MIDI specification

MIDI: **M**USICAL **I**NSTRUMENT **D**IGITAL **I**NTERFACE
Specification 1.0

INTRODUCTION

MIDI is the acronym for Musical Instrument Digital Interface.

MIDI enables synthesizers, sequencers, home computers, rhythm machines, etc. to be intercon- nected through a standard interface.

Each MIDI-equipped instrument usually contains a receiver and a transmitter. Some instruments may contain only a receiver or transmitter. The receiver receives messages in MIDI format and executes MIDI commands. It consists of an optoisolator, Universal Asynchronous Receiver/Transmitter (UART), and other hardware needed to perform the intended functions. The transmitter originates messages in MIDI format, and transmits them by way of a UART and line driver.

The MIDI standard hardware and data format are defined in this specification.

CONVENTIONS

Status and Data bytes given in Tables I through VI are given in binary.

Numbers followed by an "H" are in hexadecimal.

All other numbers are in decimal.

HARDWARE

The interface operates at 31.25 (+/- 1%) Kbaud, asynchronous, with a start bit, 8 data bits (D0 to D7), and a stop bit. This makes a total of 10 bits for a period of 320 microseconds per serial byte.

Circuit: 5 mA current loop type. Logical 0 is current ON. One output shall drive one and only one input. The receiver shall be opto-isolated and require less than 5 mA to turn on. Sharp PC-900 and HP 6N138 optoisolators have been found acceptable. Other high-speed optoisolators may be satisfactory. Rise and fall times should be less than 2 microseconds.

Connectors: DIN 5 pin (180 degree) female panel mount receptacle. An example is the SWITCHCRAFT 57 GB5F. The connectors shall be labelled "MIDI IN" and "MIDI OUT". Note that pins 1 and 3 are not used, and should be left unconnected in the receiver and transmitter.

NOTES:

1. Optoisolator is Sharp PC-900. (HP 6N138 or other optoisolator can be used with appropriate changes.)
2. Gates "A" are IC or transistor.
3. Resistors are 5%

Cables shall have a maximum length of fifty feet (15 meters), and shall be terminated on each end by a corresponding 5-pin DIN male plug, such as the SWITCHCRAFT 05GM5M. The cable shall be shielded twisted pair, with the shield connected to pin 2 at both ends.

A "MIDI THRU" output may be provided if needed, which provides a direct copy of data coming in MIDI IN. For very long chain lengths (more than three instruments), higher-speed optoisolators must be used to avoid additive rise/fall time errors which affect pulse width duty cycle.

DATA FORMAT

All MIDI communication is achieved through multi-byte "messages" consisting of one Status byte followed by one or two Data bytes, except Real-Time and Exclusive messages (see below).

MESSAGE TYPES

Messages are divided into two main categories: Channel and System.

Channel

Channel messages contain a four-bit number in the Status byte which address the message specifically to one of sixteen channels. These messages are thereby intended for any units in a system whose channel number matches the channel number encoded into the Status byte.

There are two types of Channel messages: Voice and Mode.

Voice

To control the instrument's voices, Voice messages are sent over the Voice Channels.

Mode

To define the instrument's response to Voice messages, Mode messages are sent over the instument's Basic Channel.

System

System messages are not encoded with channel numbers.

There are three types of System messages: Common, Real-Time, and Exclusive.

Common

Common messages are intended for all units in a system.

Real-Time

Real-Time messages are intended for all units in a system. They contain Status bytes only -- no Data bytes. Real-Time messages may be sent at any time -- even between bytes of a message which has a different status. In such cases the Real-Time message is either ignored or acted upon, after which the receiving process resumes under the previous status.

Exclusive

Exclusive messages can contain any number of Data bytes, and are terminated by an End of Exclusive (EOX) or any other Status byte. These messages include a Manufacturer's Identification (ID) code. If the receiver does not recognize the ID code, it should ignore the ensuing data.

So that other users can fully access MIDI instruments, manufacturers should publish the format of data following their ID code. Only the manufacturer can update the format following their ID.

DATA TYPES

Status Bytes

Status bytes are eight-bit binary numbers in which the Most Significant Bit (MSB) is set (binary 1). Status bytes serve to identify the message type, that is, the purpose of the Data bytes which follow the Status byte.

Except for Real-Time messages, new Status bytes will always command the receiver to adopt their status, even if the new Status is received before the last message was completed.

Running Status

For Voice and Mode messages only, when a Status byte is received and processed, the receiver will remain in that status until a different Status byte is received. Therefore, if the same Status byte would be repeated, it may (optionally) be omitted so that only the correct number of Data bytes need be sent. Under Running Status, then, a complete message need only consist of specified Data bytes sent in the specified order.

The Running Status feature is especially useful for communicating long strings of Note On/Off messages, where "Note On with Velocity of 0" is used for Note Off. (A separate Note Off Status byte is also available.)

Running Status will be stopped when any other Status byte intervenes, except that Real-Time messges will only interrupt the Running Status temporarily.

Unimplemented Status

Any status bytes received for functions which the receiver has not implemented should be ignored, and subsequent data bytes ignored.

Undefined Status

Undefined Status bytes must not be used. Care should be taken to prevent illegal messages from being sent during power-up or power-down. If undefined Status bytes are received, they should be ignored, as should subsequent Data bytes.

Data Bytes

Following the Status byte, there are (except for Real-Time messages) one or two Data bytes which carry the content of the message. Data bytes are eight-bit binary numbers in which the MSB is reset (binary 0). The number and range of Data bytes which must follow each Status byte are specified in the tables which follow. For each Status byte the correct number of Data bytes must always be sent. Inside the receiver, action on the message should wait until all Data bytes required under the current status are received. Receivers should ignore Data bytes which have not been properly preceeded by a valid Status byte (with the exception of "Running Status," above).

CHANNEL MODES

Synthesizers contain sound generation elements called voices. Voice assignment is the algorithmic process of routing Note On/Off data from the keyboard to the voices so that the musical notes are correctly played with accurate timing.

When MIDI is implemented, the relationship between the sixteen available MIDI channels and the synthesizer's voice assignment must be defined. Several Mode messages are available for this purpose (see Table III). They are Omni (On/Off), Poly, and Mono. Poly and Mono are mutually exclusive, i.e., Poly Select disables Mono, and vice versa. Omni, when on, enables the receiver to receive Voice messages in all voice Channels without discrimination. When Omni is off, the receiver will accept Voice messages from only the selected Voice Channel(s). Mono, when on, restricts the assignment of Voices to just one voice per Voice Channel (Monophonic.) When Mono is off (=Poly On), any number of voices may be allocated by the Receiver's normal voice assignment algorithm (Polyphonic.)

For a receiver assigned to Basic Channel "N," the four possible modes arising from the two Mode messages are:

Mode	Omni		
1	On	Poly	Voice messages are received from all Voice channels and assigned to voices polyphonically.
2	On	Mono	Voice messages are received from all Voice Channels, and control only one voice, monophonically.
3	Off	Poly	Voice messages are received in Voice channel N only, and are assigned to voices polyphonically.
4	Off	Mono	Voice messages are received in Voice channels N thru N+M-1, and assigned monophonically to voices 1 thru M, respectively. The number of voices M is specified by the third byte of the Mono Mode Message.

Four modes are applied to transmitters (also assigned to Basic Channel N). Transmitters with no channel selection capability will normally transmit on Basic Channel 1 (N=0).

Mode	Omni		
1	On	Poly	All voice messages are transmitted in Channel N.
2	On	Mono	Voice messages for one voice are sent

			in Channel N.
3	Off	Poly	Voice messages for all voices are sent in Channel N.
4	Off	Mono	Voice messages for voices 1 thru M are transmitted in Voice Channels N thru N+M-1, respectively. (Single voice per channel).

A MIDI receiver or transmitter can operate under one and only one mode at a time. Usually the receiver and transmitter will be in the same mode. If a mode cannot be honored by the receiver, it may ignore the message (and any subsequent data bytes), or it may switch to an alternate mode (usually Mode 1, Omni On/Poly).

Mode messages will be recognized by a receiver only when sent in the Basic Channel to which the receiver has been assigned, regardless of the current mode. Voice messages may be received in the Basic Channel and in other channels (which are all called Voice Channels), which are related specifically to the Basic channel by the rules above, depending on which mode has been selected.

A MIDI receiver may be assigned to one or more Basic Channels by default or by user control. For example, an eight-voice synthesizer might be assigned to Basic Channel 1 on power-up. The user could then switch the instrument to be configured as two four-voice synthesizers, each assigned to its own Basic Channel. Separate Mode messages would then be sent to each four-voice synthesizer, just as if they were physically separate instruments.

POWER-UP DEFAULT CONDITIONS

On power-up all instruments should default to Mode #1. Except for Note On/Off Status, all Voice messages should be disabled. Spurious or undefined transmissions must be suppressed.

TABLE I

SUMMARY OF STATUS BYTES

STATUS	# OF DATA	DESCRIPTION
D7---D0	BYTES	
Channel Voice Messages:		
1000nnnn	2	Note Off event
1001nnnn	2	Note On event (velocity=0: Note Off)
1010nnnn	2	Polyphonic key pressure/after touch
1011nnnn	2	Control change
1100nnnn	1	Program change

```
1101nnnn     1        Channel pressure/after touch
1110nnnn     2        Pitch bend change
```

Channel Mode Messages:

```
1011nnnn     2        Selects Channel Mode
```

System Messages:

```
11110000   *****      System Exclusive
11110sss  0 to 2      System Common
11111ttt     0        System Real Time
```

NOTES:

1. nnnn: N-1, where N = Channel #, i.e. 0000 is Channel 1, 0001 is Channel 2 ... 1111 is Channel 16.
2. *****: 0iiiiiii, data, ..., EOX;

   ```
   iiiiiii:
   ```

   Identification
3. sss: 1 to 7
4. ttt: 0 to 7

TABLE II

CHANNEL VOICE MESSAGES

```
 STATUS       DATA BYTES        DESCRIPTION
1000nnnn   0kkkkkkk 0vvvvvvv  Note Off (see notes 1-4),
                              vvvvvvv: note off velocity
1001nnnn   0kkkkkkk 0vvvvvvv  Note On (see notes 1-4),
                              vvvvvvv - 0: velocity,
                              vvvvvvv = 0: note off
1010nnnn   0kkkkkkk 0vvvvvvv  Polyphonic Key Pressure (After-Touch),
                              vvvvvvv: pressure value
1011nnnn   0ccccccc 0vvvvvvv  Control Change,
                              ccccccc: control # (0-121) (see notes 5-8),
                              vvvvvvv: control value,
                              ccccccc = 122 thru 127: Reserved, (See Table III)
1100nnnn   0ppppppp           Program Change,
                              ppppppp: program number (0-127)
1101nnnn   0vvvvvvv           Channel Pressure (After-Touch)
                              vvvvvvv: pressure value
```

```
1110nnnn   0vvvvvvv           Pitch Bend Change LSB (see note 10)
                    0vvvvvvv  Pitch Bend Change MSB
```

NOTES:

1. nnnn: Voice Channel # (1-16, coded as defined in Table I notes)
2. kkkkkkk: note # (0 - 127)
   kkkkkkk = 60: Middle C of keyboard
3. vvvvvvv: key velocity
   A logarithmic scale would be advisable.

   ```
   0     1                        64                 127
   off   ppp   pp   p   mp   mf   f       ff         fff
   ```

   vvvvvvv = 64: in case of no velocity sensors
   vvvvvvv = 0: Note Off, with velocity = 64
4. Any Note On message sent should be balanced by sending a Note Off message for that note in that channel at some later time.
5. ccccccc: control number

ccccccc	Description
0	Continuous Controller 0 MSB
1	Continuous Controller 1 MSB (MODULATION BENDER)
2	Continuous Controller 2 MSB
3	Continuous Controller 3 MSB
4-31	Continuous Controllers 4-31 MSB
32	Continuous Controller 0 LSB
33	Continuous Controller 1 LSB (MODULATION BENDER)
34	Continuous Controller 2 LSB
35	Continuous Controller 3 LSB
36-63	Continuous Controllers 4-31 LSB
64-95	Switches (On/Off)
96-121	Undefined
122-127	Reserved for Channel Mode messages (see Table III).

6. All controllers are specifically defined by agreement of the MIDI Manufacturers Association (MMA) and the Japan MIDI Standards Committee (JMSC). Manufacturers can request throught the MMA or JMSC that logical controllers be assigned to physical ones as necessary. The controller allocation table must be provided in the user's operation manual.
7. Continuous controllers are divided into Most Significant and Least Significant Bytes. If only seven bits of resolution are needed for any particular controllers, only the MSB is sent. It is not necessary to send the LSB. If more resolution is needed, then both are sent, first the MSB, then the LSB. If only the LSB has changed in value, the LSB may be sent without re-sending the MSB.
8. vvvvvvv: control value (MSB)
   - for controllers
     0: min
     127: max
   - for switches

0: off
127: on
Numbers 1 through 126, inclusive, are ignored.

9. Any messages (e.g. Note On), which are sent successively under the same status, can be sent without a Status byte until a different Status byte is needed.
10. Sensitivity of the pitch bender is selected in the receiver. Center position value (no pitch change) is 2000H, which would be transmitted EnH-00H-40H.

TABLE III

CHANNEL MODE MESSAGES>

```
 STATUS      DATA BYTES       DESCRIPTION
1011nnnn  0ccccccc 0vvvvvvv  Mode Messages
                             ccccccc = 122: Local Control
                             vvvvvvv = 0,   Local Control Off
                             vvvvvvv = 127, Local Control On

                             ccccccc = 123: All Notes Off
                             vvvvvvv = 0

                             ccccccc = 124: Omni Mode Off (All Notes Off)
                             vvvvvvv = 0

                             ccccccc = 125: Omni Mode On (All Notes Off)
                             vvvvvvv = 0

                             ccccccc = 126: Mono Mode On (Poly Mode Off)
                                                         (All Notes Off)
                             vvvvvvv = M, where M is the number of channels.
                             vvvvvvv = 0, the number of channels equals
                                          the number of voices in the receiver.

                             ccccccc = 127: Poly Mode On (Mono Mode Off)
                             vvvvvvv = 0    (All Notes Off)
```

NOTES:

1. nnnn: Basic Channel # (1-16, coded as defined in Table I)
2. Messages 123 thru 127 function as All Notes Off messages. They will turn off all voices controlled by the assigned Basic Channel. Except for message 123, All Notes Off, they should not be sent periodically, but only for a specific purpose. In no case should they be used in lieu of Note Off commands to turn off notes which have been previously turned on. Therefore any All Notes Off command (123-127) may be ignored by receiver with no possibility of notes staying on, since any Note On command must have a corresonding specific Note Off command.
3. Control Change #122, Local Control, is optionally used to interrupt the internal control path between the

keyboard, for example, and the sound-generating circuitry. If 0 (Local Off mesage) is received, the path is disconnected: the keyboard data goes only to MIDI and the sound-generating circuitry is controlled only by incoming MIDI data. If a 7FH (Local On message) is received, normal operation is restored.

4. The third byte of "Mono" specifies the number of channels in which Monophonic Voice messages are to be sent. This number, "M", is a number between 1 and 16. The channel(s) being used, then, will be the current Basic Channel (=N) thru N+M-1 up to a maximum of 16. If M=0, this is a special case directing the receiver to assign all its voices, one per channel, from the Basic Channel N through 16.

TABLE IV

SYSTEM COMMON MESSAGES

STATUS	DATA BYTES	DESCRIPTION
11110001		Undefined
11110010	0lllllll 0hhhhhhh	Song Position Pointer lllllll: (Least significant) hhhhhhh: (Most significant)
11110011	0sssssss	Song Select sssssss: Song #
11110100		Undefined
11110101		Undefined
11110110	none	Tune Request
11110111	none	EOX: "End of System Exclusive" flag

NOTES:

1. Song Position Pointer: Is an internal register which holds the number of MIDI beats (1 beat = 6 MIDI clocks) since the start of the song. Normally it is set to 0 when the START switch is pressed, which starts sequence playback. It then increments with every sixth MIDI clock receipt, until STOP is pressed. If CONTINUE is pressed, it continues to increment. It can be arbitrarily preset (to a resolution of 1 beat) by the SONG POSITION POINTER message.
2. Song Select: Specifies which song or sequence is to be played upon receipt of a Start (Real-Time) message.
3. Tune Request: Used with analog synthesizers to request them to tune their oscillators.
4. EOX: Used as a flag to indicate the end of a System Exclusive transmission (see Table VI).

TABLE V

SYSTEM REAL TIME MESSAGES

STATUS	DATA BYTES	DESCRIPTION
11111000		Timing Clock
11111001		Undefined

11111010	Start
11111011	Continue
11111100	Stop
11111101	Undefined
11111110	Active Sensing
11111111	System Reset

NOTES:

1. The System Real Time messages are for synchronizing all of the system in real time.
2. The System Real Time messages can be sent at any time. Any messages which consist of two or more bytes may be split to insert Real Time messages.
3. Timing clock (F8H)
   The system is synchronized with this clock, which is sent at a rate of 24 clocks/quarter note.
4. Start (from the beginning of song) (FAH)
   This byte is immediately sent when the PLAY switch on the master (e.g. sequencer or rhythm unit) is pressed.
5. Continue (FBH)
   This is sent when the CONTINUE switch is hit. A sequence will continue at the time of the next clock.
6. Stop (FCH)
   This byte is immediately sent when the STOP switch is hit. It will stop the sequence.
7. Active Sensing (FEH)
   Use of this message is optional, for either receivers or transmitters. This is a "dummy" Status byte that is sent every 300 ms (max), whenever there is no other activity on MIDI. The receiver will operate normally if it never receives FEH. Otherwise, if FEH is ever received, the receiver will expect to receive FEH or a transmission of any type every 300 ms (max). If a period of 300 ms passes with no activity, the receiver will turn off the voices and return to normal operation.
8. System Reset (FFH)
   This message initializes all of the system to the condition of just having turned on power. The system Reset message should be used sparingly, preferably under manual command only. In particular, it should not be sent automatically on power up.

TABLE VI

SYSTEM EXCLUSIVE MESSAGES

```
 STATUS   DATA BYTES   DESCRIPTION
11110000                Bulk dump etc.

           0iiiiiii     iiiiiii:  identification
              .
          (0*******)
              .         Any number of bytes may be sent here,
              .         for any purpose, as long as they all
          (0*******)    have a zero in the most significant bit.
              .
```

```
         11110111     EOX: "End of System Exclusive"
```

NOTES:

1. iiiiiii: identification ID (0-127)
2. All bytes between the System Exclusive Status byte and EOX or the next Status byte must have zeroes in the MSB.
3. The ID number can be obtained from the MMA or JMSC.
4. In no case should other Status or Data bytes (except Real-Time) be interleaved with System Exclusive, regardless of whether or not the ID code is recognized.
5. EOX or any other Status byte, except Real-Time, will terminate a System Exclusive message, and should be sent immediately at its conclusion.

Heini Withagen, www@eeb.ele.tue.nl
