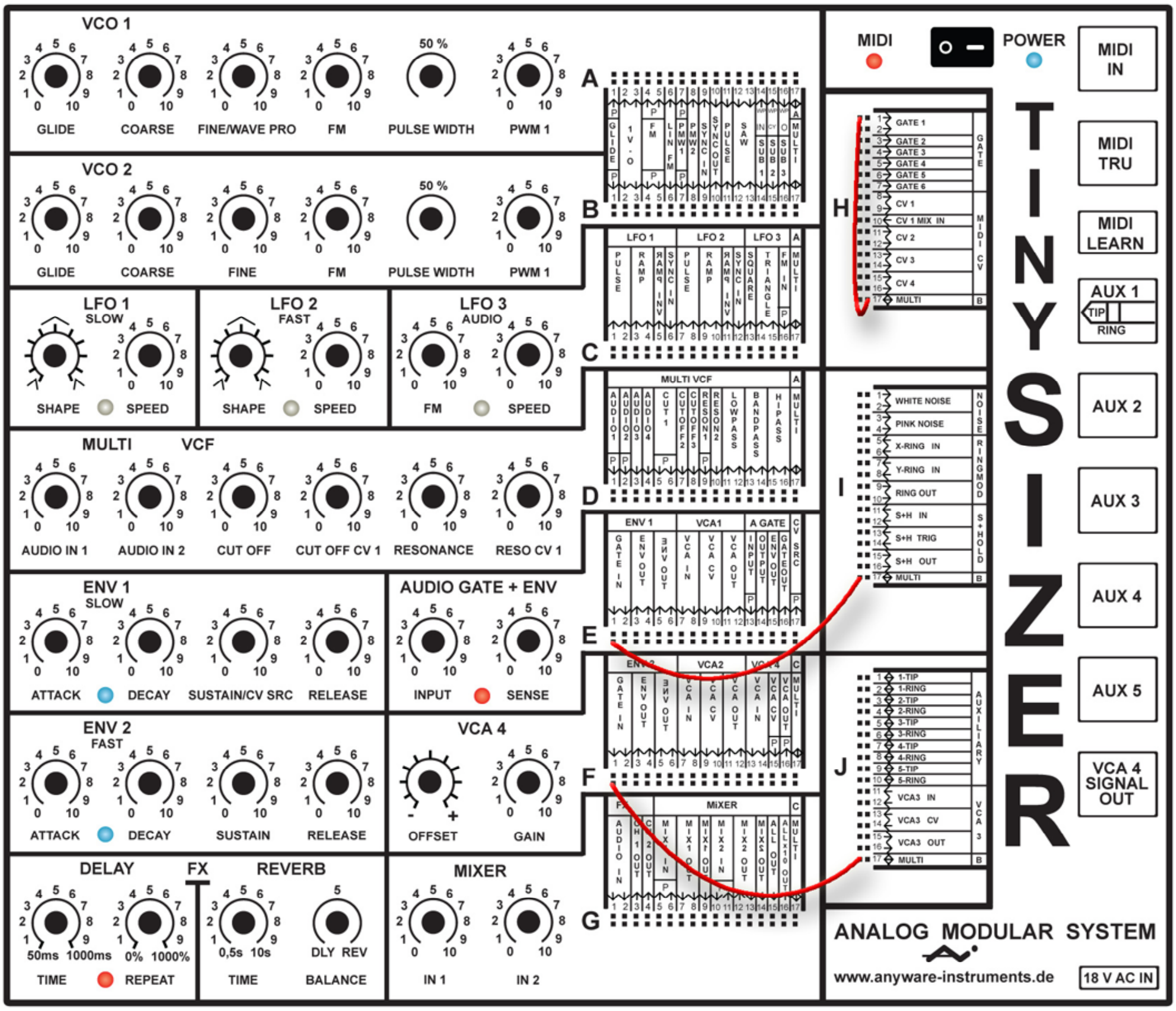
On this section of the manual, we'll guide you through the process of building some basic patches.

First of all, for a standard synth patch we have to make some standard connections, we'll begin with the MIDI to gate settings.

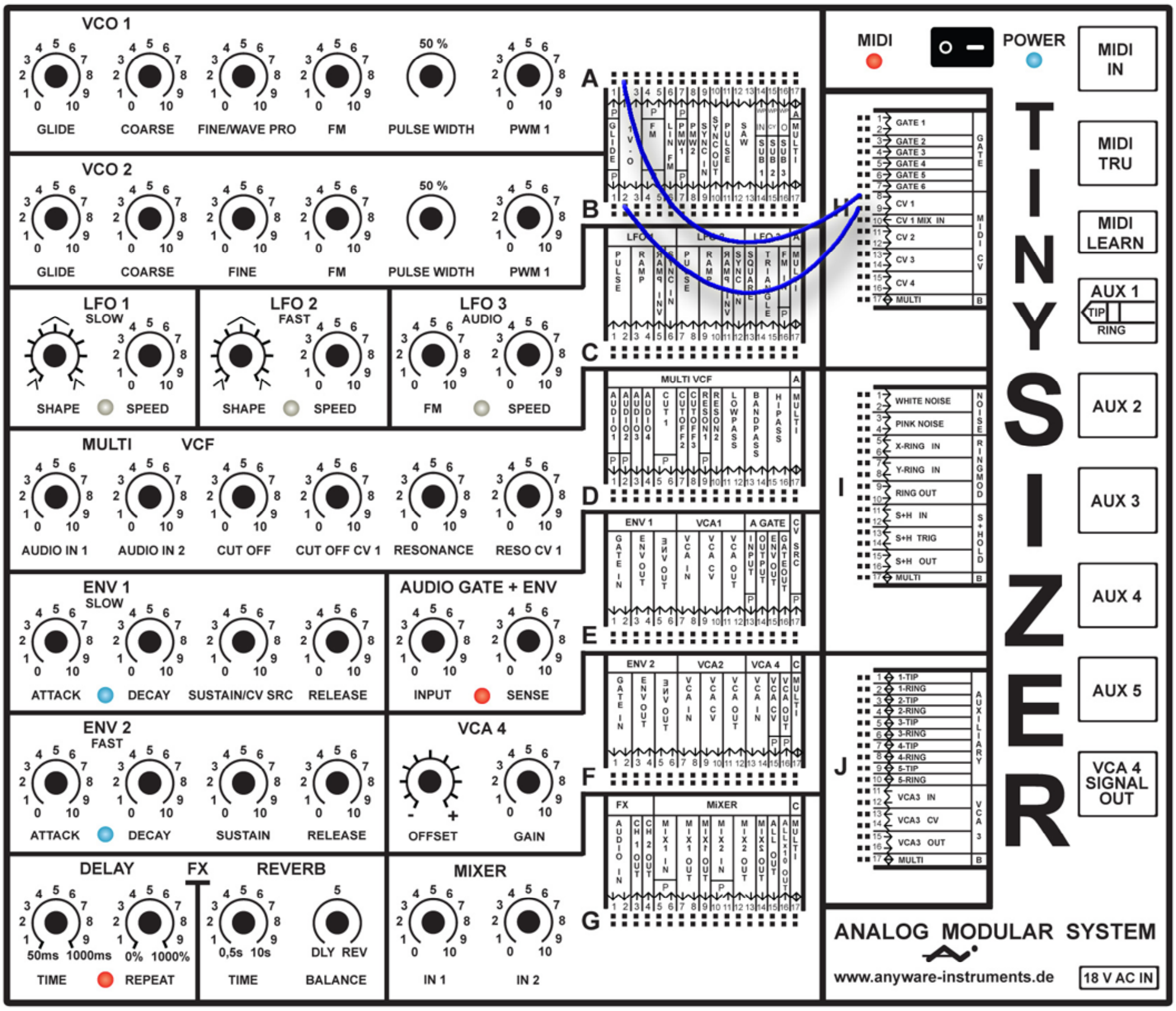
**BASIC SYNTH PATCH**



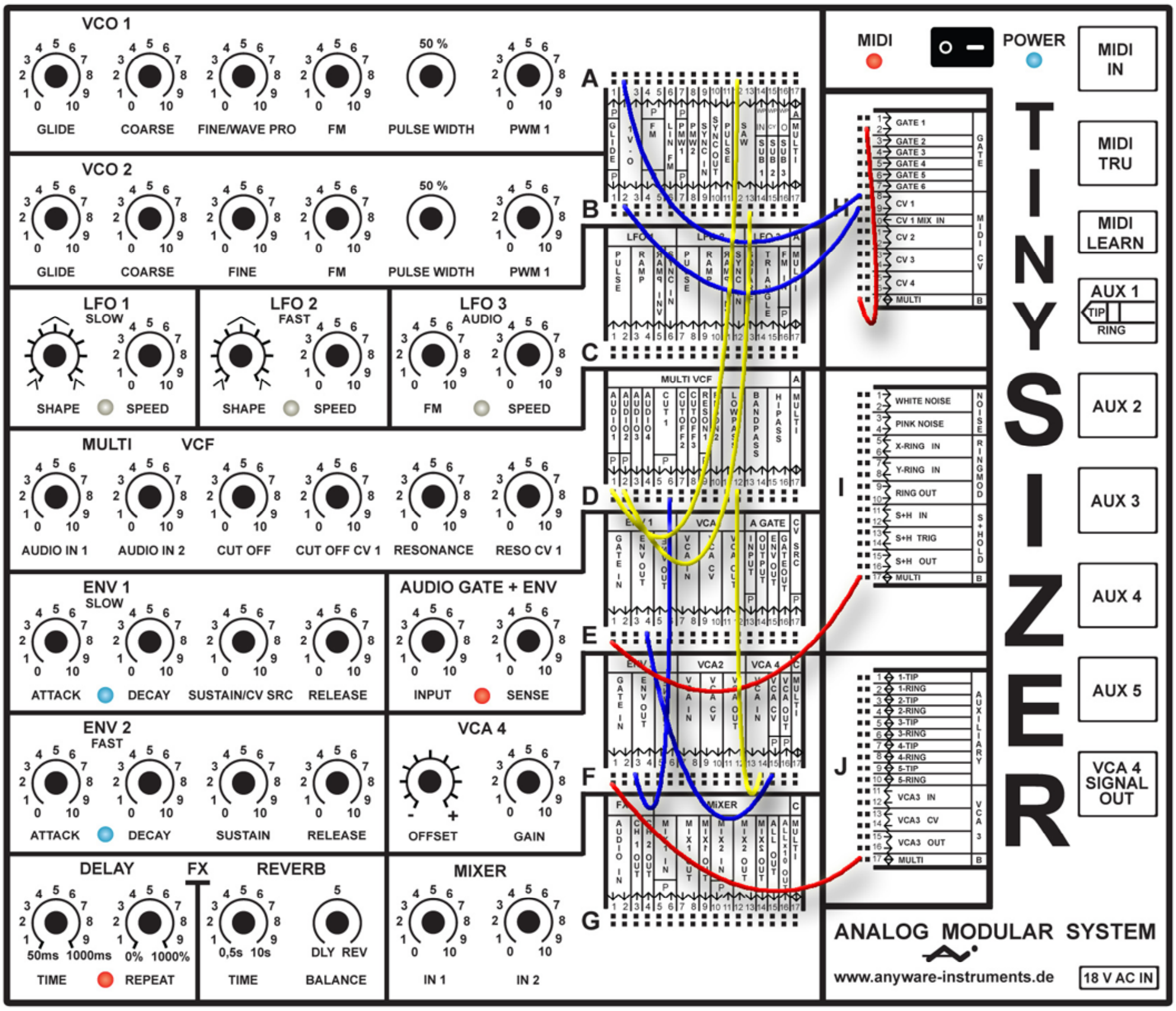

The “Gate 1” output of the MIDI to CV converter goes to the “Gate In” of both envelopes. There's another way of making these connections, that is using multiples, and that make us avoid using long cables that get along across the panel.




Next we'll be doing the CV routings so the oscillators (and optionally the filter) can track the keyboard:



Now we're going to make some important connections: First we'll patch the audio sources (in this case the oscillators) to the mixer of the filter, and then patch the Filter to the VCA.
The second patch connections will be to control the Filter and VCA with the Envelopes.
From now on, we'll represent the different kinds of signals with colours: Blue for CV, Yellow for Audio and Red for Gate.



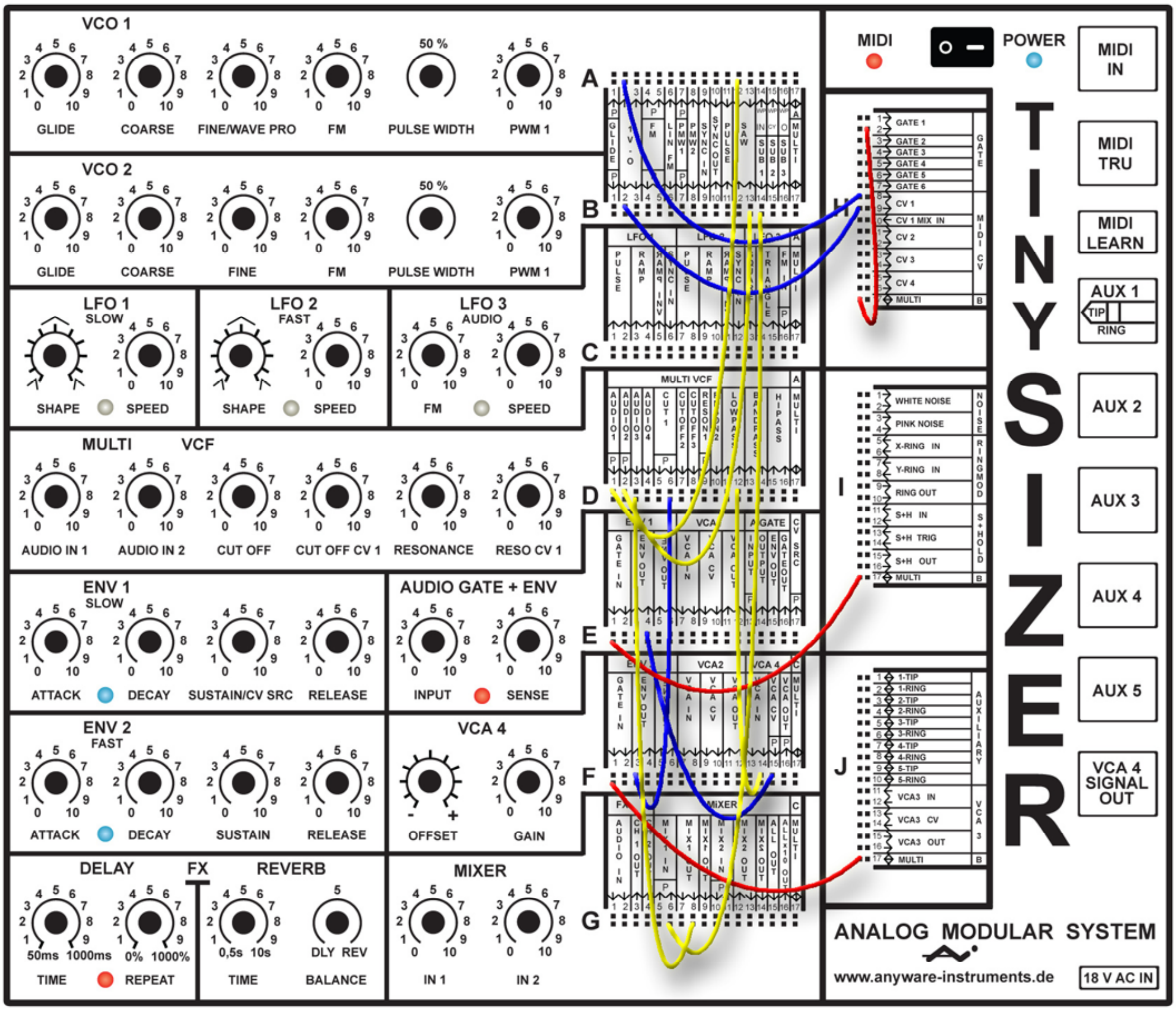

For all this to work you need to make two additional external connections: From the MIDI Out of your MIDI controller/PC to the MIDI In of the Tinysizer, and an Audio cable from VCA4 Signal Out at the side of the synth to your mixer, sound card, etc.

That's it!

But hey, all this doesn't warrant you that the synth will actually make a sound. The knob settings now play a fundamental roll. It doesn't have any sense to connect the oscillators to the filter's mixer only to have the Audio In 1 & 2 knobs down! So now apply all the information that we've been looking in the former pages.

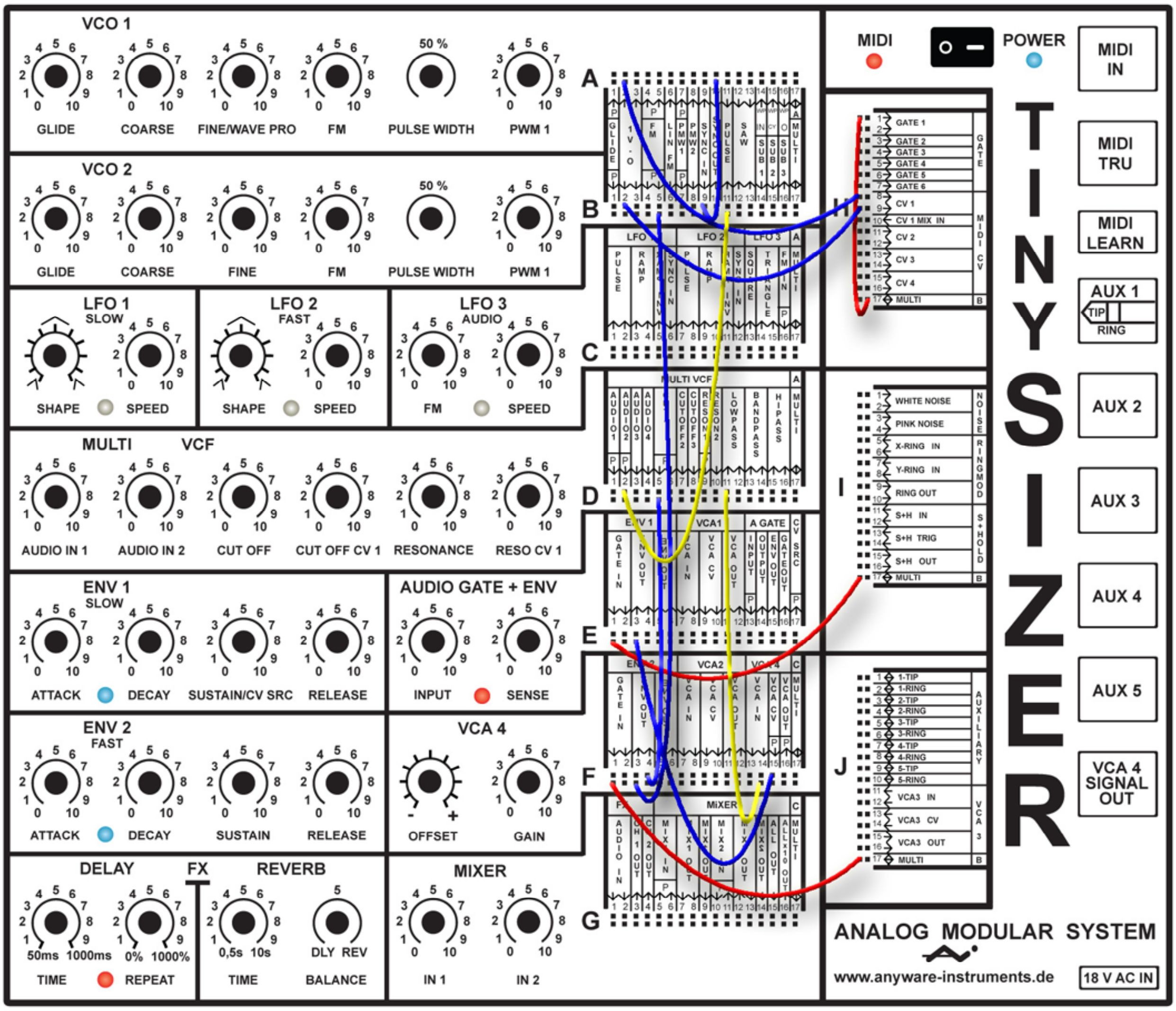
## USING THE SUB OSCILLATOR

We'll add a Sub-Oscillator to the patch, by using one channel of the MIXER to control the level of it.



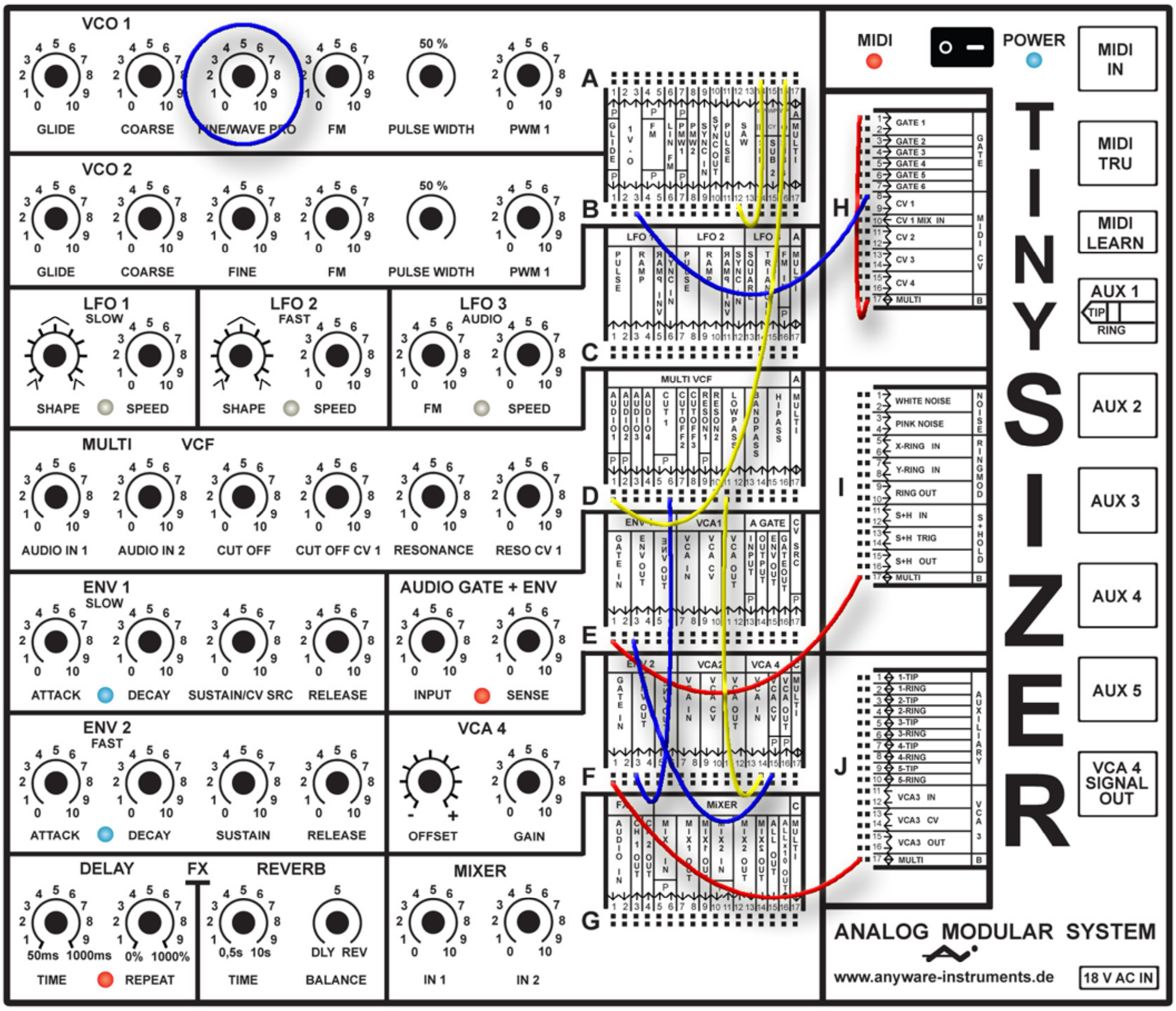
## HARD SYNC

Let's make hard sync sounds! In this example we'll control the slave's (OSC2) pitch with ENV2, for those famous hard sync sweeps. The FM knob on Osc 2 will control the depth of the sweep.



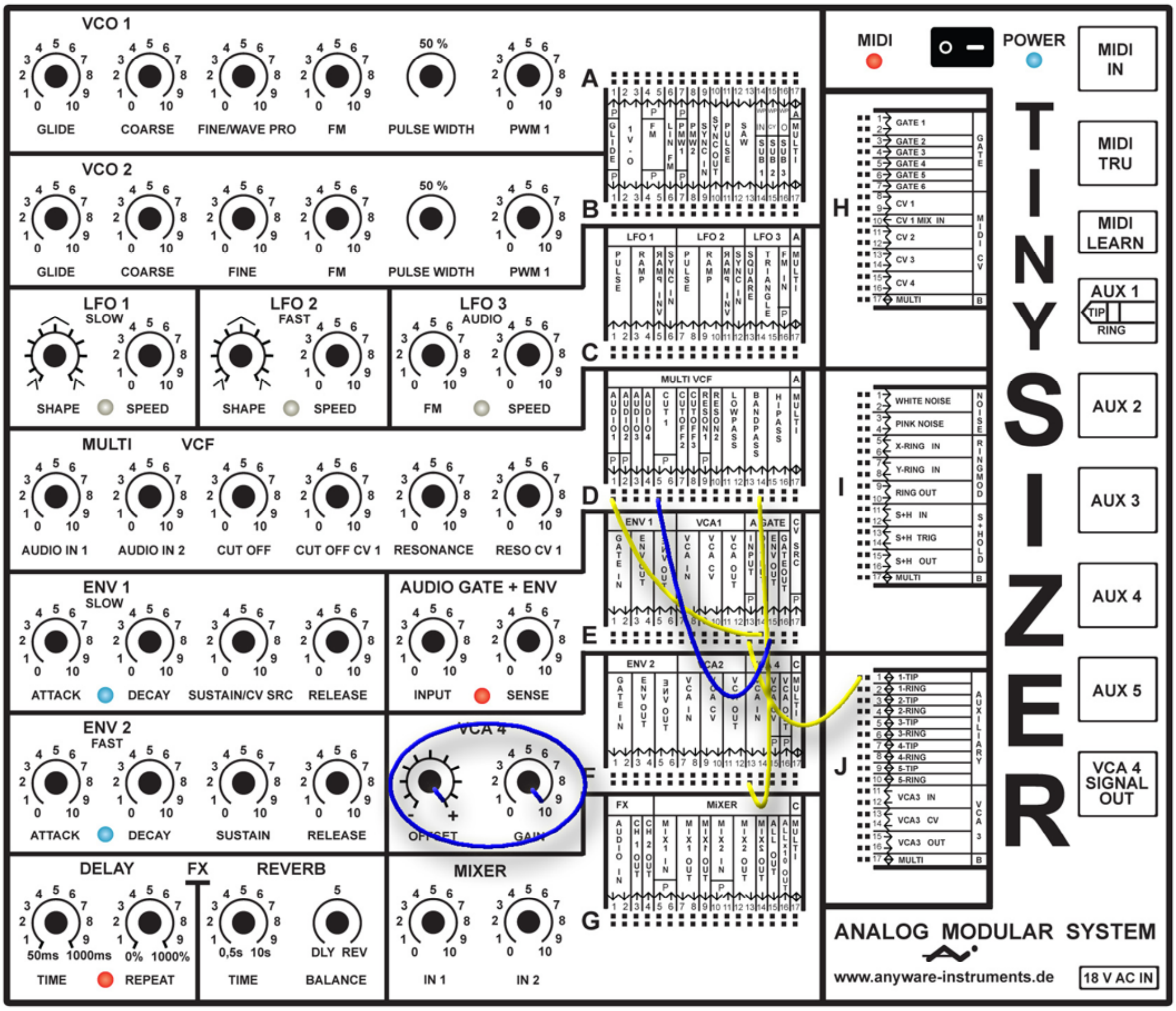
## WAVESHAPER

For this example, we'll use oscillator's 2 sawtooth wave as the waveshaper source. Now the FINE/WAVE PRO knob on oscillator 1 will control the waveshaping depth. You'll be able to sweep from saw to triangle to ramp by turning that knob. Try it!

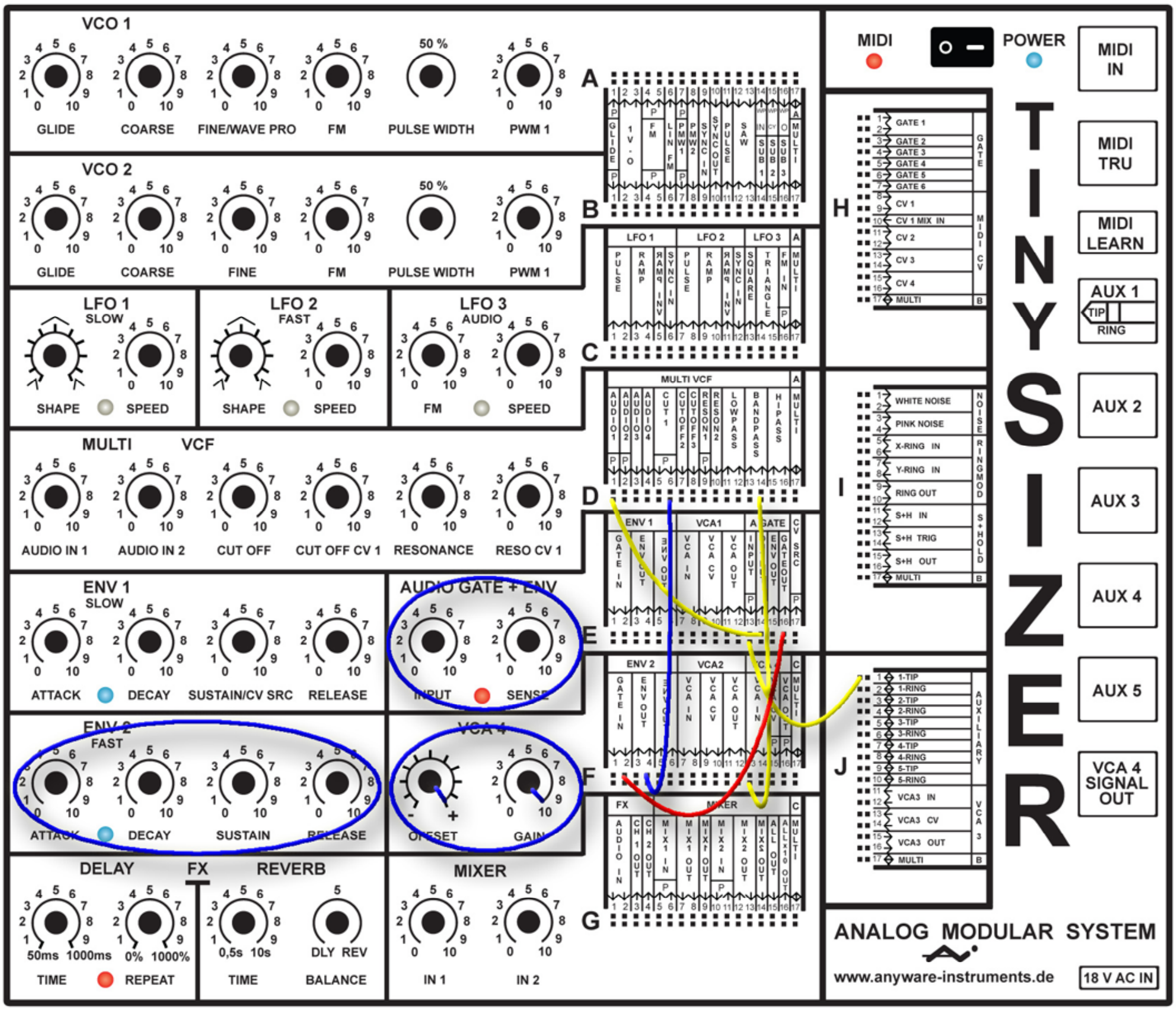


## AUTO WAH

Now let's do some external processing!  Auto wah effects are known because the filter can be controlled by the intensity of the incoming signal, which means that if you're playing a guitar the stronger you play, the brighter the sound. But other instruments may be connected, like drum machines, samplers, etc.
The sensitivity will depend on the input level, so you should adjust it with the pre amp.




Another variation on the theme is to trigger the envelopes with the gate generated by the audio. In this case the SENSE knob will determine the sensitivity of the gate generator. Also adjust the Env2 settings.


VCO 1
GLIDE
COARSE
FINE/WAVE PRO
FM
PULSE WIDTH
PWM 1
VCO 2
FINE
LFO 1
SLOW
SHAPE
SPEED
LFO 2
FAST
LFO 3
AUDIO
MULTI
VCF
AUDIO IN 1
AUDIO IN 2
CUT OFF
CUT OFF CV 1
RESONANCE
RESO CV 1
ENV 1
ATTACK
DECAY
SUSTAIN/CV SRC
RELEASE
AUDIO GATE + ENV
INPUT
SENSE
ENV 2
SUSTAIN
VCA 4
OFFSET
GAIN
DELAY
FX
REVERB
MIXER
TIME
REPEAT
BALANCE
IN 1
IN 2
MIDI
POWER
MIDI IN
MIDI TRU
MIDI LEARN
AUX 1
AUX 2
AUX 3
AUX 4
AUX 5
VCA 4 SIGNAL OUT
TINYSIZER
ANALOG MODULAR SYSTEM
www.anyware-instruments.de
18 V AC IN

## Midi Implementation Chart

**Model: Tinysizer**

| **Fuction** | | **Transmitted** | **Recognized** | **Remarks** |
|---|---|---|---|---|
| Basic Channel | Default Channel | X | 1-16 | Memorized |
| Mode | Default Messages Altered | X | 0 | Mode 3 & 4 |
| Note Number | True Voice | X | 0-127 | |
| Velocity | Note On<br>Note Off | X<br>X | 0<br>X | |
| Aftertouch | Key<br>Ch | X<br>X | X<br>0 | |
| Pitch Bender | | X | 0 | |
| Control Change | | X | 0-120 | Controller assignments are programmable |
| Program Change | | X | X | |
| System Exclusive | | X | 0 | For MIDI config |
| System Common | | X | X | |
| System Real Time | Clock Commands | X | 0 | |
| Aux Messages | | X | X | |
| Notes | | | | No MIDI out |

Mode 1:Omni On, Poly
Mode 2: Omni On, Mono
Mode 3: Omni Off, Poly
Mode 4: Omni Off, Mono

O: Yes
X: No

## Credits

Tinysizer Design and Construction: Thomas Welsch
Manual: Javier Zubizarreta
Graphic Design: Pablo Mieres
Manual Advisors: Mark Towns, Stephen Pearlman
Special thanks to: Gareth Jones, Daniel Anselmi, Ernesto Romeo, Virginia Perez Dilsizian, Iris and Wolfgang Meyer, Ignacio Zubizarreta.