

FB01 Sound Editor V2.0 : User Manual

© Frédéric Meslin 2010, fredericmeslin@hotmail.com
updated on 15/07/10

This document is part of the FB01 SE project distributed under a GNU GPL license, please read the licence.txt file located at the program root for legal stuff.

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:

- 1) Setting MIDI configuration up
- 2) Dealing with global configuration
- 3) Browsing and managing the banks
- 4) Directing the set
- 5) Tweaking and creating new voices
- 6) Meeting the automations
- 7) Getting and sending sysex
- 8) Loading and storing files
- 9) Copying, pasting and exchanging
- 10) Developer's notes

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

This section will be filled with information soon.

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 other 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) Meeting the automations

The new automation feature totally transforms the old fashion FB01 sound module into a modern synthesizer by revealing its true synthesis possibilities. It also unleashes the user from the original editing constraints by allowing him to change any FM parameter while hearing the sound evolving in real-time, providing the help of any MIDI controller.

Before playing and having fun with the automations, you must set up a MIDI compatible controller able to send MIDI control changes to the editor and possibly MIDI notes. In order to install this kind of peripheral properly, please read the *configuration section* of this manual.

Once you have configured correctly your MIDI controller, you should be able to see the messages it sends to the editor, in the *automation* tab. You will determine here which button or slider of your MIDI controller corresponds to which MIDI control change and the range of the values sent. This information is very important to configure your MIDI controller as there is no auto-learn feature included yet.

To create a new automation link, you have to select one control change from the first list, one editor parameter from the second list, set the value of both input and output ranges buttons and then validate your choice by pressing the add button. If no mistake was made, your new automation link should appear immediately in the third list as a recap entry. Whether you no longer need an automation link or if you want to replace it by a new one, choose it in the automation list and press the *delete* button. You can also completely clear the automation list by pressing the *delete all* button.

When your first automations will be configured and ready to use, you may want to see the editor parameters moving while you are turning your controller knobs. By default, this feature is disabled because it consumes some cpu time to refresh the user interface. If you want to toggle it on, just activate the *auto refresh* button. Remember that the automation function is always working, even if the button is not activated. In auto refreshing mode, you will notice that the editor parameters need some time to be updated but do not worry about this lagging, the sound module itself is updated in real time.

The input and output ranges allow you to control each editor parameter precisely. By default, the both input and output ranges are set to 0 to 127, which means the MIDI controller knob emitting the corresponding MIDI control changes directly drives the editor parameter chosen. Normally, it should be great but sometimes you will be bored of having your buttons a 360° turn to change the values from 0 to full, when precision is not required and just a quarter of turn would be enough. At the opposite for example, you may need to set one operator volume very precisely, and you just want to control the volume close to a specific value.

The way the input and output ranges work is the following one: every value coming from the MIDI controller which lies inside the input range is scaled to fit the output range. If the input value is lower than the input range, the output value will be the lower boundary of the output range whereas if the input value is higher than the input range, the output value will be the higher boundary. It works like a magnifying glass.

6) Getting and sending sysex

The getting and sending sysex functionality is located in the FB01 menu. When pressing *send bank* or *get bank*, the bank concerned is the bank selected with the *current* button. For voices and sets, the exactly same principle is used. The voice and instrument concerned are always the ones associated with the *current* button 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 sets and voices parameters (like the author and comments fields for the voices, more parameters will come in future). The sysex format is provided for bank backups and for compatibility with other MIDI tools.

9) Copying, pasting and exchanging

The purpose of this section is to clarify the original copy/paste/exchange concept and expose its limits. First, you can only copy, paste and exchange in the bank, set and operator tabs and you can only use local parameters. Do not think in opening several FB01 editors and copy parameters from one editor interface to the other one. You even cannot copy one operator from a voice, then load a new voice and paste the operator you saved before. The copy/past/exchange mechanism is definitely local.

To understand how to use the exchange feature, think of what the copy and paste functionality does. It stores parameters in a temporary buffer when copying and then it loads it back when pasting. The exchange feature swaps the buffer content with the parameters selected to be exchanged. For example, if you need to exchange the parameters of two operators, first *copy* one operator parameters then *exchange* them with the other one and finally *paste* the parameters exchanged on the first operator you copied from.

10) Developer's notes

The editor files format changed slightly since the first release because of some oversights and therefore the compatibility with the older version files is not supported yet. I will try to do something if some people really need it.

If you find some imprecise, inaccurate information or even bad English mistakes (I hope you will forgive me) in this manual please contact me 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

© Frédéric Meslin 2009 – 2010
