USER MANUAL

for

MBBS – MIDI keyboard

firmware version 1.x

www.midi-hardware.com

Roman Sowa 2010

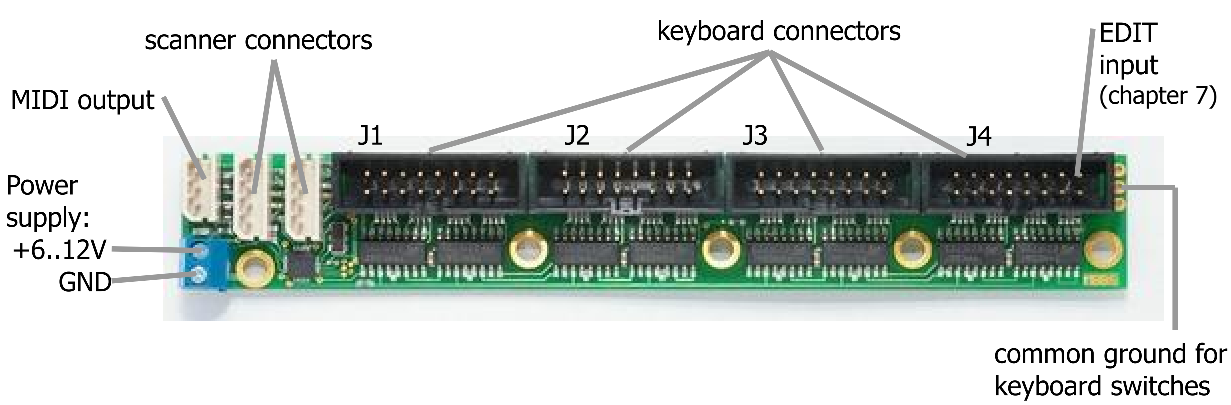
1 Overview

This manual describes the use and functionality of MBBS, a small MIDI controller for one 5-octave kayboard, that can be expanded to work with more keyboards, pots and displays.
Apart from direct inputs for 5-octave keyboard with common bus bar, there are 2 scanner inputs. Each of them can take up to 128 keys if keyboard scanner is connected, or up to 64 potentiometers if pot scanner is used. Up to 5 keyboard scanners can work together but only one input at a time can work with a chain of potentiometer scanners. It is possible to combine keyboard and potentiometer scanners on one input, thus 319 keys and 64 pots.

2 Features

- 63 direct inputs for keyboard switches
- 2 scanner inputs, for keyboards and potentiometers
- user defined split for every keyboard
- independent transposition for all keyboards/splits
- user defined MIDI channel for each keyboard/split and pot
- user defined MIDI event for each pot and keyboard split
- select patch (MIDI Program Change) from keyboard by entering patch number.
- all settings remain after disconnecting power
- DC power supply (6V-12V DC)

3 Layout

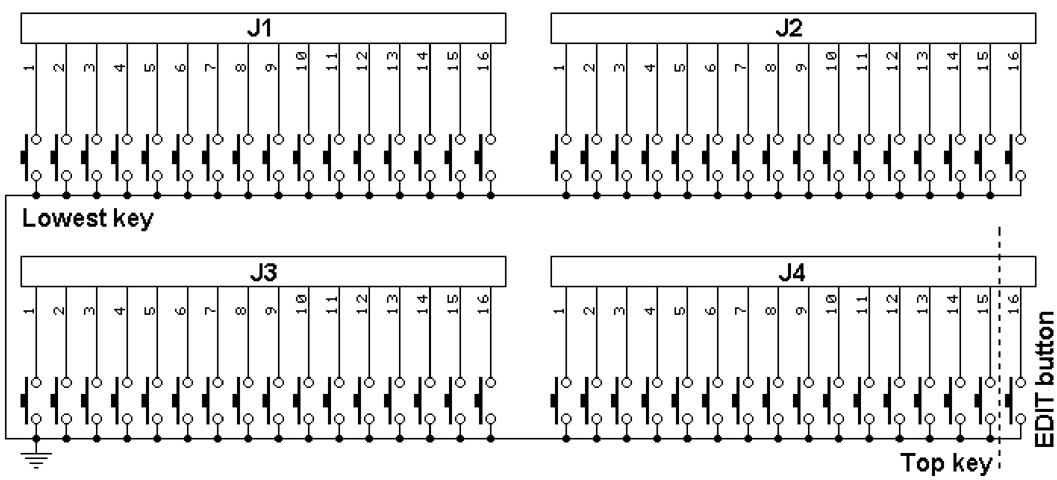


4 Power supply

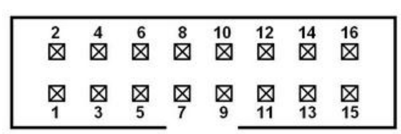
Recommended power supply range is between 6 and 12V DC. It is possible to run this board from lower voltage, but its operation is not guaranteed then. Current consumption depends on the number of attached scanners and does not exceed 10mA, making it suitable for battery operation. This figure doesn't include the load caused by potentiometers if they are used with appropriate scanners. If e.g. those are 20k pots, current consumption will rise by 0.25mA per pot. LCD module takes more current, about 20mA. It is highly recommended to connect no more than one LCD module to MBBS.
The board comes with screw terminals for power. Make sure to use proper polarity of power supply. Connecting power in reverse will not cause any damage, but of course it will only work when proper power polarity is asured.

5 Connecting keyboards

Main keyboard connects to 4 onboard IDC headers according to the schematic below.



Each black connector holds 16 inputs for keyboard switches. The pin numbering within one connector is shown here:



At the edge of the board there are 3 soldering pads (holes) to connect common ground of all swithces in a keyboard. The common may be connected there, or minus (-) supply lead of MBBS, whatever is more convenient.
Additional keyboards can only be connected via additional keyboard scanner. There are several scanners available: for 32, 64, 128 keys, with switches organized in 8x8 matrix and single-rod bus-bar. All available scanners are described in chapter 9. Type of the scanner is determined by keyboard size and the way how switches are organized.

Connection between keyboard scanner and MBBS main board is always the same, regardless of the type of scanner. MIDI settings of those keyboards can be changed by the user after all connections are in place. 8x8 scanner driver can be used if the keyboard has “scanning diode matrix”, that's very simple circuit, made of diodes forming electric XY array of 8 rows and 8 columns. Usually all modern keyboards are equipped with it. In fact it is integral part of the contacts board found beneath the keys. Those kind of keyboards can work directly with DMS-2K scanner. Its advantage is that it can connect 2 such keyboards. Older keyboards, and especially those used in old analog organs, usually don't have such a thing, so in order to use 8x8 scanner, you must build diode matrix yourself, assuming it is possible to separate the contacts into groups of 8 keys. There are also scanners especially designed for keyboards without diodes, where all keys share only one common bus, this is typical keyboard arrangement in all old organ consoles.
Below is a short table showing which scanner can be used with different keyboards:

Keyboard type	DMS-2K	BBS	PEDSCAN
Independent switches, no connections	◇	●	◇
Switches organized in 8x8 diode matrix	●	‡	●
Switches organized in 5x12 and other		‡	
One common rail for all switches		●	

● - can be used directly
◇ - with additional diodes
‡ - rewiring of original contacts circuit required

6 Connecting potentiometers

Pot inputs may be added with use of proper scanners, like POT32X, POT12, POT3 or PEDSCAN-X. They can be used as continuous controllers for things like volume, modulation etc. Usually those inputs would be connected to potentiometers, but it's possible to use them as analog inputs with range of 0..+5V. Applying voltage of 0V creates CC with lowest value, while +5V makes highest possible value of assigned MIDI parameter. Each input in the system has separately assigned MIDI event and channel. This assignment can be easily changed by the user with use of special programing keypad, or lowest 10 keys of any connected keyboard. Every potentiometer scanner is described in chapter 9, with connection schematics.
All potentiometers must be linear taper (not audio) in range 10-50k, preferably 20k.
If there are 2 or more potentiometer scanners in the system, they must be chained, i.e. first board is connected to MBBS, and second one is connected to first POT board, etc. If you use both inputs of MBBS to connect POT boards, they both will share the same settings. That means it would be like having 2 potentiometers for the same MIDI parameter on the same channel. The chain of potentiometer scanners can be

connected to any of the two MBBS inputs, or any daisy-chain input of other connected scanners, but only one. Only keyboard scanners can be connected to all inputs at the same time.
You can mix different kinds of POT scanners in one chain. There can be for example one POT32X, then two POT12, and POT3 at the end. Although this configuration is a bit strange, it will work as expected, giving you total of 59 potentiometer inputs. The total number of active inputs is limited to 64. All additional inputs will be ignored.
Every POT scanner has pins for connecting outer leads of the pots. They are described as +5V and GND. You **must NOT** connect anything except potentiometers, they are only for connection end potentiometer taps.

6.1 *Analog inputs update rate*

All analog inputs of potentiometer scanners (POT32X, POT12, POT3, PEDSCAN-X) translate input voltages, or potentiometer position into MIDI. Actual pot position is updated via MIDI every time it changes. However, the update is not immediate - this is common to any MIDI knob box. The fastest response for potentiometer movement on a single input is about 5ms. It means that when you constantly move the pot, MBBS will issue a MIDI update every 5ms. This is more than enough for most of uses. In some instruments, either hardware, or virtual, some problem may occur when there is a lot of MIDI going on. It is also sometimes desirable to limit MIDI traffic e.g. to minimize the size of recorded MIDI file. It is possible to change this setting using command "#98n" from the keypad, where "n" determines update rate according to the table below. If you don't have the keypad, use the last input of J4, marked as "EDIT input" instead of "#" button, and lowet 10 keyboard inputs as numbers 0-9.
Default factory setting is 18ms.

keypad sequence	#980	#981	#982	#983	**#984**	#985	#986	#987	#988	#989
pot latency	5ms	7ms	9ms	13ms	**18ms**	25ms	35ms	50ms	70ms	0.1s
update rate	200Hz	145Hz	115Hz	80Hz	**55Hz**	40Hz	30Hz	20Hz	15Hz	10Hz

This setting is available for all POT32X, POT12, POT3 and PEDSCAN-X scanners connected to MBBS board. Each board may have different update rates. For example you can set it to 50ms on first POT scanner, and 13ms on second one (assuming you have 2). As usual, to change any settings for particular board, select it first by moving a bit any potentiometer connected to board in question. Then using your keypad enter the code from above table. New settings will be activated and remembered.
More on user settings and general procedure can be found in next chapter.

7 MIDI settings and special functions

All settings are accessible from any keyboard connected to MBBS, either to onboard direct inputs, or with any type of compatible scanner. The last pin (64th) of onoard keyboard scanner (pin 16 of J4) should be connected to a separate momentary switch. This is the entry to EDIT mode and allows changing all settings of entire set. Detailed settings are described later in this chapter. To make the settings more ergonomic and easier, there's optional numeric keypad available, similar to phone keypad. It is connected the same way as any keyboard scanner with 4-wire cable, and works like actual keyboard, meaning it is possible to play notes with it. But the advantage is the „#” key, which duplicates the EDIT function available in mentioned 64th onboard pin.
To change any settings in EDIT mode, you have to enter new value of given parameter. To do so, use lowest 10 keys of the keyboard as numeric entry. Lowest key is digit „0”, while 10th key is digit „9”. As a general rule, any change on a controller (keyboard, or potentiometer) requires selecting this controller first before making change. For example, if you want to change MIDI channel of certain potentiometer, move it a bit, and go into MIDI channel settings mode. Or to change the split point – first play any note on the keyboard to be split, and enter split-point change mode.
In this chapter, describing how to set all parameters, whenever „#” sign is mentioned, it means the "EDIT" key, connected to 64th pin shown in the layout section, or the „#” or "*" key on numeric keypad if one is present in the system. All three have exactly the same function. Numeric entries are provided with the assumption that numeric keypad is used, but the same can be achieved with lowest 10 keys of any keyboard connected to MBBS. It helps to add a sticker over those keys with numbers from 0 to 9 if only musical keyboard is used.
If you also have dedicated LCD module - *MiDisp*, all settings activity is visible. It's much easier then to recognize what stage of programming you are in, and short prompts are displayed according to your selection.

7.1 Transposition of keyboards

Transposition of keyboards connected to MBBS is unlimited, that means any key can generate any note from full MIDI range of over 12 octaves. There are two ways of using it. Typical one is by selecting new middle C position. First you have to select the keyboard you want to edit by playing any note on it. Enter „#” then „1" on the keypad. Now, whatever key you press, it will be the new position of the middle C MIDI note afterwards. You can select new position of middle C note anywhere between 3rd and top key of the keyboard.
Another option is to use lowest 2 keys of the keyboard, or numbers "0" and "1" of the keypad. It doesn't matter if keyboard starts with key C or F or whatever, those are always two lowest keys. The 1st one shifts the keyboard one semitone down with each sequence (#10), the 2nd shifts the keyboard one semitone up (#11).
Of course the keyboard to transpose must be the last one played. This is useful when

you want to shift the keyboard in range not available by the first method, for example in bass pedals.

7.2 MIDI event assignment for keyboards and analog inputs

MIDI event assigned to given potentiometer or keyboard split can be easily changed. To perform this, turn a bit the knob, or play a key on the split you want to assign, and then select the controller type by entering keys # then 2 and then appropriate number from the list that follows. You need to enter 2 or 3 digits for each input controller depending on entered number. To assign another one, again you must turn the pot it a bit, or play the key on another split, and then start from "#2" sequence followed by event type number.
Possible MIDI event codes are from 000 to 149.
Standard setting for a keyboard is "# 2 131" - single notes, and for analog input is Continuous Controller: "# 2 nnn" - where "nnn" is number from 001 to 64. This is factory default.
Possible settings are:

7.2.1 Control Change - #2 CC

whew "CC" is any MIDI Control Change number in range from "000" up to "127". Numbers above 127 are used to generate MIDI events other than Control Change, or turn them into other functions, what is described next.

7.2.2 Pitch Bend - #2128

The pot will work then as pitch bender. If assigned to a keyboard (not recommended), each key will set pitch bender in 1/128 steps across the keyboard. Range can be adjusted with transposition settings.

7.2.3 Program Change - #2129

If assigned to a keyboard, pressing each key will generate MIDI Program Change message with different patch number. Starting number can be adjusted with transposition setting. This is useful for organ emulators, where bank of Program Change buttons can be used to work as pistons (sets of registers). Although this is rather unusual usage for a potentiometer, it will generate MIDI Program Change messages with its every move. Program Change can be also generated from the keypad, by using sequence "# 4 <number>" – this is described in 7.4. Yet another way of selecting Program Change is described in 7.2.8

7.2.4 Channel After Touch - #2130

Turning such pot will cause Channel After Touch messages to be sent out. If assigned to a keyboard, each key will set After Touch in 1/128 steps across the keyboard. Range can be adjusted with transposition settings.

7.2.5 Standard keyboard action – single notes - #2131

Whenever MIDI event 131 is assigned to a keyboard, it works as typical MIDI keyboard, playing MIDI notes. It is also possible to generate notes played in glissando, when this event is assigned to a pot. Select the pot to be edited, enter "# 2 131". This knob becomes then a note generator resembling quantized Theremin. Move the knob and a series of notes will be played. There's only one note played at a time (with velocity set like described later) and it is released just before new note is about to play. Whole knob slow rotation plays 128 notes from entire MIDI range.

7.2.6 Note on only - #2132

This mode is somehow similar to the action described just before, but only "note-on" messages are generated, that means whenever you move this pot, new notes will be played, and they will stay on forever unless proper note-off message will be issued by another means. If assigned to a keyboard, only note-on messages will be sent. It will work like with constantly depressed sustain pedal.

7.2.7 Note off only - #2133

This is like "note-on" mode described before, but instead it sends out only note-off messages. It can be used to mute part of notes already played, or as some kind of panic button – slow full rotation mutes all notes in assigned channel. If assigned to a keyboard, it will send only note-offs, so it may be used to quiet some notes played earlier.

7.2.8 One-touch Patch Recall - #2134

Keyboard in this mode serves as an array of favorite patches buttons. Each key recalls Program Change (or in another words - selects a patch/preset) that was earlier programmed. There's 64 memory locations, so you can use one full 64 key scanner. For example you can program key 1 to send Program Change 37, key 2 as PC#76, key 3 as PC#20 etc. Assigning Program Change numbers to a specific key is described later, in paragraph 7.6 "Programming Patch Recall Buttons".

7.2.9 CC keyboard - #2135

This feature has no effect on a pot, i.e. the pot will generate no MIDI event if it has this feature assigned. In this mode you can use keyboard as toggle switches selecting min/max values in a subset of CCs. All keys have increasing MIDI Continuous Controller assigned. Pressed key sends CC with max value (127), while key release generates the same CC but with minimum value (0). CCs are ordered just like there would be MIDI notes, i.e. typically they start from CC#36 at the lowest key, next key is CC#37 etc. Use transposition settings to set different starting CC.

7.2.10 MIDI channel shift for all controls - #2136

This is usable only with keyboard scanners, because using it with pot needs high precision, as full change takes about 1/8 of a turn. When assigned to a keyboard, first

16 keys work like MIDI channel selector for all controllers. After one of the keys is hit, notes played on all keyboards are played in altered channel. Individual channel settings for every keyboard and potentiometer described in chapter 7.3 work together with this setting. For example if one keyboard was set to channel 3, and you change the channel using this feature to +4 (by hitting 5th key), resulting channel is 7 (3+4). If all controllers are set to channel 1, then all 16 keys assigned to this feature are direct channel selectors from 1 to 16.

7.2.11 Small Transposer - #2137

Select the pot to be edited, enter "# 2 137". Turning such pot will shift all notes played on all connected keyboard scanners by number of semitones determined by pot position. In the middle it gives no shift, and full rotation has range from -4 to +4 semitones. It's most useful when pot is replaced by 9-position switch with 8 resistors of equal value connected between switch leads. Assigning this to a keyboard is also possible, but you cannot reach full range of transposition then.

7.2.12 Big Transposer - #2138

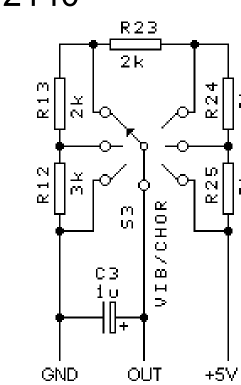
Turning such pot will shift all notes played on all connected keyboard scanners by number of semitones determined by pot position. In the middle it gives no shift, and full rotation has range from -8 to +8 semitones. Assigning this to a keyboard is also possible, but you cannot reach full range of transposition then.

7.2.13 Velocity - #2139

Position of this pot will then determine velocity parameter of all MIDI notes played. If assigned to a keyboard, each key will set velocity of all notes in 1/128 steps across the keyboard. Range can be adjusted with transposition settings. There can be only one such potentiometer for the whole system, and its settings affect all notes played on all keyboards in all channels. It's not intended for initialisation setup during installation, but rather as a way of performance expression during play. To set the velocity once, and always use chosen value, even after power cycle, use parameter #2149. It works exactly like the one described here, but additionally it remembers last position in nonvolatile memory, so it will use the same value after next power-up.

7.2.14 Native Instruments B4 chorus/vibrato - #2140

Turning this pot will be reflected in B4 as "chorus/vibrato" switch move. It has only 6 positions, and appropriate command will be sent from MBBS to B4 every time the potentiometer crosses each threshold representing another vibrato/chorus mode. That's why it is recommended to use small circuit with rotary switch shown on the right for easier B4 usage.



7.2.15 MidiTzer stops control - #2141

When assigned to a keyboard, each key becomes specific stop controller. When a key is pressed, MIDI controller 81 (51 hex) is sent, and when it is released, MIDI controller 80 (50 hex). Value of the controller is determined by the button pressed. This is default way of controlling stops in MidiTzer organ software.

7.2.16 Ahlborn Archive module stops control - #2142

If assigned to a keyboard, each key becomes stop control in Ahlborn Archive organ sound-module. When a key is pressed, MIDI controller 73 (49 hex) is sent, when released, MIDI controller 74 (4A hex). Value of the controller is determined by the button pressed. This is default way of controlling stops in Ahlborn Archive module.

7.2.17 Ahlborn Organs stops control - #2143

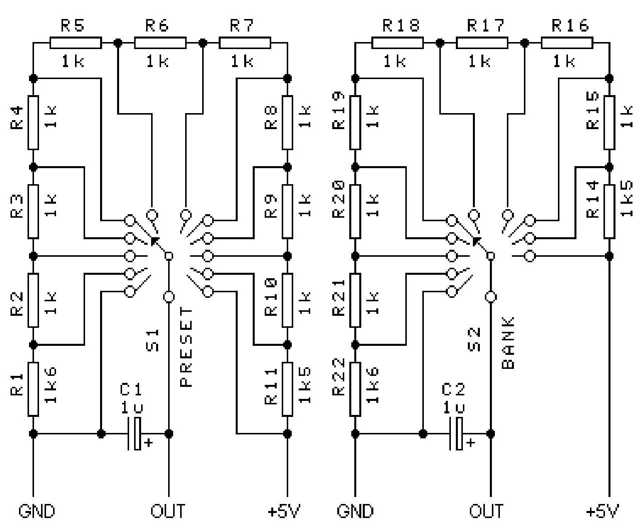
When assigned to a keyboard, each key becomes specific stop controller in Ahlborn Organs. When a key is pressed, MIDI controller 70 (46 hex) is sent with bit 6 of the value set, and when it is released, the same MIDI controller but with bit 6 of the value cleared. Other bits of the value are determined by the button pressed. In another words, pressing the button sends CC 70 with value range 0-63, and releasing a button - CC 70 with value range 64-127. This is default way of controlling stops in Ahlborn Organs.

7.2.18 Program selector - #2144

This is special mode for potentiometer, that together with another pot configured as described in 7.2.19 alows to select one of 120 available MIDI Program Changes (presets) by use of 2 rotary switches. Program selector selects one of 12 Programs within a bank. For better feel, you should rather use 12-position rotary switch. To do so, use schematics below to build appropriate resistor network around the switch.

7.2.19 Bank selector - #2145

This mode works in combination with the one described in 7.2.18. A potentiometer or switch assigned to this mode selects banks of 12 Programs. E.g. if it is in lowest position, you can select any of Program numbers from range 1-12 using Program selector described above. If you change the bank one position higher, Program selector will then select programs from range 13-24, and so on, up until program number 120. To use 10-position rotary switch instead of



potentiometer, you must add appropriate resistor network around the switch leads, as shown in schematics above.
Bank selector and Program selector are useful only if you really want to select programs by rotary switches, for example to match MIDI controller behavior to NI B4 Hammond emulator, where bank selector works exactly like that. In typical case you'd probably use different way of selecting programs, like "#4<number> described in chapter 7.4.

7.2.20 One time velocity setup - #2149

This is similar to velocity settings described in 7.2.13, but in addition, it saves last used velocity settings in nonvolatile memory. This should be used only during installation, when you want to set default velocity of notes after each power up. For expression and frequent usage, control the velocity by assigning #2139.

7.3 MIDI Channel

Channel of each device, like potentiometer or keyboard can be set individually. To change MIDI channel of certain potentiometer select the pot by simply turning it a bit. Then select the MIDI channel with the following sequence: "# 3 <channel number>". The channel number must be in range 1-16. Channels from 2 to 9 require only 1 key stroke, while 1 needs to be entered as 2 digits – namely "01". Channel 10 and above of course need 2 keystrokes too. Then turn another potentiometer and select its channel the same way. To change MIDI channel of the keyboard, first select it by playing a note, and enter mentioned sequence with channel number.

7.4 Program Change

MBBS allows you to send Program Change MIDI messages, or in another words – change patches. Three ways are available. Two were mentioned in Event Assignment chapter (7.2.), where you could program the potentiometer to act like 128-position patch rotary switch, or use keyboard assigned to Program Change to act like array of single touch patch select buttons. To change the patch on selected keyboard directly to specific number, play a note on this keyboard and enter the sequence: "# 4 <program number>". The Program Change MIDI message is sent directly after last digit of entered patch number. This may happen after 2nd or 3rd digit. You only need to enter 2 digit, when the patch number is in range 13-99. Programs lower than 13 require 3 digits, with 0s in front, for example 012, or 003. Obviously, programs with numbers higher than 99 also need 3 digits. The range of Program Change is from 000 to 127.

7.5 Keyboard split

It is possible to split each keyboard into 2 independent parts. The split point can be anywhere on the keyboard, and both parts can work with independently adjusted MIDI channel, type of event and starting note (transposition) or range of other controllers if something else than notes is assigned to a keyboard. Assuming that both scanner connectors of MBBS are populated with DMS-2K dual keyboard scanners, or chains of 2 BBS-1K, it is possible to make a system with 10 splits. To set up the split point, you have to select the keyboard by playing a note on it, and then enter sequence **„# 5"** followed by stroke of the key that you want to be the **top one of the lower part**. Since then, lower part remains on the same channel that was used for whole keyboard, while upper part switches to different settings, which by default is 5 MIDI channels higher. To change MIDI channel, type of event, transposition, or send a Program Change for split part, follow directions described before, regarding non-split keyboard, but now changes are made only in this split part, which was selected by playing a note prior entering the edit mode.

7.6 *Programming Patch Recall Buttons*

Whenever a program/patch/instrument selected from MBBS is often used, it is worth to memorize as fast recall button. There can be 64 such buttons, selected for last played keyboard, split, or turned pot. First you play on a keyboard, and to change a patch, simply select one of the buttons assigned as one-touch program recall described in chapter 7.2.8. This key can be any of 64 keys in any keyboard scanner, or digit 0-9 from numeric keypad. It's nice to have buttons with LEDs showing last selected patch. This can be realized with use of LITSW board, described in scanners section. MBBS will send the patch number in channel of last played keyboard, or last turned pot. To set any patch this way, you have to first select this patch somehow – either by entering „# 4 <patch number>", or by turning „Program Change" knob if one is assigned, or by using one of the keys in keyboard assigned to Program Change event. Then simply press "# 6", and then the key where the patch should be stored. Next time whenever you press this key/button, the MIDI Program Change message will be send, setting the patch that was programmed into that button.

8 MIDI monitor

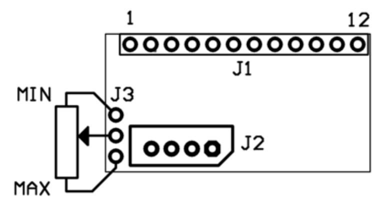
MBBS board have one outstanding feature that helps during installation and troubleshooting. You can activate MIDI monitor if you have special LCD module connected. To activate this function, select #971, and the LCD will turn red, showing "DEBUG MODE". Each activity on any scanner will be shown in the display. Every time you press or release any key, or turn the potentiometer, the display will change, and show last transmitted data. The meaning of displayed values are: "**D**" - internal device number in the system, "**Ch**" - MIDI channel, "**E**" - event number as described in chapter 7.2., "**S**" - MIDI status byte in hex, "**V1**" - value of 1st MIDI data byte in hex, "**V2**" - value of 2nd MIDI data byte in hex. MIDI monitor should not be enabled at all times, it increases MIDI latency and may cause dropped notes during very fast keyboard playing. To turn off the MIDI monitor, simply enter #970 from the keypad.

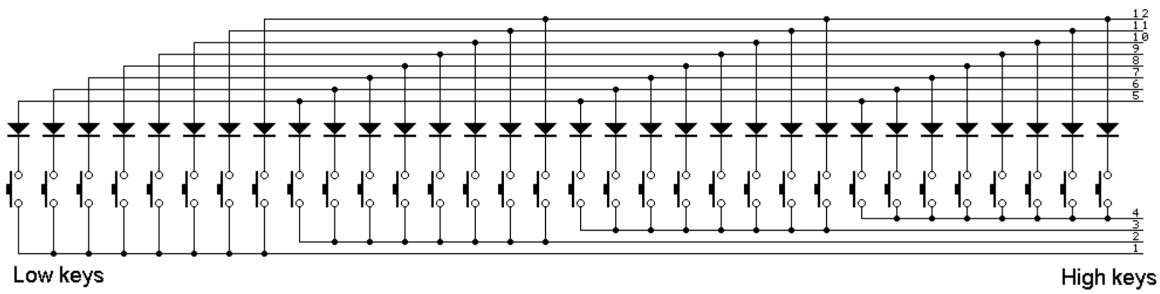
9 Scanners

MBBS board is the merging point for all keyboard and potentiometer scanners. Depending on their sizes, the whole system can cover up to 5 keyboards with 319 keys over single MIDI socket together with 64 potentiometers at the same time.
Currently available sizes are: 32, 64 and 128 keys, the last one has the 128 keys grouped in 2 keyboards. For analog (pots) there are boards with 3, 12 and 32 inputs

9.1 pedal scanner - PEDSCAN-X

Pedal board controller takes care of 32 keys and one analog input, usually all what's needed for pedals with a swell shoe. It's a small board that can be fitted inside pedal board, and it connects to the main board via supplied 4-wire cable from J2 connector. 3 pads labeled in this picture as J3, are for potentiometer. The picture shows how to connect the pot, and also in which pot position you get the maximum, or minimum value of given MIDI parameter controlled by the pot. Go to chapter 6.1 for more info. Keys must be connected in diode matrix exactly like in the schematics below:

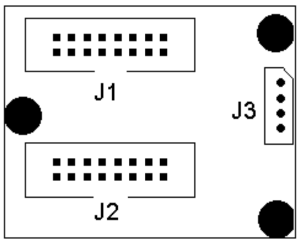




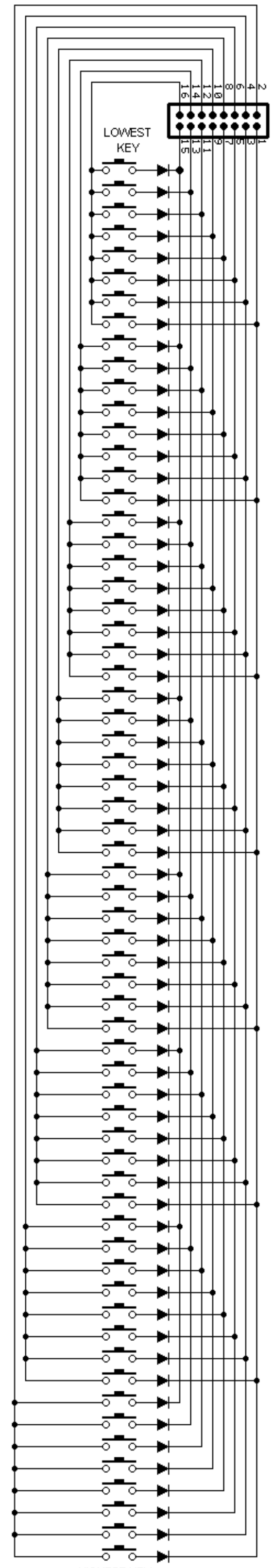
The groups of 8 switches with 1 common lead must be separate. In case of keyboard with 1 common bus bar going through entire keyboard, you have to cut the bar every 8th key.

9.2 dual keyboard scanner - DMS-2K

For keyboards with installed diode matrix ideal solution is to use DMS-2K, adding 128 inputs. This can be described as "diode matrix driver" for 2 keyboards. The layout is shown on the next page. There are two 16-way connectors, that usually are used to connect diode-matrix keyboards. Smaller connector is used to connect it with main controller board - MBBS. The keyboards must have “8x8 scanning diode matrix”, that's very simple circuit, made of diodes forming electric XY matrix. Usually all modern keyboards are equipped with it although sometimes the matrix is organized differently, sometimes it is 5x12 or 6x11. In such case you must rewire the diodes, or order customized DMS-2K.



J1 – connector of 1st group of keys (1 to 64)
J2 – connector of 2nd group of keys (65 to 128)
J3 – connector linking this board with the main board



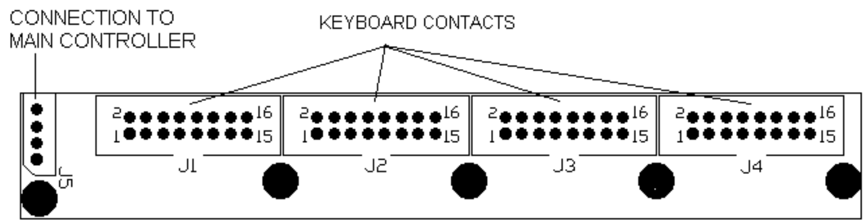
Each 16-way connector covers one keyboard.
Schematic on the right shows example of diode matrix compatible with DMS-2K board. Although this is default matrix layout, DMS-2K can be reconfigured by the user to work with 4 different diode matrix types. All 4 are need to be 8x8 but can have different order of notes. The only requirement is that cathodes are connected to odd pins of DMS-2K connector, and anodes to even pins. Either directly or via key switch. If, after the keyboard is connected, the order of keys is not correct, you can try one of the 4 modes of DMS-2K operation in the following way:

1. play any note on a keyboard connected to DMS-2K
2. enter EDIT mode either by activating 64th input of onboard MBBS scanner, or by pressing "#" keypad button
3. enter "92" from numeric keypad, or onboard scanner (pin 10 and then pin 3 of J1)
4. enter one digit from range 1-4 using keypad, or inputs 2-5 of J1. Each digit turns the scanner into different one of 4 modes of how notes are ordered.
5. play the DMS-2K keyboard to check if it is OK now. If not, start from the beginning, but now select different digit from range 1-4 in step 4.

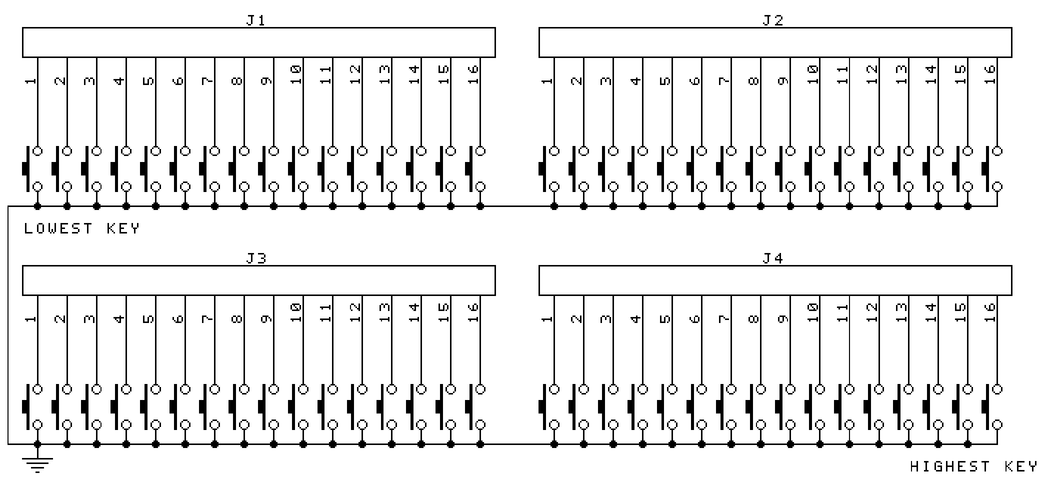
9.3 common contact scanner - BBS64X, BBS32X

There's another family of keyboard scanners, especially suitable for keyboards with single rod used as common bus for all switches in entire keyboard. This one does not use diode matrix, and can be used with almost any type of switch arrangement, it can also be controlled by logic gates. The BBSX boards come in 2 variants. They can work with either positive or negative keying, which means that key pressed is represented by 0V, or +5V, but proper BBS version must be selected when placing order. The board can have less 16-pin connectors available, covering only 32 keys. It's then called BBS32X.

In case of the big, 64-keys version, keyboard is connected to four 16-pin connectors - J1, J2, J3 and J4 shown below. Each of them covers 16 keys.



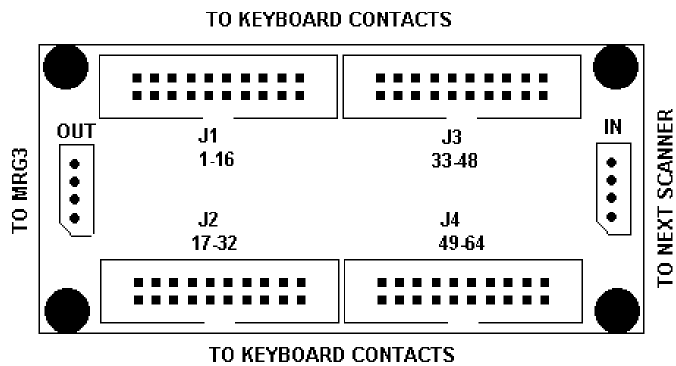
The key contacts can have one common buss bar (with GND), or it can be driven from logic IC outputs. 0V at an input means "key pressed", +5V at input or left open means "key released". The BBS board can be also ordered with reversed logic, i.e. positive voltage at input means "key pressed", 0V - "key released".
Keyboard should be connected to four 16-pin headers according to the diagram below. It's best to use 4 IDC connectors and 16-wire flat computer grade cable. Each cable connects to 16 consecutive keys.



9.4 *common contact chained scanner - BBS-1K*

This one has the same functionality as BBSX boards described in chapter 9.3, but adds a few interesting features. For basic functionality and connections description please refer to chapter 9.3, here are only outlined the differences between those 2 boards.
BBS-1K has daisy-chain input, which allows to pack more scanners into the system. For example you can use 4 BBS-1K boards, but only 2 BBSX boards. It also has a bit different connector layout, but all 4 connectors for ribbon cable are the same. In contrary to BBSX, this one comes always in 64-inputs version, there's no smaller one. Each black connector covers 16 keys, and little sticker on the board shows what range of inputs is assigned to each connector. This is also explained in the following diagram.

BBS-1K must be connected towards the MBBS with 4-way connector indicated above as "OUT". The connector marked "IN" is for the purpose of adding next optional scanner. There can be only 2 keyboard scanners (BBS-1K or other) connected in one chain. If third keyboard scanner is used on one input, it will behave like paralleled inputs of 2nd BBS-1K in the chain. Hence the limit of 5 keyboards.

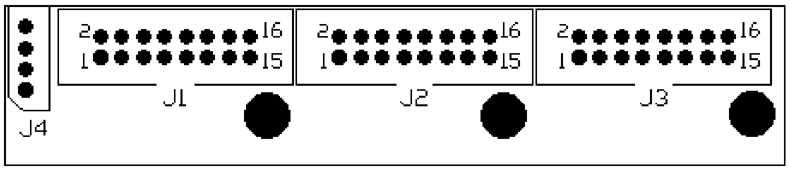


9.5 LITSW - button scanner with LED drivers

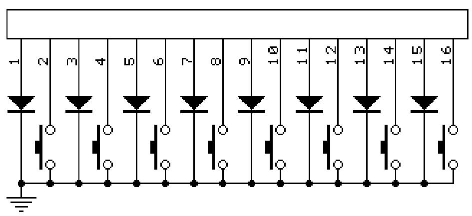
LITSW is the controller for lighted switches. It has 24 inputs for contacts, and 24 outputs for LEDs associated with them. Currently available modes of operation for this board are:

1. independent registers, where each button hit turns the LED on and off, and proper MIDI message is generated according to MBBS setting for that scanner. If this is MIDI note, a note-on is generated at LED turn-on and note-off when LED turns off. If it is Program Change, only one event is generated, when the LED turns on
2. dependent choice, where only one LED-button can be active (lights up). Pressing any other button makes all previously lighted ones to turn off, and the one pressed lights up now. If assigned to notes, only MIDI note-on is generated, so this mode is more suitable for use with Program Change
3. keyboard mode, where button action can be treated as typical keyboard scanner. So all kinds of MIDI events applicable to a keyboard are also possible here
4. potentiometer mode, where LITSW simulates potentiometer scanner. Buttons don't act like in keyboard scanner, but simulate min/max position of a potentiometer. This allows to assign MIDI events and channels separately to each of the buttons.
5. bank/select, useful for preset selectors. In this mode LITSW must be split in 2 parts. One is serving as bank select, the other one - as program select. If the split is made at 10th key, lower split is representing units of given Program number, and the upper one - tens of this number. Of course it makes more sense when MIDI event 129 is programmed in MBBS (that's Program Change)

It is possible to select more than one mode at once, although not always it makes sense. For example modes 1-2, as well as 3-4 are mutually exclusive, but you can set for example modes 1, 3 and 5 together.



J1 - LED-buttons 1-8
J2 - LED buttons 9-16
J3 - LED buttons 17-24
J4 - connector to MBBS



The pins of each LED-button connector are interlaced for easier installation. Odd pins are LED outputs, and even pins are button inputs in the following manner: pin 1 - LED 1, pin 2 - button 1, pin 3 - LED 2, pin 4 - button 2, etc. This is shown in the schematic on the left. Each 16-pin connector's layout is the same.

9.6 ***LITSW setup***

In order to change the way how LITSW operates, it MUST be connected directly to one of scanner inputs of MBBS, and not via any daisy-chaining scanners, like POT32, POT12, display, or BBS-1K. After programming is done, it can be placed anywhere in the system if it's more feasible, and only after that you can assign channels and events to the buttons. Do not try to change MIDI events or channels for LITSW until it is in its final position with regards to daisy chained modules.
With factory settings it generates note-on and note-off messages with momentary buttons and LEDs, most useful as register control in organ emulator. But it can be converted into a few variations, finding its way toward other non-typical uses. Possible controls are:

9.6.1 Split point - #905

This is different kind of split than the one described in paragraph 7.5 and is independent of that one. So you can use both kinds of split at one time, and they can be at different points. The kind described here controls only button's behavior, and not actual MIDI data transmitted by MBBS to MIDI OUT socket. Each split can work in different mode, with independent or dependent buttons, generating CCs or notes. To have different MIDI events or channels in the splits you have to also split it logically, using procedure described in 7.5.

9.6.2 Independent mode - #908

In independent mode, all buttons work without interactions to each other. If a button is pressed, associated LED lights, and note-on is sent by MBBS (only if it is configured to send notes on this input - sequence #2131). Another touch of this button and LED turns off and note-off is sent. To set this mode on a split, you have to select it first by pressing one of the buttons in that split.

9.6.3 Dependent mode - #909

In dependent mode, there can be only one LED active at a time. If you press another button, it will light up and any one that was previously lit, will now turn off. In this mode only note-on messages are generated, there is no note-off. This is most useful for selecting presets on MIDI instrument, like described in chapters 7.2.3., and 7.2.8. To set this mode on a split, you have to select it first by pressing one of the buttons in that split.

9.6.4 Bank/preset select - #910

It is possible to split the LITSW board into 2 parts by keypad sequence #5 described in MIDI settings paragraph. If you type #910 on the keypad, LITSW board will then work as patch selector with separate row of "banks" buttons and "presets" buttons. If the split is made after 10 keys, lowest 10 buttons will act like preset selector within a bank, and all buttons above 10 - as bank selectors. Although it is possible to use this mode with independent buttons, described in 9.6.2, most obvious use is together with dependent butons mode described in 9.6.3. To use it as Program Change selector, you should program appropriate MIDI event on this input, which for Program Change is - #2129. The split point doesn't have to be after 10 buttons. It can be e.g. after 6 buttons. Then first bank will select patches from 1 to 6, second bank from 7 to 12 etc.

9.6.5 Disable bank/select mode - #911

To disable bank/select mode described above, you must type #911 sequence on the keypad. It then returns to normal mode, where each button has the same weight, there's no split for banks or presets.

9.6.6 Note keyboard scanner mode - #912

This is normal way of operation for this scanner. When a button is pressed, MBBS can then generate MIDI notes, or any other MIDI event that can be programmed to a keyboard scanner. This is the oposite to CC mode described next. To set this mode on a split, you have to select it first by pressing one of the buttons in that split.

9.6.7 CC scanner mode - #913

In this mode LITSW behaves like potentiometer scanner. When button is pressed, it simulates turning a potentiometer to a maximum, and when the button is deactivated (LED is turned off) it simulates a potentiometer turned into minimum. So it works like 24 potentiometers with only 2 valid positions: min and max. This is useful for switching parameters in virtual instruments, e.g. Hammond emulators. For each button you can then setup individual MIDI event (like CC or anything else) and MIDI channel. To set this mode on a split, you have to select it first by pressing one of the buttons in that split.

9.6.8 Left split blink - #916

When you type this sequence on a keypad, LITSW will blink all LEDs in lower split of

LITSW. This can be used to determine where the split point is, and where are the LEDs connected.

9.6.9 Right split blink - #917

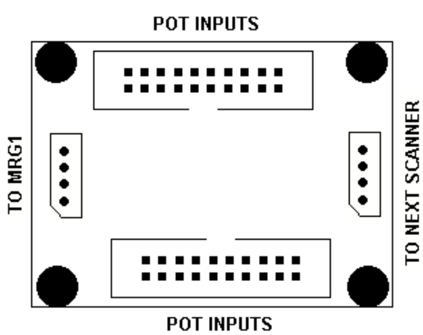
When you type this sequence on a keypad, LITSW will blink all LEDs in upper split of LITSW. This can be used to determine where the split point is, and where are the LEDs connected.

9.6.10 All LEDs blink - #918

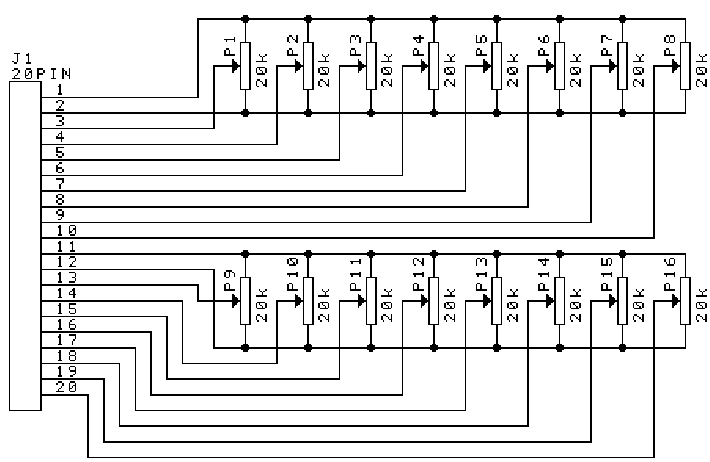
When you type this sequence on a keypad, whole LITSW will blink all LEDs. This can be used to see if every LED is connected properly and where they are located.

9.7 POT32X - potentiometer scanner

POT32X adds 32 potentiometer to MIDI system built around MBBS. It has two 20-pin black connectors for potentiometers, and 2 small 4-pin sockets, typical for all midi-hardware.com MIDI board. One of those is used to connect the potentiometer scanner to MBBS. The other one can be used to connect another scanner, be it keyboard or another potentiometer. You can chain only up to 2 POT32X scanners, giving you maximum of 64 potentiometer inputs for the whole system.
Here's layout of the POT32X potentiometer scanner:



Notice the orientation of angled corners of the small connectors in the drawing to properly connect it. 4-pin connector on the left side must be connected to MBBS, either directly, or via another scanner. Connector on the right side can be used to add another scanner, e.g. next POT32X, or keypad. If you reverse connections between those 2 sockets, the board will not work, and potentiometer movement will not result in any MIDI activity. However this does not cause any damage to the POT32X board, or MBBS.

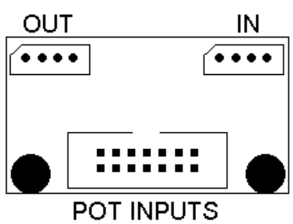


Two bigger black connectors are the potentiometer inputs. Their connection is shown on the right. Only one connector shown, 2nd one is identical.
Each sockets contains 2 pairs of contacts for potentiometer common rails, namely GND and +5V. This is only for convenience of installation, and one pair can be omitted if needed. The cable

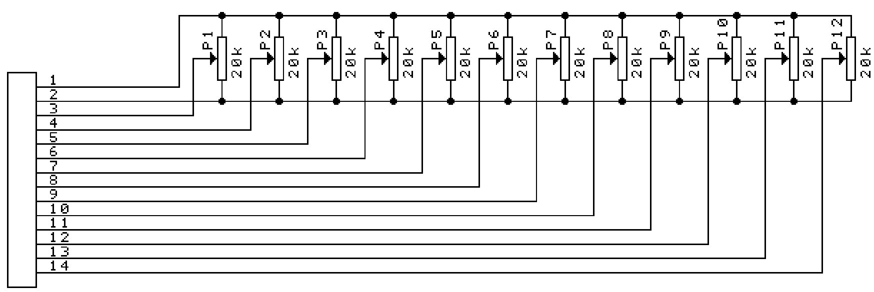
from each IDC20 connector can be split in half and routed to a group of 8 potentiometers. If you don't use all potentiometers you may want to connect unused inputs to VCC (pins 1 and 11). **Keeping any, or all inputs open is quite safe, but in some cases may result in unexpected MIDI messages.** That's especially possible with fast response settings described in chapter 6.1. With fast update rate and inputs left open, moving one potentiometer may trigger unexpected other MIDI messages. In such cases it is enough to connect all unused inputs to VCC (pin 1 or 11 of each black connector) and spurious messages will never appear again.

9.8 POT12 - potentiometer scanner

POT12 is a smaller version of POT32X. All general guidelines of POT32X apply, it differs only in the number of pot inputs, refer to chapter 9.7 for more details.
Here's layout of the POT12 potentiometer scanner:

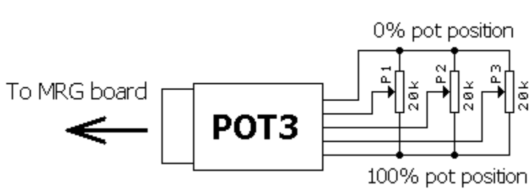


4-pin connector on the left side marked "OUT" must be connected to MBBS, either directly, or via another scanner. Connector on the right side marked "IN" can be used to add another scanner, e.g. next POT12, or keypad etc. The bigger black connector holds potentiometer inputs. Their connection is shown below:



9.9 POT3 - tiny potentiometer scanner

POT3 is the smallest board of all available at midi-hardware.com. It can be used when you need to MIDI convert only one to three swell pedals etc. In contrary to other pot scanners, it has no daisy-chain input, hence it must always be the last scanner in a chain. Below is connection diagram, looking at the components side:



There's 5 pins on one side of this micro board. They can be soldered to bigger board, or wires, or plug can be used.