

# **MIDI User Settings & Scanners GUIDE**

features and user settings of the following MIDI controllers:

- MRG2 v. 2.14
- MBBS v. 3.4
- MIDI128 v. 4.1

and use of the following scanners:

- DMS-2K
- BBS-1K
- PDS
- BBSP
- BBS24
- POT12
- LITSW
- MiDisp
- miniLCD
- KEYPAD

**[www.midi-hardware.com](http://www.midi-hardware.com)**

Roman Sowa 2016

# Table of contents

1 Introduction to user settings.....	4
2 Keyboards.....	5
2.1 Diode Matrix keyboards & order of notes.....	5
2.2 Common Rail keyboards.....	6
2.3 Keyboard contacts de-bounce - #99.....	6
3 Potentiometers & Control Voltages.....	7
3.1 Analog inputs update rate - #98.....	8
3.2 Resolution of analog inputs - #96.....	9
4 MIDI settings and advanced functions.....	9
4.1 Transposition (starting note).....	10
4.1.1 Middle "C" select - #1.....	10
4.1.2 Transpose +/- buttons.....	10
4.1.3 Transpose +/- command - #10, #11.....	11
4.1.4 Global Transpose.....	11
4.2 MIDI events for keyboards, analog inputs, and special switch inputs.....	12
4.2.1 Control Change - #2 CC.....	12
4.2.2 Pitch Bend - #2128.....	12
4.2.3 Program Change - #2129.....	12
4.2.4 Channel After Touch - #2130.....	12
4.2.5 Standard keyboard action – single notes - #2131.....	13
4.2.6 Note-on only - #2132.....	13
4.2.7 Note-off only - #2133.....	13
4.2.8 One-touch Patch Recall - #2134.....	13
4.2.9 CC keyboard - #2135.....	13
4.2.10 MIDI channel shift for all controls - #2136.....	14
4.2.11 Small Transposer - #2137.....	14
4.2.12 Big Transposer - #2138.....	14
4.2.13 Velocity - #2139.....	14
4.2.14 Native Instruments B4 chorus/vibrato - #2140.....	14
4.2.15 MidiTzer stops control - #2141.....	15
4.2.16 Ahlborn Archive module stops control - #2142.....	15
4.2.17 Ahlborn Organs stops control - #2143.....	15
4.2.18 Program selector - #2144.....	15
4.2.19 Bank selector - #2145.....	15
4.2.20 Ahlborn Common Functions - #2146.....	16
4.2.21 3-note chord collect - #2147.....	16
4.2.22 Dual note (layered sound) - #2148.....	16
4.2.23 One time velocity setup - #2149.....	16
4.2.24 Single keyboard transpose buttons - #2150 to #2153.....	17
4.2.25 Channel set buttons - #2154 & #2155.....	17
4.2.26 Global Transpose buttons - #2156 to #2159.....	17
4.2.27 MIDI notes for stops control - #2160.....	17
4.2.28 Reversed notes action - #2161.....	18
4.2.29 Program up/down - #2162 & #2163.....	18
4.2.30 Octave "-1/0/+1" select switch - #2164.....	18
4.2.31 Per-keyboard channel switch - #2165.....	18

4.3 MIDI Channel - #3.....	19
4.4 Program Change - #4.....	20
4.5 Keyboard split - #5.....	20
4.6 Programming Patch Recall Buttons - #6.....	21
5 Utilities.....	22
5.1 MIDI merge input - #972 (on/off).....	22
5.2 MIDI monitor - #971.....	22
5.3 All settings Backup / Restore - #978.....	23
5.4 Factory defaults - #979.....	23
5.5 MIDI System-Exclusive.....	24
5.5.1 Commands description.....	24
5.5.2 Starting note command - 01.....	24
5.5.3 Channel and Event - 02.....	25
5.5.4 Split Point - 05.....	25
5.5.5 Advanced Settings - 09.....	25
6 Scanners.....	28
6.1 PDS - diode matrix pedal scanner.....	28
6.2 DMS-2K dual keyboard matrix scanner.....	29
6.3 BBSP - common rail pedals scanner.....	30
6.4 BBS-1K common rail keyboard scanner.....	31
6.5 BBS24 - contacts & pots little scanner.....	31
6.5.1 Order of notes - #921 and #922.....	32
6.5.2 Contact inputs polarity - #927 and #928.....	32
6.5.3 Contact debounce - #990 to #999.....	32
6.5.4 Reset to defaults - #929.....	33
6.6 LITSW - button scanner with LED drivers.....	33
6.6.1 Split point - #905.....	35
6.6.2 Device ID shift - #906.....	35
6.6.3 Independent mode - #908.....	35
6.6.4 Dependent mode - #909.....	35
6.6.5 Bank/preset select - #910.....	36
6.6.6 Disable bank/select and Cancel mode - #911.....	36
6.6.7 Contact on/off (keyboard scanner) mode - #912.....	36
6.6.8 CC value (pot scanner) mode - #913.....	36
6.6.9 Left split blink - #916.....	36
6.6.10 Right split blink - #917.....	37
6.6.11 All LEDs blink - #918.....	37
6.6.12 Turn off all LEDs - #919.....	37
6.6.13 Cancel Button mode - #922.....	37
6.6.14 LEDs with common cathode (default) - #927.....	37
6.6.15 LEDs with common anode (reversed) - #928.....	37
6.6.16 Reset to defaults - #929.....	37
6.7 POT12 - potentiometers scanner.....	37
6.8 MiDisp & miniLCD - 2x16 characters displays.....	38
6.8.1 Momentary button input - #908.....	39
6.8.2 Toggle switch input - #909.....	40

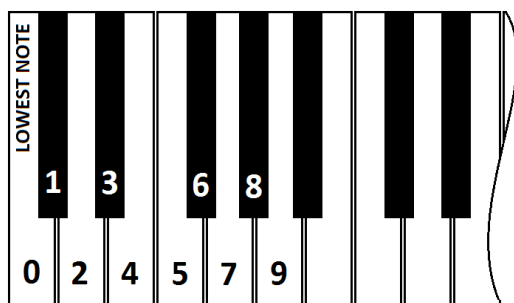
6.8.3 Display ID set - #914.....	40
6.8.4 Display blink - #918.....	40
6.8.5 Turn off display - #919.....	40
6.8.6 Display ID show - #920.....	40
6.9 KEYPAD - MIDI & setup numeric entry.....	40

## 1 Introduction to user settings

The MIDI controller set you have purchased is by default programmed to typical MIDI settings, so if you plug contact scanners, they will work as music keyboards, with typical starting note (MIDI note 36) and each keyboard in different MIDI channel. All potentiometers work as Continuous Controllers from number 1 up, all in channel 1. However, every MIDI settings, from basic things like channel and starting note, to the advanced ones that affect special features of connected scanners, are accessible the same way. The easiest one is with programming KEYPAD. To make it even easier, take advantage of visual feedback, when miniLCD, or MiDisp modules are connected to any spare scanner input (can also be chained with other scanners).

During normal performance the KEYPAD works like any musical keyboard - it plays notes. But after pressing # button it enters into EDIT mode, and becomes the entry for commands consisting of 2 to 6 digits. **Every setting applies to the keyboard/pot/input, that was last used prior entering EDIT mode.** So for example to change MIDI channel of a keyboard, you must play it first, and then without touching anything else - press the # button. No MIDI will come out until all expected digits are entered. Likewise, to change CC of a potentiometer, turn it a bit, then press #, followed by proper numbers. The procedure is exactly the same for all keyboards and pots, either for control inputs on master board (like MBBS), or connected scanners (like BBS-2K or POT12), wherever the board is connected. All available commands are described later in this manual. You may press 0 right after # to go back to normal mode if EDIT was initiated accidentally.

If there is no KEYPAD, you can still do all of this with any contact/keyboard scanner. The # button function is also available at EDIT pins on master controller board - refer to master controller hook-up manual for its location. The digits for user settings are at the lowest 10 contact inputs (notes) of any keyboard scanner either internal (MBBS, MIDI128) or external (BBS-1K, BBSP, PDS, DMS-2K, BBS24) starting with 0 up until 9 like this:



Any MIDI keyboard connected to MIDI IN of MRG2 or MIDI128 can also be used for entering setups like this, but its lowest key

must be MIDI note 36, standard for 5-octave keyboard.

Alternatively in MRG2 and MIDI128 you can change all settings via MIDI System Exclusive messages. Do not attempt to use this method if you're not familiar with Sys-Ex usage in general, as it may lead to unexpected behavior of the board.

## 2 Keyboards

Keyboard switches need a contact scanner, either internal as in MBBS and MIDI128, or with additional board. There are several scanners available: for 24, 32, 64, 128 keys, with switches organized in 8x8 matrix and single-rod bus-bar (common rail). All are described later in chapter 6. Type of the scanner required for particular application is determined by keyboard size and the way how the switches are organized. Table below shows which scanner type can be used with different keyboards:

Keyboard type	DMS-2K	MIDI128	MBBS BBSxx	PDS
Independent switches, no connections	◇	◇	●	◇
Switches organized in 8x8 diode matrix	●	●	‡	●
Switches organized in 8x12 (1 com/oct.)	●	‡	‡	
One common rail for all switches	‡◇	‡◇	●	‡◇

● - can be used directly

◇ - with additional diodes

‡ - requires rewiring of existing contacts

### 2.1 Diode Matrix keyboards & order of notes

MIDI128 and DMS-2K can be used if the keyboard utilizes "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 work like that and it is integrated with the contacts on one board beneath the keys. Older keyboards, and especially those used in old analog organs, use one common rail for entire keyboard, 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. But in such case it is generally easier to use common rail scanner (see 2.2.).

In chapter 6.2. describing DMS-2K you can find schematics of an example 8x8 diode matrix. But there are many ways how you can put diodes in 8x8 matrix and *MIDI-hardware* boards are ready for that. **MIDI128 automatically detects** the matrix orientation, as well as diodes polarity so is capable to work with 16 different diode matrix layouts. Additionally 2 modes for 12x8 are possible with DMS-2K for keyboards wired in octaves, in 12x6 diode matrix. Connecting different kind of matrix to DMS-2K results in wrong order of notes played, or keyboard being completely silent. In latter case you must use the other black 16-way plug of the FatBreak if you have one, or

rotate existing 16-way plug by 180 deg. If notes play, but order of notes is not correct you can change user settings to different matrix mode.

**To fix wrong order of notes** in DMS-2K, play any note in the "bad" keyboard, then enter EDIT mode by shorting EDIT pins for half a second (their location is described in Master Controller's manual), and using any other keyboard with correct order of notes, play at lowest octave note A, then D, followed by only one of the 4 keys from C# to E, assuming this keyboard starts with C. After that check if order is correct now. You can try 4 different note orders, each time playing different 3rd note from mentioned range. This is much easier if you have the KEYPAD connected: play the bad keyboard, enter #921, play again to check, if not OK enter #922, check again, then try #923 or #924. To make it work in 12x8 mode use command #925 or #926. This can only be done with DMS-2K. MIDI128 selects proper order of notes automatically and does not recognize any #92x commands.

## **2.2 Common Rail keyboards**

Keyboards with every contact shorting to a single rail are very easy to wire, but require a lot of wires to solder. That's the main reason why they disappeared in modern instruments. Each contact goes to individual input of bus bar scanner (MBBS, BBS-1K, BBSP, BBS24), while the common must be connected to the ground of accompanying MIDI board, which is taking care of that particular keyboard or set of switches. Use flat terminal with supplied crimp terminal where available: in BBS-1K, BBSP and MBBS, or to mounting 3.5mm hole on the board. The wires connecting common rails should be as short as possible and preferably thicker, AWG 22 at least. Try to avoid loops and windings in ground wire, place it in straight lines wherever possible. The quality of this connection will greatly influence reliability of entire instrument.

Not used inputs should be left open, **do not** short them to ground or power supply, and what's even more important - do not connect any wires there.

## **2.3 Keyboard contacts de-bounce - #99**

Every kind of contact bounces sometimes. The bouncing phenomenon is a short burst of contact closures and openings, causing instable signals for detecting key on/off state. To prevent the controller from sending MIDI notes on every bounce, a special technique is performed, called debouncing. All boards from midi-hardware.com perform contact debouncing, but MBBS and MIDI128 can also adjust the strength of this process. In MBBS default debouncing is set to 20ms, and you can change it to one of 10 steps from 0 (good for hall sensors) to about 40ms (for rusty, 50-year old, never serviced contacts). Debouncing in MIDI128 is slightly shorter, maximum 30ms. If there are no bouncing effects apparent in your keyboard, there's no need for long

debouncing, and it can be shortened to make the keyboard play more naturally. But with old corroded wire contacts it may be necessary to set the debouncing to a higher level to prevent bursts of notes whenever a key goes up or down.

By any means, debouncing time does not affect note's attack response (latency) which is usually in range of 1-3ms depending on scanner type.

To change debouncing time, you must play any note on the keyboard connected to internal inputs of MBBS or MIDI128 (this is not available for any other scanners so far) and use command #99x, where 3rd digit determines debouncing level. 0 means no debouncing, while 9 is maximum debouncing useful for on/off parametric switches rather than note keyboards. If you have dedicated LCD module - MiDisp, or miniLCD - this setting is guided with prompts after each digit.

### 3 Potentiometers & Control Voltages

Pot inputs are available in master controllers – MBBS and MIDI128, but with use of pot-capable scanners (POT12, PDS, BBS24, BBSP) you can have more of them in any setup. They can be used as continuous controllers for things like volume, modulation etc. Not only potentiometers can be connected there, it's possible to use them as analog inputs for external CV with range of 0..+5V. Applying voltage of 0V causes generation of CC with lowest value, while +5V makes highest possible value of assigned MIDI parameter. Also, for parameters with only on/off state like Sustain, or transpose steps, potentiometer inputs can accept switches. If a switch is closed to GND, value 0 is sent via MIDI, and when the switch opens, value 127. So obviously you should use normally open, or toggle switches.

Each input in the system has separately assigned MIDI event and channel. This assignment can be changed at any time with use of programing keypad, or lowest 10 keys of any keyboard (internal, on scanners, or from MIDI IN), and also via MIDI Sys-Ex. Every potentiometer scanner is described in chapter 6, 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, so first board is connected to Master Controller, and second one is connected to first POT board, etc. If you use 2 inputs of master controller to connect 2 POT boards, they both will share the same settings (channels and events). 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 straight to any one of the master's inputs, or any daisy-chain input of other connected scanners (BBS-1K, LCD), but only one. You can mix different kinds of POT scanners in one chain. There can be for example two POT12, and BBSP at the end. The **total number** of potentiometer inputs outside master controller should **not exceed 64**. All additional inputs will be ignored. Hence there is maximum 72 pots with MIDI128, 68 for MBBS. Every POT scanner has pins for connecting outer leads of the pots. They are described as +5V and GND. You **must not** connect any power to those pins, they are only for connecting potentiometer's end taps.

All cables leading from POT scanners to potentiometers should be either shielded or very short (no more than 30cm). Using long non-shielded cable will generate spurious MIDI messages when unexpected. Low cost microphone cable is good enough for preventing this. Unused POT inputs may be left open, but it's better to connect them to one of the end-taps pins described above. This will ensure no spurious MIDI will be generated under no circumstances from those inputs.

### 3.1 ***Analog inputs update rate - #98***

All analog inputs of MIDI boards translate input voltages, or potentiometer position into MIDI. Actual pot position is updated via MIDI every time it changes. This update is however not immediate - as in any MIDI knob box available today. The fastest response for potentiometer movement on a single input is less than 5ms. It means that when you constantly move the pot, MIDI parameter is updated every 5ms (200 times per second). This is more than enough for most of uses. In some instruments, either hardware, or virtual, some problem may occur when there is heavy MIDI traffic. It is also sometimes desirable to limit MIDI messages for any reason. It is possible to change this setting using command "#98n" from the keypad, where "n" determines update rate according to the following table. Default setting is 18ms.

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

Each board (master and scanners) can work with different update rates, but all inputs in one physical board share the same one. 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. With LCD module connected this procedure is clearly indicated after each digit. New settings will be activated and remembered.

The update rate can also be adjusted via MIDI System-Exclusive.

### **3.2 Resolution of analog inputs - #96**

Usually all MIDI parameters have 128 possible levels, determined by 7 bit nature of MIDI standard. In some cases it may be useful to reduce the number of possible levels, or in digital language - number of resolution bits. For example if you want to use MIDI channel selector, described in chapter 4.2.10, it's better to reduce resolution to 4 bits, and have only 16 levels over full pot's rotation. In some software synthesizers you can select parameters using only a fraction of the full CC range. Also, in organ emulators it's sometimes desirable to have only a few steps in full pedal travel if it's used for example as crescendo.

Bit resolution is user adjustable in a similar way as update rate described earlier, by using command **#96x**, where "x" is desired bit resolution. For example #967 makes the pot input 7-bits wide (default 128 levels), while #963 sets it to 3-bit resolution, allowing only 8 levels (and 8 possible MIDI messages) over full pot rotation. It's adjustable per pot, so you can set every analog input in the system individually. To change resolution of the input, turn the potentiometer connected to it, and enter #96 from the keypad, followed by number of required bits. If this potentiometer was previously set to very low resolution, like 1 or 2 bits, you have to turn the pot almost full rotation to select it for EDIT operations. To select a pot for any kind of configuration (event, channel, resolution) it must first make some action, and when you have only so little levels, it takes much more rotation to cross next valid level and indicate to master board "hi, I'm the last used pot, any setting changes apply to me now!"

## **4 MIDI settings and advanced functions**

All settings are accessible from on-board keyboard inputs (MBBS and MIDI128) or a keyboard connected with any type of contacts scanner. There is dedicated pin on every master board, that may be connected to momentary switch. This is the entry to EDIT mode and allows changing all settings of entire set. Detailed procedures for all settings are described later in this chapter. To make the settings more ergonomic and easier, there's numeric keypad available, similar to phone keypad. It is connected the same way as any keyboard scanner with 4-wire cable, and also works like a keyboard, meaning it is possible to play notes with it. One of advantages is the „#“ key, which starts the EDIT mode, just like a button connected to on-board header.

To change any settings in EDIT mode, you must 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 10<sup>th</sup> key is digit „9“. This becomes obvious with mentioned numeric keypad. As a general rule, any change to 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 enough to generate some action (see 3.2.), 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 pins shown in the layout section of Master Controller manual, or the „#“ key in numeric keypad. Both have exactly the same function. For simplicity, numeric entries are listed with the assumption that numeric keypad is used. It helps to add a sticker over lowest 10 keys with numbers from 0 to 9 if only musical keyboard is used.

If you also have dedicated LCD module - *MiDisp*, or *miniLCD*, all settings activity is clearly visible. It's much easier then to follow what stage of programming you are in, as short prompts are displayed according to your selection.

## **4.1 Transposition (starting note)**

Transposition of any keyboard is unlimited, that means any key can generate any note from full MIDI range of more than 10 octaves. Each keyboard, or each split in every keyboard scanner, can be individually set. Change is always performed for the last used controller prior entering edit mode. There are three ways of transposing individual keyboard/split, and also there is global transposition affecting all notes, including the ones received via MIDI IN. If LCD module is available, transposition is displayed after each change.

### **4.1.1 Middle "C" select - #1**

This one is usable when you want to quickly align your keyboard with actual MIDI notes. First select the keyboard to edit by playing any note in it. Enter „#“ followed by „1“ on the keypad. Now, whatever key you press, it will become the new position of the middle C MIDI note afterwards - MIDI note number 60. You can select new position of middle C note anywhere between 3rd and top key of the keyboard because lowest 2 keys are used by another transpose method. If a keyboard is split, and you want to transpose notes in upper split, still you can use entire keyboard (except 2 lowest keys) for selecting middle "C", so it can actually be outside the split range.

### **4.1.2 Transpose +/- buttons**

Second option is to use transposition buttons. There can be 4 of them, two for semitone steps up/down, and two for octave steps up/down. Such buttons can be assigned as special events for potentiometer inputs (or special switch inputs in MBBS and MIDI128). The event numbers for this feature are:

- one semitone up: 150
- one semitone down: 151
- one octave up: 152
- one octave down: 153

Pressing such button (shorting input to ground) will affect last played keyboard, and this setting will be memorized in non volatile memory. Here you don't have to enter entire command every time, instead, setting is accessible with single button stroke. Sequential usages accumulate, so it's possible to reach any transposition this way. Go to chapter 4.2. for details on how to assign events to inputs.

#### 4.1.3 Transpose +/- command - #10, #11

Third option uses 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 lowest one shifts keyboard one semitone down with each sequence (#10), the 2nd key shifts the keyboard one semitone up (#11).

This is useful if you want to slightly change the starting note once, and there's no need for +/- buttons described before. If using keyboard instead of KEYPAD for entering numbers you must always use 2 lowest keys of the keyboard, even when changing transposition of upper split.

#### 4.1.4 Global Transpose

Apart from methods just described, which are mostly used during initial setup, there's different kind of transposing, more suitable for live performance. Global transpose is applied as a number of semitones from default position, and affects ALL keyboards in the system, including the ones connected to MIDI IN of MRG2 or MIDI128. The Global Transpose is not memorized in nonvolatile memory. After power cycle, all keyboards return to initial state. To use global transpose you can assign special functions to any POT inputs (or special switch inputs in MBBS and MIDI128). Available are +/-1 semitone and octave steps, as well as options for rotary and toggle switches or potentiometers. Event numbers associated:

- rotary switch with +/- 4 semitone range (9 steps): 137
- rotary switch with +/- 8 semitone range (17 steps): 139
- button for one semitone up: 156
- button for one semitone down: 157
- button for one octave up: 158
- button for one octave down: 159
- toggle switch for -1 octave / off / +1 octave: 164

## ***4.2 MIDI events for keyboards, analog inputs, and special switch inputs***

MIDI event assigned to given potentiometer, keyboard split, or special function button can be changed at any time. To perform this, turn the knob, or play a key on the split you want to assign, and select the event type by entering sequence #2 followed by the numbers from this chapter. You need to enter 2 or 3 digits for each input controller depending on entered number. To assign another one, again you must turn selected pot 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 165.

**Standard setting for a keyboard is "# 2 131" - single notes, and for analog input it's Continuous Controller: "# 2 nnn" - where "nnn" is number from 001 up.** This is factory default.

If a keyboard is assigned to note on/off event (131 or 161 only), starting note by default is MIDI number 36, the same as most MIDI keyboards use. Those are also the only 2 modes of keyboard operation affected by Global Transposition (see 4.1.4). Any other event sends MIDI values starting from 0 at lowest contact. If, for example keyboard is assigned to After Touch event, the lowest contact will send MIDI After Touch with value 0, next - value 1 and so on. The starting point can be adjusted with transposition setting (chapter 4.1), so the first contact can actually take any possible value.

### **4.2.1 Control Change - #2 CC**

Any MIDI Control Change number, where "CC" is number in range from "000" up to "127".

### **4.2.2 Pitch Bend - #2128**

The pot will work as pitch bender. If assigned to a keyboard, each key will set pitch bender in 1/128 steps across the keyboard. Range can be adjusted with transposition settings.

### **4.2.3 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. Unusual but still working use is when assigned to a pot. It will then generate MIDI Program Change messages with its every move. Program Change can be also generated in other ways, all summarized in chapter 4.4

### **4.2.4 Channel After Touch - #2130**

Turning pot will send Channel After Touch messages. 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. It is not used in conjunction with pressed keys, but rather as general purpose MIDI messages, usable in some instruments.

#### 4.2.5 Standard keyboard action – single notes - #2131

Whenever MIDI event 131 is assigned to a keyboard, it works as typical MIDI keyboard, playing MIDI notes. So if because of mistake in settings a keyboard is not playing notes, press any key, even if it seems not to do anything, and enter #2131.

When assigned to a pot, moving it makes bursts of notes. Select the pot to be edited, enter "# 2 131". This knob may be described as quantized Theremin, spitting series of notes with every pot move. There's only one note played at a time and it is released just before new note is about to play. Whole knob slow rotation plays 128 notes from entire MIDI range. You may reduce pot's input resolution (see 3.2) to achieve more playable range, like 32 notes.

#### 4.2.6 Note-on only - #2132

This mode is similar to the action described just before, but only "note-on" messages are generated. It may be useful with bunch of dual contact momentary switches, like register tabs in organs, where one contact would send note-on, and the other contact note-off (see next). With potentiometer input, 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.

#### 4.2.7 Note-off only - #2133

This is like "note-on" mode described before, but instead it sends out only note-off messages. Both modes can be used if you need separate inputs for note-on, and note off. If assigned to a keyboard, it will send only note-offs.

#### 4.2.8 One-touch Patch Recall - #2134

Keyboard in this mode serves as an array of Program Change memory buttons. Each key recalls Program Change number (selects a MIDI preset) together with Bank Select (only for MIDI128 and MRG2) that was earlier programmed. There are 64 memory locations to fill. For example you can program key 1 to send Program Change 37 in Bank 58, key 2 as PC 76 in Bank 12, and so on. All buttons work in the channel selected for this keyboard/scanner with #3 command as described in chapter 4.3. Storing Program Change numbers to a specific key is described in chapter 4.6 "Programming Patch Recall Buttons". MBBS can only send Program Change alone, no Bank Select available there.

#### 4.2.9 CC keyboard - #2135

This can be used only with keyboards. Each key sends min and max values in a subset of CCs. The keys use a range of MIDI Continuous Controller. For example lowest key sends CC 0, next one CC 1, and so on. Pressed key sends CC with max value (127),

while key release generates the same CC but with minimum value (0). Lowest CC number can be adjusted with transposition settings.

#### 4.2.10 MIDI channel shift for all controls - #2136

This is most usable with contact scanners. First 16 keys work like MIDI channel selector for **all controllers** (keyboards and pots) and all **merged incoming MIDI** traffic. After any of those 16 keys is hit, notes played in all keyboards are played in altered channel. Individual channel settings for every keyboard and potentiometer described in chapter 4.3 are used together with this setting. For example if keyboard was originally set to channel 3, and you've changed the channel using this feature to +4 (by hitting 5th key), new 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. To use a potentiometer, you should first reduce its resolution to 4 bits (chapter 3.2). After using controls assigned this way, actual channel change is displayed in LCD module.

#### 4.2.11 Small Transposer - #2137

Select the pot to be edited and enter "# 2 137". Turning such pot will shift all notes played (but only events 131 and 161) in all keyboards and merged notes from MIDI IN, by number of semitones selected here. In the middle it makes no change, and full rotation covers range from -4 to +4 semitones. It's most useful when pot is replaced by 9-position switch with 8 equal resistors connected between switch leads, or it can be a potentiometer with 9 detents if you can find one. After using this control, current transposition is displayed in LCD module.

#### 4.2.12 Big Transposer - #2138

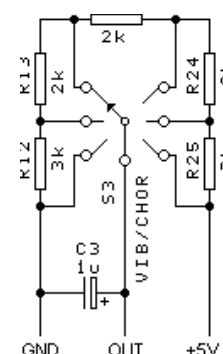
This is exactly the same feature as Small Transposer described above, but with wider range of -8 to +8 semitones. Every comment in Small Transposer description applies here as well.

#### 4.2.13 Velocity - #2139

Position of this pot determines velocity parameter of all MIDI notes played in all channels (except notes merged from MIDI IN). If assigned to a keyboard, each key will set velocity of all notes in 1/128 steps across the keyboard. It's not intended for initialization setup during installation, but rather as a way of performance expression during play. To set the velocity once, and always power up with chosen value, use event #2149 (chapter 4.2.23).

#### 4.2.14 Native Instruments B4 chorus/vibrato - #2140

Turning this pot will be reflected in NI's B4 screen as "chorus/vibrato" switch move. It has only 6 positions, and appropriate command will be sent every time the



potentiometer crosses each threshold representing another vibrato/chorus mode. The circuit on the right shows proper B4 connection. "GND" and "+5V" should be treated like pot's end taps and connected to dedicated end-tap pins in POT scanner, while "OUT" goes to any chosen analog input.

#### 4.2.15 MidiTzer stops control - #2141

This is MidiTzer organ specific, used only there for controlling stops (registers). When assigned to a keyboard, each key becomes stop controller. When a key is pressed, MIDI controller 81 (51 hex) is sent, and when released - MIDI controller 80 (50 hex). Value of the controller is determined by the button pressed. Lowest stop number can be adjusted with transposition settings.

#### 4.2.16 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. Lowest stop number can be adjusted with transposition settings. This is default way of controlling stops in Ahlborn Archive module.

#### 4.2.17 Ahlborn Organs stops control - #2143

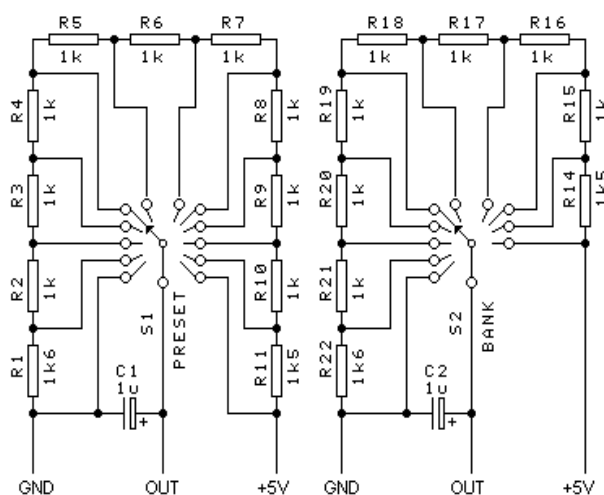
When assigned to a keyboard, each key becomes specific stop controller in Ahlborn Organs. When a key is pressed, MIDI controller 70 (46 hex) is sent with bit 6 of the value set, and when it is released, the same MIDI controller but with bit 6 of the value cleared. Other bits of the value are determined 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. Lowest value adjustable by transposition.

#### 4.2.18 Program selector - #2144

Together with 2nd pot configured as in 4.2.19 this feature allows to select one of 120 available MIDI Program Changes (presets) with 2 rotary switches. Program selector chooses one of 12 Programs within a bank. For better feel, you should use 12-position rotary switch. To do so, follow the schematics in next page to build appropriate resistor network around the switch. "GND" and "+5V" should be treated like pot's end taps and connected to dedicated end-taps pins in POT scanner, while "OUT" goes to any selected analog input.

#### 4.2.19 Bank selector - #2145

This feature works in combination with the one described before. A potentiometer or switch assigned



to this mode selects 10 banks of 12 Programs. If it is in lowest position, you can select any Program number from range 1-12 using Program selector described before. If you change the bank one position higher, Program selector will 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 in NI B4 Hammond emulator, where bank selector works exactly like that. Other ways of selecting Program Change numbers are described in 4.4.

#### 4.2.20 Ahlborn Common Functions - #2146

This is specific to Ahlborn modules only. When assigned to a keyboard, the buttons act in a way usable for group of controls in Ahlborn module called "Common Functions" like general cancel, tutti, etc. It works very much like event 143 described in chapter 4.2.17. When a key is pressed, MIDI controller 71 (47 hex) is sent with value 64-127, and when it is released, the same MIDI controller but with value lowered by 64. To reach different groups of controls, you may need to create splits in keyboards and choose different ranges with transposition settings.

#### 4.2.21 3-note chord collect - #2147

This one is suited only for keyboards. To get any MIDI note you must hold at least 3 keys together. When 3 notes are pressed, whole 3-note chord is immediately sent with no delays between notes. This assures that the chord is played clean, and not spread in time. All additional notes (4th, 5th etc.) are played in MIDI right after they are detected. This is usable in chord section of MIDI accordions, where mechanically coupled switches do not always close in the same time.

#### 4.2.22 Dual note (layered sound) - #2148

If assigned to a keyboard, every note played will be doubled in MIDI as 2 notes in user selected channels and interval. The settings of channel and starting note of doubled (layered) note are the same as would be for upper split. So in order to change defaults, you should first set the split at any point of the keyboard (*# 5 and any key*), then set the 2nd channel (*any key in upper split, # 3 and channel number*) and optionally adjust starting note of upper split (*#10 or #11 - see chapter 4.1 for details*). When finished - cancel the split by setting its position to the top key. Go to chapter 4.5 for details on split usage.

#### 4.2.23 One time velocity setup - #2149

When assigned to potentiometer, it will adjust velocity of all notes generated by MIDI scanners (but not merged notes from MIDI IN). This is similar to velocity settings described in 4.2.13, but in addition, it saves last adjusted velocity to nonvolatile



memory. This should be used only during installation, when you want to set default velocity of all notes after power up. For expression and frequent usage, control the velocity by assigning #2139.

#### 4.2.24 Single keyboard transpose buttons - #2150 to #2153

A potentiometer, or special switch input can be assigned as special button for fast keyboard transpose. There are 4 of them to choose and their event numbers are: one semitone up - 150, one semitone down - 151, one octave up - 152 and octave down - 153. Pressing such button (shorting input to ground) will affect only the last played keyboard, and this setting will be memorized in non volatile memory. For non-note events, this also changes the starting number of MIDI parameter's value range used for those events. This action is indicated in LCD module after each usage, showing newly set transposition in number of semitones.

#### 4.2.25 Channel set buttons - #2154 & #2155

MIDI channel of last used keyboard or potentiometer can be changed in single steps up or down, by using these special events. They can be assigned to any pot, or special switch input. Event 154 rises by 1 the channel number of last played keyboard, or last turned pot, while event 155 decreases the channel down by 1. The settings are stored in non volatile memory. This action is indicated in LCD module after each usage, showing current channel number for selected device.

#### 4.2.26 Global Transpose buttons - #2156 to #2159

This is similar to transposition setting described in chapter 4.2.24. Again you have:

- one semitone up: 156
- one semitone down: 157
- one octave up: 158
- one octave down: 159

But there are 2 main differences: transposition is set for ALL note keyboards in the system (including merged from MIDI IN), regardless of what was played last. And this setting is valid only as long as the power is on. After power cycle, transposition returns to initial settings, as adjusted for each keyboard by other means. This way of transposing is recommended for performance control, where you may want to frequently shift all keyboards at once in entire instrument. Go to chapter 4.1.2 for more details on available ways of transposition. This kind of transpose does NOT affect a keyboard if it was assigned to an event other than notes (131 or 161).

#### 4.2.27 MIDI notes for stops control - #2160

In this mode keyboard plays normal MIDI notes, just like with standard note event 131 (chapter 4.2.5), but Global Transposition does NOT affect this keyboard, and default starting note number is 0, like other non-note events. This is useful in organ emulators, where stops can be controlled by note-on/off messages, but you really don't want the

stops to change positions every time Global Transposition changes. Starting note can be of course adjusted with transposition settings as described in 4.1.

#### 4.2.28 Reversed notes action - #2161

If keyboard's contacts are normally closed, and open when key is pressed, you may take advantage of this mode. If keyboard is assigned to reversed action event, MIDI note-on is sent when contact opens, and note-off when contact closes. There is no need to short to ground any unused inputs of keyboard scanner, the only drawback is that during power-up you will see a burst of note-off messages, which can actually be useful sometimes. To have reversed action without the note-off burst, you need customized reversed keyboard scanner version available on request.

#### 4.2.29 Program up/down - #2162 & #2163

Those two can be assigned to any 2 analog inputs or special switch inputs. Shorting such input to ground will send next or previous combination of Bank Select and Program Change number from the user list. In MBBS only Program Change is sent alone, no Bank Select possible. There are 64 memory locations, where you can store Bank and Program Change numbers. They can be set by #6 command - see chapter 4.6. The "up" and "down" buttons browse across this sequence and send MIDI Program Change every time they are pressed. The Bank/Program Change messages are sent in the channel selected for those buttons, as in chapter 4.3. If you have LCD module connected, their action is shown as current memory position (1-64) and stored Program number (1-128).

#### 4.2.30 Octave "-1/0/+1" select switch - #2164

You can quickly transpose the keyboard 1 octave up or down with a toggle switch (3 position type, ON-OFF-ON). Add two 10k resistors, one from each end of the switch to the middle lead, and connect it to any POT input just like potentiometer: ends to GND and +5, while middle connector to POT input. In Middle position transposition is off, while other 2 give +1 octave, and -1 octave. This affects all keyboards in the system, including MIDI merged ones. If you have other controls active for global transpose, all of them will add up to final result.

#### 4.2.31 Per-keyboard channel switch - #2165

This is some kind of re-channelize device. It takes all notes played on keyboards from selected channel and moves them to another channel, higher by number selected with the switch. If resulting number is higher than 16, it wraps down and counts from 1 again. For example original channel 8 and switch at position 10 gives 18, so in fact it's now on channel 2 (18-16). It makes most sense to use 16-position rotary switch for this purpose with ladder of 15 equal resistors between positions. Connect the wiper of the

switch to POT input, and reduce its resolution to 4 bits (chapter 3.2). If you need less positions, the total value of resistors must remain, but connect less taps of the resistor ladder to the switch, or reduce POT input resolution to 3, 2 or even 1 bit. Each of 16 channels can have its own switch, and the input channel altered is the one set for the switch with #3 command (chapter 4.3).

### **4.3 *MIDI Channel - #3***

Channel of each potentiometer and keyboard or split can be set individually, and there are a few ways to do that. Also, you can change all channels at once for all controls, or keyboards working in one channel. Here's summary:

- channel setting of each keyboard split and pot, as a number: #3 <chan.>
- same as above, but with use of +/-1 stepping buttons: events 154 &155
- temporary channel shift for all controls: event 136
- temporary channel shift for keyboards within one channel: event 165

First 2 methods save channel number, so after power cycle it is recalled to previously set value. The last two are only temporary, valid as long as power is applied to master controller. To change (and save) MIDI channel of single potentiometer select the pot by turning it a bit. Likewise, to select a contact scanner, play any note or flip any switch, whatever is connected. Then you can select 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 channel 1 needs to be entered as 2 digits – namely "01". Channel 10 and above of course need 2 keystrokes too.

To be able to use 2nd method, you must first assign two POT, or special inputs to events 154 (channel +1) and 155 (channel -1) - see chapter 4.2. They become inputs for channel change buttons in steps for last touched device in the system.

Temporary channel shift control comes in 2 flavors, one of them works for all devices (keyboards and pots) and all MIDI IN signal. If you assign event 136 for example to a small contact scanner like BBS24 or even better - LITSW, you can have direct inputs for one keystroke channel changer. Every button would change the original MIDI channel of every MIDI message by fixed number: button 1 means no channel change, button 16<sup>th</sup> adds 16 to original channel. If resulting number would be greater than 16, it wraps down from 1, so if channel was 10, and you shift it by 13, resulting one is 7 (10+13-16). Go to chapter 4.2.10 for further info.

The same is possible if you want to affect only one keyboard and not entire setup. If you assign event 165 to potentiometer or LITSW like before, it will alter the channel in

the same way, but only for keyboards working in one channel. For example if you set any pot input to channel 7 and event 165, all keyboard scanners working in channel 7 will move to another channel when you move a rotary switch connected to this pot input. This does not affect keyboards from MIDI IN. Obviously there can be a switch like that for each of 16 channels.

#### **4.4 Program Change - #4**

All master controller boards allow to send Program Change MIDI messages in range 1-128. Several ways are available, here's summary of them:

- series of Program Change buttons in continuous numbering range using event 129 described in chapter 4.2.3
- same as above but with each input programmable to a specific Bank and Program number, using event 134 described in 4.2.8
- "up" and "down" buttons for browsing through sequence of Bank + Program Change numbers programmed by the user as Patch Recall Buttons in chapter 4.6. Use events 162 and 163 to any analog inputs
- rotary switches for Bank and Program select, chapters 4.2.18-19
- numeric entry of Bank Select MSB (CC#00) and Program Change

The last one can be used to directly select Bank and Program in one go, but also it is the only way to setup Bank Select number to be stored Patch Recall buttons. Enter "#4" followed by the Bank Select number from range 000 to 127. Next digits select Program Change number from range 001 to 128. Proper MIDI messages are sent afterwards to set required instrument in selected bank **in the MIDI channel of last played keyboard**, either internal (on scanners) or merged from MIDI IN. Selected bank and patch number within a bank are indicated in LCD module.

#### **4.5 Keyboard split - #5**

It is possible to split each keyboard into 2 independent parts. The split point can be placed anywhere in the keyboard, and both parts can work in separately adjusted MIDI channel, type of event and starting note (transposition) or range of values for events other than notes. To set the split point, first you have to select the keyboard by playing a note in it, and then enter sequence „# 5” followed by stroke of the key that you want to be the top of lower part. Since then lower part remains at previous settings, while upper part changes the channel. 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 to the split last played** prior entering the edit mode.

To cancel existing split of a keyboard, play any note in that keyboard, and set the split

at the top note like this: # 5 <top note>. This sets the end of 1st split at the top note, so in fact the upper one is not visible anymore.

Channel and starting note settings of upper split are also used in dual note mode described in 4.2.22.

## **4.6 *Programming Patch Recall Buttons - #6***

Whenever a program/patch/instrument selected from this MIDI controller is often used, it is worth to memorize as fast recall button. There can be 64 such buttons, working in single MIDI channel. Those buttons can be connected to any keyboard scanner. Master board will send the patch number in channel previously assigned to those buttons with #3 command (see chapter 4.3.). To memorize any patch this way, you have to first select a patch using any method described in chapter 4.4. Then simply press "# 6", and the key where the patch should be stored. Next time whenever you press this key/button, the MIDI Bank Select MSB and Program Change messages will be transmitted, with the numbers previously stored.

The numbers stored this way can only have range 1-128 for Program Change and 0-127 for Bank Select. Bank Select is memorized only in MRG2 and MIDI128. MBBS can store only Program change numbers.

To have Program Change memory in multiple channels, you may use multiple keyboard scanners or splits, but total number of PC recall buttons must not be greater than 64 in entire MIDI system built around one Master Controller. Also, if 2 or more scanners are used this way, the same input in every scanner will generate the same Bank/Program number, although you may define their channels separately.

In order to have one keystroke inputs for Bank Select & Program Change stored this way, you should assign a keyboard scanner to event 134 (see chapter 4.2.8).

## 5 Utilities

### 5.1 *MIDI merge input - #972 (on/off)*

There's MIDI merging input in MRG2 and MIDI128 boards. It adds possibility to connect any existing MIDI controller, and use it together with all scanners connected to them. All activity on MIDI controllers, and your existing controller connected to MIDI IN, are available then at one MIDI output. This is called MIDI merge function, and it is similar to audio mixing in analog world.

All MIDI notes incoming to MIDI IN are **affected by global transposition** (see chapter 4.1.4) and **global channel shift** (see 4.3 and 4.2.10). Other than that all MIDI traffic leaves the board unaltered. You cannot affect any parameters described in chapter 4 other than transposition and channel.

MIDI data received at MIDI IN has priority over all scanners connected to master board. If there is heavy traffic coming in, like sys-ex dumps, you will not see any action from the scanners until the dump is finished. In normal situations, like playing MIDI files, time codes etc., you should experience no problems with increased latency or missing notes. The only case when you can notice some latency is when MIDI monitor is on.

The merger can be turned off if required, by entering #972 from the keypad. This is needed for example to prevent loops if both MIDI IN and OUT are connected to the same instrument. With merger turned off incoming MIDI is still recognized, but not transmitted via OUT socket. MIDI IN is used then only to receive messages **controlling LCD modules and LITSW LEDs**. To turn the merger on again, repeat #972 from the keypad. This toggles merger's on/off state. Actual state is displayed in LCD after each change. Without LCD you can check it by looking at MIDI OUT LED. If it does not blink when MIDI IN LED blinks, that means the merger is off. Merger's last selected on/off state is memorized, and restored after power cycle.

### 5.2 *MIDI monitor - #971*

Master Controllers from midi-hardware.com have one outstanding feature that helps during installation and troubleshooting. You can use MIDI monitor if you have MiDisp, or miniLCD modules (chapter 6.8). To activate this function, select #971 from KEYPAD, and the LCD will show "DEBUG MODE". Additionally MiDisp module will turn red. Each activity on any scanner, as well as incoming MIDI where applicable, will be shown in

the display. Every time you press or release any key, or turn the potentiometer, the display will indicate last transmitted data.

The meaning of displayed values are:

- "D" - internal device number in the system
- "Ch" - MIDI channel
- "E" - event number as listed in chapter 4.2.
- "S" - MIDI status byte in hexadecimal format
- "V1" - value of 1st MIDI data byte in hexadecimal format
- "V2" - value of 2nd MIDI data byte in hexadecimal format.

MIDI monitor **should not be enabled during normal performance**, it noticeably increases MIDI latency and may cause dropped or hung notes during fast keyboard playing, especially if LCD is connected in one chain with other scanners. To turn off the MIDI monitor, enter #970 from the keypad. MIDI monitor is always disabled at power-up.

### ***5.3 All settings Backup / Restore - #978***

When entire settings process is finished and everything works as required, it is worth to save your work results externally, so they can be easily recalled. Also, if you need different setups of the MIDI controller adjusted for 2 or more instruments, you can change all events, channels and more in a manner of seconds by playing a MIDI file. To have full backup of current MRG2 or MIDI128 configuration, connect MIDI OUT to any kind of sequencer or Sys-Ex editor, start recording, and dial #978 from the keypad. Master Controller will then send a burst of MIDI System Exclusive messages with all settings. After about 2 seconds it's over and you can save the sequence as standard MIDI file. This is indicated in LCD module.

Whenever you wish to recall specific configuration, simply connect sequencer or MIDI file player to MIDI IN of your Master Controller and play selected MIDI file. Configuration will be then saved just like it would be done by entering numerous setting commands from the Keypad. This process is also indicated at the LCD. Settings covered by backup/restore procedures are: all channels, events, splits, starting notes, velocity, 64 user programmable Patch Recall memory locations, update rate and resolution of Master Controller's analog inputs.

NOT included are: update rate and resolution of analog inputs in scanners, settings of LITSW (chapter 6.6) and MiDisp/miniLCD (chapter 6.8). Those can only be done manually or via separate Sys-Ex commands.

### ***5.4 Factory defaults - #979***

The MIDI controllers have a lot of features to set and you may sometimes go one step

too far, so it may be useful to have last resort in form of reset to factory defaults. To do so, enter command #979 either from the keypad if you have one, or from any keyboard, using lowest 10 keys as numbers from 0 to 9, in a way described in chapter 4. About one second later the master board is ready to use with all factory settings in place. If a LCD module is connected, this process is clearly indicated at the display. Settings cleared to defaults by factory reset are: all channels, events, splits, starting notes, velocity, 64 user programmable Patch Recall memory locations, update rate and resolution of internal MIDI128 and MBBS analog inputs. NOT included are: update rate and resolution of analog inputs in scanners, settings of LITSW (chapter 6.6) and MiDisp/miniLCD (chapter 6.8). Those can only be reset manually or via Sys-Ex for each scanner separately.

## 5.5 ***MIDI System-Exclusive***

For master controllers with MINI IN, all settings are also available via MIDI Sys-Ex messages. The *<product ID>* shown in every command is different for each type of board. MRG2: *<product ID>* = 04, MIDI128: *<product ID>* = 06. The *<input ID>* parameter points to specific keyboard or potentiometer in the MIDI system and is present in every command.

Valid input IDs and their assignment to actual inputs is described in separate document, specific for each Master Controller.

**All numbers are shown in hexadecimal format**, typical for Sys-Ex editors.

### 5.5.1 Commands description

Available commands are:

- **01** - starting note, with 1 parameter byte
- **02** - channel and event, with 3 parameter bytes, one for channel, and 2 for event (MSB and LSB)
- **05** - split point with 1 parameter byte
- **09** - advanced settings, with 1 parameter byte
- **10** - write 32-byte chunk of Program Change and Bank Select numbers used in "patch recall" buttons

### 5.5.2 Starting note command - 01

*F0 00 20 7A <product ID> <input ID> 01 <starting note> F7*

this is available only for splits of keyboards, command will be ignored if selected input ID is POT type. It sets the starting note at given MIDI note number. So if you want standard lowest C in 5-octave keyboard, the *<starting note>* parameter should be 24 (hex). Any other value from 00 to 7F is accepted.



### 5.5.3 Channel and Event - 02

*F0 00 20 7A <product ID> <input ID> 02 <channel> <event MSB> <event LSB> F7*

this is available for all controls, so every keyboard split and pot can have different setting. The <channel> is given in straight channel number from 1 to 16, which in hexadecimal means from 01 to 10. Followed by 2 bytes of event. In general, Continuous Controller is set when <event MSB> is 0, then CC number is passed in <event LSB>. All other events have MSB set to 1, and LSB is the event number lowered by 128, for example note on/off is MSB=1, LSB=3 (1\*128+3=131).

### 5.5.4 Split Point - 05

*F0 00 20 7A <product ID> <input ID> 05 <key position> F7*

this Sys-Ex string sets the split at specific point. Lowest key of the keyboard is selected when <key position> parameter equals 00, and rising up to 3F for the last key of any contact scanner. Setting split point with parameter = 00 means that one lowest key will be in 1st split, and all remaining keys in 2nd split.

To turn off the split, simply set it beyond connected keys range.

### 5.5.5 Advanced Settings - 09

*F0 00 20 7A <product ID> <input ID> 09 <advanced command> F7*

using 09 command gives access to most of advanced settings typically available from keypad's sequence #9x. Possible options to set are:

- analog inputs update rate of all analog inputs in one board (5-100ms)
- bitwise resolution of each analog input (1-7 bits, 2-128 levels)
- reset to factory defaults
- setup of LITSW modes
- selected options of MiDisp and miniLCD setup

This is described here in detail:

#### **Analog inputs update rate, latency**

*from F0 00 20 7A <product ID> <input ID> 09 **50** F7 (fastest)  
to F0 00 20 7A <product ID> <input ID> 09 **59** F7 (slowest)*

To select update rate, the device ID must be within range belonging to selected pot scanner (or MBBS, or MIDI128 on-board inputs), so for example update rate of

MIDI128 internal inputs can be changed if you send this command with ID from range 0C - 13 (hex). This setting is valid for all inputs in one board, so doing it for more than one input from this range makes no sense. To set different update rate in connected scanner, use appropriate ID from its range.

Actual scan rate is defined by specific *<advanced command>*. 50 means the fastest scan, while 59 is the slowest. 50-59 in hexadecimal represents 80-89 decimal, go to section 3.1 to check available scan rates.

### **Resolution of analog inputs**

from *F0 00 20 7A <product ID> <input ID> 09 **3D** F7* (1-bit)  
to *F0 00 20 7A <product ID> <input ID> 09 **43** F7* (7-bit)

Bits of resolution can be adjusted for each input in the system individually, so here input ID points to specific one. Use guidelines from specific master controller manual to find proper input ID. This setting is available for all analog inputs in MBBS, MIDI128 and all scanners in the chain if they contain analog inputs.

### **Reset to factory defaults**

*F0 00 20 7A <product ID> 00 09 4F F7*

Send the above MIDI string to master controller, and it will turn back to default factory settings. Affected are all MIDI channels, events and splits, note velocity, as well as list of 64 user programmable Program Numbers. Analog inputs update rate and bits resolution is restored only for the internal inputs of MBBS and MIDI128, because those settings are actually stored in each board individually.

### **LITSW setup**

from *F0 00 20 7A <product ID> 3F 09 **05** F7*  
to *F0 00 20 7A <product ID> 3F 09 **1D** F7*

All available commands are described in LITSW section, see chapter 6.6. Before sending any LITSW related command, you should press any of its buttons first.

### **LCD modules setup**

from *F0 00 20 7A <product ID> <input ID> 09 **08** F7*  
to *F0 00 20 7A <product ID> <input ID> 09 **14** F7*

Before you send any Sys-Ex regarding LCD module, it has to be selected first, and the only way to do it is to press its button, in another words temporarily short the switch input pads at the back of selected LCD module. Also keep in mind that all LCDs in chain

share the same two input IDs, not like all other scanners, so every LCD can be set using the same input ID. It is not possible to set display ID with MIDI Sys-Ex, it can only be done from the KEYPAD.

### ***Patch Recall memory fill***

*F0 00 20 7A <product ID> 00 10 [32 Program Change numbers] F7*  
writes Program Change numbers in locations 1 to 32

*F0 00 20 7A <product ID> 01 10 [32 Program Change numbers] F7*  
writes Program Change numbers in locations 33 to 64

*F0 00 20 7A <product ID> 02 10 [32 Bank Select numbers] F7*  
writes Bank Select MSB numbers in locations 1 to 32 (only MIDI128 and MRG2)

*F0 00 20 7A <product ID> 03 10 [32 Bank Select numbers] F7*  
writes Bank Select MSB numbers in locations 1 to 32 (only MIDI128 and MRG2)

MIDI128 and MRG2 can store 64 pairs of Bank Select Program Change numbers, that can be later recalled either by a button click, or in +/-1 sequence. MBBS stores only Program Change numbers, no banks. You can quickly store all numbers in 4 blocks independently, 2 for Program Change and 2 for Bank Select (only MIDI128 and MRG2), 32 numbers in each block. The block of 32 bytes can contain any numbers in range 00 to 7F (in decimal that's 0-127), which is just what you need for Bank Select, but Program Change numbers must be lowered by 1, for example Program Change 1 should be sent in sys-Ex as 00, and 128 as 127 (7F in hex).

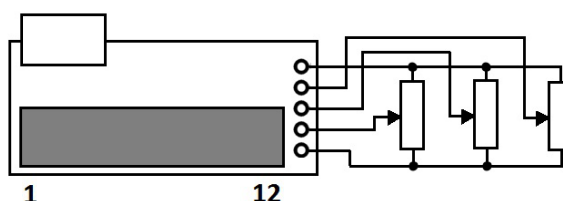
## 6 Scanners

Master controller boards described in this manual are the merging point for all keyboard and potentiometer scanners. Depending on their sizes, the whole system can cover up to 10 keyboards with 640 keys and 96 potentiometers over single MIDI cable. This chapter describes all scanners available. Each one works with each master controller. Most of the boards use 16-way black connectors, and their pinout numbering is always the same, shown here:

2	4	6	8	10	12	14	16
⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
1	3	5	7	9	11	13	15

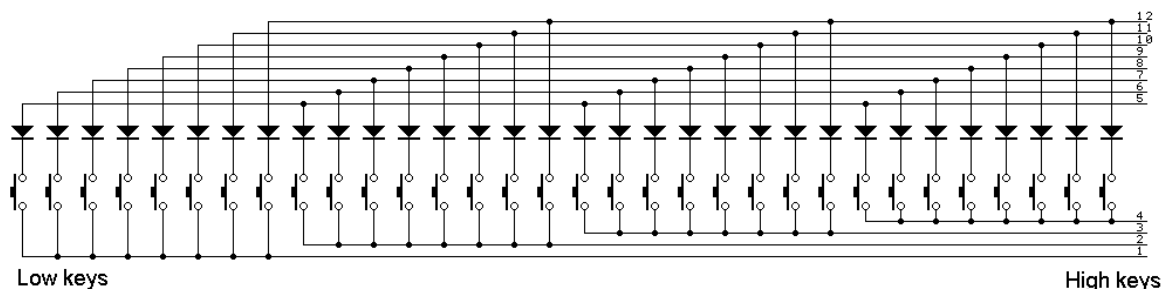
### 6.1 PDS - diode matrix pedal scanner

Pedal board scanner takes care of 32 keys and three analog inputs, usually all what's needed for pedals and 3 swell shoes (2 prior version 3.0). It's a small board that plugs on 12-pin male header in 0.1 inch pin spacing, and connects to



the main board via supplied 4-wire. Dark rectangle in the picture is 12-pin female socket. When plugged to the header, PDS stands vertically, taking very little space, about an inch high including the plug from 4-wire cable (upper small rectangle in the picture). The 5 pins at the shorter edge of PDS are for 3 potentiometers. Optional "PotsTail" cable simplifies installation of potentiometers. The picture shows how to connect the pots. Upper slider position in the picture generates max MIDI value, and lowest position – minimum value. Go to chapter 4 for more about potentiometers usage.

Keys must be connected in diode matrix, an example is shown below. To change note order if it's not correct due to different diode matrix, use the procedure described in chapter 6.2 about DMS-2K (commands #921, #922, #923, #924). To maintain compatibility with currently available master controllers, use PDS in version at least 3.0, otherwise note order setup will not be possible.

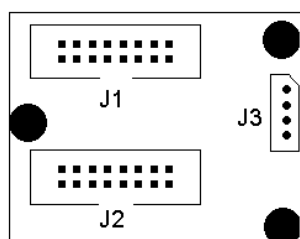


Leads to PDS in above schematics are counted 1 to 12 from the bottom.

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 or 4th key (depending on diodes direction) or better use BBSP scanner instead.

## 6.2 *DMS-2K dual keyboard matrix scanner*

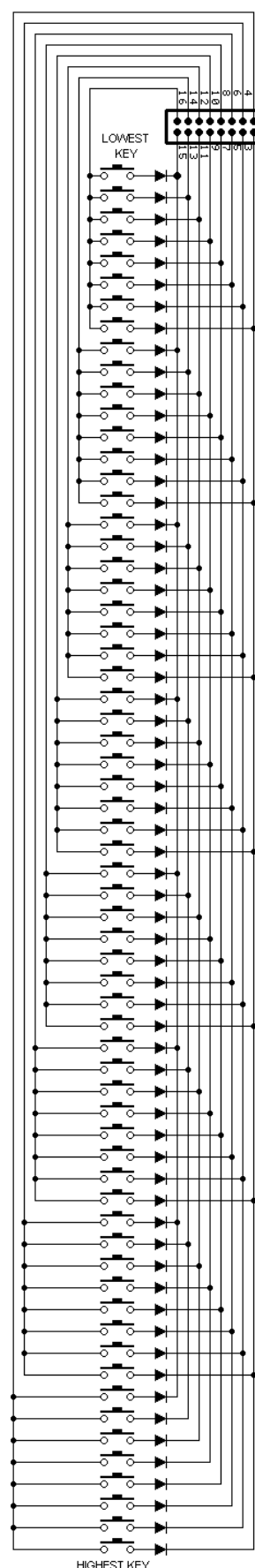
Ideal solution for keyboards with diode matrix is DMS-2K, adding 128 inputs. Example matrix layout is shown here - notice that the **connector is shown from the bottom**, so use actual pin numbers for guidance. Lowest key is at pins 15-16. If your keyboard has different matrix, notes will not play in correct order, but it can be easily fixed with user settings - read on. There are two 16-way connectors for two 8x8 matrix keyboards. Smaller connector for master controller. The matrix is very simple circuit, made of diodes forming electric XY array. Usually all modern keyboards are equipped with just that, but sometimes organized differently, in 6x12 or 6x11. For those you may try 8x12 mode of DMS-2K but in such case only one keyboard connects to DMS-2K.



J1 – connector of 1st group of keys (1 to 64)  
 J2 – connector of 2nd group of keys (65 to 128)  
 J3 – connector for cable to the main controller

DMS-2K can be reconfigured by the user to work with 4 different 8x8 diode matrix types, and also as a single keyboard scanner using 12x8 matrix. 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 8x8 keyboard plays **notes in wrong order**, try one of the 4 modes of DMS-2K:

1. play any note on "bad" DMS-2K keyboard
2. enter EDIT mode either by momentarily shorting proper 2 pins of header on master board, or by



pressing "#" keypad button

3. enter "92" from numeric keypad, or any other well known working keyboard scanner
4. enter one digit from range 1-4 the same way as you entered "92". Each digit turns the scanner into different order of notes.
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.

Both keyboard in one DMS-2K work in the same mode, so both must use the same kind of diode matrix. To work with 12x6 (or up to 12x8) keyboard, you need to utilize both connectors (J1 and J2) to connect one keyboard. Use all 8 even pins of J1, and 4 even pins of J2 to connect anodes of the matrix, and connect 8 odd pins of either J1 or J2 (whichever convenient) to the cathode side of the matrix. Then follow setup procedure above, to set matrix mode to one of two 12x8 layouts, but now use digit 5 or 6 in step 4.

If there are more than 64 keys, all the ones above will play in different channel and start from low C. You only need to adjust the channel and transposition of the upper part, like you'd normally do for any keyboard (chapter 4.1 and 4.3).

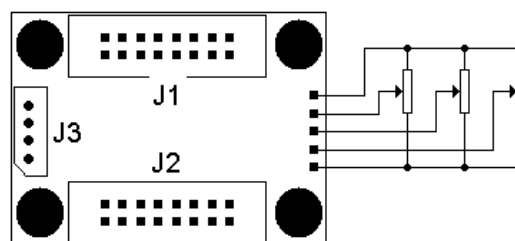
12x8 mode is available since version 1.7 of DMS-2K.

### 6.3 **BBSP - common rail pedals scanner**

This one is especially suitable for small keyboards (or pedals) with single wire used as common bus for all switches in entire keyboard. This one does not need diode matrix, and can be used with almost any type of switch arrangement.

Keyboard goes to two 16-pin connectors - J1 and J2 shown below. The key contacts must use one common buss bar (with GND), or logic IC may control BBSP inputs. 0V at an input means "key pressed", +5V at input or open circuit means "key released". Each of the connectors covers 16 keys, while **common wire** of all contacts must be connected **to flat terminal** located by one of the mounting holes. Proper crimp terminal is provided in the bag.

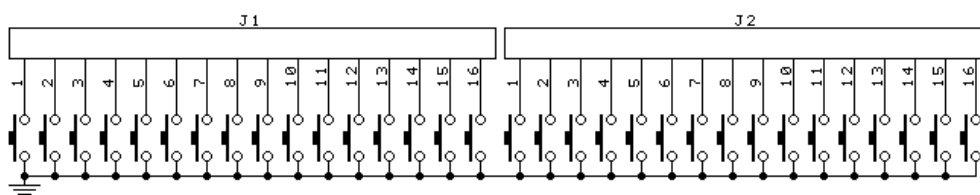
Potentiometers connect to 5-pin header at the side, according to the diagram.



Potentiometers connect to 5-pin header at the side, according to the diagram. Optional "PotsTail" matches this connector to simplify installation. When using pot resistance larger than 20k or wires longer than 30cm, a **shielded cable is required**. The shields of 3 cables (from 3 pots' wipers) should be tied together at the lowest of the 5 pins when oriented as in the diagram here. See chapter 4 for more info on potentiometers usage.

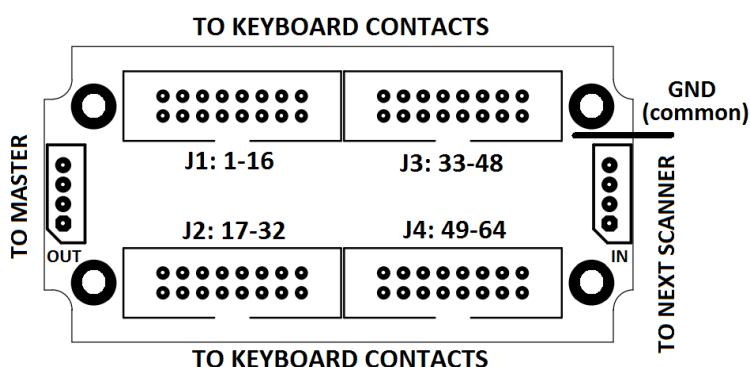
Keyboard should be connected to two 16-pin headers according to the diagram below. Works best with 2 IDC plugs and 16-wire ribbon cable, ordered separately.

Each black connector holds 16 inputs for keyboard switches.



## 6.4 ***BBS-1K common rail keyboard scanner***

This one is mainly used in 5-octave keyboards. It has 64 contact inputs, and daisy-chain input, so you can use 2 of those connected to 1 scanner input of Master Controller. For example with MRG2 you can use 10 BBS-1K

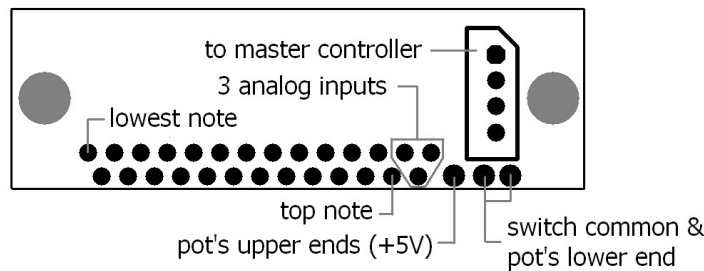


boards, but only 5 BBSP. All 4 connectors for ribbon cable and the pin out are the same as used in BBSP - see above. Each black connector covers 16 keys. Common wire of all contacts goes to GND flat terminal located by one of the mounting holes. Proper crimp terminal is provided in the bag. BBS-1K must be connected towards the Master Controller with 4-way connector indicated "OUT". The one marked "IN" is for adding next optional scanner. There can be only 2 keys scanners (BBS-1K, BBSP, DMS-2K, PDS, BBS24, LITSW) in one chain. If third keyboard scanner is chained, it will work like paralleled inputs of 2nd keyboard in the chain. But you can still connect there numerous LCD, or POT12 and KEYPAD at the end and.

Special breakout board is available for BBS-1K if you wish to use existing wires instead of ribbon cable to connect keyboard contacts. Two of those boards, called WIPLUG, can be plugged on both sides of BBS-1K, and instead of header pins you have nice push-in terminals for inserting bare wires. No soldering or tool is needed for connecting wires, maybe except small screwdriver, pencil, or tooth stick to push the lever when inserting the wire. This can be also done with finger nail.

## 6.5 ***BBS24 - contacts & pots little scanner***

This one is especially suited for the bass/chord part of MIDI accordions. It works with 24 switches, and 3 analog sources like pots, or voltages from pressure sensor. If split is set in the middle (chapter 4.5.) 12 inputs can be assigned to basses, and 12 to chord contacts in accordion and play in different channels.



The board is small enough to put anywhere. Unlike other scanners, all user inputs are bare pads to solder wires. This further reduces space required for installation, because typical pin headers would be too high. Of course it may work in any other instrument/controller requiring limited (24) number of contacts, and up to 3 potentiometers. The diagram describes all connections to BBS24.

BBS24 accepts a few configuration commands to make it more suitable for demanding installations. To use any of them you must first make some action on any of 3 analog inputs. This may seem awkward as changes are made to the contacts, not analog inputs, but that's the way it works in current setup. Settings for analog inputs are described in chapter 3, as they are always the same regardless at which board such board is located. Here are only described additional options changing behavior of contact inputs:

#### 6.5.1 Order of notes - #921 and #922

It is possible to reverse default order of notes as shown in BBS24 diagram, so the lowest note is at the side of 4-way connector. To do so, enter #922 after any analog input of that BBS24 was generating MIDI traffic. #921 returns to normal note order.

#### 6.5.2 Contact inputs polarity - #927 and #928

In default state note-on is send whenever BBS24 contact input is closed to GND, and note-off, when contact is open. This can be reversed by applying #928. Here also analog inputs must generate some MIDI prior this command. After that note-on will be triggered by opening a contact, and note-off by closing it.

#### 6.5.3 Contact debounce - #990 to #999

When closing contacts results in bursts of notes instead of clean attack, it means contacts are scratchy and suffer from bouncing. This is taken care of by any keyboard scanner described in this book. BBS24 has the ability to adjust this process, so you can get faster response when needed. To change it, you must generate some action from BBS24 analog input, and then choose one of 10 levels of debouncing. From no processing at all with #990 command, to very long debouncing of about 40ms with #999. In any setting the attack phase of each note is not delayed, it always appears as fast as possible in each setting.



#### 6.5.4 Reset to defaults - #929

Like with other scanners, this command returns BBS to default state with regards to options mentioned above and also settings for scan rate and resolution of analog inputs.

### 6.6 **LITSW - button scanner with LED drivers**

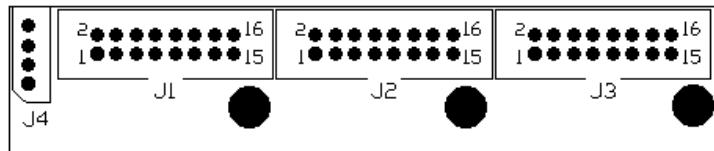
LITSW is the controller for lighted switches. It has 24 inputs for contacts, and 24 outputs for LEDs associated with them. The LEDs can be also controlled via MIDI if LITSW is connected to MRG2 or MIDI128. To turn any LED on or off, master controller must receive the same MIDI note in the same channel as it sends when appropriate LITSW button is manually pressed.

Available modes of operation for this board are:

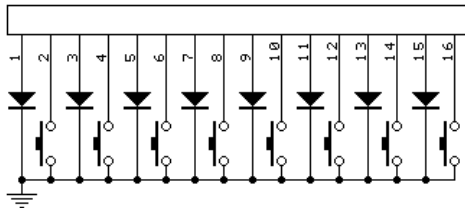
1. independent registers ("check boxes"), where each button click lights up or turns off the LED, and proper MIDI message is generated according to Master Controller's setting for that scanner. So note-on (or CC with value 127) is sent at LED turn-on and note-off (or CC with value 0) when LED turns off. If it is Program Change, only one event is generated, when the LED turns on
2. dependent choice ("radio buttons"), where only one LED-button can be active (lights up). Pressing any other button causes previously lighted to turn off, and the one pressed lights up now. If assigned to notes, only MIDI note-on is generated, this is more suitable for use with Program Change
3. contact on/off mode, where button action is recognized as typical contact scanner (BBS). So all kinds of MIDI events applicable to a keyboard are also possible here giving predictive results.
4. CC value mode, where LITSW works like potentiometer scanner, but with only min and max values. The advantage of this mode is that you can assign totally different MIDI channel and event to each button.
5. bank/select, useful for preset selectors. First it must be split in 2 parts (#905). One part works as bank select, the other one - as program select. If the split is made on 10th key, lower part is representing units, and the upper one - tens of selected value. Of course it makes more sense when MIDI event 129 is assigned to that LITSW (that's Program Change), but any event from chapter 4.2 can be assigned
6. Cancel Button - in this mode last 24th button cancels everything that is on within the LITSW. Press it and all other 23 LEDs go dark, while associated MIDI messages are sent, like when turning them off manually (available since LITSW version 3.0).

It is possible to select more than one mode at once, although it makes sense not for all

combinations. For example modes 1-2, as well as 3-4 are mutually exclusive, but you can set for example modes 2, 4 and 6 together. The mode of operation can be assigned to all buttons or any the 2 splits if it was divided in a way described later (6.6.1.).



J1 – LEDs/buttons 1-8  
 J2 – LEDs/buttons 9-16  
 J3 – LEDs/buttons 17-24  
 J4 - connector to Master Controller



The pins of each LED and buttons connector are interleaved 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 this schematics here.

Each 16-pin connector layout is the same. The common wire must be connected to the flat terminal marked "COM". Proper mating terminal is provided with the board. It is possible to reverse the on/off state of the outputs, so existing LEDs wired with common anode can be connected also with no modification. In reversed mode the buttons still are connected like in the schematic, shorting to GND, but LEDs need common anode connected to +5V power, and cathodes to odd pins of LITSW connectors. The reversed mode can be activated with #928 command (chapter 6.6.15).

LITSW is driving the LEDs at about 3.5mA per output. This is more than enough for modern indicator LEDs, but may be too small for backlighting applications, for example if you want to control 12V light bulb in lighted toe piston. For that purpose there is current booster available, called LITBO. It plugs directly into LITSW 16-way connector providing another one on top, and screw terminal for high current ground return. When using LITBO, the buttons must be wired the same way as usual, shorting to GND, but outputs are now open drain, so in active state they sink current from load to ground. So whatever you connect, high power LED, light bulb or little motor, it must be connected between LITBO output (any odd pin of the 16-way connector) and positive power supply, that can go up to +30V. Each output is capable of 0.5A continuous current draw.

**WARNING!** do NOT ever unplug LITBO from LITSW when power is on. Failing to comply to this requirement will result in smoking board damage, and possibility of components exploding right into your eye.

Out of the box LITSW generates note-on and note-off messages with momentary buttons in independent mode, 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:

#### 6.6.1 Split point - #905

This is totally different kind of split, non related to the one described in chapter 4.5. So you can use both kinds of split at one time, and they can be at different points. The #905 split controls only button's behavior, and not actual MIDI data transmitted via MIDI OUT socket. Each split can work in different mode, with independent or dependent buttons, generating CC values or contact on/off's. To activate split, enter #905 from the KEYPAD and press a button connected to the LITSW module in question, where you want the top key of lower split. You may check where is the split point by using blink commands – LEDs will blink in lower or upper split only. See chapters 6.6.9-6.6.11. To have different MIDI events or channels in the splits you have to also split it logically, using procedure described in 4.5.

#### 6.6.2 Device ID shift - #906

This is usable only when CC value mode is used in multiple LITSW in the system, or if POT scanners are in different chain than LITSW. In all other cases it's better not to use it.

When LITSW works in CC value mode, every input can be assigned to send different MIDI event. But the device ID of each input in the system is counted from the point of connection to master board. So if you have 2 LITSW connected directly to 2 Master Controller's inputs, all 24 buttons on 2 LITSW will have the same device IDs, and sending exactly the same MIDI messages. Using #906 command allows to shift the device IDs in one LITSW and have 2 ranges that do not overlap. To use it, press any button on the LITSW to change, enter #906 from the keypad, and press a button in this LITSW. First button means no shift from default state, 2nd button - shift by 1, and so on, till button 24. Available in ver. 3.x

#### 6.6.3 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 Master Controller (in case it is configured to send notes on this input). 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.

#### 6.6.4 Dependent mode - #909

In dependent mode, there can be only one LED active at a time within one split of LITSW (or entire LITSW if there is no split set). 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 4.2.3., and 4.2.8. To set this mode on a

split, you have to select it first by pressing one of the buttons in that split.

#### 6.6.5 Bank/preset select - #910

This mode requires prior split of the LITSW keypad sequence #905 described in chapter 6.6.1. If you then type #910 on the keypad, LITSW board will 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. Most obvious use is together with dependent buttons mode described in 6.6.4.

Since version 3.0 LITSW in bank/select mode is working only in CC value mode, so you may easier select any value from 0-127 range using 24 buttons. To use it then as Program Change selector, you should program appropriate MIDI event - #2129. The split point doesn't have to be after 10 buttons. If it's after 6 buttons, first bank will select patches from 1 to 6, second bank from 7 to 12 and so on.

#### 6.6.6 Disable bank/select and Cancel mode - #911

To disable bank/select mode described above, or Cancel button from chapter 6.6.13 you should type #911 sequence on the keypad after any button of this LITSW was pressed. After this command LITSW returns to normal mode, where each button is treated equally.

#### 6.6.7 Contact on/off (keyboard scanner) mode - #912

This is normal way of operation for this scanner. When a button is pressed, Master Controller can then generate MIDI notes, or any other MIDI event typical for keyboard scanner. This is the opposite to CC value 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.

#### 6.6.8 CC value (pot scanner) mode - #913

In this mode LITSW works a bit like potentiometer scanner. When button is on, it simulates turning a potentiometer to a maximum, and when the button is off (LED is turned off) - a potentiometer turned to minimum. So it's like 24 pots with only 2 valid positions: min and max. This is useful for switching parameters in virtual instruments, commonly using 2-state CCs. For each button you can then assign 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.

#### 6.6.9 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. To make it work, you must first press any button of the LITSW in question.

#### 6.6.10 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. To make it work, you must first press any button of the LITSW in question.

#### 6.6.11 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. To make it work, you must first press any button of the LITSW in question.

#### 6.6.12 Turn off all LEDs - #919

Whenever #919 is entered, all LEDs that were previously on, will be turned off, and appropriate MIDI messages are sent, just like you'd manually press the buttons one by one (available since v3.3).

#### 6.6.13 Cancel Button mode - #922

It makes the last, 24th button a Cancel-Button for all remaining 23 inputs, regardless if they are split or not. When pressed, all buttons that were previously on, will be turned off, together with sending MIDI message assigned for this input's off state. To disable Cancel button and have 24 equal ones, enter #911.

#### 6.6.14 LEDs with common cathode (default) - #927

It makes the LITSW outputs work in default positive logic, meaning on-state is indicated with present voltage at the output, so if LED is connected with cathode to GND, it lights up then.

#### 6.6.15 LEDs with common anode (reversed) - #928

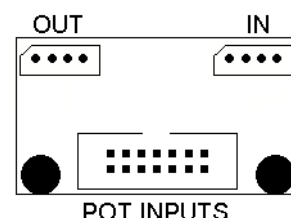
It makes the LITSW outputs work in reversed logic, meaning on-state is indicated with zero voltage at the output, and 5V when in off-state. So in order to use LEDs, they need to be connected to LITSW outputs with their cathodes, and all LEDs anodes should be connected to +5V power.

#### 6.6.16 Reset to defaults - #929

If your LITSW has been set with a strange combination of modes, it may be useful to return to the state where it came out of the box. To do so, press any of the LITSW buttons, and enter command #929. LITSW then returns to factory defaults, which is independent mode, no split or cancel, and contact scanner mode. This is confirmed with short blink of all LEDs.

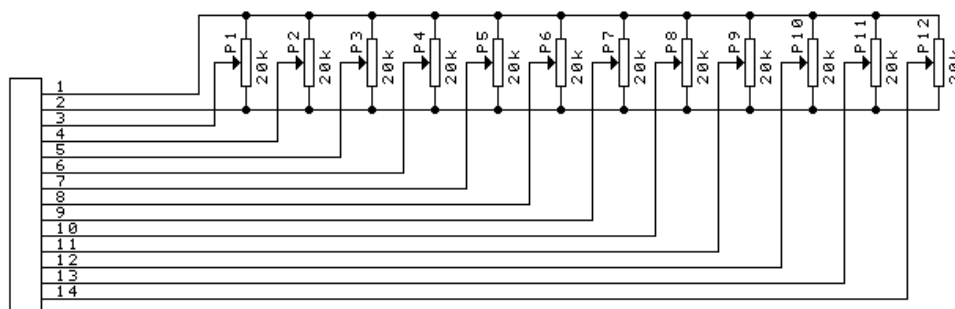
## 6.7 **POT12 - potentiometers scanner**

POT12 adds 12 analog inputs (potentiometers or



voltages) to MIDI system built around any Master Controller. It has one 14-pin black connectors for potentiometers, and 2 small 4-pin sockets, typical for all scanner boards described here. The one indicated "OUT" goes to Master Controller. The other one can be used to connect another scanner, be it keyboard or another POT board. 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 any board.

You can chain up to 5 POT12 scanners and make 60 inputs for the whole system. The bigger black connector holds potentiometer inputs according to the following schematics:



There are 12 inputs for pot wipers, and 2 for top and bottom taps. The pin numbering within one connector is shown here. Notice orientation of the keying slot. If you don't use all potentiometers you may want to connect unused inputs to VCC (pin 1).

2	4	6	8	10	12	14
⊗	⊗	⊗	⊗	⊗	⊗	⊗
1	3	5	7	9	11	13
⊗	⊗	⊗	⊗	⊗	⊗	⊗

**Keeping any, or all inputs open is accepted, but in some cases may result in unexpected MIDI messages.** That's most likely to happen with fast response settings described in chapter 4.1. At fast update rate and inputs left open, moving one potentiometer may trigger unexpected additional MIDI messages. In such cases connect all unused inputs to VCC (pin 1 of the black connector) and spurious messages will disappear.

## 6.8 MiDisp & miniLCD - 2x16 characters displays

All Master Controller boards have lots of useful features, and all are available to the user without need of any PC or custom programmer. Simply because there are so many parameters, it may sometimes become difficult to follow every setup procedure without any clear indication what is happening. To help in user settings and add yet another functionality, you may use MiDisp or miniLCD module, with the following features:

- one input for external switch, assignable the same way as any other scanner with separate setting (event and channel) for on and off state

- text messages indicating every stage of user settings and performance controls like transposer, program change and so on
- back lit color changing depending on current EDIT, or other modes (MiDisp only)
- MIDI monitor useful when checking communication and MIDI settings
- displaying text messages from Hauptwerk Sys-Ex

MiDisp module is standard size 2x16 LCD text display, and can change back lit color. On the other hand miniLCD is about half smaller, with white letters on black only, but much better contrast when viewing from +/-20 degrees vertical. Other than that both LCD are functionally equivalent.

LCD modules can be connected anywhere in the system, directly to Master Controller, or in the chain with other scanners. There may be several LCD modules connected to one Master Controller, but due to power limitations there should be no more than 10 MiDisp or 30 miniLCD modules connected directly. To have more, you should use power booster inserted in the chain of LCDs. Available on request.

Only one display, connected the closest to Master Controller, will become system display during startup. It will show all EDIT messages and MIDI monitor. Other displays may be used only as 1-button controls with visual indication, or displaying text received via Sys-Ex from Hauptwerk.

**To display messages from Hauptwerk, you must first disable the MIDI merger by command #972** - see chapter 5.1. This is also required by Hauptwerk to prevent MIDI lops. For best results with Hauptwerk messaging it is recommended that LCD is in version at least 2.7, and MRG2 in version 2.9 or MIDI128 4.0.

You may change the way how LCD operates to some extent. To do any settings on particular display module, you must select it first, and the only way of doing that is to press the button connected to display's switch input (or simply short those 2 pads temporarily with a piece of wire). The pads to connect LCD button are located at the back of it. More general guidelines about entering commands can be found in chapter 4. Possible options are:

#### 6.8.1 Momentary button input - #908

In this mode, the module can toggle between on and off if momentary switch is connected. If a button is momentarily pressed, the display shows "ON" and turns white. Appropriate MIDI message is sent via master board's MIDI OUT. At next button press the display shows "OFF" (MiDisp also turns blue), and another MIDI message is sent. You can assign completely different MIDI events and channels for "ON" state and "OFF" state, like described in chapters 4.2. and 4.3. Both states work like potentiometer action, but the value of both is always the same, determined by display ID, see 6.8.3 for details.

### 6.8.2 Toggle switch input - #909

This mode is needed when you connect toggle switch, or reed sensor in draw knob. It works almost like a keyboard, keeping "on" state only as long as the switch is closed, turning "off" when switch opens. Apart from this, all other functionality is exactly like in the other mode described just above - ON state generates different event than OFF state, and both can be programmed.

### 6.8.3 Display ID set - #914

Each LCD in the system has individually set parameter called "Display ID". It is used as value for on/off events of the button connected to the display, and also as display ID used in Hauptwerk, so you can send different text messages to multiple LCDs. To change display ID, you have to select it by pressing its button first. Then enter command #914, and the display will show current ID, waiting for new digits. Possible values are from 00 to 99. This parameter is associated with the display itself, and does change when you replace the module to another master board input, or another position in a chain.

### 6.8.4 Display blink - #918

This is simply to make the LCD module blink for a while. If you have chain of displays, they will blink in a sequence determined by their position in the chain.

### 6.8.5 Turn off display - #919

This command clears all text from all LCDs in a chain and turns off the backlit. In this state LCD takes much less power than in normal operation. The display will light up again at first opportunity, whenever anything would be displayed for any reason like entering setup, etc. (available since version 2.7).

### 6.8.6 Display ID show - #920

You can check display IDs of all MiDisp and miniLCD modules in the system at once. First you have to press a button of any LCD in the chain, and then enter command #920. All displays will show their current display ID.

## 6.9 **KEYPAD - MIDI & setup numeric entry**

This small numeric keypad is mostly needed for setup procedures, but it has more uses:

- 10-note keyboard
- numeric entry for user settings in EDIT mode
- value entry (0-127) for any MIDI event
- input for one rotary encoder

When the master board is not in edit mode, buttons 0-9 work like any music keyboard, playing notes by default. Button "#" works like momentary switch connected to the



EDIT header on the master board. It switches the MIDI system into EDIT mode. If MiDisp or miniLCD module is connected, it is clearly indicated there with short prompts for each procedure. To quit from EDIT mode without changing anything - enter "0". There's no command starting with 0, so it immediately returns to normal playing mode. The KEYPAD, as setup entry, does not have to follow the rule of "only 2 keyboards in a chain". It may be connected at the end of 2 BBS-1K chain and it will still work correctly as user settings dial pad.

The electronic board mounted at the back of the keypad (since version 7.0) has pads for **one rotary encoder**, called sometimes "endless potentiometer". It is treated like any other POT input, meaning it can be assigned to any possible event. Most obvious use is Program Change knob (event 129), but it may be transposition, channel shift, or any Continuous Controller.

The encoder can only be with "no detents", or "2 detents per pulse". If 1 detent per pulse is used, every step of the encoder will result in 2 MIDI messages, crossing 2 steps. The encoder must be soldered to the only 3 empty pads by the side of scanner connector. Pin out is the same as in encoder itself: common lead in the middle, and pulse inputs at the sides.

Since version 8.0 there is new feature called "value mode". When you press the "\*" key, the keypad turns into value mode, and every number typed there is assumed to be a value of assigned MIDI event. Just like you'd have a potentiometer, but instead of turning it, directly enter it's position in numbers from 0 to 127. Any kind of MIDI event from chapter 4.2 can be assigned to KEYPAD in value mode. An example is event 129 - Program Change, just type the Program number anytime. Different device IDs are used in KEYPAD, so you can have completely different MIDI events/channels assigned to the encoder, keypad in normal note-like mode, and keypad in value mode.

To return from value mode to regular mode, press "#0". Keypad always powers up in notes mode. You have to press the "\*" key only once after power up to switch it to value mode.

This manual describes all boards and their features with assumption they were delivered in the same box with it, meaning the firmware is in most recent version. Older boards may have some features changed or unavailable.

MIDI boards described here are products of:

"MIDI-hardware" Roman Sowa  
ul. Azotowa 15B, 41-503 Chorzów, Poland  
phone +48 532 425 835  
email [info@midi-hardware.com](mailto:info@midi-hardware.com)  
[www.midi-hardware.com](http://www.midi-hardware.com), [www.midimplant.com](http://www.midimplant.com)

© 2009-2016, Roman Sowa  
boards made and manual printed in Poland, EU