

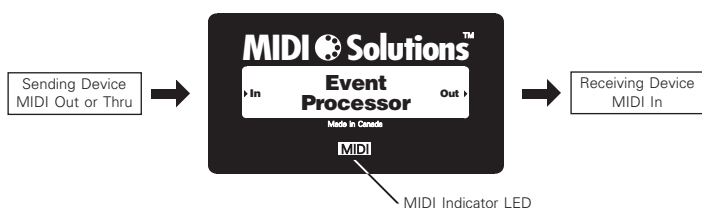
Event Processor

OPERATING INSTRUCTIONS

INTRODUCTION

Thank you for purchasing the MIDI Solutions Event Processor. The MIDI Solutions Event Processor offers 10 settings, each of which can be programmed with a unique MIDI processing function. The Event Processor is MIDI-powered and requires no batteries or power supply to operate.

These operating instructions provide complete details on the System Exclusive messages needed to program the Event Processor. **If you are using the Programming Tools software these messages are created automatically for you.**



CONNECTIONS

The MIDI **Out** of the Event Processor can be left disconnected during programming.

Once the Event Processor is programmed, it can be inserted anywhere in your MIDI setup. Connect the MIDI **Out** or **Thru** of the sending MIDI device to the MIDI **In** of the Event Processor, and the MIDI **Out** of the Event Processor to the MIDI **In** of the receiving MIDI device. It is recommended that the number of MIDI Solutions products powered by a single MIDI **Out** or **Thru** be limited to four.

OPERATION

The Event Processor's MIDI Indicator LED will light as soon as the sending device is turned on, and flashes whenever MIDI data passes through the unit. Incoming MIDI events are processed according to the Event Processor's programmed settings as described in these instructions. All other MIDI events are sent unchanged to the MIDI **Out**.

PROGRAMMING

The Event Processor contains 10 settings, each of which can be programmed with a unique MIDI processing function. The settings are programmed by sending the Event Processor MIDI System Exclusive programming messages from a computer with a MIDI interface, as described in detail in the following instructions. All values must be entered in hexadecimal - for decimal to hexadecimal conversions, see the chart at the end of these instructions. Upon receipt of a System Exclusive programming message, the MIDI indicator LED flashes rapidly for about one second to indicate that the setting has been stored. Settings are retained in memory even after power is removed from the unit.

Clear Memory

To **clear all** of the Event Processor's settings, send it the following System Exclusive message:

F0 00 00 50 28 00 F7

To **clear a single setting** of the Event Processor, send it the following System Exclusive message:

F0 00 00 50 28 00 ss F7

where **ss** = setting to be cleared (00 - 09)

Filter MIDI Event

To program the Event Processor to filter "MIDI Event A", send it the following:

F0 00 00 50 28 01 ss (aa bb cc dd) F7

followed by **"MIDI Event A"**

where **ss** = setting to be programmed (00 - 09)

(aa bb cc dd) is optional, specifies value ranges and variables (see reverse)

"MIDI Event A" can be a maximum of 32 bytes

Example: To program setting #3 to filter all Program Change events on MIDI channel 16, send the Event Processor the following:

F0 00 00 50 28 01 02 00 02 00 7F F7

CF 00

Map MIDI Event

To program the Event Processor to map "MIDI Event A" to "MIDI Event B", send it the following:

F0 00 00 50 28 02 mm (aa bb cc dd) F7

followed by **"MIDI Event A"**

followed by **"MIDI Event B"**

where **ss** = setting to be programmed (00 - 09)

(aa bb cc dd) is optional, specifies value ranges and variables (see reverse)

"MIDI Event A" and **"MIDI Event B"** combined can contain a maximum of 32 bytes.

Example: To program setting #7 to map the System Exclusive message F0 00 11 22 33 F7 to the System Exclusive message F0 44 55 66 77 F7, send the Event Processor the following:

F0 00 00 50 28 02 06 F7

F0 00 11 22 33 F7

F0 44 55 66 77 F7

Trigger MIDI Event

To program the Event Processor to trigger "MIDI Event B" when the specified values of "MIDI Event A" move into a selected range, send it the following:

F0 00 00 50 28 03 ss (aa bb cc dd) F7

followed by **"MIDI Event A"**

followed by **"MIDI Event B"**

where **ss** = setting to be programmed (00 - 09)

(aa bb cc dd) specifies value ranges and variables (see reverse)

"MIDI Event A" must contain at least one value range. **"MIDI Event B"** cannot contain any value ranges. **"MIDI Event A"** and **"MIDI Event B"** combined can contain a maximum of 32 bytes.

Example: To program setting #1 to trigger a middle C (Note-on #60) of velocity 127 on MIDI channel 1 whenever the value of Control Change #7 on MIDI channel 1 moves above 63 (i.e. into the range 64 - 127), send the Event Processor the following:

F0 00 00 50 28 03 00 01 03 40 7F F7

B0 07 00

90 3C 7F

Define a Sequence of MIDI Events

To program the Event Processor to cycle through a sequence of MIDI events in response to "MIDI Event A", send it the following:

F0 00 00 50 28 04 ss ff ee tt jj (aa bb cc dd) F7

followed by "MIDI Event A"

where **ss** = setting to be programmed (00 - 09)

ff = first setting of sequence (00 - 09)

ee = #events/step (max. 15) **tt** = total #steps (max. 31)

jj = jump to step# (01 - 09), DEC step# (7E), INC step# (7F)

(aa bb cc dd) is optional, specifies value ranges and variables (see reverse)

Defining events within this sequence is described below.

Define Events with a Sequence

To define an event within a sequence, send the Event Processor the following:

F0 00 00 50 28 05 ss (aa bb cc dd) F7

followed by "MIDI Event B"

where **ss** = setting to be programmed (00 - 09)

(aa bb cc dd) is optional, specifies value ranges and variables (see reverse)

"Event B" is the event in the sequence.

Note: If ranges of values are specified, ensure that the incoming event contains those values. Failure to do this will result in unpredictable data.

Example: To program setting #1 to cycle through Program Change 1 on MIDI channel 1 followed by Program Change 2 on MIDI channel 1 in response to Note-on #60 events on MIDI channel 1, send the Event Processor the following:

F0 00 00 50 28 04 00 01 01 02 7F 01 03 01 7F F7

90 3C 40

F0 00 00 50 28 05 01 F7

C0 00

F0 00 00 50 28 05 02 F7

C0 01

(in the above example, Program Change 1 and 2 are stored in settings #2 and #3 respectively)

Turn Setting On/Off

To program the Event Processor to turn a setting on or off when a value of "MIDI Event A" moves into a specified range, send it the following:

F0 00 00 50 28 06 ss tt xx (aa bb cc dd) F7

followed by "MIDI Event A"

where **ss** = setting to be programmed (00 - 09)

tt is set according to table below, **xx** = setting# to turn On/Off

(aa bb cc dd) specifies values and variables (see reverse)

	If values are out of range →	Turn setting ON	Turn setting OFF	Do nothing
If values are in range:	Turn setting ON	N/A	tt = 06	tt = 04
	Turn setting OFF	tt = 09	N/A	tt = 08
	Do nothing	tt = 01	tt = 02	N/A

Example: To program setting #1 to turn on setting #10 when Control Change #64 on MIDI channel 1 is in the range 64 - 127, and turn it off setting #10 when the Control Change value is out of the range, send the Event Processor the following:

F0 00 00 50 28 06 00 06 09 01 03 40 7F F7

B0 40 00

Store Incoming Value in a Variable

To program the Event Processor to store an incoming value in a variable, send it the following:

F0 00 00 50 28 07 ss (aa bb xx yy) F7

followed by "MIDI Event A"

where **ss** = setting to be programmed (00 - 09)

aa = 00: Event A, Value X (**bb** ignored)

aa = 01: Event A, Value Y (**bb** ignored)

aa = 02: store Value X in Variable **bb** (bb = 00 or 01)

aa = 03: store Value Y in Variable **bb** (bb = 00 or 01)

xx yy = incoming (aa = 00, 01) or stored (aa = 02, 03) range

Example: To program setting #1 to store volume (CC#7) on MIDI channel 5 in Variable 1, with the incoming range 0 - 127 scaled to the stored range 0 - 100, send the Event Processor the following:

F0 00 00 50 28 07 00 01 03 00 7F 03 00 00 64 F7

B4 07 00

Value Range and Variable Specification

Incoming and outgoing events can each include up to two value ranges or variables (defined by **aa bb cc dd**), where **aa** is set as shown below, **bb** specifies the byte# (placement in incoming or outgoing event), and **cc dd** specifies the range.

aa = 00: Event A (incoming event), Value X

aa = 01: Event A (incoming event), Value Y

aa = 02: Event B (outgoing event), Value X

aa = 03: Event B (outgoing event), Value Y

aa = 1x: Event A (incoming event), Variable x = 0 or 1

aa = 2x: Event B (outgoing event), Variable x = 0 or 1

aa definitions must appear in order of **aa** lowest to highest. If **aa** = 1x or 2x then **cc dd** is ignored.

MIDI CONTROL CHANGE TABLE

Decimal	Hex	Control Function
0	00H	Bank Select
1	01H	Modulation wheel or lever
2	02H	Breath Controller
3	03H	Undefined
4	04H	Foot controller
5	05H	Portamento time
6	06H	Data entry MSB
7	07H	Channel Volume
8	08H	Balance
9	09H	Undefined
10	0AH	Pan
11	0BH	Expression Controller
12-13	0C-0DH	Effect Controls 1-2
14-15	0E-0FH	Undefined
16-19	10-13H	General Purpose Controllers (#s 1-4)
20-31	14-1FH	Undefined
32-63	20-3FH	LSB values for 0-31
64	40H	Damper pedal (sustain)
65	41H	Portamento On/Off
66	42H	Sostenuto
67	43H	Soft pedal
68	44H	Legato Fsw (vv=00-3F: Normal, 40-7F: Legato)
69	45H	Hold 2
70	46H	Sound Controller 1 (default: Sound Variation)
71	47H	Sound Controller 2 (default: Timbre/Harmonic Content)
72	48H	Sound Controller 3 (default: Release Time)
73	49H	Sound Controller 4 (default: Attack Time)
74	4AH	Sound Controller 5 (default: Brightness)
75-79	4B-4FH	Sound Controllers 6-10 (no defaults)
80-83	50-53H	General Purpose Controllers (#s 5-8)
84	54H	Portamento Control
85-90	55-5AH	Undefined
91	5BH	Effects 1 Depth (formerly External Effects Depth)
92	5CH	Effects 2 Depth (formerly Tremolo Depth)
93	5DH	Effects 3 Depth (formerly Chorus Depth)
94	5EH	Effects 4 Depth (formerly Celeste (Detune) Depth)
95	5FH	Effects 5 Depth (formerly Phaser Depth)
96,97	60H,61H	Data increment, Data decrement
98,99	62H,63H	Non-Registered Parameter Number LSB, MSB
100,101	64H,65H	Registered Parameter Number LSB, MSB
102-119	66-77H	Undefined
120-127	78-7FH	Reserved for Channel Mode Messages

HEXADECIMAL CONVERSION TABLE

Dec/Hex	16	10	32	20	48	30	64	40	80	50	96	60	112	70
1 01	17	11	33	21	49	31	65	41	81	51	97	61	113	71
2 02	18	12	34	22	50	32	66	42	82	52	98	62	114	72
3 03	19	13	35	23	51	33	67	43	83	53	99	63	115	73
4 04	20	14	36	24	52	34	68	44	84	54	100	64	116	74
5 05	21	15	37	25	53	35	69	45	85	55	101	65	117	75
6 06	22	16	38	26	54	36	70	46	86	56	102	66	118	76
7 07	23	17	39	27	55	37	71	47	87	57	103	67	119	77
8 08	24	18	40	28	56	38	72	48	88	58	104	68	120	78
9 09	25	19	41	29	57	39	73	49	89	59	105	69	121	79
10 0A	26	1A	42	2A	58	3A	74	4A	90	5A	106	6A	122	7A
11 0B	27	1B	43	2B	59	3B	75	4B	91	5B	107	6B	123	7B
12 0C	28	1C	44	2C	60	3C	76	4C	92	5C	108	6C	124	7C
13 0D	29	1D	45	2D	61	3D	77	4D	93	5D	109	6D	125	7D
14 0E	30	1E	46	2E	62	3E	78	4E	94	5E	110	6E	126	7E
15 0F	31	1F	47	2F	63	3F	79	4F	95	5F	111	6F	127	7F

WARRANTY

MIDI Solutions Inc. warrants this product to be free from defects in material and workmanship for a period of one (1) year from date of purchase. This warranty is void if the product has been damaged by accident, misuse, alteration, unauthorized repairs or other causes not arising out of defects in material or workmanship. Under no circumstances will MIDI Solutions be liable for any loss of profits, benefits, time, interrupted operation, commercial loss, or consequential damages arising out of the use or inability to use the product. MIDI Solutions specifically disclaims any implied warranties of merchantability and fitness for a particular purpose. If the product requires service, a Return Merchandise Authorization (RMA) number must be obtained from MIDI Solutions and the product must be shipped prepaid to a specified Service Center. MIDI Solutions will repair or replace the product at our discretion and will pay return shipping fees. The customer is responsible for any damage or loss sustained during shipment in any direction.