

FB3

Digital Wave Memory Synthesizer Plug-In Version 1.0

© 2026 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.
AAX is a trademarks of Avid Technology, Inc.

Table of Contents

Introduction	3
Kraftfelder	3
K3 – Kawai Goes Digital and Additive	4
Inner Values	4
The Good, the Bad, and the Plug-in	4
A Word About AI	5
Acknowledgments	5
Overview	6
User Interface	6
Program Memory	7
Program Editing	7
Memory Protect	7
Additional Information	7
Transpose Mode	8
Portamento and Mono Mode	8
Differences Between K3/K3m and FB3 Programs	8
Master Page	9
Sound Programming	10
Oscillators	10
Voltage Controlled Filter (VCF)	10
Voltage Controlled Amplifier (VCA)	11
Low Frequency Oscillator (LFO)	11
Touch Sensitivity	11
Keyboard Control Voltage (KCV)	12
Chorus	12
Tweak Parameters	12
User Waves	13
Creating User Waves the Hard Way	13
Writing and Saving	14
Little Helpers	15
MIDI	16
MIDI Learn	16
Plugin Handling	17
LINK	17
The <i>fb3.ini</i> Configuration File	17
Control FB3 Via MIDI	18
Options Menu	18
Plug-In Parameters	20
Common	20
Oscillator	20
VCF	20
VCA	21
LFO	21
Touch Sensitivity	21
KCV	21
Chorus	21
Tweak	22
Oscillator Waveforms	22
Frequently Asked Questions	23

Introduction

FB3 is a software instrument plug-in (VST2/VST3/AU/CLAP/AAX) for Microsoft Windows and Apple macOS simulating the sound generation part of the classic Kawai *K3/K3m* synthesizer from 1986. It is written in native C++ code for high performance even on “lighter” systems. The main features are:

- Simulation of the original wave generator unit
- Original factory sounds
- Up to 64 voices polyphony
- User interface can be used to edit the original *K3/K3m* units
- SysEx and User Wave import and export
- Supports [MTS-ESP by ODDSound](#) dynamic micro-tuning
- All parameters can be controlled by MIDI controllers
- Plug-in supports Windows and macOS

FB3 is based on the **iPlug2** framework maintained by **Oli Larkin and the iPlug2 team**. Big thanks, guys!!! Without your work it would not have been possible to create a resizable *FB3* user interface.

To resize the plug-in you just grab the yellow triangle at the bottom right of the window and drag it. You can save the current window size using the menu entry “Save Window Size” in the *Options Menu*.

If you have trouble with the standard version of *FB3*, please grab the (sound-wise identical) “N” version of the plug-in which is based on the original **iPlug** framework.

Kraftfelder

In April 2026, I started to wonder why my website and myself got so much more traffic and emails than usual. It did not take long to find out that **Espen Kraft** had released a video show-casing my plug-ins on YouTube! Watching lots of his other great videos¹, I finally got in touch with him to say “Thank You”. And after a brief chat he said that if I’d ever consider doing something like the Kawai *K3* I should let him know – he could deliver sounds and samples and insights from his own *K3m* (the MIDI module version of the *K3* keyboard synthesizer).

At that time I was working on a different (but similar) project, so I considered the idea as a break from routine. In fact, back in the 80s I really was dying to buy myself a *K3* (but somehow never did) – obviously there was a connection here...

I prepared a prototype plug-in that emulates the *K3m*’s computer unit plus analog circuitry and sent it over to Espen to check whether I am right on track. After exchanging samples and thoughts I started the “augmented” plug-in; I wanted the *FB3* to have increased polyphony, a decent parameter editor grid, User Wave import etc. And here we are: *FB3* might not be a replacement for a real *K3/K3m*, and I doubt that Espen will sell his unit now, but it is close enough for both of us.

1 Espen’s video about patchbays convinced me that after more than 40 years of making music it was high time for me to get a patchbay myself!

K3 – Kawai Goes Digital and Additive

The *K3* was Kawai's first "hybrid" synthesizer: A digital sound generation module combined with analog filters and amplifiers. Korg had released similar machines, the *DW-6000* and the *DW-8000*. PPG pursued the same approach with their famous *Wave* series, while featuring a much more sophisticated (and expensive) way of sound generation, the groundbreaking Wave-Table Synthesis. And there are many, many more to follow (Ensoniq *ESQ-1*, Sequential *Prophet VS*, to name a few...).

The *K3* and the later *K3m* module version were sometimes called "the poor man's PPG" because they were able to produce new sounds that somehow "felt digital". The key to this sound is the capability to playback single-cycled waveforms that were either sampled from original instruments or artificially generated with Additive Synthesis. Additive Synthesis (re)combines a waveform by using the weighted sum of sine waves with frequencies that are integer multiples of the base frequency. These sine waves are the *harmonics*; the frequency factor is called the *harmonic number*, and the "weight" in the sum of harmonics is the *intensity* of the harmonic.

Besides pre-calculated waveforms in the ROM, the *K3* allowed for one (!) User Wave that could be created by the user using a rather tedious way of Additive Synthesis. However, at this price segment, that was something that we call nowadays a "unique selling point"!

Inner Values

The heart of the *K3/K3m* is the 8-bit digital wave generator which runs at a fixed sample rate of 41667Hz. It is capable of creating 12 independent oscillator signals from 31 multi-sampled ROM waveforms plus 1 User Wave. The length of a wave is 512 samples and the machinery will do linear interpolation to smooth away (to a certain degree) the biggest artifacts. But of course these artifacts are exactly what we love, right? In fact the wave generator is a very sophisticated circuit that uses almost only standard components. The trade-off is the non-intuitive way the multi-sampled waveforms (seven octaves as for the Korg *DW* series) are encoded. Luckily, Scott Nordlund aka acreil has done a great job of [uncovering this secret](#)!

The analog filters are realized using SSM 2044 chips known from the Korg *Polysix* and *Mono/Poly*, the PPG *Wave 2.2* and *2.3* and many others. Along with a funky Chorus effect, this lays the foundation for a 6-voice synthesizer that still has its fans today.

The Good, the Bad, and the Plug-in

So *FB3* is it. There are just a few difference to the original *K3/K3m*:

- Polyphony can be something from 1 to 64 voices.
- Programs have their own copy of User Waves.
- Internal and cartridge User Waves can both be used in programs.
- User Waves can be imported from WAV files.
- Parameters can be directly accessed via the parameter grid.
- Additional parameters like VCF Envelope Invert etc.

Of course, *FB3* sound nothing like the hardware, and especially the filters are crap. That said I hope that all the non-nitpickers will have fun with my plug-in. ☺

A Word About AI

I did not use AI (Artificial Intelligence) to create *FB3*.

This is not because I am a hater – not at all: In my daily work I do use AI tools frequently, and AI can definitely be of great help. But creating plug-ins is my very personal thing: I love to know what I am doing, and I want to do it all by myself.

The journey is the reward. And my journey is making plug-ins.

Acknowledgments

- **Oli Larkin** and the **iPlug2** team.
- **Espen Kraft** (youtube.com/@EspenKraft) – musician, influencer, legend!
- **kraftraum** (soundcloud.com/kraftraum) – “my” Beta tester!
- **Scott Nordlund** aka **acreil** (acreil.wordpress.com) for deciphering the K3 wave generator.
- **Vincent** from dbwbp.com.
- **My family** for bearing me and my crude hobby.

No, I am not affiliated with Kawai (nor KORG) in what relation ever except that I find myself entangled with their instruments. ☺

Overview

FB3 is a polyphonic synthesizer with up to 64 voices. Each voice has two digital oscillators, one “voltage-controlled” filter (VCF) and two envelope generators. A global low-frequency oscillator (LFO) for modulation purposes and a Chorus effect completes the setup.

FB3’s program memory consists of 100 programs which are organized in 50 *internal* and 50 *cartridge* programs – this reflects the memory structure of the original K3/K3m.

User Interface

The user interface resembles the one known from the K3/K3m hardware. Central part is the main display with three seven-segment LED displays along with 51 buttons to select programs and parameters (and more) plus 10 buttons for mode selection, editing and other functionality.



The original layout has been expanded to include a grid of controls for direct manipulation of sound parameters, as well as a number of buttons for convenience functions to handle the plugin.

Important note #1: Sometimes it is required to “press” two buttons at the same time. This can be done by right-clicking one button (which then remains highlighted) and subsequently clicking the other button.

Important note #2: For negative values the LED display will show a dot at the top left of the respective number.

Program Memory

The *K3/K3m* features 50 internal programs which can be expanded by another 50 programs using a cartridge module. I took the liberty to permanently add a cartridge to the *FB3* – thus there are 100 programs available. To access one of the programs, first make sure that no other than either the INTERNAL or the CARTRIDGE LED in the top right section is lit. Now click on one of the buttons labeled with the numbers from 1 to 50. If you click one of the two buttons INTERNAL or CARTRIDGE before that, the corresponding memory bank will be selected. The LED above these buttons shows which section is currently selected. Furthermore, the leftmost LED display denotes the current program number.



To save or overwrite the current program, you first have to set the PROTECT switch to "OFF" (see below). Activate the blue WRITE button so that the respective LED will get lit. Now click the INTERNAL or CARTRIDGE button (if required) and then the desired target program number. That's it.



Program Editing

To edit the current program, click on the PARAMETER button (top right section). The buttons 1 to 39 will select the desired parameter to edit – note that the buttons also show the respective parameter name. Use the +/YES and -/NO buttons to change the parameter value.



Alternatively you can click & drag the parameter controls in the parameter grid at the bottom of the user interface. *K3/K3m* users sure would be happy to have that grid on their machines!

Memory Protect

If you are familiar with my plug-ins you know that I usually don't add any memory protection to them. However, when I fiddled around with *FB3* I found that this feature might be pretty useful (and the original hardware has it, too).



Memory protection is set using the PROTECT switch, and its state is stored globally in the *fb3.ini* (see section *The fb3.ini Configuration File*). If PROTECT is set to ON, you cannot overwrite any program nor User Wave data of the *FB3*. In this case the display will show the cryptic message



Additional Information

Besides the parameter grid, *FB3* adds some more features which I summarize under the label "Additional Information". If you are a purist you can turn them off using the respective menu entry "Show "Additional Information" in the *Options Menu*.

First of all, *FB3* introduces a small message display area at the top right of the screen. As expected it will show some hints and outcomes of executed functions.

Next, each program can have a program name which is shown right above the PROGRAM LED display. You can edit the name by clicking the field and start typing.

When you edit the wave parameters of oscillator 1 or 2, the name of the selected wave (which cannot be changed) is displayed right above the VALUE LED display. Furthermore, instead of program number and name, the wave itself will be shown.

Transpose Mode

It does exactly what it should: You can transpose the incoming MIDI notes 6 notes up or 5 notes down. The display will show the transpose values as note names (G, G#, A, A', B, C, C#, D, D#, E, F, F#) where "C" means no transpose at all. When transpose is active but Transpose Mode is not selected, the TRANSPOSE LED will blink. Nice.



Portamento and Mono Mode

To active Portamento, just press the respective button. Of course the current program should have a Portamento time greater than 0 or else there would be no effect.



The MONO button puts the *FB3* into a monophonic six-voice unison mode. The 12 oscillators are slightly detuned against each other (the amount is given by the tweak parameter MONO DETUNE).

Both Portamento and Mono mode are saved per preset.

Differences Between *K3/K3m* and *FB3* Programs

An original *K3/K3m* program stores 39 program parameter values. There is just one User Wave for the whole machine² (plus a second one when a cartridge is present), and all 50 programs have to share that single wave (it is not possible to use the cartridge User Wave in an internal program and vice versa).

FB3 overcomes this problem: Every *FB3* program stores its own internal and cartridge User Wave along with the program parameter values. Furthermore, the Master Volume, Stereo switch, Tune, Transpose, and Note Priority values are also saved per program. And of course it is possible to use any User Wave, internal or cartridge, in the same program. Thus you now can set each oscillator independently to any waveform you want.

2 Remember that memory was expensive in the 80s!

Master Page

The Master page opens the door to some more functionality. When activated, you can click on one of 12 buttons in the MASTER³ section (buttons 40 to 50, and LINK) to select the required function.



- **40: LEARN**
Starts MIDI Learn. See section *MIDI Learn*.
- **41: UNLEARN**
Starts MIDI Unlearn. See section *MIDI Learn*.
- **42: FUNCTION [1–5]**
This value defines what MIDI data is accepted by FB3 (see section *MIDI*).
- **43: PRIORITY**
Selects the note priority in MONO mode: Last or highest note.
- **44: EXCLUSIVE [0–3]**
This value defines what MIDI SysEx data will be sent when clicking one of the 50 program buttons or the INTERNAL/CARTRIDGE buttons (see section *MIDI*).
- **45: LFO SYNC**
Activates synchronization of LFO speed to the host tempo (see section *Low Frequency Oscillator (LFO)*).
- **46: HARMON/INTEN**
Enters the User Wave editing mode. See section *User Waves*.
- **47: COPY**
User Wave editing mode. See section *User Waves*.
- **48: ERASE**
User Wave editing mode. See section *User Waves*.
- **49: SAVE**
Copies all 50 internal programs to the cartridge program area.
- **50: LOAD**
Copies all 50 cartridge programs to the internal program area.
- **TUNE**
Here you can change the tuning about ± 50 cent.

³ The K3/K3m offers some more functions like *MIDI channel* and *OMNI Mode* which I omitted since they are not suitable for the plug-in version.

Sound Programming

Each program has a parameter that can be edited immediately in PROGRAM mode by clicking the +/YES and -/NO buttons. To start editing the other parameters, one has to click the PARAMETER button and select the desired parameters via the number buttons. However, this is a tedious way of editing, thus I added the parameter grid which allows for accessing and viewing parameter values directly and without further ado.



Oscillators

There are two oscillators, OSC 1 and OSC 2. Each oscillator can playback one of the 31 ROM waveforms, the internal or cartridge User Wave, or white noise. It is also possible to switch an oscillator off. Parameters 1 (OSC 1 WAVE) and 7 (OSC 2 WAVE) will select the respective waveforms – for a complete list see *Oscillator Waveforms*.

RANGE sets the range of both oscillators (16', 8', and 4') while OSC 2 COARSE displaces the second oscillator about ± 24 notes; OSC 2 FINE detunes it about ± 30 cents. The value of PORTA SPEED defines the Portamento speed.

However, to enable the Portamento effect one has to activate the PORTAMENTO button (see *Portamento and Mono Mode*), too.

PITCH BEND sets the range of the MIDI Pitch Bend controller. AUTO BEND does the same for the glide added (or subtracted) to the beginning of a note, but the auto bend time is controlled by the DELAY parameter of the LFO (see *Low Frequency Oscillator (LFO)*) which is a bit odd. The mix of both oscillators is controlled by parameter 4 (BALANCE). Negative values raise the output of OSC 1 with respect to OSC 2 while positive values decrease it.

OSC		
1 OSC 1 WAVE	4 BALANCE	7 OSC 2 WAVE
2 RANGE	5 PITCH BEND	8 OSC 2 COARSE
3 PORTA SPEED	6 AUTO BEND	9 OSC 2 FINE

Voltage Controlled Filter (VCF)

The filter has the standard parameters CUTOFF and RESONANCE as well as its own ADSR envelope with the parameters ATTACK, DECAY, SUSTAIN and RELEASE. ENV (parameter 13) controls the amount of modulation by the envelope. Here I took the liberty to add parameter 16 (ENV INV – not present on the K3K/3m and thus labeled in dark gray) which can be used to invert the envelope modulation.

LOW CUT sets the cutoff frequency of a global high-pass filter as known from many Roland synthesizers. This is helpful to remove unwanted bass frequencies and/or “thin-out” the sound.

VCF		
10 CUTOFF	13 ENV	16 ENV INV
11 RESO- NANCE	14 ATTACK	17 SUSTAIN
12 LOW CUT	15 DECAY	18 RELEASE

Voltage Controlled Amplifier (VCA)

Besides the global LEVEL parameter, the amplifier also has its own ADSR envelope with the parameters ATTACK, DECAY, SUSTAIN and RELEASE.

The *K3/K3m* has an enormous low end which is fantastic but might be unwanted in some situations. Again I took the liberty to add parameter 22 (LOW BOOST – not present on the *K3K/3m* and thus labeled in dark gray) where you can reduce or even increase the amount of low boost (-18dB to +6dB).

VCA	
19 LEVEL	22 LOW BOOST
20 ATTACK	23 SUSTAIN
21 DECAY	24 RELEASE

Low Frequency Oscillator (LFO)

The LFO offers seven waveforms (parameter 25 SHAPE):

1 – *Triangle*, 2 – *Sawtooth Up*, 3 – *Sawtooth Down*, 4 – *Square*,
5 – *Inverted Square*, 6 – *Random*, 7 – *Chromatic Random*

Chromatic Random only differs from *Random* when applied to the oscillator pitch modulation. In this case the LFO produces random semitone “notes”. Thus, you will not hear an effect when the modulation amount is too low (because it might be less than a semitone).

SPEED and DELAY control the obvious functions. Note that DELAY is single-note triggered i.e. it does not restart with legato playing. Furthermore, DELAY also defines the Auto Bend time (see section *Oscillators*).

In contrary to the original hardware, the speed of the LFO can be synchronized to the host tempo. Therefore, the Master Page function 45 LFO SYNC has to be activated (see section *Master Page*).

The modulation amount can be controlled independently for the oscillators’ pitch, the VCF cutoff frequency and the VCA level – nice! A funny thing is that the MIDI Modulation controller CC#1 (“Modulation Wheel”) directly changes the value of parameter 28 (OSC) – but only in the range of 0 to 15. This is an artifact of the original devices and possibly due to the fact that the *K3* keyboard version does not have a dedicated modulation wheel but only a pitch bend wheel.

LFO	
25 SHAPE	28 OSC
26 SPEED	29 VCF
27 DELAY	30 VCA

Touch Sensitivity

Velocity and Aftertouch/Channel Pressure can be applied to modulate the VCF and VCA. The amount of modulation is set with parameters 31, 32, 34, and 35 (VELO VCF, VELO VCA, PRES VCF, PRES VCA).

A unique feature is the modulation of the balance between OSC1 and OSC 2 via Channel Pressure, which is controlled by parameter 33 PRES OSC BAL: The higher the pressure, the more the “tilt” of the balance towards OSC 2. Finally, parameter 36 PRES LFO-OSC sets the amount of LFO pitch modulation via pressure. For the *K3* keyboard version this is a vital performance parameter since there is no modulation wheel on board...

TOUCH SENS	
31 VELO VCF	34 PRES VCF
32 VELO VCA	35 PRES VCA
33 PRES OSC BAL	36 PRES LFO-OSC

Keyboard Control Voltage (KCV⁴)

Both VCF and VCA can be modulated by the keyboard i.e. the note played. Positive values of parameter 37 VCF will increase the cutoff frequency for higher notes and decrease it for lower notes. Negative values will turn this upside down (higher notes cause lower cutoff frequency and vice versa). Parameter 38 VCA does the same for the level of the amplifier.

Note that F#4 (MIDI note number 66) is the pivot key where keyboard modulation does not have any effect on VCF nor VCA.



Chorus

0	<i>Off</i>	Chorus is disabled.
1	<i>Chorus I</i>	Slow modulation
2	<i>Chorus II</i>	Faster modulation
3	<i>Chorus III</i>	Stepped modulation
4	<i>Tremolo</i>	Fast+deep modulation
5	<i>Chorus IV</i>	Very short delay
6	<i>Chorus V</i>	Short delay
7	<i>Delay</i>	Longer delay

The Chorus has 8 modes and actually consists of three bucket-brigade delay (BBD) lines. The last three modes 5, 6 and 7 are in fact short delays.



When the STEREO switch is set to "STEREO", one channel will be delayed with respect to the other (Kawai calls these modes "Ambient").

Tweak Parameters

There are some tweaks that I added to FB3 which are not part of the K3/K3m specification. The number of voices can be set to 1, 6, 12, 32, or 64 using the VOICES parameter. Note here that 1 voice is different to Mono Mode where 6 voices are played in unison!

Talking about Mono Mode: MONO DETUNE controls the detuning of the 6 voices when Mono Mode is chosen.

Since FB3 was developed based on the original hardware specification, there is a good chance that for identical parameter values the VCF cutoff frequency might differ⁵ from that of your K3/K3m device (this is what happened to Espen Kraft as well). Thus, I added the VCF OFFSET parameter which allows you to tune down or up the cutoff frequency.

The two other tweak parameters are ENV INV (invert VCF envelope, see section *Voltage Controlled Filter (VCF)*) and LOW BOOST (see section *Voltage Controlled Amplifier (VCA)*).



4 It's funny that this is called "Control Voltage": Even on the original K3/K3m the keyboard modulation signal is not a voltage but calculated by the internal micro computer.

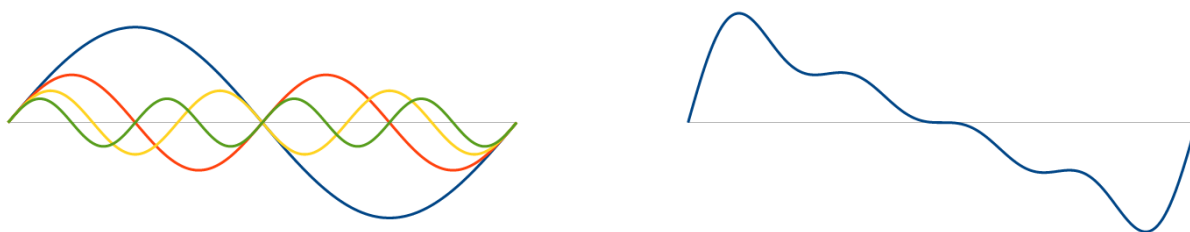
5 Possible reasons are component aging and incompetence of the developer.

User Waves

As already mentioned, the oscillators of the *K3/K3m* have the ability to playback a user-defined waveform, a User Wave. There is only one such waveform available, and the 50 programs have to share it (the 50 cartridge programs have to share the single cartridge User Wave). The *FB3* uses its resources a bit more lavishly and allows for an internal and a cartridge User Wave *per program*. Furthermore, you can use both User Waves in a program, so OSC 1 and OSC 2 can have different User Waves at the same time.

Creating User Waves the Hard Way

User Waves are created by Additive Synthesis: Up to 32 *harmonics* (sine waves with a frequency that is an integer multiple of the base frequency) can be added to shape the final waveform, while each harmonic is weighted with a value from 0 to 31 (the *intensity*).



Adding some sine waves (left) results in an odd waveform (right).

This operation is called *Inverse Sine Transform*, a special case of the Inverse Fourier Transform where only sine waves are used. It can only create *odd* waveforms (the second half of the waveform is the inverted, negative version of the first half) – the phase of the harmonics is ignored. But this is no real problem because humans cannot really hear the difference. In fact the *harmonic content* is what counts.

The original workflow of creating a User Wave starts with selecting the Master mode and clicking button 46 HARMON/INTEN. Now the display shows



- 1) the number of harmonics already defined,
- 2) the current harmonic to edit and
- 3) the intensity of the harmonic.

First, the harmonic number is flashing, and you can change it using the +/YES and -/NO buttons. Click HARMON/INTEN again to edit the intensity (now the intensity number is flashing) and again to toggle between those values. This way you can setup the harmonic content of your User Wave – the COUNT display show you how many non-zero harmonics are already used (maximum is 32).

There are two helper functions: Copy and Erase.

COPY

When you click this button, you copy the intensity of the current harmonic and increase or decrease it for the next harmonic with a *step value* from -3 to 3 (a step value of 0 means that the current intensity will be copied unmodified). Now **right-click** the COPY button again (to simulate a "held" button) and change the harmonic with the +/YES and -/NO buttons. You will immediately see the change of the of the harmonic intensities. Here's an example:

1. Enter Master mode and click HARMON/INTEN.
2. Select the first harmonic using the +/YES and -/NO buttons.
3. Click HARMON/INTEN again. Set the intensity to 31.
4. Now click COPY. The display says "STEP=?" (with a little imagination).
5. Select a *step value* of -1 using the -/NO button. The display shall show a "1" with a dot (for a negative value) at the top left.
6. Right-click the COPY button; it will remain highlighted ("pressed").
7. Select the next harmonic #2 with the +/YES button. The display now shows an intensity of 30 since the *step value* is 1 ($31 - 1$).
8. Select harmonic #3. Now the intensity is 29.
9. You can repeat this procedure until all 32 possible harmonics are defined and COUNT shows "32".
10. When you are done, click COPY again to "release" the button.

Feel free to experiment with the Copy function. You may start copying with any harmonic backwards and forwards and any intensity.

ERASE

Click this button when you want to erase a harmonic i.e. set its intensity to 0. When you right-click the ERASE button it will remain highlighted ("pressed"), and you can erase the subsequent harmonics by selecting them with the +/YES and -/NO buttons (similar to the copying procedure). When you are done, click ERASE again to "release" the button.

Writing and Saving

Once you have defined your User Wave you can listen to it (provided that your current sound uses the User Wave you just edited) by pressing the blue WRITE button – the LED right above the button should now be lit. *FB3* immediately calculates the waveform and transfers it to the wave buffer.

If you want to permanently save the User Wave, click either the INTERNAL button to save it as the internal User Wave, or the CARTRIDGE button to save it as the cartridge User Wave. However, if you want to restore the original User Wave of the program, click WRITE again so that its LED goes off, and the click either INTERNAL or CARTRIDGE. *FB3* will then load the stored waveform from the program.

Little Helpers

So far, *FB3* just mimics what the original *K3K/3m* does, too. But there is some helper functionality that might facilitate the work with User Waves.

Given that Additional Information is activated (see section *Additional Information*) and either parameter 1 (OSC 1 WAVE) or 7 (OSC 2 WAVE) is selected, the edited User Wave will be displayed in the display section.



Of course the display will get updated when you click WRITE (see section above).

Even more exciting is the possibility to load a User Wave from an existing WAV file. The format of the WAV has to be mono, but the sample rate is irrelevant. Furthermore, it must not be too large (I am talking about mega bytes here...). Adventure Kid's waveforms⁶ are perfect candidates.

Loading a User Wave can be done either by using the *Options Menu* or the WAVE→INT and WAVE→CR. Buttons (see section *Plugin Handling*). *FB3* will transform the WAV data into the internal format and calculate the harmonic intensities. And this is where it gets interesting: Since the waveform generation from harmonics and intensities will only use sine waves, the resulting waveform will be "odd" (see *Creating User Waves the Hard Way*). But the loaded waveform isn't necessarily "odd"! Thus, when you click WRITE to re-calculate the loaded waveform from the harmonics, it will likely be different. By the way: This is also true for "odd" waveforms since the resolution of the harmonic intensities is rather poor (only 32 values).

The good thing is that you don't have to recalculate your waveform from the harmonics: *FB3* will happily accept and playback any waveform imported from a WAV.



Left: A "non-odd" waveform imported from a WAV file.
Right: The "odd" waveform after recalculation from its harmonics.



Left: An "odd" waveform (triangle).
Right: The waveform after recalculation from its harmonics.
The difference comes from the poor resolution of the intensities.

6 Get them here: <https://github.com/KristofferKarlAxelEkstrand/AKWF-FREE>

MIDI

The types of MIDI messages that *FB3* accepts are defined by the value of the Master Page function 42 FUNCTION:

- 1 Note On/Off and Sustain pedal only
- 2 like 1 plus Velocity
- 3 like 2 plus Channel Aftertouch, Pitch Bend and Modulation Wheel
- 4 like 3 plus Program Change
- 5 like 4 plus SysEx data (this is the default)

Given that parameter 42 FUNCTION is set to 5, *FB3* will understand sound and User Wave MIDI SysEx dumps sent to it.

FB3 can also send sound and User Wave MIDI SysEx dumps when clicking one of the 50 program buttons (for program sound data) or the INTERNAL/CARTRIDGE buttons (for User Wave data). The Master Page function 44 EXCLUSIVE defines what MIDI SysEx data will be sent:

- 0 Nothing (this is the default)
- 1 User Wave data
- 2 Program sound data
- 3 Program sound and User Wave data

After sending User Wave data, the value of EXCLUSIVE will be set from 3 to 2 or from 1 to 0.

When the option "Send SysEx on Parameter Change" is active (see section *Options Menu*), *FB3* will transmit MIDI parameter change SysEx data. This way you can edit your hardware *K3/K3m* device⁷.

Important note: *FB3* will send MIDI data on channel #1 only!

Of course you can import and save SysEx data from/to files, too. To do so you can either use the Options Menu or the PROG→SYX and BANK→SYX buttons (see section below). Saving or loading User Wave SysEx data will only export/import the harmonic intensity values, not the wave samples themselves.

MIDI Learn

The easiest way to (re)assign MIDI controllers to *FB3* is to use the *MIDI Learn* function. To activate MIDI Learn, click on the MASTER button (if its LED is not lit) followed by the LEARN button.



Now wiggle both the MIDI controller and *FB3*'s parameter or button that you want to link. If you want to unlearn the assignment, click the UNLEARN button instead. Now wiggle the MIDI controller or the parameter that you want to unlearn.

⁷ Due to technical limitations, this does not work for the Audio Units format.

Plugin Handling

The *FB3* plug-in has some handy shortcuts explained in this section.



- **MENU:** Opens the *Options Menu*.
- **FXP → PROG:** Loads the contents of a FXP file into the current program (this will replace the original data of the current program).
- **FXB → BANK:** Loads the contents of a FXB file (this will replace the original data of all 100 programs).
- **PROG → FXP:** Saves the current program to a FXP file.
- **BANK → FXB:** Saves all 100 programs to a FXB file.
- **LOAD SYX:** Loads a *K3/K3m* compatible SysEx (SYX; MIDI System Exclusive) file. The contents of the file (program or User Wave data) will be determined automatically.
- **INIT:** Initializes the current program data to default values.
- **PROG → SYX:** Saves the current program to a *K3/K3m* compatible SysEx file.
- **BANK → SYX:** Saves the current bank (50 programs; internal or cartridge) to a *K3/K3m* compatible SysEx file.
- **WAVE → INT:** Loads the internal User Wave from a mono WAV file.
- **WAVE → CR.:** Loads the cartridge User Wave from a mono WAV file.

Note that you have the option to set a default path for FXP/FXB program files as well as for SYX files. The MENU function will be described below.

LINK

The *K3/K3m* has the option to create a list of at most 32 programs which can be stepped through by clicking the LINK button. I added the same to the *FB3* (although not sure what it might be good for).



To program the link list, click WRITE (so that the LED is on) and LINK. Now click on the program buttons in the desired order. When you are done, click the WRITE button again (LED is off now).

The link list is saved globally in the `fb3.ini` file...

The *fb3.ini* Configuration File

FB3 is able to read some settings from a configuration file (`fb3.ini`). The exact location of this file depends on your operating system and will be displayed when you click on "Reload" or "Save Configuration".

Control **FB3** Via MIDI

It is possible to fully control **FB3** by MIDI controllers, or more precise: Each MIDI controller (except *Modulation Wheel* and *Sustain Pedal*) can control one of **FB3**'s parameters or buttons. The mapping is defined in the `fb3.ini` for example like this:

```
[MIDI Control]
CC7  = 0  # Master Volume
CC65 = 17 # VCF Cutoff
CC71 = 33 # LFO Speed
...
```

The syntax is straight forward:

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

Given the above example, controller 7 directly controls the overall *Master Volume* parameter, controller 65 the VCF's Cutoff frequency etc. As you can see, comments are introduced by the Pound sign (#); they are here just for description purposes and completely optional. 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.

Options Menu

This menu open when you click the MENU button in the common section.

Copy Program	Copy current program to internal clipboard
Paste Program	Paste internal clipboard to current program
Init Program	Initialize the current program
Load Program	Load a FXP program file containing a program to FB3 's current program
Save Program	Save FB3 's current program to a FXP program file
Load Bank	Load a FXB bank file containing 100 programs into FB3
Save Bank	Save FB3 's 100 programs to a FXB bank file
Restore Internal Factory Sounds	Restores the 50 internal programs with the original factory programs of the <i>K3</i>
Restore Cartridge Factory Sounds	Restores the 50 cartridge programs with the original factory programs of the <i>K3m</i>
Load SysEx File	Loads a <i>K3/K3m</i> compatible SysEx (SYX; MIDI System Exclusive) file. The contents of the file (program or User Wave data) will be determined automatically.
Save SysEx Program File	Saves the current program to a <i>K3/K3m</i> compatible SysEx file.
Save SysEx Bank File	Saves the current bank (50 programs; internal or cartridge) to a <i>K3/K3m</i> compatible SysEx file.

Load User Wave...		
	Load User Wave (I) from File	Loads the internal User Wave from a mono WAV file.
	Load User Wave (C) from File	Loads the cartridge User Wave from a mono WAV file.
Save User Wave...		
	Save User Wave (I) to File	Saves the internal User Wave to a mono WAV file.
	Save User Wave (C) to File	Saves the cartridge User Wave to a mono WAV file.
	Save User Wave (I) to SysEx File	Saves the harmonic coefficients of the internal User Wave to a <i>K3/K3m</i> compatible SysEx file.
	Save User Wave (C) to SysEx File	Saves the harmonic coefficients of the cartridge User Wave to a <i>K3/K3m</i> compatible SysEx file.
Default Path for Program Files		Sets the default path for program and bank files
Default Path for SysEx/WAV Files		Sets the default path for SysEx and WAV files
Select Startup FXB		Select the FXB file that should always be loaded when <i>FB3</i> is started
Load Startup FXB		Load the Startup FXB file; can also be used to check what the current Startup bank is
Unselect Startup Bank		Unselect the current Startup FXB file
Global MIDI Settings...		
	MIDI Thru	Set globally if MIDI data sent to <i>FB3</i> should be sent through to its MIDI output (stored in configuration file)
	Ignore Program Change	Set globally if MIDI Program Change data sent to <i>FB3</i> should be ignored (stored in configuration file)
	Send Parameter Change SysEx	Set globally if SysEx data should be sent when a <i>FB3</i> parameter is changed
Show Additional Information		Shows helpful
Reload Configuration		Reload <i>FB3</i> 's configuration file
Save Configuration		Save <i>FB3</i> 's configuration file
Show All Knob Values		Set globally if the knob values should always be shown
Go Online...		
	Check Online for Update	When connected to the Internet, this function will check if a newer version of <i>FB3</i> is available at fullbucket.de
	Visit fullbucket.de	Open fullbucket.de in your standard browser

Plug-In Parameters

Common

parameter	id	description
<i>Master Volume</i>	0	Master (overall) volume
<i>Stereo</i>	1	Sets mono or stereo operation of the Chorus effect
<i>Master Tune</i>	2	Master tune (± 50 cents)
<i>Transpose</i>	3	Transpose (-6 to +5 notes)
<i>Portamento</i>	4	Portamento on/off
<i>Mono</i>	5	Mono Mode on/off
<i>Priority</i>	6	Mono Mode note priority (<i>Last/Highest</i>)
<i>Working Parameter</i>	7	The parameter you can immediately access

Oscillator

parameter	id	description
<i>OSC 1 Wave</i>	8	OSC 1 waveform (0 to 34)
<i>Range</i>	9	Oscillator range (16'/8'/4')
<i>Porta Speed</i>	10	Portamento speed
<i>Balance</i>	11	Balance between OSC 1 and OSC 2
<i>Pitch Bend</i>	12	Pitch Bend amount (0 to 12 notes)
<i>Auto Bend</i>	13	Auto Bend amount (± 31)
<i>OSC 2 Wave</i>	14	OSC 2 waveform (0 to 34)
<i>OSC 2 Coarse</i>	15	OSC 2 pitch (± 24 notes)
<i>OSC 2 Fine</i>	16	OSC 2 fine tune ($\pm 10 \rightarrow \pm 30$ cents)

VCF

parameter	id	description
<i>Cutoff</i>	17	VCF cutoff frequency (0–99)
<i>Resonance</i>	18	VCF resonance (0–31)
<i>Low Cut</i>	19	High-pass filter (0–31)
<i>Envelope</i>	20	VCF envelope modulation amount (0–31)
<i>VCF Attack</i>	21	VCF envelope attack time (0–31)
<i>VCF Decay</i>	22	VCF envelope decay time (0–31)
<i>VCF Env. Invert</i>	23	VCF envelope inverted (on/off)
<i>VCF Sustain</i>	24	VCF envelope sustain level (0–31)
<i>VCF Release</i>	25	VCF envelope release time (0–31)

VCA

parameter	id	description
<i>Level</i>	26	Program level (0–31)
<i>VCA Attack</i>	27	VCA envelope attack time (0–31)
<i>VCA Decay</i>	28	VCA envelope decay time (0–31)
<i>Low Boost</i>	29	Low boost (-18 to +6dB)
<i>VCA Sustain</i>	30	VCA envelope sustain level (0–31)
<i>VCA Release</i>	31	VCA envelope release time (0–31)

LFO

parameter	id	description
<i>Shape</i>	32	LFO waveform (1 to 7)
<i>Speed</i>	33	LFO speed (0–99)
<i>Delay</i>	34	LFO delay and Auto Bend time (0–31)
<i>LFO to OSC</i>	35	LFO to OSC modulation amount (0–31)
<i>LFO to VCF</i>	36	LFO to VCF modulation amount (0–31)
<i>LFO to VCA</i>	37	LFO to VCA modulation amount (0–31)

Touch Sensitivity

parameter	id	description
<i>Velo VCF</i>	38	Velocity to VCF amount (0–31)
<i>Velo VCA</i>	39	Velocity to VCA amount (0–31)
<i>Pres. OSC Bal.</i>	40	Pressure to OSC balance amount (0–31)
<i>Pres. VCF</i>	41	Pressure to VCF amount (0–31)
<i>Pres. VCA</i>	42	Pressure to VCA amount (0–31)
<i>Pres. LFO OSC</i>	43	Pressure to LFO-OSC modulation amount (0–31)

KCV

parameter	id	description
<i>KB VCF</i>	44	Keyboard to VCF modulation amount (± 15)
<i>KB VCA</i>	45	Keyboard to VCA modulation amount (± 15)

Chorus

parameter	id	description
<i>Chorus</i>	46	Chorus type (0 to 7)

Tweak

parameter	id	description
<i>Voices</i>	47	Number of voices (1, 6, 12, 32, 64)
<i>Mono Detune</i>	48	Voice detuning in Mono Mode (0–15)
<i>VCF Freq.Offset</i>	49	VCF cutoff frequency offset (± 24)
<i>LFO Sync</i>	50	LFO speed to host synchronization off/on

Oscillator Waveforms

value	description	value	
0	Oscillator off	18	Pipe Organ
1	Grand Piano I	19	Wood Bass
2	Bell	20	Resonant Strings
3	Strings	21	E-Piano II
4	E-Bass	22	Jazz Organ
5	Oboe	23	Glocken
6	E-Piano I	24	Oriental Bell
7	Organ	25	Harpsichord
8	Brass I	26	Trumpet
9	Digital Bell	27	Sax
10	Slap Bass	28	Grand Piano II
11	Plucked String	29	Square
12	Classic Guitar	30	Sawtooth
13	Hammered Tine	31	Sine
14	Metallic Wave	32	User wave (Internal)
15	Vox Humana	33	White Noise
16	Sharp Wave	34	User Wave (Cartridge)
17	Digital Harmonics		

Frequently Asked Questions

How do I install FB3 (Windows VST2 32 bit version)?

Copy the files `fb3.dll` from the ZIP archive you have downloaded to your system's or favorite DAW's VST2 plug-in folder. Your DAW should automatically register the *FB3* VST2 plug-in the next time you start it.

How do I install FB3 (Windows VST2 64 bit version)?

Copy the file `fb364.dll` from the ZIP archive you have downloaded to your system's or favorite DAW's VST2 plug-in folder. Your DAW should automatically register the *FB3* VST2 plug-in the next time you start it.

Note: You may have to remove any existing (32 bit) `fb3.dll` from your VST2 plug-in folder or else your DAW may screw the versions up...

How do I install FB3 (Windows CLAP 32/64bit version)?

Copy the file `fb332.clap` (32 bit) or `fb364.clap` (64 bit) from the ZIP archive you have downloaded to the `C:\Program Files\Common Files\CLAP` folder. If your DAW supports the CLAP format, it will automatically register the *FB3* CLAP plug-in the next time you start it.

How do I install FB3 (Windows VST3 64 bit version)?

Copy the file `fb3.vst3` from the ZIP archive you have downloaded to your system's or favorite DAW's VST3 plug-in folder. Your DAW should automatically register the *FB3* VST3 plug-in the next time you start it.

How do I install FB3 (Windows AAX 64 bit version)?

Copy the file `fb3_AAX_installer.exe` from the ZIP archive you have downloaded to any of your system's folder and run it. Your AAX-enabled DAW (Pro Tools etc.) should automatically register the *FB3* AAX plug-in the next time you start it.

How do I install FB3 (Mac)?

Locate the downloaded PKG package file 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 FB3?

The ID is `fb - 3`.