FB01 Sound Editor V2.0 : User Manual

© Frédéric Meslin 2010, fredericmeslin@hotmail.com
updated on 09/07/10

Finally, here comes the awaited user manual. This manual describes all the functionalities available in the editor. You will also find a complete reference of the parameters that can be edited and their role. The whole information is split in sections regarding the action intended by the user.

Summary:

Manual contents:

1) Setting MIDI configuration up

Before you start editing parameters and creating amazing new sounds, you must setup your system configuration.

First, plug your MIDI interface(s) and MIDI controller(s) to your computer and switch them all on. Refer to your devices proper manuals for installation instructions.

Then, you need to configure the links between the editor, the FB01 sound module and your potential MIDI controllers. In order to do so, choose the MIDI devices in the *configuration* tab with the provided lists of MIDI compatible peripherals. At least you must select one device for *MIDI IN* and one device for

MIDI OUT. The *MIDI IN* device should be connected to the *MIDI OUT* port of the FB01 sound module and the *MIDI OUT* device to the MIDI IN port reciprocally. To do advance editing including the use of automations, you must configure a MIDI controller able to send MIDI control changes. If this controller can also send MIDI notes (like keyboards and pads), the MIDI routing system will allow you to play notes while editing.

To recognize MIDI devices plugged after the editor start, you can push the *refresh* button. It will close all MIDI connections pending and refresh the three device lists.

When you activate the *IN -> OUT* button*,* every MIDI data received on *MIDI IN* device, excepting the incoming sysex messages, is send back to the *MIDI OUT* device. The *CTRL-> OUT* button provide the same functionality but transmits MIDI data from *MIDI CTRL.*

To fully use the virtual keyboard, you can change its channel and the velocity value sent when a note is hit with the *MIDI channel* and *velocity* buttons. If you own a *QWERTY* or an *AZERTY* keyboard, you should press the button relative to the model of your computer keyboard. If none of them is selected, keyboard playing will be disabled.

2) Dealing with global configuration

The global configuration contains the top level parameters available in the FB01 sound module. The *system channel* is a virtual channel listened by the sound module. Before editing any parameters, the editor's system channel must be the same channel as the sound module. It allows editing several sound modules chained on the same MIDI cable. *Memory protect* is the only lock to prevent you from accidentally erasing parameters stored in the sound module memory. The *set number* is the number of the current set and *master detune* and *master level* controls the tune and the volume of the overall output.

3) Browsing and managing the banks

4) Directing the set

The FB01 set is a kind of mixing console. This console has up to eight channels each one connected to a voice. To go further and learn more about every set's parameters meaning, you should read the A1 annex.

In the *configuration* tab, there are global parameters for the set. They define the set *name*, the way it receives MIDI notes and the LFO configuration.
The *combine mode* button tells the sound module to take in account some particular parameters from the voice loaded rather than the same name parameters from the instrument. It works like a preference choice for the voice parameters when it is activated and a preference choice for the instrument parameter when not. The parameters managed by the *combine mode* are the *Mono / Poly*, the *porta. time*, the *pitch bender* and the *PMD control.* parameters from the voice configuration.

The *reception mode* tells the sound module which MIDI notes it has to play. This feature allows you to easily double the polyphony if you own two FB01 sound modules. In that case one module should have this parameter set to odd, the over to even and they should share the same system channel. On normal operation, this parameter should be set to all.

The *LFO speed* represents the frequency of the LFO and the *LFO waveform* the curb of the LFO output signal. *AMD* parameter stands for Amplitude Modulation Depth and *PMD* parameter for Pitch Modulation Depth. These two values set how deeply the LFO influences the amplitude and the pitch of the instrument sounds if their respective *LFO enable* are activated.

In the *current set* section you find your virtual mixing console. A console part is called an instrument. Each instrument can reserve from one to eight notes considering one rule, the sum of the number of notes reserved by all the instruments must not exceed the eight notes limit. You can reserve the notes with the *num. of notes* button. Each instrument responds to one *MIDI channel.* To place different voices on a keyboard scale you can configure the *Upper limit* value and the *Lower limit* value. Only the notes received inside this interval will be played by the instrument. The *bank* and *voice* parameters set together the voice associated with the instrument. To retrieve a particular voice and its position inside memory, you might use either the bank manager or the front panel of the sound module. The *transpose* parameter sets the degree of

transposition in octaves and the *detune* parameter sets the fine tuning of the instrument. The *enable LFO* button allows the LFO to modulate the pitch and the amplitude of the instrument sound. The *Mono / Poly* button determines the instrument response to a multiple notes play. The *Porta. time* value represents the duration of the pitch glide between the first note held to the next pressed simultaneously. To set the amplitude in semitone of the pitch controller, also called the *PMD control.,* you have to edit the *pitch bender* parameter . The two last buttons *volume* and *pan* are the instrument's volume and position in stereophony.

5) Tweaking and creating new voices

This is funniest and the most interesting part of the FM synthesis process, sadly, it is also the most difficult part to understand. One advice, keep it simple at first and experiment a lot.

a) The voice parameters

In the *current voice* tab there are the global voice parameters like the voice *name,* the voice *style* (which replace the user code in MIDI implementation) and the *feedback* level. This last parameter control the amount of signal from the output reinserted as a modulation input for operator four. The voice *transpose* button defines the base voice transposition degree but unlike the set *transpose* parameter, the value is in semitones. Keep in mind that the final sound transposition degree is the result of the sum of the instrument *transpose* parameter and the associated voice *transpose* parameter.

The following four parameters are the ones that depend of the *combine mode* state (see the *Managing the set* section for more information). They all figure on the set relative instruments and they define both the same values. So why do they need to appear twice in the editor? When the *combine mode* button is activated, the individual parameters from the voice are used, when not, it is the parameters from the set instruments which are taken in account. For a detailed description of this parameters, read the *Managing the set* section.

The *algorithm* parameter sets the connection schematic for the four operators. To understand the way it works, a small figure shows the real links between each operator. The red operators produce the final sound whereas the green intermediary operators modulate the signal for other operators. The operator four with a small F on a corner, is the only operator which can be modulated by the *feedback* signal, if the *feedback* level is not null. This special operator may produce distortion and noise effects depending on the *feedback* amount and its output volume.

The *author* and *comments* fields are not parameters from the sound module and therefore they are not stored in its memory. These parameters are only here to help the user to classify his sounds and remember their intended use. Consider that this fields are only stored in editor files.

The lonely LFO present in the sound module can be configured by either the set LFO parameters or by the voice LFO parameters which have the same effect on it. Like with the *combine mode* feature, the *LFO load* option selects which configuration should be used in preference, in other words, the set one or the voice one. There is something very important concerning the sound module limits you need to understand now. As there is only one LFO, only the LFO parameters of the last voice loaded, with the *LFO load* button activated, drive the LFO. If none of the voices loaded by the set have the *LFO load* parameter on, it is the LFO set configuration which is finally used.

The *LFO sync* parameter synchronizes the beginning of the LFO wave with key pressing. If this parameter is not set, the LFO continues to oscillate even if there are no more notes played.

The next *AMD* and *PMD* parameters are strictly the same parameters found in the set configuration (if you want to know more about them, read the *managing the set* section). These values may replace the same name set parameters whether the *LFO load* button is activated or not.

The *PMS* and *AMS* parameters respectively stand for Pitch Modulation Sensitivity and Amplitude Modulation Sensitivity. They are the limits of modulation intensity for pitch and volume. To clarify the difference

between the *AMS* and *AMD* parameters (or between the *PMS* and *PMD* parameters), understand that the intensity of the modulation is bound to 0 and AMS (or 0 and PMS) and the current modulation depth is set by *AMD* or *PMD* which represent the ratio between 0 and AMS (or 0 and PMS).

b) The operator parameters

In the *current operators* tab are located all the parameters related to the four operators that forge the sound shape. These somehow complex parameters describe mathematically the tone and the envelope of the sound.

The *lvl. vel.* parameter (which means level velocity) determines how much the note hitting velocity influence the operator output volume. Typically, this parameter is useful to produce a louder sound when the velocity increases. It can also lead to special effects if set on an intermediary operator for example. The *AR vel.* parameter defines the influence of the velocity on the envelope attack rate, specified by *AR*, pretty much like the *lvl. vel*. acts on the output volume.

The *AR* (attack rate), *D1R* (decay 1 rate), *SL* (sustain level), *D2R* (decay 2 rate) and *RR* (release rate) parameters define the envelope shape. An envelope is an overall curb that a parameter tracks. The FB01 sound module gives to each operator a four stage envelope controlling its output level. The different phases of the envelope are drawn on the little figure associated with the operator. Remember that *AR*, *D1R*, *D2R* and *RR* are rates and not durations, so the higher they are the quicker the envelope phases are.

The *coarse* parameter sets the operator transposition in semitones and the *mult.* parameter represents the transposition in octaves.

The *Car. / Mod.* (carrier vs. modulator) button allows or not the signal to saturate the next stage. This saturation is generally unwanted for the operators that produce the output sound but can be useful to create noise and creative artifacts for an intermediary operator.

5) Exploring the automations

The automation feature completely unleashes the user from the original editing constraints allowing him to change nearly any FM parameter available, in real-time, with the help of any MIDI compatible controller.

6) Getting and sending sysex

The getting and sending sysex functionality is located in the FB01 menu. When pressing send or get bank, the bank concerned is the bank selected with the *current* button. For the voices, the exactly same principle is used. The voice concerned is always the voice associated with the *current* instrument selected.

As you can notice, there is no way to receive the global configuration of the sound module. This is due to limits in the MIDI implementation of the FB01.

7) Loading and storing files

There are two kind of file the editor can handle. Pure sysex files and editor's own format files. You should use this last format if you want to add extra information to your banks, sets and voices configuration (like the author and comments fields for the voices, more parameters will come). The sysex format is provided for compatibility with other MIDI tools.

A1) The set parameters reference

A2) The voice parameters reference

A3) File formats

The editor files format changed slightly since the last release but the compatibility with the previous version is maintained. This information is addressed to other MIDI tool developers in order to support these files. Remember these formats are subject to changes and this data arrangement is not guaranteed.

a) FBB files :

b) FBS files :

c) FBV files :

If you find some imprecise, inaccurate information or even English mistakes in this manual please contact me soon so I can update it quickly. If you have any question regarding this user manual, the software or the synthesis process in general, feel free to contact me by email at:

fredericmeslin@hotmail.com

_____

_____