Eventide
the next step

Technical Note #94

MIDI Sysex Messages on the DSP4000

This technote assumes familiarity with the MIDI System Exclusive Message format, and the use of hex (hexadecimal) numbers. It largely applies to Orville and the 7000 family.

The system exclusive format used by the 4000 series is of the form;

0xF0 <EVENTIDE> <H4000> <id> <message_code> <lots-o-bytes> 0xF7

the 0xF0 and 0xF7 are standard MIDI for start of system exclusive, and end of system exclusive. Note that 0xF0 (for example) is a hexadecimal representation of the decimal value 240, while 0xF7 is decimal 247.

<**EVENTIDE**> is 0x1C (decimal 28)
<**H4000**> is 0x70 (decimal 112).
<**id**> is the device id number. If this is zero, all DSP4000s will listen to the message.
<**message_code**> tells us what message this is. The various messages are described below.
<**lots-o-bytes**> is the rest of the message. This **data** depends on the type of message. Not all messages have 'lots-o-bytes'.

With many messages, a "byte" is actually two bytes. Since MIDI allows only 7 bits of data, we split an 8 bit byte into two 4 bit *nibbles* and send the nibbles. The most significant nibble is sent

first.

Message codes:

SYSEXC_OK 0x00

Data: None. This message is returned by the DSP4000 in response to assorted commands. It simply says "everything was ok with that last command".

Response: none

SYSEXC_KEYPRESS 0x01

Data: The next 4 bytes (8 nibbles) say what keys are to be pressed. There are 32 bits in the message, one for each key. This allows for multiple keys being pressed at the same time. See Appendix A for the individual key values.

Response: none

SYSEXC_USEROBJECT 0x02

Data: "lots-o-bytes" contains a userobject message, the description of which is beyond the scope of this note. You can setup the DSP4000 to send out these messages when paramater changes are made. These can then be recorded by a MIDI sequencer, and when replayed, will cause the control changes to be duplicated. Every userobject message gets a response.
Response: none

SYSEXC_BANKCHANGE 0x03

Data:

First byte is 0 for internal, 1 for external.
Second byte is bank number. (0 thru 99)
These two bytes are broken into 4 nibbles. (ok, I know we don't need to this, but it makes the software easy).

Response: none

SYSEXC_PROGRAM_DUMP_OLD 0x04

Obsolete - do not use

SYSEXC_SETUP_DUMP_OLD 0x05

Obsolete - do not use

SYSEXC_PROGRAM_WANT 0x06

Data: none.

Response: Upon receiving this message, the DSP4000 will send out a SYSEXC_PROGRAM_DUMP message for the currently loaded program.

SYSEXC_SETUP_WANT 0x07

Data: none

Response: Upon receiving this message, the DSP4000 will send out a SYSEXC_SETUP_DUMP message for the state of the unit.

SYSEXC_SIGFILE_DUMP 0x08

Data: a sigfile representing the currently loaded program. A sigfile is a human readable form of a program. It is a series of text characters (Not split into nibbles) that form words, numbers, and strings. (a string is some text bounded by "".) See Appendix B for more information on sigfiles.

Upon receiving this message, the DSP4000 will encode the program, compile it, and load it. This does take time. If there are errors, they will be reported on screen.

Response: none.

SYSEXC_SIGFILE_WANT 0x09

Data: none.

Response: Upon receiving this message, the DSP4000 sends a SYSEXC_SIGFILE_DUMP message. The sigfile is heavily commented as to data type, name, min, maxs, etc. It does take some time to output this message.

SYSEXC_SIGFILE_DUMP_REMOTE 0x0A

Data: Just like SYSEXC_SIGFILE_DUMP.

Response :if no error, will return a SYSEXC_OK message. If error, will return a SYSEXC_ERROR message instead of putting a message on the screen. This message is intended for remote editors.

SYSEXC_SIGFILE_WANT_QUICK 0x0B

Data: none

Response: Take the current program, convert to a sigfile, and send it back thru MIDI. Upon receiving this message, the DSP4000 sends a SYSEXC_SIGFILE_DUMP_REMOTE message. With this command, the sigfile has no comments.

SYSEXC_SIGDBASE_DUMP 0x0C

Data: A sigfile database message sent by the DSP4000. This message contains the information that a remote editor might need to operate on sigfiles.

Note: This dump is very big (currently 160K). We also put delays to help things keep up. So, this take a long time. Sending this message to the DSP4000 does nothing.

SYSEXC_ERROR 0x0D

Data: This message is returned by the DSP4000 in response to assorted commands. The message indicates that an error occured with the last command. <lots-o-bytes> may contain a ascii text error message (not split into nibbles)

SYSEXC_SIGDBASE_WANT 0x0E

Data: none.

Response: The DSP4000 outputs a SYSEXC_SIGDBASE_DUMP.

SYSEXC_FILES_DUMP 0x0F

Data: This is a set of files to replace the ones in the machine. The format of the data is a string of bytes split into nibbles.

The first 8 nibbles tell the size of the block, ie how many bytes are actually loaded.

Next is the actual block to be loaded. This has a variable size.

At the end is two nibbles the form a 1 byte checksum. This byte added to all the bytes including the size bytes should equal zero. If the sum does not equal zero, then the DSP4000 will prompt you to send the dump again.

Response: none.

SYSEXC_FILES_WANT 0x10

Data: none

Response: Upon receiving this message, the DSP4000 will send out a SYSEXC_FILES_DUMP message for the current presets.

SYSEXC_INTERNAL_DUMP 0x11

This message replaces the complete contents of the internal NV ram.

Data: The format of the data is the same as SYSEXC_FILES_DUMP.

Response: none

SYSEXC_INTERNAL_WANT 0x12

Data: none

Response: Upon receiving this message, the DSP4000 will send out a SYSEXC_INTERNAL_DUMP message for the current presets.

SYSEXC_CARD_DUMP 0x13

Data: The contents of a memory card. The format is the same as SYSEXC_SYSEXC_FILES_DUMP.

Response: none.

SYSEXC_CARD_WANT 0x14

Data: none

Response: Upon receiving this message, the DSP4000 will send out a SYSEXC_CARD_DUMP message for the current card.

SYSEXC_PROGRAM_DUMP 0x15

Data: A program in binary form, the format being the same as SYSEXC_FILES_DUMP.

Response: none

SYSEXC_SETUP_DUMP 0x16

Data: This loads a system setup into the DSP4000, the format being the same as SYSEXC_FILES_DUMP.

Response: none

SYSEXC_SCREEN_DUMP 0x17

Data: This is a dump of the current screen. The format of the data is a string of bytes split into nibbles.

First 8 nibbles are screen width in pixels.

Next 8 nibbles are screen height in pixels.

Next 8 nibbles are screen dump size in bytes.

Next is screen bitmap as nibbles. This has a variable size given by the product of width (rounded up to nearest 8) and height.

At the end is two nibbles the form a 1 byte checksum. This byte added to all the bytes including the size bytes should equal zero.

Response: none

SYSEXC_SCREEN_WANT 0x18

Data: none

Response: Upon receiving this message, the DSP4000 will send out a SYSEXC_SCREEN_DUMP message.

SYSEXC_INFO_DUMP 0x19

Data: This is a list of system information. It is in ASCII and is much like the information you get when you press "information" under SETUP:service. At the begining of a page, there is: **********(cr), followed by the information. For example:

Name: System ROM(cr)
Revision: 1.067(cr)
Time: Wed Nov 23 10:58:32 1994(cr)
Size: 524288(cr)

There is a 0 at the end of the dump before the end of the SYSEX message. The DSP4000 does nothing when receiving this message.

Response: none

SYSEXC_INFO_WANT 0x1A

Data: none

Response: Upon receiving this message, the DSP4000 will send out a SYSEXC_INFO_DUMP message.

Appendix A - Key press values

Key values are a 32 bit word, with one bit corresponding to each key. This bit is zero is the key is pressed, 1 if released. Because multiple keys could be pressed at the same time (or stuck down due to faulty switches) more than one key bit could be zero at the same time.

Key Name	***Bit #*** (0 is LSB)	***MIDI number when pressed*** *(Hexadecimal)*
no key	none	FFFFFFFF
LEVELS	0	FFFFFFFE
BYPASS/MUTE	9	FFFFFDFF
SOFT1	26	FBFFFFFF
SOFT2	18	FFFBFFFF
SOFT3	10	FFFFFBFF
SOFT4	2	FFFFFFFB
PROGRAM	27	F7FFFFFF
SETUP	11	FFFFF7FF
PATCH	3	FFFFFFF7
PARAMETER	19	FFF7FFFF
< LEFT	16	FFFEFFFF
SELECT	8	FFFFFEFF
RIGHT >	24	FEFFFFFF
USER1	25	FDFFFFFF
USER2	17	FFFDFFFF

Key Name	***Bit #*** (0 is LSB)	***MIDI number when pressed*** *(Hexadecimal)*
no key	none	FFFFFFFF
ONE	31	7FFFFFFF
TWO	23	FF7FFFFF
THREE	15	FFFF7FFF
FOUR	30	BFFFFFFF
FIVE	22	FFBFFFFF
SIX	14	FFFFBFFF
SEVEN	29	DFFFFFFF
EIGHT	21	FFDFFFFF
NINE	13	FFFFDFFF
ZERO	20	FFEFFFFF
DOT	28	EFFFFFFF
MINUS	12	FFFFEFFF
UP	7	FFFFFF7F
DOWN	6	FFFFFFBF
CXL	5	FFFFFFDF
ENTER	4	FFFFFFEF

Appendix B: Sigfile Format

A sigfile is an ascii text representation of a DSP4000 preset. Each entry represents a module, or *operator*.
For each operator, there is the name of the operator type, a name for that particular operator, and the data required by the operator. Each operator has a different data set, which can be different for the different instances of the same operator.

A semicolon, ';' starts a comment field that bounded by the end of the line.

See the 4000 Operators Manual for information on the operators and their data sets. Sigfiles are complex and should be viewed as being for experts only. As a result, further detail is considered beyond the scope of this Technote.

Example:
Note that:
- *the names of the operators are in capitals*
- *operator entries may be split across multiple lines*
- *The first module is always HEAD, the last is always TAIL.*
- *Text does not need to be in quotes if it contains no spaces.*

```
HEAD Hed LMix-out RMix-out "Universal Matrix" "Matrix" 2 Setup-obj info-obj
MENUPAGE Setup "Operation" " Setup" 7 no-obj MatName-obj FactName-obj no-obj no-obj
MatKnob-obj WideKnob-obj
TEXTLINE MatName "     Matrix Output:"
TEXTLINE no " "
TEXTKNOB MatKnob "%s" "shnam" 7 0.0000 "Normal L/R" "Reverse R/L" "MS - Left is Mid"
"MS - Right is Mid" "L to both" "R to both" "Mono sum"
TEXTLINE FactName "   MS Stereo Width:"
KNOB WideKnob "%2.1f (1 is normal)" "shnam" 0.0000 2.0000 0.1000 1.0000
C_BOUND Sfact WideKnob-out 0.0000 1.0000
C_MULTIPLY negSfact Sfact-out -1.0000
C_BOUND Mbnd WideKnob-out 1.0000 2.0000
C_SUBTRACT Mfact 2.0000 Mbnd-out
C_SWITCH LfactL 7 MatKnob-out 1.0000 0.0000 Mfact-out Sfact-out 1.0000 0.0000 0.7070
C_SWITCH LfactR 7 MatKnob-out 0.0000 1.0000 Sfact-out Mfact-out 0.0000 1.0000 0.7070
C_SWITCH RfactL 7 MatKnob-out 0.0000 1.0000 Mfact-out negSfact-out 1.0000 0.0000 0.7070
C_SWITCH RfactR 7 MatKnob-out 1.0000 0.0000 negSfact-out Mfact-out 0.0000 1.0000 0.7070
MIX LMix Hed-left Hed-right LfactL-out LfactR-out
MIX RMix Hed-left Hed-right RfactL-out RfactR-out
TEXTBLOCK info 7 "M/S (mid/side) recording lets you air" "stereo events with complete mono
com-" "patibility. This setting decodes M/S" "recordings & controls their stereo" "width. It also
lets you fix mono and" "stereo routing." "Stereo in, stereo out."

TAIL JayRose
```