

**CHROMA POLARIS**  
**MIDI IMPLEMENTATION**

Copyright (c) 1986  
Fender Musical Instruments Corporation  
1130 Columbia Street  
Brea, California 92621

CHROMA POLARIS MIDI IMPLEMENTATION

TABLE OF CONTENTS

Polaris Architecture . . . . . 1

MIDI Implementation Overview . . . . . 1

Input Commands Recognized . . . . . 4

Output Commands Generated . . . . . 7

System Exclusive Format . . . . . 9

Polaris Object Numbers . . . . . 11

Data Stream Numbers and Messages . . . . . 12

Parameter Numbers and Values . . . . . 13

MIDI Implementation Chart . . . . . 14

MIDI Implementation Chart Notes . . . . . 15

CHROMA POLARIS MIDI IMPLEMENTATION**Polaris Architecture**

The internal architecture of the Polaris has three primary sources of musical event data:

**Main** -- This generates information from the entire keyboard, or part of the keyboard if it is split. It generates information from the various performance inputs (levers, pedal and footswitch). It also generates program changes and parameter changes.

**Link** -- This generates keyboard information and performance control information, but only if a link is in effect. It can also generate program changes when a link is set up or cleared, and can generate parameter changes if the Edit Link function is invoked on the control panel.

**Sequencer** -- This generates all information when a sequence is played.

The Polaris has eight destinations for information, called "logical instruments". The system of logical instruments is characterized by the following features:

Each logical instrument may either exist or not. If a logical instrument exists, it contains the definition of a sound.

A logical instrument that exists may have voices allocated to it to generate the sound. The allocation of voices to instruments is dynamic, allowing each instrument to play polyphonically, subject to competition from other instruments.

The logical instruments are numbered from 0 to 7, and the first three are additionally named the Main, Link and Sequencer instruments.

The first three instruments can receive data from the internal sources, the MIDI Interface and the Chroma Interface. The remaining five are only accessible to the MIDI Interface and the Chroma Interface.

The connections between the Main, Link and Sequencer sources and the Main, Link and Sequencer instruments can be individually broken.

The sequencer is capable of recording whatever is sent to the Main instrument, regardless of whether it comes from the keyboard, the MIDI Interface or the Chroma Interface.

**MIDI Implementation Overview**

The implementation of MIDI in the Chroma Polaris is distinguished by the following characteristics:

The basic channel number (hereafter referred to as "b") is assignable by the user to MIDI channel 1 through 16. The scale of MIDI channel numbers is considered circular, so if b equals 16, b+1 equals 1.

Channel b corresponds to the Main data stream in the Polaris, channel b+1 corresponds to the Link data stream, and channel b+2 corresponds to the Sequencer data stream.

There are three software switches, called the MIDI Input Switches, that selectively enable reception on each of the three channels. There are three more software switches, called the MIDI Output Switches, that selectively enable transmission on each of the three channels.

There are three switches, called the Local Control Switches, that selectively enable the internal data flow on these channels. (There are also three switches, called the Chroma Out Switches, that selectively enable transmission of each channel over the Chroma interface.)

There is a configuration parameter, called MIDI Extra Channels, that allows reception on from 0 to 5 more MIDI channels, numbered b+3 through b+7. This has no effect on transmission. These channels feed information into logical instruments 3 through 7.

There is a software switch, called the MIDI Panel Switch, that determines how much information is received and transmitted over all the MIDI channels. When off, control panel related information (specifically program changes and parameter changes) is filtered out of all MIDI input and output. When on, this information is received and transmitted normally. This switch should be off if the Polaris is communicating directly with a different type of instrument. It should be on if the Polaris is connected to another Polaris or to a computer that understands the Polaris panel information. There is a similar switch for the Chroma Interface.

Sending a Program Change command with a non-zero program number into the Polaris on a particular MIDI channel tells the Polaris to create the corresponding logical instrument using the specified program. Sending a Program Change command with a program number of zero tells the Polaris to delete the corresponding logical instrument. A Program Change 0 can therefore be thought of as a command to "select no program", hence disabling that MIDI channel from further sound generation.

The Main Instrument is set up at all times unless it has been explicitly deleted through the MIDI Interface or the Chroma Interface. The Link Instrument is created whenever a link is set up and deleted whenever the link is cleared. The Sequencer Instrument is created whenever a sequence is played, and deleted when the sequence is stopped.

Instruments 3 through 7 don't exist until they are created through the MIDI Interface or the Chroma Interface. This implies that the MIDI Panel Switch must be on if the Extra MIDI Channels are to be used (as it is necessary to select a non-zero program for a channel before it will respond to notes).

The Polaris MIDI supports Poly, Mono, Poly/Omni and Mono/Omni modes. The system always powers up in Poly/Omni mode; the mode can be changed from the panel as well as through the MIDI interface.

When either Omni Mode is in effect, information received on all 16 MIDI channels is routed into the Main instrument. Information from the Main,

Link and Sequencer sources will be transmitted on channel b. Note that if a Link Unison is in effect, each performance event will be sent in duplicate; as the Main and Link information will both come out on channel b.

When either Mono Mode is in effect, received Note On messages are assigned to voices with a different algorithm than when Poly Mode is in effect. This algorithm causes a single voice to be used repeatedly. This does not affect notes played on the same logical instrument from the keyboard, sequencer or Chroma Interface. This differs slightly from the conventional interpretation of the MIDI Mono Mode. In particular, there is no fixed relationship between voices and channels in Mono Mode. Playing polyphonically on the keyboard can still steal voices that have been played monophonically through the MIDI Interface. The only effect that Mono mode has is that MIDI Note On commands will be directed to the most recently played voice that is already allocated to the correct logical instrument, instead of looking for the oldest released voice.

Mono Mode has no effect on transmission.

The third byte of the Mono Mode command is ignored. When Mono mode is selected, all MIDI input channels are placed in Mono mode.

Each input channel that is programmed to receive MIDI input responds to Local Control and All Notes Off commands independently. According to the true MIDI definition, then, each of these channels is a separate "basic channel". An All Notes Off command or a Local Control command to one channel will only affect that channel. The routing of these commands is the same as for Voice Status commands; that is, if Omni Mode is on a Local Control or All Notes Off command from any channel will be routed to the Main Instrument.

Each input channel that is programmed to receive MIDI input responds to Mode Change commands. There is, however, only one global mode for the whole Polaris, rather than a separate mode for each channel. Therefore, it doesn't matter which of the channels receives the Mode Change. If Omni Mode is in effect, a Mode Change will be acted on regardless of what channel it is received on.

The Polaris can store 132 programs, numbered internally 1 to 132. The last five of these are inaccessible to the MIDI interface, as MIDI only allows numbers as high as 127. If the Polaris ever has to send a program number higher than 127, it will send zero instead.

All parameters in a Polaris program are accessible through Control Change commands. Their ranges are not normalized to 0..127. In particular, on-off parameters are considered the same as variable parameters, except that their range is only 0..1. No control has more than seven bits, so no Control Change LSB Commands are used.

The MIDI standard did not allow enough Control Change numbers to cover all the parameters in the Polaris, so some of the undefined Control Change numbers have been appropriated. As stated before, though, parameter changes are only received and transmitted if the MIDI Panel Switch is on.

The internal metronome that runs the sequencer can be synchronized to

received MIDI clocks, and can also generate MIDI clocks. The MIDI clock interface can be used concurrently with the sync interface. For instance, the metronome can be slaved to the MIDI clock input and can generate a sync output. Alternatively, it can be slaved to the sync input and generate a MIDI clock output (in addition to a sync output at a different ratio, if desired). It is best, though, to use the internal metronome as the master, as this allows the metronome's tempo footswitch to set the tempo for the entire system.

The sequencer Stop and Start functions are implemented. They are transmitted only when the MIDI clock output is being used and are interpreted when received only when the MIDI clock input is being used. The Continue, Song Position Pointer and Song Select functions are not implemented.

The physical interface uses two connectors, one for MIDI IN and the other for both MIDI OUT and MIDI THRU, selectable by the user. The MIDI output is in THRU mode if all MIDI Output Switches are off and the MIDI clock output is disabled; it is in OUT mode if any MIDI Output Switch is on or if the MIDI clock output is enabled. The output can also be placed into OUT mode through the use of a System Exclusive command.

The Fender System Exclusive packet format includes a channel number byte after the id byte. A received packet will be interpreted only if it has the Fender id (8) and the number of one of the channels that the Polaris is currently programmed to respond to (any channel if in Omni Mode). It doesn't matter which particular channel, if the Polaris is listening to more than one. All transmitted packets will indicate channel number b.

All configuration parameters are non-volatile, except that the MIDI Output Switches are always turned off, the Local Control Switches are turned on and the MIDI Mode is set to Poly/Omni whenever the power is turned on (or whenever a MIDI Reset is received).

### Input Commands Recognized

Each input channel that the Polaris is listening to understands the following set of commands. The dashes indicate ignored bits and "cccc" indicates where the channel number minus one is to be substituted.

1000cccc Okkkkkkk Ovvvvvvv Note Off

k = key number

v = velocity

The velocity is ignored, except that the sequencer will record it.

1001cccc Okkkkkkk 00000000 Note Off

k = key number

The implied velocity is 01000000

1001cccc Okkkkkkk Ovvvvvvv Note On

k = key number

v = velocity

1010cccc Okkkkkkk Oppppppp      Key Pressure  
 k = key number  
 p = pressure  
 This is equivalent to a Channel Pressure command, except that it only affects voices playing key k. Key Pressure and Channel Pressure commands will fight with each other if used concurrently. Pressures affect the sound according to the various "pedal" related parameters.

1011cccc 00000000 0vvvvvvv      Volume  
 v = volume (0 = off, 1..127 = -48..+16db)  
 A Program Change will set this to 96, which is 0db. This is not the same as the Volume parameter.

1011cccc 00000001 Oppppppp      Modulation Lever  
 p = position (0 = center, 127 = push)  
 The Polaris Lever 1 input is bipolar, while this command is unipolar. Therefore, only the positive half of the range is accessible. A Program Change will set this to 0.

1011cccc 01000000 00000000      Sustain Footswitch Release  
 1011cccc 01000000 01111111      Sustain Footswitch Press  
 1011cccc 01000000 C-----      Ignored  
 A Program Change will release the footswitch.

1011cccc 01111010 00000000      Local Control Off  
 1011cccc 01111010 01111111      Local Control On  
 1011cccc 01111010 C-----      Ignored

1011cccc 01111011 00000000      All Notes Off  
 1011cccc 01111011 C-----      Ignored

1011cccc 01111100 00000000      Omni Mode Off  
 1011cccc 01111100 0-----      Ignored

1011cccc 01111101 00000000      Omni Mode On  
 1011cccc 01111101 C-----      Ignored

1011cccc 01111110 0-----      Mono Mode

1011cccc 01111111 00000000      Poly Mode  
 1011cccc 01111111 0-----      Ignored

1011cccc 0nnnnnnn 0vvvvvvv      Parameter change  
 n = parameter number  
 v = parameter value  
 The meanings of n and v are shown in a later section. This command is ignored if the MIDI Panel Switch is off.

1011cccc 0----- C-----      Ignored  
 Any combination not covered above is ignored.

1100cccc 00000000      Program Change 0  
 This deletes the logical instrument, stopping further sound generation on this channel. This command is ignored if the MIDI Panel Switch is off.

- 1100cccc Oppppppp                      Program Change p  
     p = program number  
 This creates the logical instrument if it does not already exist and enables further sound generation, using program number p. This command is ignored if the MIDI Panel Switch is off.
- 1101cccc Oppppppp                      Channel Pressure  
 This is internally converted into a Key Pressure command to all channels allocated to the logical instrument. Key Pressure and Channel Pressure commands will fight with each other if used concurrently. Pressures affect the sound according to the various "pedal" related parameters. A Program Change will set this according to the Pedal Initial parameter.
- 1110cccc 0----- Oppppppp              Pitch Bend Lever  
     p = position (0 = push, 64 = center, 127 = pull)
- 11110000 00001000 0---cccc ...      System Exclusive Prefix  
 11110000 0----- 0----- ...      Ignored
- 11110110                                  Tune Request  
 This works by simulating the LOWER FUNCTION, TUNE ALL switch presses. It will not work if the control panel is already in the middle of a multi-switch sequence.
- 11111000                                  Timing Clock  
 This is only interpreted when the metronome is enabled and slaved to the MIDI clock input.
- 11111010                                  Start  
 This is only interpreted when the metronome is slaved to the MIDI clock input and the sequencer is stopped. The sequencer and metronome are started.
- 11111100                                  Stop  
 This is only interpreted when the metronome is slaved to the MIDI clock input and the sequencer is playing. The sequencer and metronome are stopped.
- 11111111                                  Reset  
 All logical instruments are deleted and then the main and possibly the link instruments are recreated. The Local Control switches are turned on, the MIDI Output switches are turned off, and Poly/Omni mode is selected. This also disables further interpretation of data bytes until another status byte is received.
- 1-----                                  Ignored  
 All commands not mentioned above are ignored.

**Output Commands Generated**

Each output channel whose MIDI Output Switch is on can generate the following set of commands. The channel number minus one will be substituted for "cccc".

1000cccc Okkkkkkk Ovvvvvvv Note Off

k = key number

v = velocity

This will only be generated if the sequencer plays a sequence that was recorded on a keyboard that measures release velocity.

1001cccc Okkkkkkk 00000000 Note Off

k = key number

This is the Note Off that the Polaris normally generates, and implies a release velocity of 01000000.

1001cccc Okkkkkkk Ovvvvvvv Note On

k = key number

v = velocity

1010cccc Okkkkkkk Oppppppp Key Pressure

k = key number

p = pressure

This will never be transmitted unless a sequence containing pressure changes is played back. The Main and Link internal sources never generate key pressure information, as the Polaris has no pressure sensor.

1011cccc 00000000 Oppppppp Volume

This will never be transmitted unless a sequence containing volume changes is played back. The Main and Link internal sources never generate a volume change.

1011cccc 00000001 Oppppppp Modulation Lever

p = position (0 = center, 127 = push or pull)

The Polaris Lever 1 input is bipolar; this command is unipolar. Therefore, the absolute value is taken.

1011cccc 01000000 00000000 Sustain Footswitch Release

1011cccc 01000000 01111111 Sustain Footswitch Press

1011cccc 01111011 00000000 All notes off

This is sent in place of a Program Change 0 if the MIDI Panel Switch is off to make sure that other types of instruments squelch their sound properly. It is also possible for this to be sent if a sequence is played that has a Squelch Instrument or Release Instrument message in it, but this isn't normal.

1011cccc Onnnnnnn Ovvvvvvv Parameter change

n = parameter number

v = parameter value

The meanings of n and v are shown in a later section. This command is not transmitted if the MIDI Panel Switch is off.

- 1100cccc 00000000                      Program Change 0  
This is transmitted when a link is cleared, or when a sequence is stopped. This command is not transmitted if the MIDI Panel Switch is off; instead, an All Notes Off command will be transmitted.
- 1100cccc Oppppppp                      Program Change p  
p = program number  
This is transmitted whenever a program is selected, including when a link is set up or when a sequence is started. It is not transmitted if the MIDI Panel Switch is off.
- 1101cccc Oppppppp                      Channel Pressure  
p = pressure  
This is transmitted whenever the foot pedal is moved, as the Polaris has no pressure sensor.
- 1110cccc 00000000 Oppppppp              Pitch Bend Lever  
p = position (0 = push, 64 = center, 127 = pull)
- 11110000 00001000 0000cccc ... System Exclusive Prefix  
This is transmitted by the Polaris in response to certain incoming System Exclusive packets, and when the user invokes the MIDI Send Program function on the control panel.
- 11110111                                  End Of System Exclusive  
This is transmitted by the Polaris when the MIDI Send Program control panel function is invoked. This function causes several System Exclusive packets to be transmitted, followed by this byte. This is also transmitted by the Polaris at the end of a System Exclusive Packet that is a response to a received System Exclusive Packet.
- 11111000                                  Timing Clock  
This is only transmitted when the metronome is enabled and the MIDI clock output is enabled.
- 11111010                                  Start  
This is only transmitted if the MIDI clock output is enabled and the Play switch is pressed while the sequencer is stopped.
- 11111100                                  Stop  
This is only transmitted if the MIDI clock output is enabled and the Stop switch is pressed while the sequencer is playing.

### System Exclusive Format

The Fender System Exclusive format allows objects inside the Polaris to be created, deleted, opened, read, written, and added to the system as software extensions. It also allows messages in the Polaris' internal message format (which bears no resemblance to MIDI) to be inserted into the various internal data streams. Certain of these functions requires knowledge of proprietary details of the inner workings of the Polaris; if misused, the Polaris' computer will almost definitely crash, and in doing so will probably clobber a few bytes in the middle of your favorite program or sequence. In other words, experiment at your own risk. Other functions, though, are documented here, as they are fairly simple and presumed to be generally useful.

A System Exclusive packet begins, as mentioned above, with the System Exclusive status byte, the Fender id (8), and a channel number byte. Beyond that, all bytes are interpreted as ASCII characters, and the following rules hold:

- 1 SCII control characters are ignored. This includes codes 0C through 1F and 7F hex.

The first printable character indicates the packet type. Generally, this is a capital letter that suggests the function; a response packet, though, will start with an equals sign. An unknown packet type causes the entire packet to be ignored.

Remaining characters are interpreted according to the packet type. Most data is represented in ASCII hex notation, where each byte of data is transmitted as two characters from the set 0123456789ABCDEF, most significant nibble first. This implies that there will always be an even number of such characters in a packet. When inputting ASCII hex, any characters not in the above set (including lower case abcdef) will be ignored.

Four digit hex numbers are transmitted most significant byte (and digit) first.

The length of each packet is implied by or encoded in the packet itself. The End Of System Exclusive status byte is extraneous, and is not needed to determine the end of a packet. Any bytes beyond the expected end of the packet are ignored. If a packet has too few bytes, the packet will either not be interpreted if part of a command is missing or will be partially interpreted if data is missing.

The System Exclusive system allows an external device to gain access to one internal data object at a time. At any given time, one object (possibly non-existent) is "open" to the interface and this object (if it exists) is either write-enabled or write-protected. All objects are numbered, using a four-digit hex number, and the particular association between object numbers and particular physical objects is instrument-dependent. It is, however, possible to find out if an object exists and how big it is through the interface. The only reserved object number is 0, which represents all of the RAM space starting at location 0004 (locations 0000 through 0003 are inaccessible). Initially, this object is open and write-protected.

The following System Exclusive packet types are understood by the Polaris:

Create Object                    Cnnnnlllltt  
     nnnn -- The object number  
     llll -- The desired length in bytes  
     tt -- The object type

If object nnnn already exists, or if nnnn is too big (the Polaris allows numbers up to 00FF), or if the length is zero or larger than the available memory, the operation will not take place. The object type is normally 00; other types are proprietary. Performing this operation does not open the object for further access (see Open Object below), nor does it return any response.

Delete Object                    D  
     If the open object exists and is write-enabled, it is deleted.

Extension                        E  
     The open object is added to the list of software extensions if it exists, is not object zero, and is write-enabled. This is a proprietary function that will definitely crash the computer if you try it.

Identification                  I  
     A response packet is returned that looks like  
                                   =iirr"name"  
     ii -- The instrument id (03 for the Polaris)  
     rr -- The software revision level  
     name -- The name of the instrument ("Polaris") in quotes  
     In addition, this forces the output into OUT mode.

Led                                Lnn  
     nn -- An LED number, 00..3B  
     The current state of LED number nn is returned in the form  
                                   =ss  
     ss -- 00 (off) or 01 (on)  
     This is particularly useful in conjunction with the Send Message packet, below. By sending certain switch presses into the Polaris, the LEDs can be made to display various configuration parameters. These can then be read using this packet type and possibly modified by further Send Message packets. This is not useful if the LED is flashing! If the output is in THRU mode, the response will be lost.

Open                              Onnnnmmm  
     nnnn -- The object number  
     mm -- The mode  
     Object number n is selected as the open object. If mm is 00 (or any even number), it will be write-protected. If mm is 01 (or any odd number), it will be write-enabled. A response packet is returned that looks like

                                  =l111  
     l111 -- The length of the object in bytes  
     The length is the length at the time the packet is interpreted. If the object does not exist, the length will be zero. A non-existent object can be selected; if it is subsequently created (either through the interface or by another process in the Polaris) the new object will be open without having to repeat this command. If the output is in THRU mode, the response will be lost.

**Read** Rnnaaaa

nn -- The number of bytes  
 aaaa -- The starting address

The open object is read starting at location aaaa. A response packet is returned containing nn bytes. If nn is 00, 256 bytes are read. Bytes beyond the end of the object (or all bytes if the object does not exist) will be zero. The response looks like:

=nndd..dd

nn -- The number of bytes  
 dd..dd -- The data bytes

If the output is in THRU mode, the response will be lost.

**Send Message** Sssiimmoo

ss -- Stream number  
 ii -- Instrument number  
 mm -- Message  
 oo -- Operand

Message mm and operand oo are sent via data stream ss. If an instrument number is required, it is provided by ii. If an instrument number is not required, ii is ignored. If you send a message to an unused stream, the computer will crash. Certain streams and messages are documented below.

**Write** Wnnaaaadd..dd

nn -- The number of bytes  
 aaaa -- The starting address  
 dd..dd -- The data bytes

The data bytes are written into the open object starting at location aaaa. If nn is 0, 256 bytes are written. Bytes beyond the end of the object (or all bytes if the object does not exist or is write-protected) will be ignored.

### Polaris Object Numbers

The following objects can be safely created, deleted, open, read and written:

0001..0084 -- Programs A1, A2, A3, ..., K11, K12. Each of these must be 44 bytes long if it exists.

0094..009F -- Sequences 1 through 12.

The following objects can be safely open and read.

0085 -- The Main Instrument (Instrument 0). The Main Workspace program is in the first 44 bytes.

0086 -- The Link Instrument (Instrument 1). This normally only exists if a link is set up.

0087 -- The Sequencer Instrument (Instrument 2). This normally only exists while a sequence is being played.

0088..008C -- Instruments 3 through 7. These normally only exist if they have been created through the MIDI or Chroma interface.

**Polaris Data Stream Numbers and Messages**

In general, if you send a message to a stream that goes someplace and the message is not understood, the message will be ignored. If, however, you send a message to a stream that doesn't go someplace, the computer will crash. The following stream numbers (in hex) can be safely used:

16: Panel -- This is where the panel sends switch press and slider messages. It understands the following messages:

0C $ss$  -- Switch Press

Switch  $ss$  is pressed, where  $ss$  is in the range 01..3B. The switches are numbered basically from left to right.

ss $pp$  -- Slider Move

Slider  $ss$  - 80, where  $ss$  is in the range 81..93 (the slider number is in the range 01..13) is moved to position  $pp$  (00 is bottom, FF is top). The sliders are numbered from left to right, not including the Master slider.

17: Master -- This is where the panel sends Master slider messages. It only understands:

80 $pp$  -- Master Slider Move

The master slider is moved to position  $pp$  (00 is bottom, FF is top). The effect of this depends upon the current function of the master slider.

1E: Instrument -- A message sent here will be played by the logical instrument, if it exists. An instrument number (00..07) must be specified. Some of the messages understood by instruments are:

7D $pp$  -- Swap Program

The program contained in the instrument is swapped with program  $pp$  (where  $pp$  is in the range 01..84).

7DFF -- Release Instrument

All notes are released.

7E $pp$  -- Store Program

The program contained in the instrument is stored over program  $pp$  (where  $pp$  is in the range 01..84).

7EFF -- Squelch Instrument

All notes are released and their envelopes are chopped off.

7F00 -- Recreate Instrument

The instrument (which is expected to already exist) is recreated using the sound it contains. This is useful after using Open and Write to write a 44 byte program into the instrument's workspace.

7F $pp$  -- Create Instrument

The instrument is created if it does not already exist, and given program number  $pp$  (where  $pp$  is in the range 01..84). An instrument object number (85..8C) can also be used in place of  $pp$ , in

which case the contents of that instrument's workspace will be used as the source of the program data.

7FFF -- Delete Instrument  
The instrument is deleted.

**Parameter Numbers and Values**

Each Polaris program contains fifty parameters, corresponding to MIDI Control Change numbers 2..31 and 96..115. Each parameter has a range appropriate to its function. On/off parameters have a range of 0 to 1. Some parameters have bipolar ranges.

Parameters are divided into two categories, tonal and panel. The tonal parameters directly control the sound generation and so are of most interest to the MIDI interface user. The panel parameters do nothing to the sound by themselves; rather, they affect the settings of certain control panel related functions, or establish certain conditions when a program is selected.

The range given for each parameter in the following list is the range of values that the Polaris will generate when sending a Control Change command. On input, if a value is out of range, it will either be truncated to the correct number of bits, or will be limited to one end of the range. Negative numbers are written in seven-bit two's complement notation, so, for instance, -5 would appear as 01111011; the eighth bit is, of course, always zero.

#	Name	Range	#	Name	Range
2	VOLUME	0..127	27	DETUNE	-64..+63
3	GLIDE	0..63	28	RING MOD	0..1
4	sweep RATE	0..127	29	SYNC	0..1
5	RATE PEDAL depth	-64..+63	30	osc 1 SAWS/PULSE	0..1
6	sweep SINE/SQUARE	0..1	31	osc 2 SAWS/PULSE	0..1
7	VIBRATO DELAY	0..63	96	osc 1 PULSE WIDTH	-64..+63
8	MOD LEVER RANGE	0..15	97	osc 2 PULSE WIDTH	-64..+63
9	VIBRATO PEDAL depth	0..15	98	osc 1 SWP PWM/ENV PWM	0..1
10	PITCH BEND RANGE	-16..+15	99	osc 2 SWP PWM/ENV PWM	0..1
11	PITCH PEDAL depth	-16..+15	100	osc 1 PULSE WIDTH MOD	-64..+63
12	env FIXED/TOUCH	0..1	101	osc 2 PULSE WIDTH MOD	-64..+63
13	env ATTACK	0..63	102	NOISE	0..1
14	env DECAY	0..63	103	filter CUTOFF	0..127
15	env SUSTAIN	0..63	104	filter RESONANCE	0..7
16	env SUSTAIN DECAY	0..63	105	filter SWEEP DEPTH	-64..+63
17	env RELEASE	0..63	106	filter ENV DEPTH	-64..+63
18	vol env FIXED/TOUCH	0..1	107	filter KYBD TRACK	-64..+63
19	vol env ATTACK	0..63	108	CUTOFF PEDAL depth	-64..+63
20	vol env DECAY	0..63	109	VOLUME PEDAL depth	0..63
21	vol env RELEASE	0..63	110	Pedal Initial	0..127
22	osc 1 TRANSPOSE	0..60	111	Slider Assignment	0..13
23	osc 2 TRANSPOSE	0..60	112	Link Mode	0..3
24	OSC 1 VIBRATO	-64..+63	113	Link Program Number	0..127
25	OSC 2 VIBRATO	-64..+63	114	Keyboard Split	0..60
26	OSC 2 ENV depth	-64..+63	115	Keyboard Range	0..1

MIDI Implementation Chart (version 3)

Function ...		Transmitted **	Recognized ***	Remarks
Basic Default		1-16	1-16	SEE NOTE 1.
Channel Changed		1-16	1-16	
Mode Default		1	1	SEE NOTE 2.
Messages Altered		None transmitted *****	OMNI On/Off M/P X	
Note Number	True voice	24 to 96 *****	9 to 120 9 to 120	
Velocity Note ON		0	0	SEE NOTE 3.
Note OFF		X(90H, V=0)	0	
After Touch	Key's Ch's	X 0	0 0	Press. Interpreted as Pedal. SEE NOTE 4.
Pitch Bender		0	0	Lever pulled 127.
Non-Inclusive:	0	X	0	Perf. Volume
	1	0*	0	Mod Lever
Control Change	2	0*	0	Volume
	4	0*	0	Sweep Rate
	13	0*	0	Env Attack
SEE NOTE 5	24	0*	0	Osc 1 Vibrato
FOR A COMPLETE	25	0*	0	Osc 2 Vibrato
LISTING.	103	0*	0	Filter Cutoff
	64	0*	0	Sustain Ftsw
Prog Change : True #		0* *****	0 0 to 127	Prog 0 Stops Sound in Ch. SEE NOTE 6.
System Exclusive		0	0	ID = 8 SEE NOTE 7.
System : Song Pos		X	X	
: Song Sel		X	X	
Common : Tune		X	0	
System : Clock		0	0	SEE NOTE 8.
Real Time : Commands		Start/Stop	Start/Stop	
Aux : Local ON/OFF		0	0	SEE NOTE 9.
: All Notes Off		0	0	
Mes- : Active Sense		X	X	
sages: Reset		X	0	

**MIDI Implementation Chart Notes**

Mode 1 : OMNI ON, POLY  
 Mode 2 : OMNI ON, MONO  
 Mode 3 : OMNI OFF, POLY  
 Mode 4 : OMNI OFF, MONO

O : Yes  
 X : No

\*Only when the MIDI Panel Switch is ON.  
 \*\*Only when the MIDI Out Switch is ON.  
 \*\*\*Only when the MIDI In Switch is ON.

## Note 1:

Main on basic channel (b), Link on b+1, Sequencer on b+2. Channels are on a circular scale (if b = 16 then b+1 = 1). There is a soft switch to recognize channels b+3 through b+7 (MIDI Extra Channels). Basic channel selection is stored in non-volatile RAM.

## Note 2:

Mono mode has no effect on transmission. In Mono mode reception, a single voice is used repeatedly for each MIDI channel. The algorithm used is 'most recently used' instead of 'least recently used'. Mode commands received on any channel that is listened to affect all channels that are being listened to.

## Note 3:

Main, Link, and normally the Sequencer, send Note On with velocity of 0 for Note Off. In receiving, a Note On with velocity of 0 is interpreted as a Note Off with a release velocity of 40H. Note Off velocity is normally ignored. However, if the Sequencer is recording the MIDI input and receives Note Off commands with some velocity, then the commands and their velocities are remembered and retransmitted.

## Note 4:

Key Pressure is interpreted as a pedal change for the voice playing that key. Channel Pressure is interpreted as a pedal change for all channels allocated to the logical instrument. Key Pressure and Channel Pressure commands will fight each other if used concurrently. Channel Pressure will be transmitted whenever the pedal is moved. Key Pressure commands are not transmitted unless the sequencer is playing back a sequence recorded through the MIDI or Chroma Interface with pressure data.

## Note 5:

The following control changes are transmitted and received:

cc#	Name	Range	Remarks
0	Performance Volume	0..127	96 at Prog selection. 1/2dB changes 0 = Center, 127 = Full Push
1	Modulation Lever	0..127	
2	VOLUME	0..127	These are program parameters and are transmitted only when a MIDI panel switch is on.
3	GLIDE	0..63	
4	sweep RATE	0..127	
5	RATE PEDAL depth	-64..+63	
6	sweep SINE/SQUARE	0..1	
7	VIBRATO DELAY	0..63	
8	MOD LEVER RANGE	0..15	
9	VIBRATO PEDAL depth	0..15	
10	PITCH BEND RANGE	-16..+15	
11	PITCH PEDAL depth	-16..+15	
12	env FIXED/TOUCH	0..1	
13	env ATTACK	0..63	
14	env DECAY	0..63	
15	env SUSTAIN	0..63	
16	env SUSTAIN DECAY	0..63	
17	env RELEASE	0..63	
18	vol env FIXED/TOUCH	0..1	
19	vol env ATTACK	0..63	
20	vol env DECAY	0..63	
21	vol env RELEASE	0..63	
22	osc 1 TRANSPOSE	0..60	
23	osc 2 TRANSPOSE	0..60	
24	OSC 1 VIBRATO	-64..+63	
25	OSC 2 VIBRATO	-64..+63	
26	OSC 2 ENV depth	-64..+63	
27	DETUNE	-64..+63	
28	RING MOD	0..1	
29	SYNC	0..1	
30	osc 1 SAWS/PULSE	0..1	
31	osc 2 SAWS/PULSE	0..1	
64	Sustain Footswitch	0..127	0 = Release, 127 = Press
96	osc 1 PULSE WIDTH	-64..+63	These are program parameters and are transmitted only when a MIDI panel switch is on.
97	osc 2 PULSE WIDTH	-64..+63	
98	osc 1 SWP PWM/ENV PWM	0..1	
99	osc 2 SWP PWM/ENV PWM	0..1	
100	osc 1 PULSE WIDTH MOD	-64..+63	
101	osc 2 PULSE WIDTH MOD	-64..+63	
102	NOISE	0..1	
103	filter CUTOFF	0..127	
104	filter RESONANCE	0..7	
105	filter SWEEP DEPTH	-64..+63	
106	filter ENV DEPTH	-64..+63	
107	filter KYBD TRACK	-64..+63	
108	CUTOFF PEDAL depth	-64..+63	
109	VOLUME PEDAL depth	0..63	
110	Pedal Initial	0..127	
111	Slider Assignment	0..13	
112	Link Mode	0..3	
113	Link Program Number	0..127	
114	Keyboard Split	0..60	
115	Keyboard Range	0..1	

All control change commands not implemented are ignored (mode control changes are implemented, as well as Local Control, All Notes Off, OMNI ON/OFF and MONO /POLY). Negative numbers are written in 7-bit two's complement notation.

#### Note 6:

Program 0<sup>2</sup> is transmitted when a link is cleared or when a sequence is stopped. It is not transmitted if the MIDI Panel Switch is off; instead, an All Notes Off command is transmitted. Program 0, when received, deletes the logical instrument, stopping further sound generation on this channel. Programs 1-127 correlate to Polaris programs A1-K7, so programs K8-K12 are inaccessible via MIDI. If MIDI extra channels happens to be enabled, non-zero Program Change commands (including Program 0) are not transmitted or listened to.

#### Note 7:

The Fender System Exclusive format allows software objects inside the Polaris to be created, deleted, opened, read, written, and added to the system as software extensions. These objects include the Main, Link, Sequencer and extra MIDI instruments as well as Programs and Sequences. The format also allows messages in the Polaris' internal message format to be inserted into the various internal data streams. This allows a computer to simulate multiple key depressions, among other things.

#### Note 8:

Timing Clock is transmitted when the metronome is enabled and the MIDI clock output is enabled. Timing clock is interpreted only when the metronome is enabled and slaved to the MIDI clock input. Start is transmitted only when the metronome is enabled and the MIDI clock output is enabled. Start is interpreted only when the metronome is slaved to the MIDI clock input and the sequencer is stopped. Stop is transmitted only if the MIDI clock output is enabled and the stop switch is pressed while the sequencer is playing. Stop is interpreted only when the metronome is slaved to the MIDI clock input and the sequencer is playing. When start or stop is received the sequencer and the metronome are started or stopped.

#### Note 9:

Each input channel that is programmed to receive MIDI input responds to Local Control and All Notes Off independently. If OMNI is on however, a Local Control or All Notes Off command from any channel will be routed to the main instrument (basic channel). Upon receipt of Reset, all logical instruments are deleted and the Main and Link (if applicable) instruments are recreated. The local control switches are turned on, the MIDI output switches are turned off and POLY/OMNI ON mode is selected. This disables further interpretation of data bytes until another status byte is received.