miniMIDI-JOYstick

Advanced User Guide (vers 1b)

Before you read this Guide, please be sure you are familiar with all of the material in the Basic User Guide. Even if you are already used to more advanced MIDI devices, there may be some details in the Basic Guide you'll need. At least give it a skim….

Despite its simple appearance, small size and seemingly simple control, the miniMIDI JOYstick is an amazingly powerful MIDI device! It is capable of over 320 million setup variations. Here's a list of some of the more powerful and advanced features:

- 16 user-programmable Preset memory locations, as well as Realtime memory (which can hold different messages and values than any of the Presets).

- first 5 Presets can be easily selected at power-up.

- can recall any preset instantly by sending it MIDI Program Change messages.

- can report out different messages for any of 8 distinct joystick moves: North, South, East, West, Return from North, Return from South, Return from East, Return from West.

- 9 different message types that can be assigned independently to the 8 joystick moves.

- positions of the joystick can be set to have no output, allowing for more complex combinations of movements and sequences, as well as latching versus toggle behaviors.

- all message types and data values held in Preset memory, or Realtime memory, can be changed by sending it Continuous Controller MIDI messages.

- Program Change Mode (PGM mode) whereby MIDI Program Change messages can be sent out, incrementing and decrementing from a user-definable starting value.

Read on for more on all of these….

Preset Selection

Position of the Joystick on power cycle

Any of the first five Presets (0-4) can be selected by the position of the joystick at the moment the MMJ is connected (powered up). Here's how the presets map to each direction:

Center – Preset 0
East – Preset 1
West – Preset 2
South – Preset 3
North – Preset 4

MIDI Program Change messages

Once the unit is powered up, any of the 16 Presets can be recalled by sending the MMJ one of the first 16 MIDI Program Change messages. These messages can be sent from most any MIDI sequencer, controller or utility application. It should be noted, however, that there are two different ways manufacturers have decided to implement the numbering of Program Changes; 0-127 and 1-128. Because of this, Preset 0 on the MMJ may be recalled by sending it Program Change 0 on one platform, but by Program Change 1 on another. This mildly annoying discrepancy will be obvious quickly, however, as the numbers will either match all be off by one with any particular setup.

Preset Programming

While the MMJ comes with 16 factory presets installed, users are likely to get much more out of it if they change these to match their own gear and musical/artistic intentions. Really, with all of the options available on any one preset (nearly 20 million) and all of the different connected technologies and user interests, these presets represent just a drop in a vast ocean of possibilities.

So, what other types of setups might you want? Well, maybe…

- you need to match the pitchbend range of your specific synth or pitch shifter, or set up your own custom intervals: mapped to the directions of the joystick

- you'd like some particular performance combo, like making the South direction play a low drone for you to improvise over while using East to drop your pitch an octave whenever you like, using West to control the amount of reverb on your instrument and North to control an electronic vibrato.

- you'd like to use each of the four directions to pan your instrument in a surround sound space

- you'd like to control both lighting and audio mixer presets simultaneously, changing them at particular moments during a musical performance or between tunes in a musical set.

- you'd like to use the joystick to both trigger sampling and playback of various loops.

Any of these examples, and so much more, are possible when MIDI-controllable devices and software are coupled with the right MIDI messages from the MiniMIDI JOYstick!

All MMJ Presets can be programmed by sending it specific MIDI Continuous Controller codes (CCs). Just as with recalling Presets, these messages can be sent from most any MIDI sequencer, DAW, controller or utility application. Below, we will look at examples of how to program new Presets some people may want, which will also give insight into how to create other possible variations. For this, you will need to refer to the MIDI Implementation Chart at the end of this Guide (which should make more sense after you read through the examples).

Example 1a: Changing the values of a Preset's Controller Codes

OK, let's start by looking at Factory Preset 2. This preset contains Modulation (CC1) on North/South and Expression (CC 11) on East/West. Here's what is sent by the MMJ (with all values returning to 0 in the center):

North = Modulation (Controller 1), value 64
South = Modulation (Controller 1), value 127

East = Expression (Controller 11), value 127
West = Expression (Controller 11), value 64

Well, what if the South modulation effect, which is set to full, is too much? Maybe you'd rather it be set to 100 rather than 127? To do this takes sending the MMJ two short MIDI messages….

Looking at the "Presets (write)" section of the MIDI Implementation Chart, we see that CC40 is used to select both which Preset you want to write to and which of the 8 directions you are changing. So, according to the chart, to change the South value on Preset 2 we'd send CC40 with a value of 18 (see "Preset & Direction" part of the Chart).

Next, to set the value to 100 we look again to "Presets (write)" and see that the value of a CC is changed by sending the MMJ CC43, so we'd send CC43 with a value of 100. Now, the next time we recall Preset 2, South will send Modulation 100.

Note: any new data values will now be sent to South on Preset 2 until CC40 is updated (which specifies where new data values are to go). This means updating this value again will only take one message!

Example 1b: Changing the values of a Preset's Controller Codes – non zero centers

Now let's look at another aspect of this same Preset (#2) that may need a change. If you are using a synth where Expression (CC 11) maps to a volume related function, rather than just a tonally related one, like brightness, you may not want it to go to 0 in the center, as that may cause no audio to be output at all! So, let's remap the return from West to center and East to center to output a value of 64 instead of 0. This takes sending four short MIDI messages: (2 for each direction)

1) Send CC40, 20 then CC43, 64 to set "release East" on Preset 2 to 64

2) Send CC40, 21 then CC43, 64 to set "release West" on Preset 2 to 64

Again, this change will happen the next time Preset 2 is recalled. It does not happen right away, as it is a changed to the *stored* values.

Of course, as the West value of Preset 2 is 64, you'd probably want to increase it some as well to make sure there's an audible difference between West and Center for Expression, let's say 100…. So, send CC40, 17 followed by CC43, 100.

Example 1c: Changing a Controller Code (controller #)

If you liked Preset 2 with Modulation North/South and Expression to the West, but wanted East to be the Sustain Pedal (CC 64), you would send the MMJ CC40, 16 (to choose East on Preset 2) then CC42, 64 (to set the CC # to 64).

Example 2: Changing message type

The MMJ joystick messages can be any of eight types: Note On, Note Off, Key Aftertouch, Channel Aftertouch, Program Change, Continuous Controller, Pitchbend, and NA (not assigned). These are listed in the chart under the “Type Config” heading.

Message types, and their designated MIDI channels, are chosen using CC 41 (as per the “Presets (write)” section of the chart. To assign South on Preset 7 to Channel Aftertouch (on Channel 5) with a Value of 80 you’d send the MMJ CC 40, 58 (Preset 7, South) then CC 41, 84 (Chan Aftertouch on Channel 5) followed by CC 43, 80 (value of 80).

Example 3: Pitchbend Presets

Pitchbend messages are different from the other types available to map to the MMJ joystick in two important ways.

First, they have a much larger range of values (16384 instead of 128), and to do this it combines two 7-bit values (from two bytes of data) to get a 14-bit result. This extra data is necessary to get the precision needed to tune exact intervals over a wide range of octaves.

Second, pitchbends are relative functions. The interval that will be output for a given pitchbend value can vary depending on the setting of the *pitchbend range* on the receiving device. Think about it this way: if the device is set to a pitchbend range of +/- 3 semitones, sending it a full value pitchbend (16384) will cause an output of 3 semitones (a minor 3rd) higher than the input, but if that same device is set to range of +/- 12 semitones then the same value will cause an output of 12 semitones (an octave) higher.

So, before you determine what pitchbend data to send you must know what the pitchbend range of the receiving device is set to (which can be different for each preset on that device.)

NOTE: To make it easier to figure out what the two bytes of data should be to get a desired interval over a specified pitchbend range, this manual includes a chart of four of the most common ranges (below).

Let’s consider how to change Preset 0, which is a 4-way Pitchbend preset that assumes a range of +/- 3 semitones, to a Preset for a +/- 12 semitone range with intervals of -12 (down an octave), - 7 (down a perfect fifth), -5 (down a perfect 4th), and -2 (down a whole step). Looking at the pitchbend chart, go to the column that says “+/- 12 semitones Bend Values” and note the values in the rows for -12, -7, -5 and -2 (0, 3413, 4778, and 6826). Then look to the righthand section of the chart that shows the MSB and LSB pairs of bytes for each of these values.

0 = 0, 0
3413 = 26, 85
4778 = 37, 42
6826 = 53, 42

Looking back at the MIDI Implementation Chart again, you can see that these MSB and LSB values are used with CC 43 and 42 respectively to get the full 14-bit value.

So, as Preset 0 is already Pitchbend, we do not need to change the message type, just the values of MSB and LSB for each direction. So, to get South = -12, East = -7, West = -5 and North = -2 we send the following codes:

CC 40, 0 (Preset 0, East)
CC 43, 26
CC 42, 85

CC 40, 1 (Preset 0, West)
CC 43, 37
CC 42, 42

CC 40, 2 (Preset 0, South)
CC 43, 0
CC 42, 0

CC 40, 2 (Preset 0, North)
CC 43, 53
CC 42, 42

Now Preset 0 is transformed!

Realtime Programming

When a Preset is recalled on the MMJ, it is loaded into the Realtime (or Runtime) memory. In the previous section we looked at how to change what is stored in these Preset memory locations, but now we will see how to change the messages and values in the Realtime memory directly, without changing or recalling Presets at all. Once again, this is done through the use of Continuous Controller codes, and the particulars of these can once again be found in the MIDI Implementation Chart (in the "Realtime Variables" section).

Programming Realtime memory works very similarly to programming presets. To change type of message associated with a particular direction of the joystick, CCs 16-23 are used (see chart). The data values are CCs 24-31 (LSB) and 32-39 (MSB).

So, let's go back to example 1 from the Preset Programming section and say we wanted to change the South Modulation value to 100, from 127. We'd send CC 34, 100. Just one message and it's done. Keep in mind, however, that this new value will not be remembered after a power cycle as it is not stored in Preset memory.

Let's consider a more complicated change now, requiring changing message type and all data values. If you wanted to make pressing North send a low-low c, which is MIDI Note 36 (second ledger below the bass clef staff), on channel 1… with a dynamic level (velocity) of 110, you'd send three CC messages. First, the type change: CC 19, 16 (Note On for North on channel 1).

Next the note number: CC 33, 36 (South MSB = note 36). Finally, the velocity: CC 26, 100 (South LSB = velocity 100).

Program Change Mode

The MiniMIDI JOYstick can send MIDI Program Change messages in the 0-127 range. The types of messages are commonly used to remotely change presets on receiving devices (examples are: to change sounds on a synthesizer, to change effects on an effects unit, to recall stored settings on a digital mixer, etc…).

While Program Change messages can be added to Presets and/or loaded into Realtime memory, we avoided talking about them until now due to the fact that they can be used in two different ways. One way is to assign them to any direction and have them send out a fixed value (like all the other message types). The more specialized (and powerful) way is to use them in a special Program Change Mode, which can be activated and deactivated at any time by sending a CC 15, 127 message.

The reason for the special mode is that Program Changes work a little differently than all of the other message types here in that their value can be incremented and decremented from a user selectable staring value rather than outputting fixed values. Pressing East causes the program number to be incremented by 1, pressing West causes the value to be decreased (decremented) by 1. This is sometimes referred to as an increment/decrement (INC/DEC) mode.

If, for example, 17 is set as the starting value in Pgm mode, then pressing East will send a Program Change 18 message, pressing West will send a Program Change 16 message. If West is repeatedly pressed, the number will lower by 1 each time until 0 is reached, at which time further West presses will start the number descending from 127. Similarly, if the number is increased to 127 by pressing East, further presses East will start the count ascending from 0.

The Program Change mode state (on or off) can be saved within the presets or, as stated before, turned on and off through CC 15, 127. It should be noted, however, that the starting value of the INC/DEC function is a system wide setting, meaning independent starting values are not saved as part of the 16 presets. The starting value is updated by way of CC 46, but it does not take effect until the unit is power cycled.

MMJ units are shipped with their stored Program Change starting value set to 0.

Joystick Calibration

All MiniMIDI JOYsticks are calibrated at the factory and may never need further calibration (that is the intent). Over time and/or a lot of use, however, it is possible the mechanical and/or electrical nature of the joystick part itself may change. If this change is great enough, one or more directions may no longer respond correctly. If this happens, the joystick may be able to be brought back into full use through recalibration (changing its sensitivity to changes in direction).

So, how can one recalibrate the joystick? Why, by using a Continuous Controller message of course! In this case, that CC is 47. The factory value is usually around 45, so if the joystick is not responding to a particular direction, lower values could be tried. If the joystick is reporting too many values, or thinks center is a move in one of the four directions, then a higher value may solve the issue.

If a joystick cannot be brought back into working order through calibration, it likely needs to be replaced. This is actually a relatively simple, inexpensive process. The part used is the same as for the popular PSP 3000 (Sony PlayStation) so replacement parts are easy to come by.

A Note on LSB and MSB data bytes

The MMJ sets aside two bytes of data to accompany every MIDI message type. These are noted in the MIDI Implementation Chart as MSB and LSB, which are broken down even further in the notes on in the lower left of that chart (as well as below).

As already touched upon, these bytes set things like Continuous Controller code number (MSB) and Continuous Controller value (LSB). One common use example is that Modulation (CC 1) is set through the LSB (rather than, say, Expression which is CC 11, and then the value (amount) of modulation is set through the value in the LSB.

If the message type is a Note On, however, the LSB determines which note is to be played, and the LSB sets the dynamic level (aka. MIDI velocity).

Here's an overview of what each data byte with different message types:

MSB

Controller number (type)
Note On velocity
Pitchbend MSB (course value)
Polyphonic (key) Aftertouch value

LSB

Controller value
Note On # (pitch)
Note Off # (pitch)
Pitchbend LSB (fine value)
Program Change value
Channel Aftertouch (monophonic) value
Polyphonic (key) Aftertouch note number (pitch)

MiniMIDI JOYstick - MIDI Implementation Chart (v1b)

Realtime Variables		
Controller Code 16	Type Config (0-127)	set press East type
Controller Code 17	Type Config (0-127)	set press West type
Controller Code 18	Type Config (0-127)	set press South type
Controller Code 19	Type Config (0-127)	set press North type
Controller Code 20	Type Config (0-127)	set release East type
Controller Code 21	Type Config (0-127)	set release West type
Controller Code 22	Type Config (0-127)	set release South type
Controller Code 23	Type Config (0-127)	set release North type
Controller Code 24	data, LSB (0-127)**	set press East argument 1
Controller Code 25	data, LSB (0-127)**	set press West argument 1
Controller Code 26	data, LSB (0-127)**	set press South argument 1
Controller Code 27	data, LSB (0-127)**	set press North argument 1
Controller Code 28	data, LSB (0-127)**	set release East argument 1
Controller Code 29	data, LSB (0-127)**	set release West argument 1
Controller Code 30	data, LSB (0-127)**	set release South argument 1
Controller Code 31	data, LSB (0-127)**	set release North argument 1
Controller Code 32	data, MSB (0-127)**	set press East argument 2
Controller Code 33	data, MSB (0-127)**	set press West argument 2
Controller Code 34	data, MSB (0-127)**	set press South argument 2
Controller Code 35	data, MSB (0-127)**	set press North argument 2
Controller Code 36	data, MSB (0-127)**	set release East argument 2
Controller Code 37	data, MSB (0-127)**	set release West argument 2
Controller Code 38	data, MSB (0-127)**	set release South argument 2
Controller Code 39	data, MSB (0-127)**	set release North argument 2
Controller Code 15	127	toggle Program Change Mode

Presets (recall)		
Program Change	0-15	select a Preset (0-15)

Presets (write)		
Controller Code 40	Preset and direction	select Preset / direction to configure
Controller Code 41	type config (0-127)	set message type and channel
Controller Code 42	LSB data (0-127)**	CC value, Note, Bend LSB, Pgm, Touch val
Controller Code 43	MSB data (0-127)**	CC #, velocity, Bend MSB, PolyTch val
Controller Code 44	Preset # (0-15)	enable Pgm Mode
Controller Code 45	Preset # (0-15)	disable Pgm Mode

System wide (changes after power cycle)		
Controller Code 46	0-127	select starting Program # (for Pgm Mode)
Controller Code 47	0-127	Calibrate stick (trigger distance from center)*

* Calibration is set at the factory (to around 45) and should not need to be changed. If the joystick drifts over time, however, the joystick can be recalibrated manually. If calibration values are set too high, moving the joystick will stop triggering new values; if set too low it may trigger incorectly or unintentionally. Note: eventually these joystick wear out abut they can be replaced fairly easily and cheaply. They are the same as used in the PSP 3000 game system.

**MSB/LSB Data Outline
MSB - CC #, velocity, Bend MSB, Poly Touch value
LSB - CC value, Note #, Bend LSB, Pgm #, Chan Touch value, Poly Touch Note

Preset & Direction		
Sel #	preset #	direction
0	0	press East
1	0	press West
2	0	press South
3	0	press North
4	0	release East
5	0	release West
6	0	release South
7	0	release North
8	1	press East
9	1	press West
10	1	press South
11	1	press North
12	1	release East
13	1	release West
14	1	release South
15	1	release North
16	2	press East
17	2	press West
18	2	press South
19	2	press North
20	2	release East
21	2	release West
22	2	release South
23	2	release North
24	3	press East
25	3	press West
26	3	press South
27	3	press North
28	3	release East
29	3	release West
30	3	release South
31	3	release North
32	4	press East
33	4	press West
34	4	press South
35	4	press North
36	4	release East
37	4	release West
38	4	release South
39	4	release North
40	5	press East
41	5	press West
42	5	press South
43	5	press North
44	5	release East
45	5	release West
46	5	release South
47	5	release North
48	6	press East
49	6	press West
50	6	press South
51	6	press North
52	6	release East
53	6	release West
54	6	release South
55	6	release North
56	7	press East
57	7	press West
58	7	press South
59	7	press North
60	7	release East
61	7	release West

Type Config		
Sel #	midi type	channel
0	Note Off	1
1	Note Off	2
2	Note Off	3
3	Note Off	4
4	Note Off	5
5	Note Off	6
6	Note Off	7
7	Note Off	8
8	Note Off	9
9	Note Off	10
10	Note Off	11
11	Note Off	12
12	Note Off	13
13	Note Off	14
14	Note Off	15
15	Note Off	16
16	Note On	1
17	Note On	2
18	Note On	3
19	Note On	4
20	Note On	5
21	Note On	6
22	Note On	7
23	Note On	8
24	Note On	9
25	Note On	10
26	Note On	11
27	Note On	12
28	Note On	13
29	Note On	14
30	Note On	15
31	Note On	16
32	Key Aftertouch	1
33	Key Aftertouch	2
34	Key Aftertouch	3
35	Key Aftertouch	4
36	Key Aftertouch	5
37	Key Aftertouch	6
38	Key Aftertouch	7
39	Key Aftertouch	8
40	Key Aftertouch	9
41	Key Aftertouch	10
42	Key Aftertouch	11
43	Key Aftertouch	12
44	Key Aftertouch	13
45	Key Aftertouch	14
46	Key Aftertouch	15
47	Key Aftertouch	16
48	Controller Code	1
49	Controller Code	2
50	Controller Code	3
51	Controller Code	4
52	Controller Code	5
53	Controller Code	6
54	Controller Code	7
55	Controller Code	8
56	Controller Code	9
57	Controller Code	10
58	Controller Code	11
59	Controller Code	12
60	Controller Code	13
61	Controller Code	14

MIDI Implementation Chart, continued

62	7	release South
63	7	release North
64	8	press East
65	8	press West
66	8	press South
67	8	press North
68	8	release East
69	8	release West
70	8	release South
71	8	release North
72	9	press East
73	9	press West
74	9	press South
75	9	press North
76	9	release East
77	9	release West
78	9	release South
79	9	release North
80	10	press East
81	10	press West
82	10	press South
83	10	press North
84	10	release East
85	10	release West
86	10	release South
87	10	release North
88	11	press East
89	11	press West
90	11	press South
91	11	press North
92	11	release East
93	11	release West
94	11	release South
95	11	release North
96	12	press East
97	12	press West
98	12	press South
99	12	press North
100	12	release East
101	12	release West
102	12	release South
103	12	release North
104	13	press East
105	13	press West
106	13	press South
107	13	press North
108	13	release East
109	13	release West
110	13	release South
111	13	release North
112	14	press East
113	14	press West
114	14	press South
115	14	press North
116	14	release East
117	14	release West
118	14	release South
119	14	release North
120	15	press East
121	15	press West
122	15	press South
123	15	press North
124	15	release East
125	15	release West
126	15	release South
127	15	release North

62	Controller Code	15
63	Controller Code	16
64	Program Change	1
65	Program Change	2
66	Program Change	3
67	Program Change	4
68	Program Change	5
69	Program Change	6
70	Program Change	7
71	Program Change	8
72	Program Change	9
73	Program Change	10
74	Program Change	11
75	Program Change	12
76	Program Change	13
77	Program Change	14
78	Program Change	15
79	Program Change	16
80	Chan. Aftertouch	1
81	Chan. Aftertouch	2
82	Chan. Aftertouch	3
83	Chan. Aftertouch	4
84	Chan. Aftertouch	5
85	Chan. Aftertouch	6
86	Chan. Aftertouch	7
87	Chan. Aftertouch	8
88	Chan. Aftertouch	9
89	Chan. Aftertouch	10
90	Chan. Aftertouch	11
91	Chan. Aftertouch	12
92	Chan. Aftertouch	13
93	Chan. Aftertouch	14
94	Chan. Aftertouch	15
95	Chan. Aftertouch	16
96	Pitch Bend	1
97	Pitch Bend	2
98	Pitch Bend	3
99	Pitch Bend	4
100	Pitch Bend	5
101	Pitch Bend	6
102	Pitch Bend	7
103	Pitch Bend	8
104	Pitch Bend	9
105	Pitch Bend	10
106	Pitch Bend	11
107	Pitch Bend	12
108	Pitch Bend	13
109	Pitch Bend	14
110	Pitch Bend	15
111	Pitch Bend	16
112	Do Nothing	X
113	Do Nothing	X
114	Do Nothing	X
115	Do Nothing	X
116	Do Nothing	X
117	Do Nothing	X
118	Do Nothing	X
119	Do Nothing	X
120	Do Nothing	X
121	Do Nothing	X
122	Do Nothing	X
123	Do Nothing	X
124	Do Nothing	X
125	Do Nothing	X
126	Do Nothing	X
127	Do Nothing	X

Pitchbend Values (14-bit to MSB/LSB to intervals)

Semitones	+/- 2 semitones Bend Values	+/- 3 semitones Bend Values	+/- 12 semitones Bend Values	+/- 24 semitones Bend Values
24				16383
23				16042
22				15700
21				15359
20				15018
19				14676
18				14335
17				13994
16				13653
15				13311
14				12970
13				12629
12			16383	12287
11			15700	11946
10			15018	11605
9			14335	11263
8			13653	10922
7			12970	10581
6			12287	10239
5			11605	9898
4			10922	9557
3		16383	10239	9215
2	16383	13653	9557	8874
1	12287	10922	8874	8533
0	8192	8192	8192	8192
-1	4096	5461	7509	7850
-2	0	2731	6826	7509
-3		0	6144	7168
-4			5461	6826
-5			4778	6485
-6			4096	6144
-7			3413	5802
-8			2731	5461
-9			2048	5120
-10			1365	4778
-11			683	4437
-12			0	4096
-13				3754
-14				3413
-15				3072
-16				2731
-17				2389
-18				2048
-19				1707
-20				1365
-21				1024
-22				683
-23				341
-24				0

Bend Val	MSB	LSB
16383	127	127
16042	125	42
15700	122	84
15359	119	127
15018	117	42
14676	114	84
14335	111	127
13994	109	42
13653	106	85
13311	103	127
12970	101	42
12629	98	85
12287	95	127
11946	93	42
11605	90	85
11263	87	127
10922	85	42
10581	82	85
10239	79	127
9898	77	42
9557	74	85
9215	71	127
8874	69	42
8533	66	85
8192	64	0
7850	61	42
7509	58	85
7168	56	0
6826	53	42
6485	50	85
6144	48	0
5802	45	42
5461	42	85
5120	40	0
4778	37	42
4437	34	85
4096	32	0
3754	29	42
3413	26	85
3072	24	0
2731	21	43
2389	18	85
2048	16	0
1707	13	43
1365	10	85
1024	8	0
683	5	43
341	2	85
0	0	0