MIDI BANK MASTER

MBM

MBM DEVICE

MBM is a programmable MIDI keyboard that can store and send MIDI sequences autonomously.

It has 16 in-line keys, a MIDI DIN output, a USB HOST type A input/output and another USB DEVICE type B through which the device is powered. A switch changes the MBM to HOST or DEVICE mode.

Three types of scenarios can be proposed regarding the connection.

1. Only connect the MIDI DIN output directly to the equipment in question, powering the MBM through the USB B input.
2. Connect the USB B input to a computer (HOST) instead of a power supply. To do this we will have the switch in the DEV position, the MBM is now a USB MIDI Class Compliant device. What is received by the USB B input is sent to the MIDI output. The MIDI sequences when the keys are pressed can be output both through USB and through the MIDI output.
3. With the switch in the HOST position, we can connect a USB MIDI Class Compliant device to the USB A input, or several through a HUB.

This third option is the most interesting and complex at the same time, although easy to understand and configure with the software that we will talk about later.

If we connect several devices with a HUB, the MBM, in addition to being able to discriminate which keys from each bank are sent to which devices, can act as a router. As we said, the USB HOST input supports a USB HUB (or two in cascade) where we can connect up to eight USB Class Compliant MIDI devices (controllers, keyboards, sound modules, MIDI interfaces, etc...) and communicate them with each other or with the MIDI DIN output.

Once configured to our wishes, operation in practice is very simple.

The system consists of 15 banks of 15 keys each. The rightmost black key [B], is used to select which bank to use. When you turn on the device, bank 1 is default. When pressed, the [B] key lights up, the next key pressed from the first 15 selects the bank, and the [B] key goes out. At all times the key that represents the selected bank flashes every 3 seconds. Each key within a bank is illuminated if it is programmed to send some sequence, or unlit if it is empty.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [B]

A MIDI sequence can be sent when pressing and a different one when releasing a key. For example, sending a program change and a chord, and changing the program again and disabling the chord on release. Activate a certain effect and deactivate it when releasing, etc... Thus, we can have 15 banks x 15 keys x 2 = 450 different sequences.

Physically, the keys are MX type (like those on a mechanical keyboard) and have a double transparent layer to be able to change their labeling. The first 15 have a white background and the last [B] has a black background.

To configure and update the MBM firmware, we have developed cross-platform software for Windows, Mac and Linux with the most current programming techniques, ReactJS/Electron in Typescript, which we detail below.

SOFTWARE MBM

To configure or update MBM, we download the version corresponding to our operating system from https://www.sevillasoft.com and install it.

We put the switch in DEV mode and, holding down the [B] key, connect the USB cable to the computer. All keys are illuminated indicating that the device is ready to be configured.

We open the application, and if we have done the right thing, the following message appears:

Otherwise this message will appear:

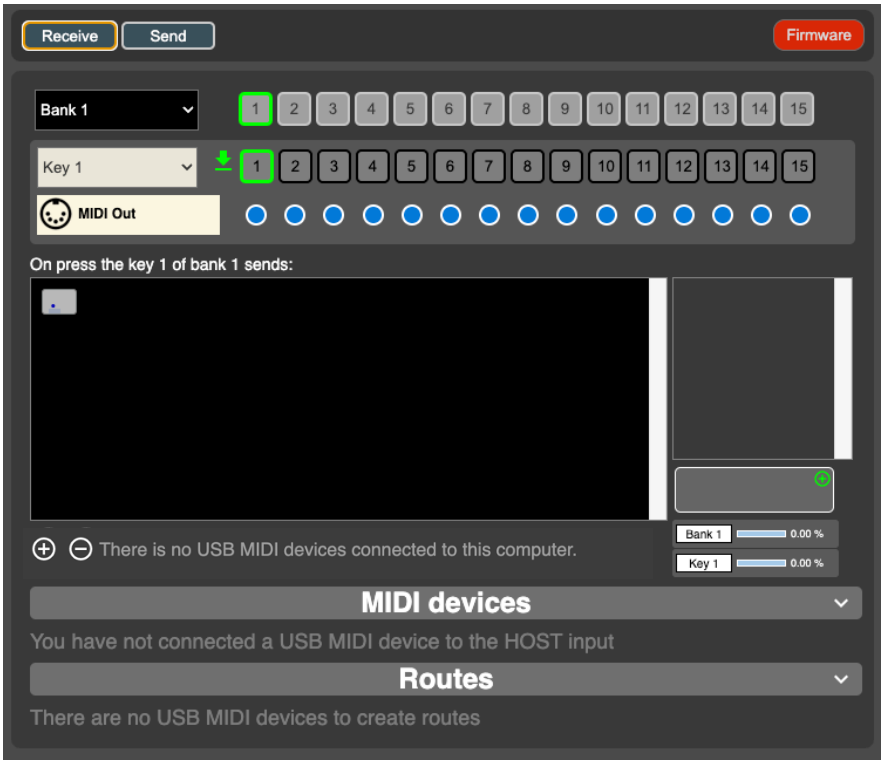
And the button with the option to reconnect.

Once connected, the screen will show these buttons:

In the program menu we will have the option to open a previously saved configuration file with the CFG extension, example: mbm_20_10_2023.cfg

When you press Receive, the MBM sends its current configuration and this is shown on the screen in addition to the message:

In principle, since we have not yet connected any USB MIDI device to the HOST input, we will get an empty configuration like this.



At the top we see that there is a new button: Send. When you press it, the changes we make are transferred to the MBM. The following box contains the buttons to select the bank. Below we have the key selection buttons and whether the action is to press or release .

In the MIDI Out line, each check indicates which keys of the selected bank will use the MIDI output.

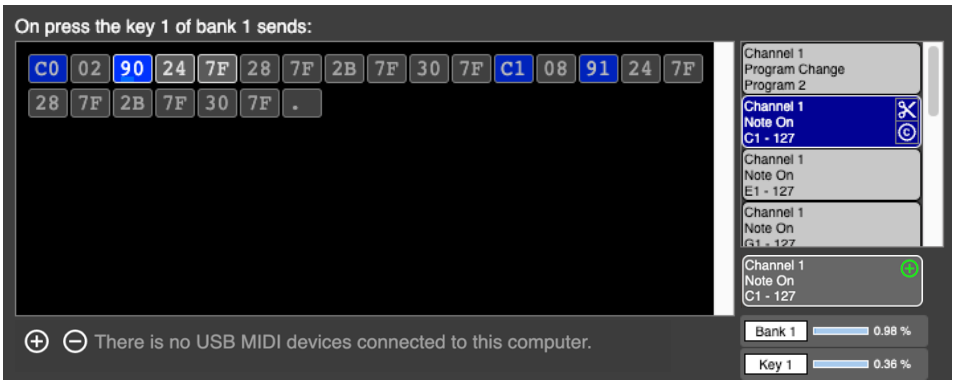
Next, we have the MIDI editor. Here you edit the sequence to be sent for the selected bank/key according to the type of action press/release.

The MIDI Editor

- We can directly write the MIDI data.
- Learn from USB MIDI devices connected to or computer.
- Copy/paste from an external source.
- Import it from a SYX file with the Open SYX menu option.

We can also save the content of the editor to a file with the **Save SYX** menu option.

The list on the right shows the meaning of the edited hexadecimal values. Use the arrows to move around both the editor and the list.



With the ✂ and © buttons we can cut or copy the selected MIDI sequence.

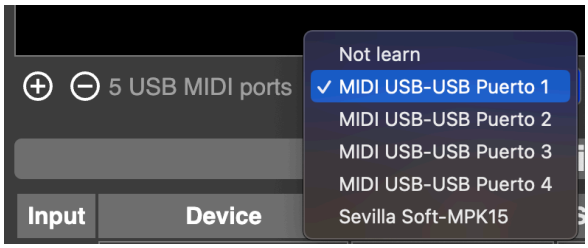
The box below the list shows said sequence and the button ⊕ inserts it at the cursor position in the editor.

In this example, we are editing key 1 of bank 1. The program is changed to channel 1 and a C Major chord with 4 notes is sent. The program is then changed to channel 2 and the same chord is sent.

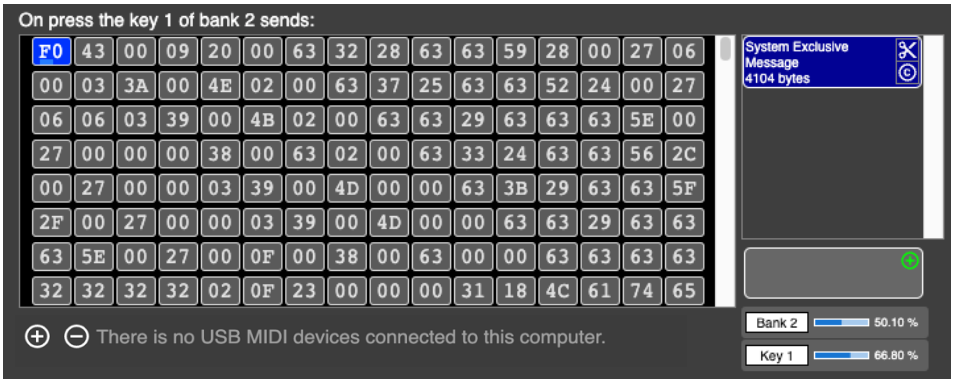
Here, releasing the key 1 of bank 1 deactivates the notes that were activated when you pressed.

The editor buttons ⊕ ⊖ are used to insert or delete MIDI data.

We can take information from MIDI devices that we connect to the USB ports of our computer, which is known as Learn.



Each device that we connect appears in a list, we can select Do not learn or we can select a specific device and the information that it sends is copied to the editor.



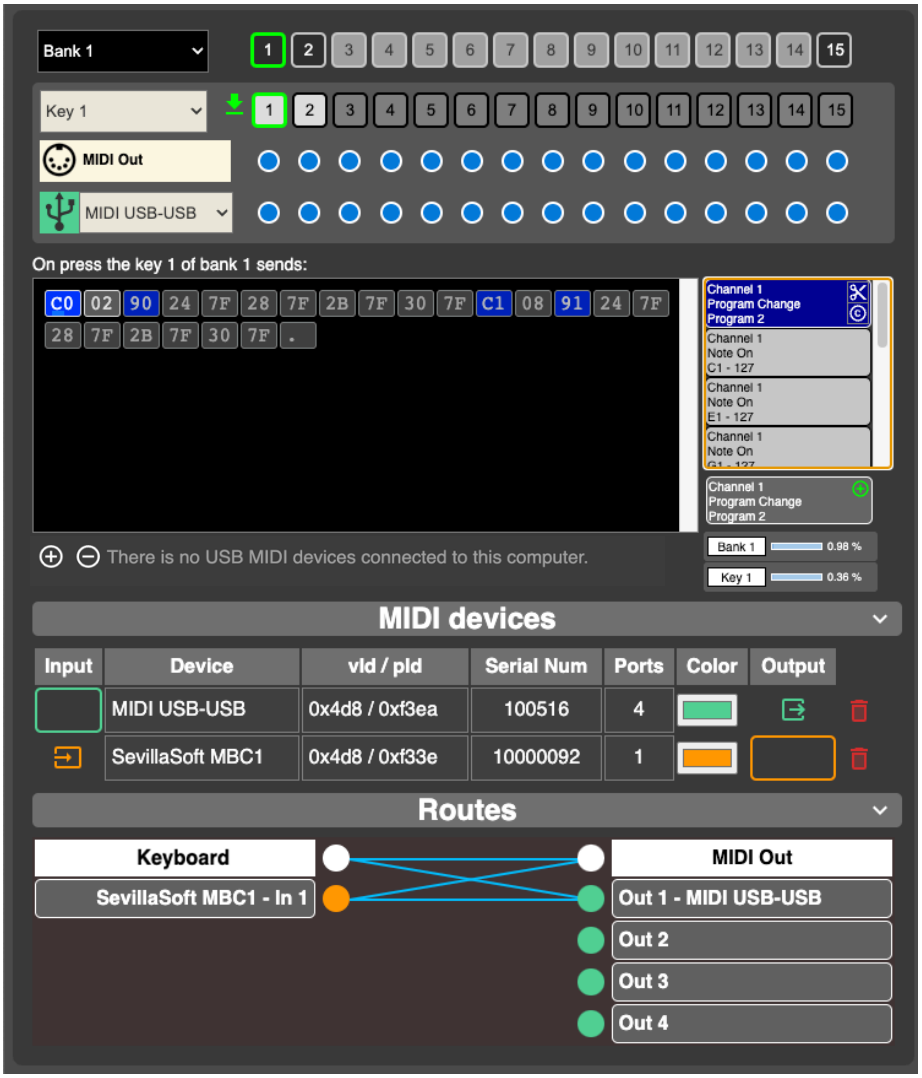
In this example, the editor contains a system exclusive message that is sent when you press key 1 of bank 2.

As we mentioned earlier, editor data can be saved or read from files with the **syx** extension. They are standard files used by programs such as Sysex Librarian or Mios Studio. These files can contain one or more system exclusive messages, or in our case also, any set of MIDI data.

The program menu also gives us the option to change the language and save or read all the configuration in a CFG file.

USB MIDI devices

With the switch in the HOST position, each compatible USB MIDI device that is connected to the MBM is memorized and assigned a default configuration. All device inputs (a USB MIDI device can have multiple ports) are routed to the MIDI Out. When we receive the configuration, in our program we can easily modify it, do routing, that is, what is received by one device is sent to another device and/or to the MIDI output. And also, we can indicate through which outputs to send the information programmed in each key of each bank.



Bank 1

Key 1 on press.
All keys in **Bank 1** send to the MIDI output. All keys in **Bank 1** send to the USB-USB MIDI device output except **Key 2**.

MIDI sequence to be sent when pressing key 1 of bank 1.

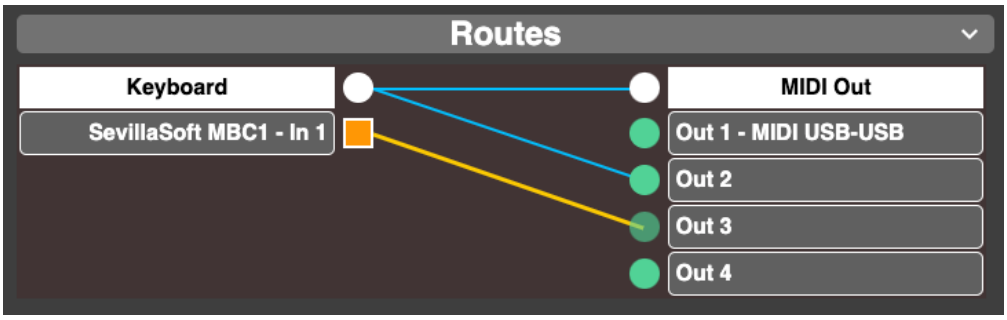
A bank can have a maximum of 8Kb of data. One key up to 6Kb. The percentage used is shown on the screen.

In the Input and Output columns, we select the pair of devices to route. In the example, we have selected the Sevilla Soft MBC1 input which has one port and the USB-USB MIDI output which has 4 ports.

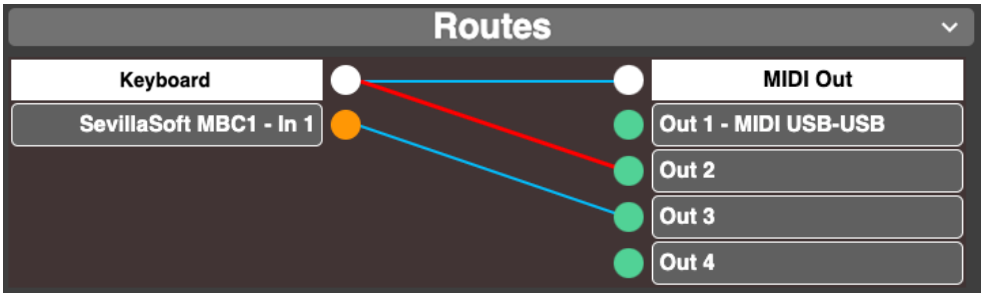
The data sent by the keyboard and the data sent by the Sevilla Soft MBC1 device come out through the MIDI output and through port 1 of the MIDI USB-USB.

Routes

To create a route, click on the circle that represents the input, bring the line to the circle that represents the output and click again.



To delete a route, place the mouse over the line to delete and click when it is red.



Send configuration

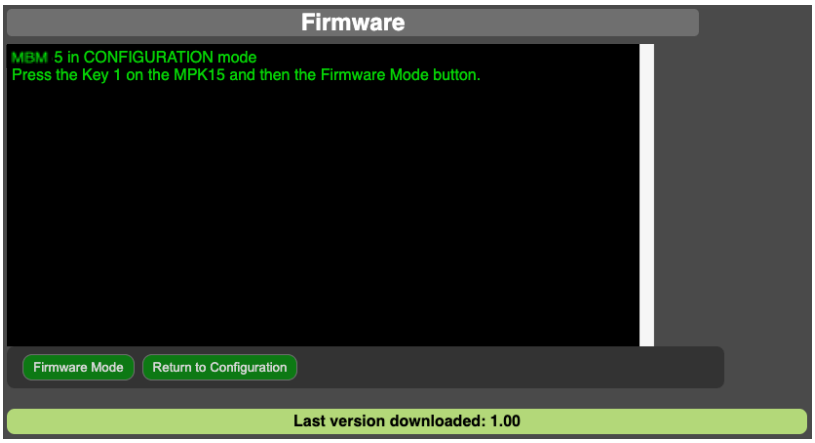
Once configured to our needs, we should save the information in a CFG file with the Save CFG menu option. For our MBM to be configured, press the **Send** button and wait for the sending to be completed.

When this operation is finished, this message is displayed and the MBM is configured.

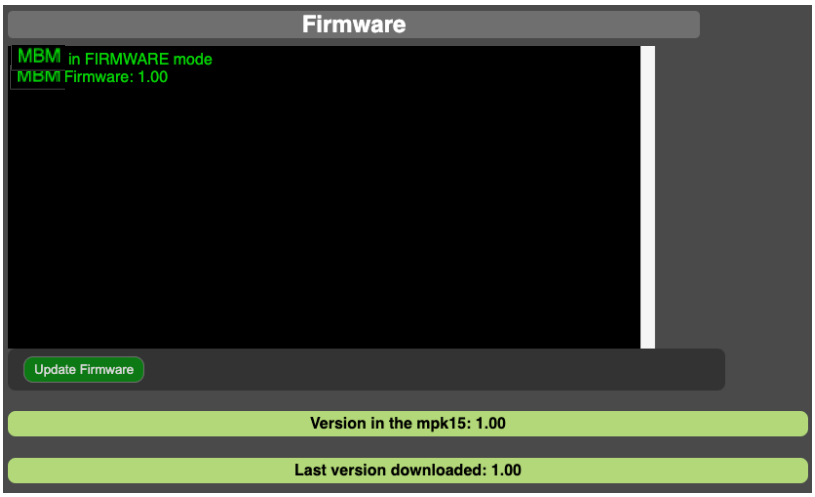
Firmware update

To update the firmware you must be connected to the internet. Pressing the red Firmware button at the top right downloads the latest version of firmware from our servers. This message is displayed:

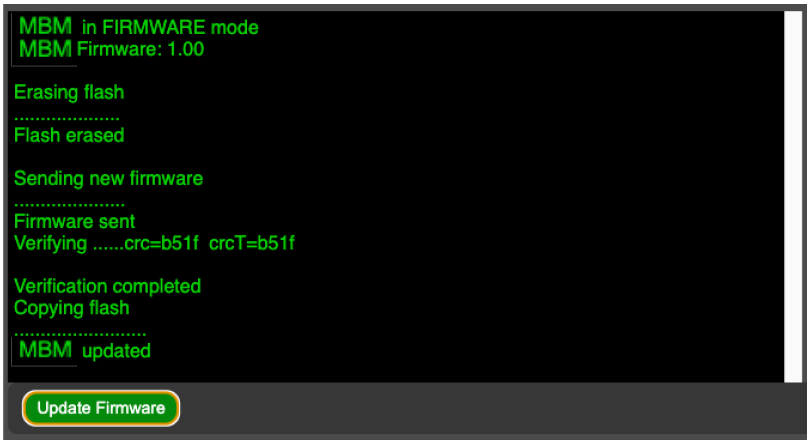
The screen gives us instructions on how to proceed:



In this state, we can return to the previous screen by pressing **Return to Configuration**, or press **Key 1** on the **MBM**, leaving only the black key illuminated, and now press **Firmware Mode**.



In the two lines below we see the current version in the **MBM** and the version that you downloaded from the internet. In this case the **MBM** is updated and it would not be necessary to update it.



When we update the firmware the process is shown. It is important that you do not disconnect the MBM during operation. Which ends with the message:

MBM updated.