

# Kern

## Performance Synthesizer

Version 1.1

© 2015-2019 by Björn Arlt @ Full Bucket Music  
<http://www.fullbucket.de/music>



VST is a trademark of Steinberg Media Technologies GmbH

Windows is a registered trademark of Microsoft Corporation

The Audio Units logo is a trademark of Apple Computer, Inc.

## Table of Contents

Introduction.....	3
Why Kern?.....	3
User Interface.....	4
Sound Engine.....	5
Oscillators.....	5
Filter and Amp.....	5
LFO and Envelopes.....	5
Chorus.....	5
Performance Controls.....	6
Program Menu.....	6
MIDI Learn.....	6
Options Menu.....	6
The kern.ini Configuration File.....	8
Skinning Support.....	8
MIDI Control Change Messages.....	8
Parameters.....	9
Oscillators.....	9
Filter.....	9
LFO.....	10
Amplifier.....	10
Chorus.....	10
Frequently Asked Questions.....	11

## Introduction

*Kern* is a software synthesizer plug-in for Microsoft Windows (VST) and Apple macOS (VST/AU) designed to run with and to be fully controlled by modern MIDI keyboard controllers like the Novation® Impulse™. It is written in native C++ code for high performance and extremely low CPU consumption. The main features are:

- Streamlined to be used with MIDI keyboard controllers; all parameters can be controlled by MIDI CC
- MIDI Learn
- Two alternative user panels
- 32 voices polyphony
- Two band-limited oscillators including Hard Sync
- 4-pole zero-delay feedback lowpass filter (two types)
- Two envelopes, one LFO
- Chorus effect
- Double precision audio processing
- Plug-in supports Windows and macOS (32 bit and 64 bit)

## Why Kern?

Ask yourself:

- Do you have a MIDI controller with all those shiny sliders, knobs, and buttons?
- Do you feel the urge to use it to twiddle the parameters of your favorite (software) synth?
- Do you get frustrated because moving a knob here changes a knob there, but the mapping seems not to be intuitive?
- Or maybe the parameter you want to access isn't even mapped?
- And, to even increase frustration, do you remember the good old days when synthesizers had exactly one dedicated slider/knob/button for each parameter?

If your answer is always "No" then ask yourself:

- Do you want a light-weight, easy-to-use, CPU-friendly, cool sounding synth?

If it is "No" again then *Kern* may be not the right thing for you.

... but now you know why I created *Kern*. Together with my V-Machine (which is grateful for CPU-friendly plug-ins!) I have a fully controllable stand-alone synthesizer that does not need a PC.

Of course there are drawbacks: Since today's MIDI master keyboards typically do not have more than 30 hardware controls I had to limit the number of *Kern*'s parameters to (what I believe – you may have a different opinion here, that's OK –) the minimum of what is absolutely required. That is why *Kern* is named "Kern" which is German for "core".

## User Interface

Two alternative user panels (“views”) are available: The standard (“traditional”) view is in line with the architecture of subtractive synthesizers while the second view reflects the typical layout of the sliders, knobs, and buttons of today's MIDI hardware controllers. If you own a Novation Impulse (like I do) or a similar machine you will find the latter view very helpful since it visually maps the hardware controls to *Kern's* parameters.

You can switch between the views via the *Options* menu or via the *Switch View* button (only available on the standard view).



Kern's standard view



Kern's alternative view

## Sound Engine

### Oscillators

*Kern* has two band-limited oscillators that can create *Sawtooth* or *Square* waves; the waveform has to be selected for both oscillators together. Oscillator 2 can be transposed by  $\pm 24$  notes and detuned by  $\pm 1$  note. Furthermore, it is possible to hard-synchronize Oscillator 2 to Oscillator 1.

The frequency of the oscillators can be modulated either by the LFO or the filter envelope (positively or negatively). If Hard Sync is activated, only Oscillator 2 will be modulated to produce the classic rich harmonic "Sync" spectra we all love. Apart from that, frequency modulation of both oscillators by the LFO ("Vibrato") can always be applied via the modulation wheel. Portamento is on board, too.

Finally, it is possible to switch *Kern* into monophonic mode (e.g. for lead and/or bass sounds). By default the envelopes are *single triggered* meaning that they are not restarted when playing *legato* (also known as "Minimoog mode"). However you can change the trigger mode to *multiple* using the context menu that opens when you click on the *Mono* switch.

### Filter and Amp

The filter is based on a (attention: buzz words!) *Zero-Delay Feedback* design and provides two modes: *Smooth*, a 4-pole lowpass with moderate non-linearities and potential self-oscillation, and *Dirty*, a punchy 2-pole lowpass with potential but no self-oscillation. *Cutoff* and *Resonance* of course are editable.

The cutoff frequency of the filter can be modulated simultaneously and both positively or negatively by four sources: filter envelope, LFO, key track, and velocity.

The amplifier just offers *Volume* and *Velocity* parameters; the latter controls the influence of the velocity to the output volume.

### LFO and Envelopes

The LFO offers three waveforms: *Triangle*, *Square*, and *S/H* (random); its' speed rate ranges from 0 to 100 Hz.

The filter envelope is a simplified ADS generator: The *Decay* parameter controls both Decay and Release rates together while *Sustain* can only be switched on or off. The amplifier envelope is similar with the exception that here *Release* can be controlled independently from the Decay rate.

### Chorus

The Chorus can be switched on or off. Furthermore it is possible to set the speed rates of the two triangle-shaped LFOs modulating the Chorus as well as the modulation depth.

## Performance Controls

### Program Menu

If you know my other plug-ins then there will be no surprises: To select one of the 64 patches just click on the program number, and edit its' name by clicking in the text field.

### MIDI Learn

Every parameter of *Kern* can be controlled by one MIDI controller. If you want to change the assignment of MIDI controller (CC; *MIDI Control Change*) to *Kern* parameter the *MIDI Learn* function comes in quite handy: Just click the *MIDI Learn* button on *Kern's* control panel (caption turns red) and wiggle both the MIDI controller and the parameter you want to assign (you can abort *MIDI Learn* by clicking the red button). To save the controller assignments use "Save Configuration" in the *Options* menu (see next section).

### Options Menu

When clicking on the *Options* button, a context menu opens with these options:

<b>Copy Program</b>	Copy current program to internal clipboard
<b>Paste Program</b>	Paste internal clipboard to current program
<b>Load Program</b>	Load a program file containing a patch to the <i>Kern's</i> current program
<b>Save Program</b>	Save the <i>Kern's</i> current program to a program file
<b>Load Bank</b>	Load a bank file containing 64 patches into the <i>Kern</i>
<b>Save Bank</b>	Save the <i>Kern's</i> 64 patches to a bank file
<b>Init Program</b>	Initialize the current program
<b>Reload Configuration</b>	Reload the <i>Kern's</i> configuration file (see section <i>The kern.ini Configuration File</i> )
<b>Save Configuration</b>	Saves the <i>Kern's</i> configuration file (see section <i>The kern.ini Configuration File</i> )
<b>Select Startup Bank</b>	Select the bank file that should always be loaded when the <i>Kern</i> is started
<b>Load Startup Bank</b>	Load the Startup bank file; can also be used to check what the current Startup bank is
<b>Unselect Startup Bank</b>	Unselect the current Startup bank
<b>Check Online for Update</b>	When connected to the Internet, this function will check if a newer version of the <i>Kern</i> is available at fullbucket.de
<b>Switch View</b>	Switches between the views (see section <i>User Interface</i> )
<b>Visit fullbucket.de</b>	Open fullbucket.de in your standard browser

**Note:** If the *Kern* DLL and the `kern.ini` configuration file are located in a “protected” Windows folder (for example somewhere under “C:\Program Files\...”) then it may not be possible to save the configuration. In this case it is strongly recommended to use a `VstPlugins` directory that lies “outside” the protected folder.

## The kern.ini Configuration File

*Kern* is able to read some settings from a configuration file (`kern.ini`) located in the same directory as the as the *Kern* VST DLL (`kern.dll` or `kern64.dll`) or Mac VST/AU (`Kern.component` or `Kern.vst`) itself. After you have edited this INI file in a text editor, you have to reload it using the *Reload Configuration* command from the *Options* menu (see section *Performance Controls*).

## Skinning Support

*Kern* features some very rudimentary skinning support. Within the INI file, each of the bitmaps used in the UI can be replaced by specifying the full path to an alternative bitmap file; bitmaps that you don't want to replace can be set to default (see example below), or left out of the INI file completely.

```
[Editor]
Bitmap_Back = C:\myCoolSkins\flashy.png
Bitmap_Back2 = C:\myCoolSkins\nerdy.png
Bitmap_Handle = default
Bitmap_Knob = default
Bitmap_SmallKnob = default
Bitmap_Button = default
Bitmap_MidiLearn = default
```

**Important:** The replacing graphics need to have the same size and control placement to work correctly. Also knob/button/slider graphics need to have the exact same size and number of frames to work correctly. It is not possible to alter the placement of controls, as that is handled by code in the plug-in.

## MIDI Control Change Messages

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

```
[MIDI Control]
CC41 = 12 # Filter Cutoff
CC42 = 13 # Filter Resonance
CC43 = 21 # Filter Env Attack
CC44 = 22 # Filter Env Decay
CC45 = 24 # Amp Env. Attack
CC46 = 25 # Amp Env. Decay
CC47 = 27 # Amp Env. Release
...
```

The syntax is straight forward:

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

Given the above example, controller 41 directly controls the overall *Filter Cutoff* parameter, controller 42 the *Filter Resonance* 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 *Kern*'s parameters is given in the section *Parameters* below. Note that the *controller number* can run from 0 to 119, with the exception of 1 (*Modulation Wheel*) and 64 (*Sustain Pedal*); the latter two are simply ignored.

Of course, instead of editing the controller/parameter assignments in `kern.ini` with a text editor it is much easier to use the *MIDI Learn* function and save the configuration (see sections *MIDI Learn* and *Options Menu* above).

## Parameters

### Oscillators

parameter	ID	description
<i>Mono</i>	1	Switches between polyphonic and monophonic mode ( <i>Single</i> or <i>Multiple Trigger</i> )
<i>Master Tune</i>	4	Master tune (hidden parameter)
<i>Wave</i>	5	Selects the waveform ( <i>Sawtooth</i> or <i>Square</i> )
<i>P.Bend</i>	2	Pitch Bend range (in notes)
<i>Porta</i>	3	Portamento time
<i>FM</i>	6	Frequency modulation depth
<i>FM Src.</i>	7	Frequency modulation source
<i>Trans.</i>	8	Oscillator 2 transpose (in notes)
<i>Tune</i>	9	Oscillator 2 tuning
<i>Sync</i>	10	Oscillator 2 Hard Sync

### Filter

parameter	ID	description
<i>Cutoff</i>	12	Cutoff frequency
<i>Reso.</i>	13	Resonance
<i>Mode</i>	11	Filter mode ( <i>Smooth</i> or <i>Dirty</i> )
<i>Env</i>	14	Cutoff frequency modulation by filter envelope
<i>LFO</i>	15	Cutoff frequency modulation by LFO
<i>Key</i>	16	Cutoff frequency modulation by note number
<i>Velocity</i>	17	Cutoff frequency modulation by velocity
<i>Attack</i>	21	Attack time of filter envelope
<i>Decay</i>	22	Decay/Release time of filter envelope
<i>Sustain</i>	23	Sustain of filter envelope ( <i>Off</i> or <i>On</i> )

## LFO

parameter	ID	description
<i>Rate</i>	19	Rate of the LFO (0 to 100Hz)
<i>Wave</i>	20	Waveform ( <i>Triangle, Square, S/H</i> )

## Amplifier

parameter	ID	description
<i>Attack</i>	24	Attack time of amplifier envelope
<i>Decay</i>	25	Decay time of amplifier envelope
<i>Release</i>	27	Release time of amplifier envelope
<i>Sustain</i>	26	Sustain of filter amplifier ( <i>Off or On</i> )
<i>Volume</i>	0	Master volume
<i>Velocity</i>	18	Velocity amount

## Chorus

parameter	ID	description
<i>Enable</i>	28	Chorus on/off
<i>Rate 1</i>	29	Rate of first Chorus LFO
<i>Rate 2</i>	30	Rate of second Chorus LFO
<i>Depth</i>	31	Depth of Chorus modulation

## Frequently Asked Questions

### ***How do I install the Kern (Windows 32 bit version)?***

Just copy the files `kern.dll` and `kern.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 *Kern* VST plug-in the next time you start it.

### ***How do I install the Kern (Windows 64 bit version)?***

Just copy the file `kern64.dll` and `kern.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 *Kern* VST plug-in the next time you start it.

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

### ***How do I install Kern (Mac VST/AU universal 32/64 bit)?***

Locate the downloaded PKG package file `kern_1_1_4_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 plug-in ID of the Kern?***

The ID is `k e r n .`

### ***I spent a lot of time customizing the MIDI controller/parameter assignments. Can I save these assignments?***

Yes, by using "Save Configuration" in the *Options* menu (see section *Options Menu*).

### ***Why doesn't the configuration file (kern.ini) get saved?***

Maybe you have placed the *Kern* files `kern.dll/kern64.dll` and `kern.ini` into a "protected" folder somewhere under "`C:\Program Files\...`"? In this case you should use a `VstPlugins` directory that lies "outside" the protected folder (also see section *Options Menu*).

### ***How do I know if a new version of the Kern is available?***

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