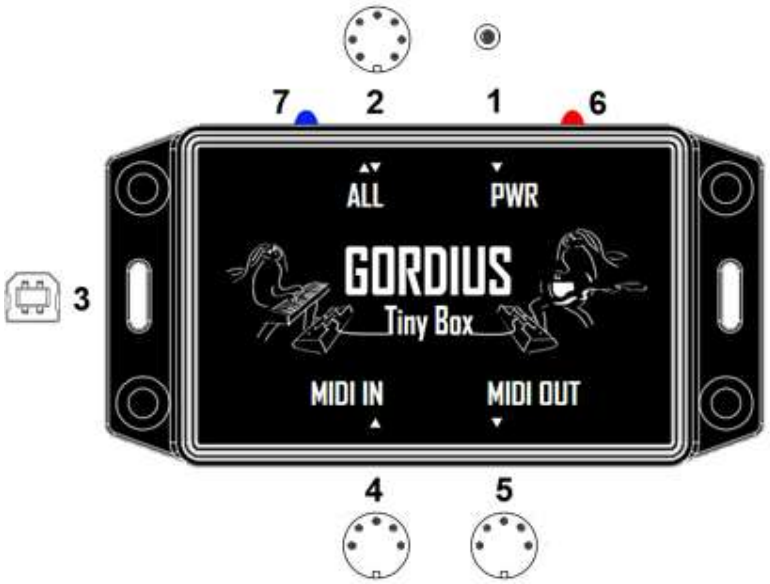
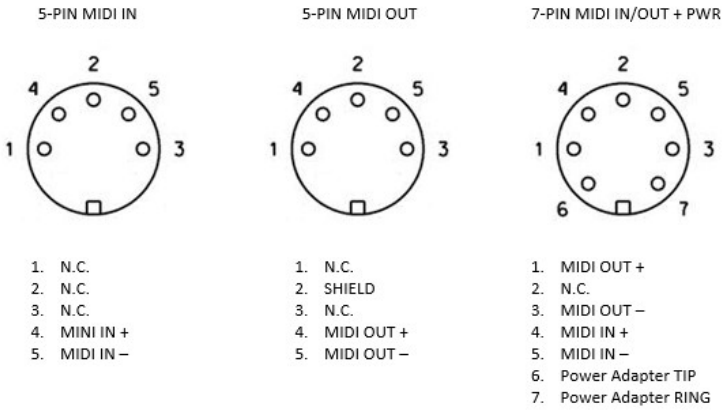
The Tinybox MIDI routings – a detailed rundown

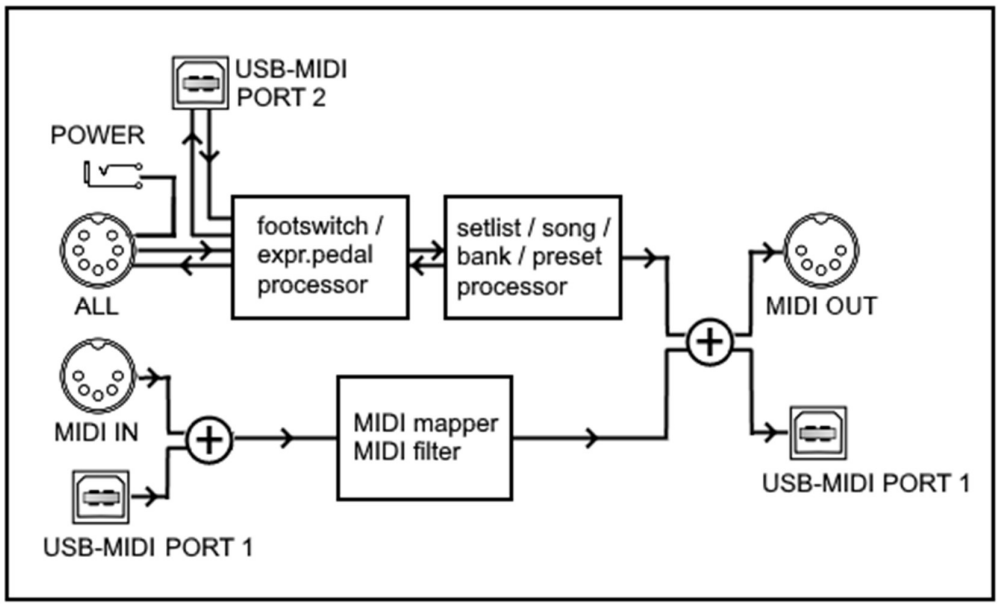


1. Power input: connect any power adapter delivering 9V AC or DC, 500mA or more
2. 7-pins MIDI connector for two-way connection with the Behringer FCB1010
3. USB connector for connection with a Windows or Mac computer
4. 5-pins MIDI IN connector
5. 5-pins MIDI OUT connector
6. Red power LED
7. Blue USB status LED

The FCB1010 connects to the TinyBox through a 7-pins connector. This connector actually behaves as the combination of a power jack, a 5-pins MIDI IN connector and a 5-pins MIDI OUT connector.



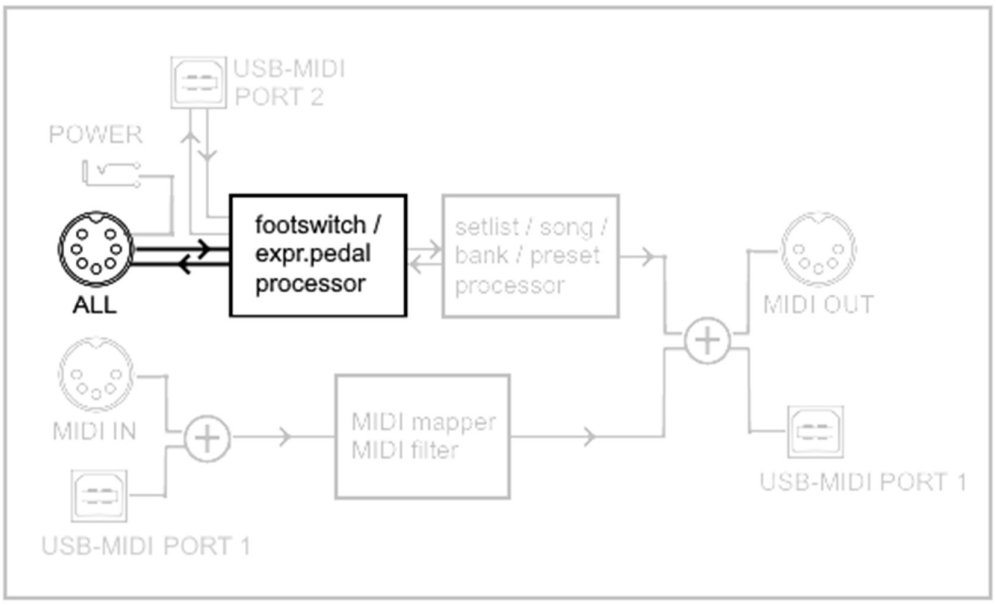
the 7-pins MIDI connector compared to a regular 5-pins MIDI IN / MIDI OUT connector



The TinyBox internal routing

Remark : in the schematic representations you will see several USB connectors labeled “USB-MIDI PORT 1” or “USB-MIDI PORT 2”. In reality these 2 MIDI-USB ports share the same physical USB connector. Actually, the TinyBox contains 3 MIDI-USB ports in total. Port 3 is a dedicated port for patchdumps and firmware updates, and is therefore not shown in the schematic representations of the TinyBox MIDI routing.

Communication between FCB1010 and TinyBox



The TinyBox purchase includes a specific firmware chip, to be installed in the FCB1010, which turns it into a dummy slave for the TinyBox. With this firmware the FCB1010 no longer contains any logic or setup, it just sends a MIDI command on each switch press and switch release and on each movement of the expression pedals, and it listens to incoming MIDI messages in order to turn LEDs on or off, or show info on the 7-segment display.

The TinyBox will proceed to a next or previous setup bank (or song) when it receives a command from the FCB1010 which indicates a click on the Up or Down switch. It will select a preset within the current bank when it receives a command from the FCB1010 which indicates a click on one of the ten footswitches. It will send whatever “sweep” is programmed in the TinyBox setup when it receives commands from the FCB1010 indicating a movement of one of the two expression pedals.

Whenever the TinyBox has selected a different bank or song, it will send a command to the FCB1010 to update the current bank number on the 7-segment display. Whenever a preset is selected or an effect is activated or deactivated, the TinyBox will send the necessary commands to the FCB1010 to turn the corresponding LEDs on or off. So again, it’s the TinyBox which is in full control of the FCB1010, which is just a “dummy” slave showing on its LEDs whatever the TinyBox tells it to show.

Next page shows the details of this FCB1010 – TinyBox communication.

FCB1010 to TinyBox :

```
MIDI Channel 13

NoteOn/NoteOff 0x00-0x09 0x7f  = switch 1-10    press/release
NoteOn/NoteOff 0x0e/0x0f 0x7f  = switch DOWN/UP press/release
ControlChange  0x04/0x07 value = expr.pedal A/B position
```

TinyBox to FCB1010 :

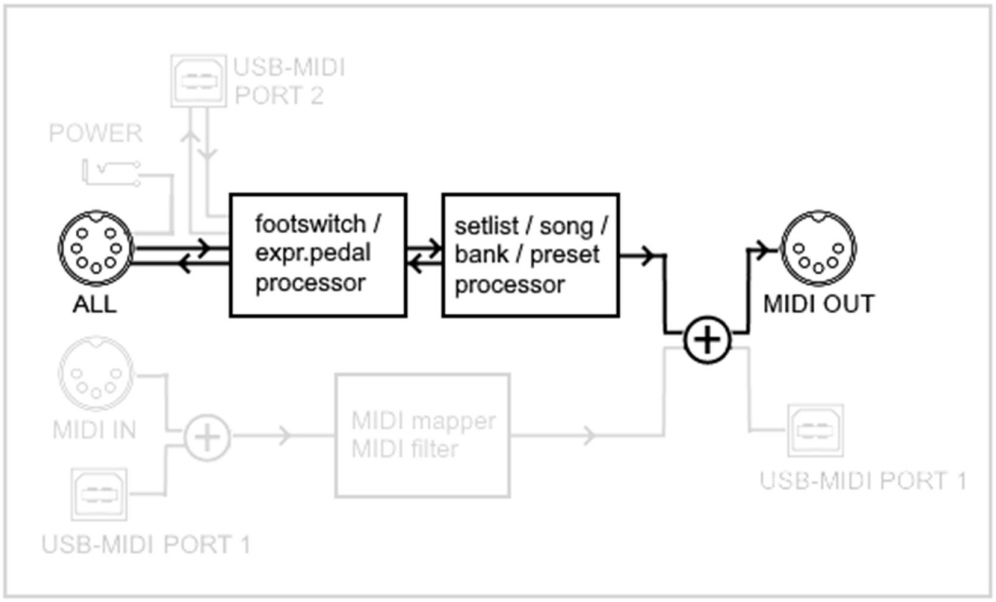
```
MIDI Channel 13

ControlChange 0x0c value 0x00-0x09 = LED ON  for switch 1-10
                  value 0x0a-0x0b = LED ON  for expr.pedal A/B
                  value 0x0c-0x16 = LED ON  for fcb "menu" LEDs
                  value 0x18-0x19 = RELAY1/2 ON
                  value 0x20-0x29 = LED OFF for switch 1-10
                  value 0x2a-0x2b = LED OFF for expr.pedal A/B
                  value 0x2c-0x36 = LED OFF for fcb "menu" LEDs
                  value 0x38-0x39 = RELAY1/2 OFF
ControlChange 0x0d value 0x00-0x7f = DIGIT001 7-seg (*)
ControlChange 0x0e value 0x00-0x7f = DIGIT010 7-seg (*)
ControlChange 0x0f value 0x00-0x7f = DIGIT100 4-seg + points (**)
ControlChange 0x10 value 0x00-0x7f = show value 000-127 on display
ControlChange 0x11 value 0x00-0x47 = show value 128-199 on display
ControlChange 0x12 value 0x00-0x7f = as 0x10 but with '+' added
ControlChange 0x13 value 0x00-0x47 = as 0x11 but with '+' added

               g
               --
             b|  |f           (*) bit6..0 =  g   f   e   d   c   b   a
               --   --> a     (**) bit6..0 = dp3 dp2 dp1 '+'  f   e   a
             c|  |e
               --
               d
```

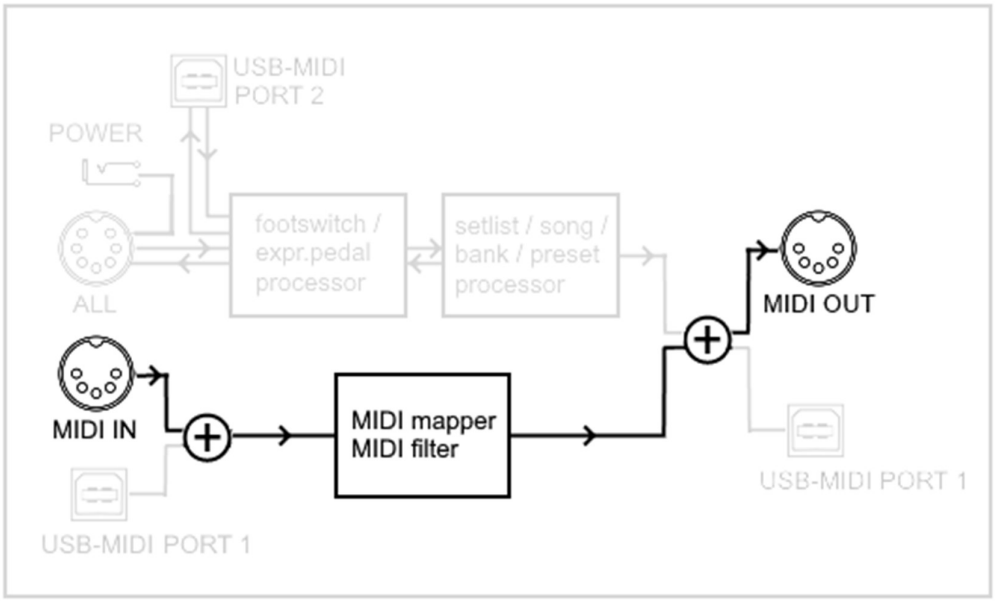
Communication protocol FCB1010 - TinyBox

Controlling your MIDI enabled hardware



This can be considered the main function of the TinyBox: you create the necessary presets to control your gear, you organize them in banks, and you download this setup to the TinyBox. FCB1010 switch presses are processed and result in a certain preset or effect to be activated. The preset content is a series of MIDI commands, which are sent to the MIDI OUT connector of the TinyBox.

The MIDI IN to MIDI OUT routing

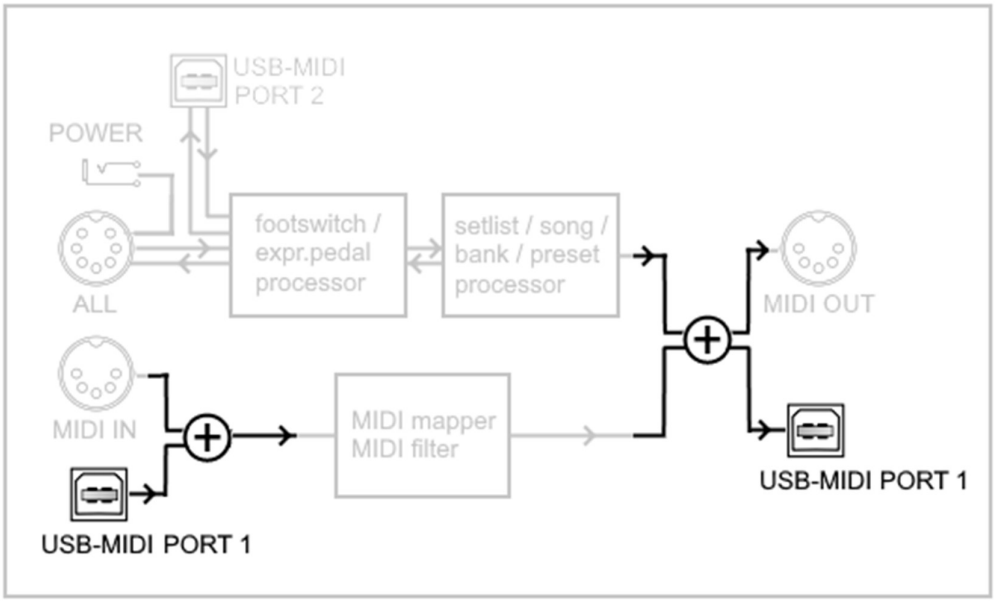


Apart from the MIDI messages generated by your TinyBox setup, you can also forward MIDI messages to your gear coming from another MIDI source. These messages are merged with the MIDI messages generated by the TinyBox, and both streams are sent to the MIDI OUT connector.

One example of a device connected to the MIDI IN connector could be an extra MIDI controller which sends commands to your gear, next to the FCB1010 – although in most cases the combination of FCB1010 and TinyBox is powerful enough to take simultaneous control over many devices.

Another more useful example would be to have a MIDI keyboard connected to the MIDI IN, and your synths or sampler modules connected to the MIDI OUT. This way you can control all your sounds with the FCB1010 at your feet while playing the keyboard. For this scenario the TinyBox adds a powerful MIDI mapper and MIDI filter to this MIDI routing. It allows you to block certain note ranges, move them to a different MIDI channel, transpose them up or down, and so on. This can turn a very simple MIDI keyboard into a powerful master keyboard with different zones controlling different synths.

MIDI-USB port 1



When connecting the TinyBox to your computer, you will see 3 extra MIDI ports appear, labeled “TinyBox port1”, “TinyBox port2” and “TinyBox port3”.

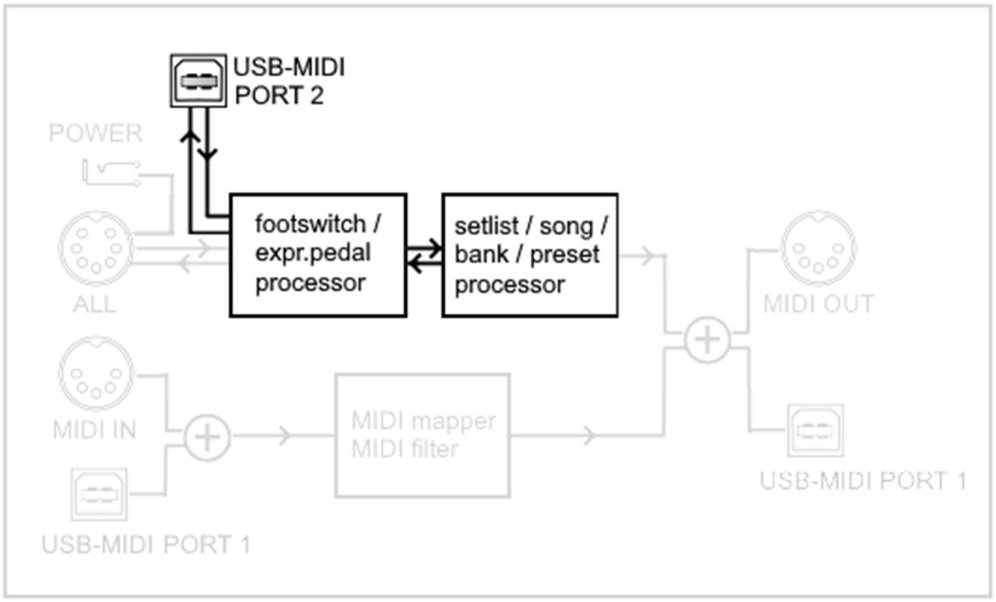
Port 1 is available for general use. Thanks to this port the TinyBox can control not only your MIDI hardware but also any MIDI enabled software application: DAW, soft synth, VST plugin, guitar amp modeler, … As you can see on the schematic above any MIDI message sent to the MIDI OUT port is also sent to USB-MIDI port 1.

On the other hand the TinyBox can also be used as MIDI-USB interface to send messages from your computer to your MIDI hardware. As can be seen on the schematic above all messages leaving your computer on MIDI-USB port 1 run through the MIDI mapper and filter and leave the TinyBox on the MIDI OUT connector.

Important remark – LOOP ALERT !

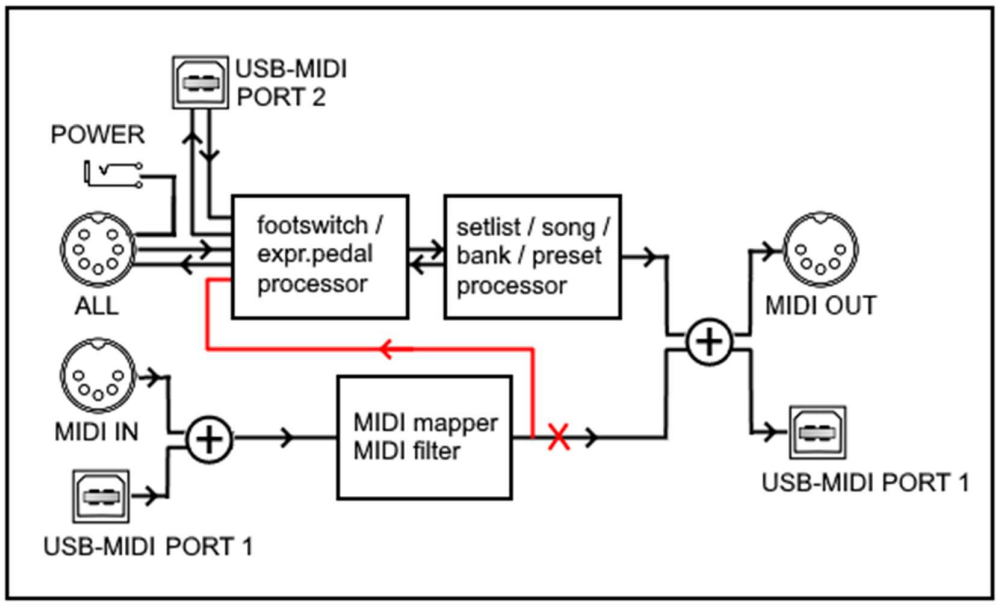
What the schematic also shows is that there is a routing from MIDI-USB PORT 1 to MIDI-USB PORT 1. This means that all messages sent from your computer on PORT 1 OUT will again arrive at PORT 1 IN. Therefore it is important that you disable any “MIDI THRU” or “MIDI forwarding” functionality in your software application. If you don’t do that it is obvious that you will create a “MIDI loop”, and MIDI messages will start running around from computer to TinyBox and back, making the complete MIDI system unstable.

MIDI-USB port 2



Although you will see “TinyBox Port 2” as an available MIDI-USB port when connecting the TinyBox to your computer, this port cannot be used by your music software. It is reserved for use by TinyBox ControlCenter. The schematic shows that this USB port connects to the TinyBox internal system just like the FCB1010 does through the 7-pins port. Indeed, this allows TinyBox ControlCenter to receive the current FCB1010/TinyBox status (which effects are activated, which bank is currently active, and so on) and it also allows you to use the FCB1010 status screen on your iPad as a touch-enabled remote control, sending footswitch messages to the TinyBox just like the real floorboard does.

A special case : “UseKeyboardControl”



The TinyBox also supports another way of control, instead of using the FCB1010. This alternative routing, indicated in red in the schematic above, might be attractive to keyboard players for instance. Indeed, it allows to use the keys of a keyboard to scroll through banks, select presets or activate effects in the TinyBox.

On a previous page we mentioned the communication protocol between FCB1010 and TinyBox. More specifically the footswitch presses and expression pedal movements cause following MIDI messages to be sent by the FCB1010:

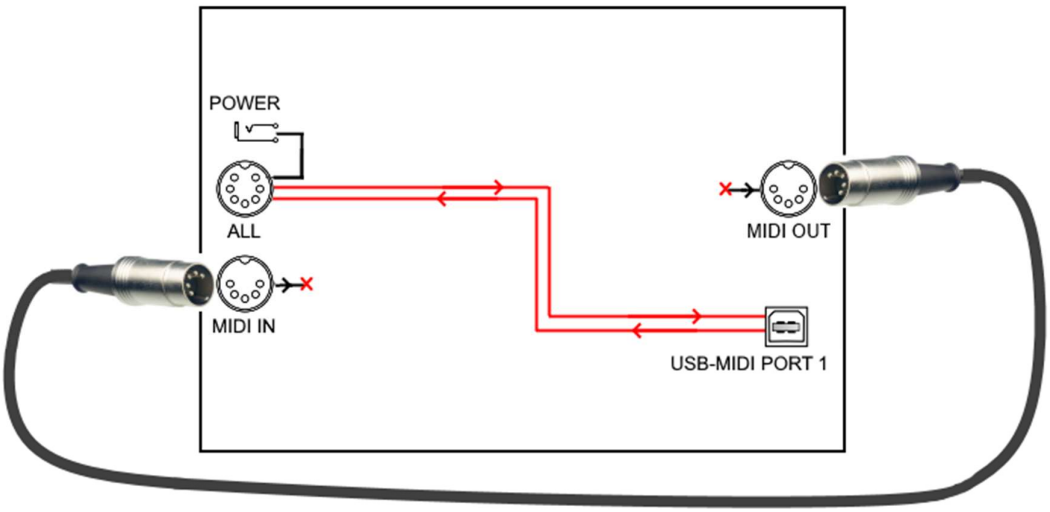
```
Channel 13 NoteOn/NoteOff 0x00-0x09 0x7f  = switch 1-10    press/release
Channel 13 NoteOn/NoteOff 0x0e/0x0f 0x7f  = switch DOWN/UP press/release
Channel 13 ControlChange  0x04/0x07 value = expr.pedal A/B position
```

With the adapted routing above, the FCB1010 switch presses can be simulated by sending these same NoteOn/NoteOff messages to the TinyBox from a keyboard connected to MIDI IN, or even from a computer application, through TinyBox MIDI-USB port 1. It is important to notice that the TinyBox MIDI mapper is still available in this routing. This means that you can easily map the lowest octave of your keyboard to MIDI channel 13, note range 0-15, and this way turn this low region of your keyboard into a remote control for your entire rig. A joystick or other continuous control can be mapped to the expression pedal movement, and activate any of the sweeps in the TinyBox setup.

This special routing is activated by sending the TinyBox command “`UseKeyboardControl`”. This command will typically be part of the “`INIT_TINYBOX`” initialization preset, which is triggered as soon as the TinyBox is powered.

One last important fact: if an incoming MIDI message does not simulate any switch press or expression pedal movement, it is simply forwarded to MIDI OUT. So you can still play all your synths with the keyboard as usual, while reserving that one lower octave for remote control.

A special case : troubleshooting



You can force the TinyBox in “troubleshooting” mode by powering it up with a MIDI cable connected directly from MIDI OUT to MIDI IN connector. In this mode the full TinyBox functionality is omitted, all MIDI messages coming from the FCB1010 are directly forwarded to MIDI-USB port 1, and all messages coming from MIDI-USB port one are directly forwarded to the FCB1010.

This mode helps troubleshooting the FCB1010 connection. If you suspect the FCB1010 has problems sending or receiving MIDI messages you can use a MIDI monitor application and check if a footswitch press or expression pedal movement results in the expected messages being sent. In the opposite direction a MIDI monitor application can send LED control messages to the FCB1010 and you can check if the floorboard reacts to these messages.

Apart from this specific routing, the troubleshooting mode also clears the setup stored in the TinyBox. If for some reason a corrupted setup would make the TinyBox unstable, you can always revert it to factory setting with no setup loaded by connecting a MIDI cable directly between MIDI IN and MIDI OUT during power-up.