Mono/Fury

Software Synthesizer Plug-In

Version 2.2

http://www.fullbucket.de/music

Table of Contents

Introduction

The *Mono/Fury* is a software synthesizer plug-in for Microsoft Windows (VST) and Apple macOS (VST/AU) simulating the classic KORG *Mono/Poly* analog synthesizer of the early 1980's. It is written in native C++ code for high performance even on "lighter" systems. The main features are:

- Close emulation of behavior and all controls
- Monophonic/quadrophonic keyboard action
- Four band-limited oscillators
- Four-pole lowpass filter with self-oscillation
- "Effects" section including Cross modulation and hard sync
- Arpeggiator with sync-to-host option
- Additional tweak section
- Plug-in supports Windows and macOS (32 bit and 64 bit)

A Warning to the Potential User

1. The *Mono/Fury* does not sound like it's famous analog archetype.

I own the *Mono/Poly* since 1984 and I am aware that there *is* a difference – however, I think the *aura* is captured pretty well. While testing and recording some songs with the *Mono/Fury* I immediately felt home.

2. The *Mono/Fury* is no extended polyphonic oversized buzz-and-bells has-nothing-to-do-with-the-original recreation.

Why that? Because the *Mono/Fury* is a *simulation* – I wanted to do something *close* to the original, not something more or less. The layout of the *Mono/Poly* offers some unique features but also has some limitations. To me this combination was and is a source of creative inspiration.

3. Hey, but neither "real" Polyphony nor Panorama control!?

Yes. Because the *Mono/Poly* has only one voice and a single mono output. Be grateful, I do spare you the five to ten minutes warm-up time until all four VCOs would be in tune. And I'm pretty sure you will find a pan in your favorite host – or in the kitchen.

4. Damn, why should I ever download or even use this crap?

Damn, you're right. Better go and download the real thing.

Architecture

The *Mono/Fury* features four band-limited oscillators (*VCO*[1]), each with four selectable waveforms (triangle, sawtooth, pulse with modulated and fixed width), four octave ranges (16' to 2'), individual tuning and volume controls. The VCOs and an optional white noise generator feed a classic four-pole low-pass filter (*VCF*) with resonance up to self-oscillation. Two envelope generators (*EG*) control the contour of the filter cutoff and the amplifier (*VCA*) respectively.

In the *Effects* section, the VCOs can be Cross-modulated and/or hard-synchronized as well as frequency-modulated by the VCF EG or the first modulation generator (*MG1*). MG1 features four waveforms (triangle, falling and rising sawtooth, square) and is also capable of modulating the overall pitch, the pulse width of the VCOs, and the VCF cutoff frequency.

Modulation generator 2 (*MG2*) produces a triangle waveform used to modulate the VCOs' pulse width. Additionally, it is used to control the speed of the built-in arpeggiator. The latter offers three modes (up, down, and up & down), three key ranges (1, 2 or 4 octaves), and can be latched.

Key Assignment Modes

Although the *Mono/Fury* is a monophonic instrument, it offers two "polyphonic" keyboard modes where the VCOs can be played individually (resulting in a four-voice "pseudo-polyphony"; mind that there still is only *one* VCF and only *one* VCA!). Furthermore, the *Chord Memory* mode allows to play a chord of up to four notes (one per VCO) by a single key. This function compensates the lack of a dedicated control for interval-tuning of the VCOs.

1 VCO = **V**oltage **C**ontrolled **O**scillator. Of course the modules of the *Mono/Fury* are not voltage-controlled in the same way as in analog synthesizers, but I will stick to this neat old-fashioned acronyms.

The following key assignment modes are available:

Unison	Monophonic keyboard action: The VCOs are stacked and play the same key.
Unison/Share	Polyphonic keyboard action: The pressed keys *share* the VCOs, i.e. if one key is pressed all VCOs will play the same note; if two keys are pressed each note will be played by two VCOs. If more than two keys are pressed each note will be played by one VCO.
Poly	Polyphonic keyboard action: Each key is played by one VCO. If the arpeggiator is activated each note will be played by one VCO while the next note will be played by the next VCO in row.
Chord Memory	Monophonic keyboard action: The VCOs are stacked and play the relative notes defined by the Chord Memory function.
Round Robin	*Tweak mode – see section* Tweaks*!* Monophonic keyboard action: The pressed key is played by one VCO while the next key is played by the next VCO in row (similar to *Poly* when the arpeggiator is activated).

Trigger Mode and Auto Damp

Since there is only one filter and one amplifier the parameters *Trigger* and *Auto Damp* are of special interest when playing one of the two polyphonic modes (*Unison/Share* or *Poly*).

On key press, *Single Trigger* will restart the envelopes only when no other key is pressed (legato) while *Multiple Trigger* will restart them always.

If *Auto Damp* is set to *Off* the VCOs will play until all keys have been released (even if their assigned keys are already released). If set to *On* a VCO will be muted as soon as its assigned key is released.

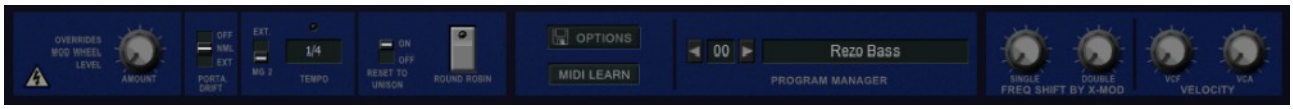
Programming the Chord Memory

Of course this has to be done as with the *Mono/Poly*! Play a chord on the keyboard and click the *Chord Memory* button at the same time. Or activate the *Hold* button, play the desired keys one for one, and then click the *Chord Memory* button. Or set the relative chord notes directly via the hosts' built-in plug-in parameter dialog (if available).

Tweaks

At some places I decided to insert some minor additions or *tweaks* to the *Mono/Fury* which are not part of the original design. The purist will cry out loud but I think these tweaks are quite useful. Some tweaks are obvious: You can save a plug-in program, you have an integrated MIDI interface, it does not take minutes until the VCOs have warmed-up, the plug-in weights just some hundered kilobytes and not kilograms... ;-)

Here, I'm talking about some new *controls*. These tweak controls will be shown when the "Danger / High Voltage" symbol at bottom left of the panel is clicked.



Mod Wheel Override

The amount of modulation by MG 1 is controlled by the modulation wheel. However, the position of the mod wheel cannot be stored with the plug-in's programs. This tweak parameter fixes this issue by emulating an initial position of the mod wheel.

Portamento Drift

The Portamento of the original *Mono/Poly* was set by a four-gang potentiometer – for higher settings this resulted in slightly different Portamento times of the four VCOs. The *Portamento Drift* tweak parameter controls the amount of this effect (*Off*, *Normal*, *Extreme*).

Arpeggiator Trigger Source

The arpeggiator is usually triggered by MG 2 while the *Mono/Poly* provided a clock input jack for external triggering. This tweak parameter can be used to synchronize the arpeggiator to the host tempo.

Arpeggiator Tempo

If the arpeggiator is synchronized to the host, this tweak parameter is used to set the tempo in note lengths.

Key Assign Mode Reset By Effects Button

When you activate the Effects section, the key assignment mode will be reset to *Unison*. On the *Mono/Poly* it was possible to avoid this behavior by setting a specific jumper on one of the circuit boards. Here, you just set this tweak parameter to *Off*.

Round Robin

This is an additional monophonic key assignment mode not to be found on the *Mono/Poly* (see section *Key Assignment Modes*). *Round Robin* schedules only one VCO to a key; however, when the next key is pressed it will be scheduled to the *next* VCO in row (similar to the *Poly* mode when the arpeggiator is activated).

Frequency Shift of Slave VCOs by X-Mod

Actually this consists of two tweak parameters for the *Single* and *Double* Effects modes respectively and needs a bit more explanation.
The Effects section provides Cross modulation which is equivalent to amplitude modulation of one or more slave VCOs by a master VCO; the modulation intensity is controlled by the *X-Mod* parameter (for maximum *X-Mod* the Cross becomes Ring modulation).
However, when I compared the sound of the *Mono/Fury* to the original, it turned out that on the *Mono/Poly* the *X-Mod* parameter does not only control the Cross mod intensity but also causes a *pitch shift* of the slave VCOs, i.e. the higher *X-Mod* the higher the pitch[2], and the amount is different for *Single* and *Double* mode.
This may not happen on every model of the *Mono/Poly*, thus I decided not to hard-code the pitch shift amount into the Cross mod algorithm but to add these two little tweak parameters.

Velocity to VCF/VCA

The original *Mono/Poly* does not react to Velocity – the *Mono/Fury* v2.0 can!

Patches

The *Mono/Poly* does not have any preset or patch memory to store and recall sounds by a few clicks (thanks to advances in quantum physics, the *Mono/Fury* does have this capability). Instead, the manufactures often provided *patch sheets* for their products – small diagrams on real paper with parameter settings of "factory presets".

The *Mono/Poly* was delivered with a booklet of such *Setting Charts*, and I have integrated those 24 presets into the program memory of the *Mono/Fury*. They are identified by the prefix "Chart:" in the patch name followed by the original English preset name as given in the booklet (except on the patch "Elektronengewehr" where I used the German preset name).

In some cases I had to tweak the patches a little bit since the original charts missed one or more parameter values (for example "Synthetic Guitar" or "Elektronen-gewehr"); I hope I have captured the original sound designer's intention here.

2 A quite similar effect will occur if one pushes the panel of the *Mono/Poly* close to the *X-Mod* control. This is obviously no feature but related to the analog circuit board mounted directly under the panel. Anyway, I decided not to emulate this behavior.

Program and File Menu

When clicking on the *File* button (disk symbol in the Program section), a context menu opens with the following options:

Copy Program	Copy current program to internal clipboard
Paste Program	Paste internal clipboard to current program
Load Program	Load a VST program file containing a patch to the *Mono/Fury's* current program
Save Program	Save the *Mono/Fury's* current program to a VST program file
Load Bank	Load a VST bank file containing 64 *Mono/Fury* patches
Save Bank	Save the *Mono/Fury's* 64 patches to a VST bank file
Init Program	Initialize the current program
MIDI Thru	Set globally if MIDI data sent to the *Mono/Fury* should be sent through to its MIDI output (stored in configuration file)
Ignore Program Change	Set globally if MIDI Program Change data sent to the *Mono/Fury* should be ignored (stored in configuration file)
Reload Configuration	Reload the *Mono/Fury's* configuration file (see section *The monofury.ini Configuration File*)
Save Configuration	Save the *Mono/Fury's* configuration file (see section *The monofury.ini Configuration File*)
Select Startup Bank	Select the VST bank file that should always be loaded when the *Mono/Fury* is started
Load Startup Bank	Load the Startup VST bank file; can also be used to check what the current Startup bank is
Unselect Startup Bank	Unselect the current Startup bank
Check Online for Update	When connected to the Internet, this function will check if a newer version of the *Mono/Fury* is available at fullbucket.de
Visit fullbucket.de	Open fullbucket.de in your standard browser

The *monofury.ini* Configuration File

The *Mono/Fury* is able to read some settings from a configuration file (`monofury.ini`) located in the same directory as the *Mono/Fury* VST DLL (`monofury.dll` or `monofury64.dll`) itself. After you have edited this INI file in a text editor, you have to reload it using the *Reload Configuration* command from the *File* menu (see section *Program and File Menu*).

MIDI Control Change Messages

All parameters of the *Mono/Fury* can be controlled by MIDI controllers, or more precise: Each MIDI controller (except *Modulation Wheel* and *Sustain Pedal*) can control one of *Mono/Fury's* parameters. The mapping is defined in the `monofury.ini` for example like this:

```
[MIDI Control]
# trying to follow GM2 definitions here ;-)
CC7  = 8  # Volume
CC70 = 38 # Effects X-Mod
CC71 = 44 # VCF Resonance
CC72 = 57 # VCA Release
CC73 = 54 # VCA Attack
CC74 = 43 # VCF Cutoff
CC75 = 55 # VCA Decay
CC76 = 48 # MG 1 Rate
CC77 = 39 # Effects FM
CC84 = 13 # Portamento
CC94 = 12 # Detune
...
```

The syntax is straight forward:

```
CC<controller number> = <parameter ID>
```

Given the above example, controller 7 directly controls the overall *Volume* parameter, controller 74 the *VCF Cutoff* etc. As you can see, comments are introduced by the Pound sign (#); they are here just for description purposes and completely optional.

The *parameter ID* of one of the *Mono/Fury's* parameters is given in the section *Parameters* below. Note that the *controller number* can run from 0 to 110, with the exception of 1 (*Modulation Wheel*) and 64 (*Sustain Pedal*); the latter two are simply ignored.

MIDI Learn

The easiest way to assign MIDI controllers to *Mono/Fury* parameters is to use the MIDI Learn function. To activate *MIDI Learn*, click on the respective button and wiggle both the MIDI controller and the *Mono/Fury*'s parameter that you want to link. If you want to unlearn the assignment, right-click the *MIDI Learn* button (the label now reads "UNLEARN") and activate it. Now wiggle the MIDI controller or the parameter that you want to unlearn.

Parameters

Global

parameter	ID	description
Volume	8	the overall volume
Range	9	*Off* (no output), *Low* and *High*
Master Tune	11	overall tuning of the VCOs
Detune	12	detunes VCO2 in negative and VCO4 in positive direction
Portamento	13	portamento time (0 to 5 seconds)

Keyboard Assignment Modes

parameter	ID	description
Key Assign Mode	0	five modes are available: *Unison*, *Unison/Share*, *Poly*, *Chord Memory*, and *Round Robin* (tweak mode)
Hold	3	if set to *On* the pressed keys will be held until *Hold* is set to *Off*
Trigger	1	two trigger modes are available: *Single* and *Multiple*
Auto Damp	2	switches *Auto Damp* mode on or off
Chord Memory 1	4	relative pitch of the respective notes of the Chord Memory; a CC value of 127 switches the note off
Chord Memory 2	5	
Chord Memory 3	6	
Chord Memory 4	7	

Wheels Section

parameter	ID	description
Bend Intensity	15	intensity of pitch bend modulation (source is the pitch bend wheel)
Bend Destination	14	destination of pitch bend modulation: *VCO 1* or *Slave VCOs* (depending on effects *On/Off* state), *Pitch*, or *VCF*
MG1 Intensity	17	intensity of modulation by MG1 (controlled by the mod wheel)
MG1 Destination	16	destination of modulation by MG1: *VCO 1* or *Slave VCOs* (depending on effects *On/Off* state), *Pitch*, or *VCF*

VCO 1

parameter	ID	description
Waveform	18	four types are available: triangle, sawtooth, pulse with modulated and fixed width
Octave	19	footage (16', 8', 4', or 2')
Volume	20	output volume

VCO 2

parameter	ID	description
Tune	21	fine tuning
Waveform	22	four types are available: triangle, sawtooth, pulse with modulated and fixed width
Octave	23	footage (16', 8', 4', or 2')
Volume	24	output volume

VCO 3

parameter	ID	description
Tune	25	fine tuning
Waveform	26	four types are available: triangle, sawtooth, pulse with modulated and fixed width
Octave	27	footage (16', 8', 4', or 2')
Volume	28	output volume

VCO 4

parameter	ID	description
Tune	29	fine tuning
Waveform	30	four types are available: triangle, sawtooth, pulse with modulated and fixed width
Octave	31	footage (16', 8', 4', or 2')
Volume	32	output volume

All VCOs

parameter	ID	description
PW	35	controls the fixed pulse width for all VCOs
PWM	33	amount of pulse width modulation for all VCOs
PWM Source	24	source of the pulse width modulation (*VCF EG*, *MG1*, or *MG2*)

Noise

parameter	ID	description
Volume	36	output volume of the white noise generator

VCF

parameter	ID	description
Cutoff	43	cutoff frequency of the filter
Resonance	44	feedback amount of the filter; high values will cause self-oscillation
EG Intens	45	amount of cutoff frequency modulation by VCF EG (positive or negative)
KBD Track	46	amount of keyboard tracking

VCF EG

parameter	ID	description
Attack	50	attack time of the envelope
Decay	51	decay time of the envelope
Sustain	52	sustain level of the envelope
Release	53	release time of the envelope

VCA EG

parameter	ID	description
Attack	54	attack time of the envelope
Decay	55	decay time of the envelope
Sustain	56	sustain level of the envelope
Release	57	release time of the envelope

MG 1

parameter	ID	description
Waveform	47	four types are available: triangle, falling and rising sawtooth, and square
Rate	48	rate of the MG1

MG 2

parameter	ID	description
Rate	49	rate of the MG2

Effects Section

parameter	ID	description
On/Off	37	switches the effects on and off
X-Mod	38	amount of Cross modulation of slave VCOs (if *Mode* is set to *S&X* or *XMod*)
Freq Mod	39	frequency modulation of slave VCOs
Freq Mod Source	40	source of frequency modulation (*VCF EG* or *MG1*)
Mode 1	41	three types affecting the slave VCOs are available: *Sync*: hard sync *S&X*: hard sync and Cross modulation *XMod*: Cross modulation
Mode 2	42	selection of slave VCOs: *Single*: VCO 1 modulates slave VCOs 2 – 4 *Double*: VCO 1 and 3 modulate slave VCO 2 and 4 resp.

Arpeggiator

parameter	ID	description
Range	58	range of the arpeggiator: *1 Octave*, *2 Octaves*, or *Full*
Mode	59	arpeggiator mode: *Up*, *Down*, or *Up/Down*
State	60	arpeggiator state: *Off*, *On*, or *Latch*

Tweaks

For a detailed description of the tweak parameters see section *Tweaks*.

parameter	ID	description
Mod Wheel Override	61	overrides the mod wheel to provide a MG1 modulation intensity parameter value of the plug-in
Portamento Drift	69	sets the amount of "drift" (*Off*, *Normal*, *Extreme*)
Arpeggiator Trigger Source	65	sets the arpeggiator clock source to *MG2* (internal) or *External* (VST host)
Arpeggiator Tempo	66	if *Arpeggiator Trigger Source* is set to *External this parameter* selects the tempo of the arpeggiator in note lengths
Key Assign Mode Reset By Effects Button	62	if switched to *Off* the *Key Assign Mode* will not be automatically reset to *Unison* when the *Effects* section is activated
Round Robin	–	additional key assignment mode with cyclic VCO scheduling (see *Key Assign Mode*)
Freq Shift of Slave VCOs by X-Mod (Single Mode)	63	controls the frequency shift of the slave VCOs by the *X-Mod* parameter for *Single* mode
Freq Shift of Slave VCOs by X-Mod (Double Mode)	64	controls the frequency shift of the slave VCOs by the *X-Mod* parameter for *Double* mode
Velocity to VCF	67	amount of Velocity applied to VCF cutoff
Velocity to VCA	68	amount of Velocity applied to VCA volume

Frequently Asked Questions

Before you read on here, please make sure that you also have read my statements regarding the *Mono/Fury*'s capabilities and limitations at the beginning of this manual (see section *A Warning to the Potential User*).

How do I install the Mono/Fury *(Windows 32 bit version)?*

Just copy the files `monofury.dll` and `monofury.ini` from the ZIP archive you have downloaded to your system's or favorite DAW's VST plug-in folder. Your DAW should automatically register the *Mono/Fury* VST plug-in the next time you start it.

How do I install the Mono/Fury *(Windows 64 bit version)?*

Just copy the file `monofury64.dll` and `monofury.ini` from the ZIP archive you have downloaded to your system's or favorite DAW's VST plug-in folder. Your DAW should automatically register the *Mono/Fury* VST plug-in the next time you start it.

Note: You may have to remove any existing (32 bit) `monofury.dll` from your VST plug-in folder or else your DAW may screw the versions up...

How do I install the Mono/Fury *(Mac VST/AU universal 32/64 bit)?*

Locate the downloaded PKG package file `monofury_2_1_3_mac.pkg` in Finder (!) and do a right- or control-click on it. In the context menu, click on “Open”. You will be asked if you really want to install the package because it comes from an “unidentified developer” (me ☺). Click “OK” and follow the installation instructions.

What is the VST plug-in ID of the Mono/Fury*?*

The ID is 1416.

Will you support the Mono/Fury*?*

Yes. If you have problems, found a bug, or have some suggestions about the *Mono/Fury* please send me a mail: `full.bucket@gmx.net` .

How can I prevent output clipping (for example when I use high Resonance values)?

Turn down the volume or select the *Low* output range (*Output Range* switch on the left of the panel).

How can I tune the VCF frequency to the note played on the keyboard?

Turn the *KBD Track* parameter to 6.4 and play MIDI key F#1 to adjust the *Cutoff* parameter so that the VCF is in tune with the oscillators. Or use the patches 29 ("Chart: Whistle") or 45 ("Duophonic Madness") as a base class of your sound.

Ups, I cannot find the tweak parameters on the panel...!?

Click the "Danger / High Voltage" symbol at bottom left of the panel and you will.

If the MG rates are pretty high their associated LEDs on the panel will not blink in sync! Why?

This effect is called *aliasing*. Never mind, I never thought they would work at all!

Will you someday provide a "true" polyphonic version of the **Mono/Fury***?*

No. I would add a more flexible modulation infrastructure and velocity sensitivity as well, but that would be a new synth like the *blooo* and not a "bigger" *Mono/Fury*.

How do I know if a new version of the **Mono/Fury** *is available?*

When connected to the Internet, open the Options menu (see section *Program and File Menu*) by clicking the disk icon and select the entry "Check Online for Updates". If a new version of the *Mono/Fury* is available on fullbucket.de the respective information will be shown in a message box.