MIDI
MUSICAL INSTRUMENT DIGITAL INTERFACE

"MIDI data is simply a series of number values, from 0 to 255, that allow control events to be universally understood by different hardware and software."

Peter Kirn, *Real World Digital Audio*

Terms and Concepts

Note Messages	**Devices**	**Important Control Change Numbers**
Note-on/Note-off pair	Keyboard	#1 - Modulation wheel
- Note-on command	Controller keyboard	#7 - Volume
– Note-off command	Alternate controller	#10 - Pan
Note number (0-127)	Broadcast channel	#64 - Sustain pedal
Key velocity (0 and 1-127)	Polyphony	#123 - All notes off
- Attack velocity	Modes: Global/System, Instrument, Multitimbral, Song, and Sequencing	
- Release velocity		**Interconnection**
Channel number (1-16)		5-pin DIN connector
	Adding Expressivity	Apple's AudioMIDI Setup
Channel Messages	Expression pedal	Master controller/slave
Aftertouch	Modulation wheel	Daisy chain
- Polyphonic	Pitch-bend wheel	MIDI IN/OUT/THRU ports
- Channel	Sustain pedal	MIDI interface (2x2, 4x4, etc.)
Control change		
- Continuous (0-63)	**Data**	**Synchronization**
- Noncontinuous (64-127)	Baud rate	EBU
Pitch bend	Bit	MIDI Time Code (MTC) and MIDI Clock
- Course	Byte (8 bits)	SMPTE
- Fine	- Data (0-127)	
Program change	- Status	**Standards**
- Bank	Numeric notation	OpenSound Control (OSC)
- Program number	- Binary	General MIDI (GM)
	- Decimal	- GM Instrument Library
System Messages	- Hexadecimal	- GM Perc. Key Map
System common	MIDI bus	MIDI 1.0 Specification
System real-time	MIDI data format	MIDI Implementation Chart
System-exclusive (SYSEX)	MIDI data rate (31.25 kb)	National Association of Music Merchants (NAMM)
	MIDI event	
	MSB/LSB	
	Pulse per quarter (PPQ)	
	Standard MIDI File (SMF)	
	- Type 0	
	- Type 1	

Reading

Charles Dodge and Thomas Jerse, "Standard Interfaces for Musical Devices" in *Computer Music: Synthesis, Composition, and Performance* (New York: Schirmer, 1995), 407-12.
Peter Kirn, *Real World Digital Audio* (Berkeley: Peachpit Press, 2006), 285-91, 293-303, 324-25.

Quotable

"Striking a key on a clavier produces a three-byte MIDI data sequence: a *Note On* command, followed by the *note (key)* number to indicate pitch…and then a numerical value that is a measurement of the *key velocity* imparted by the performer. When multiple keys are struck to play a chord, the commands are sent in the order the keys are depressed. Releasing a key also initiates a three-byte sequence: a *Note Off* command, followed by the number of the note that has been turned off, and a numerical value that indicates how quickly the key returns to its original position."

"System real-time messages are used to synchronize MIDI devices with start, stop and continue commands that control a sequence of events. There is also a MIDI timing clock that transmits a pulse 24 times per quarter note [*relative* to] the current tempo."

"The MIDI standard provides a definition of MIDI Time code (MTC) which can be used to synchronize MIDI-compatible devices. MTC contains an *absolute* description of the time in hours, minutes, seconds, and fractions of a second."

"To make the timing of the data unambiguous to the receiver, an 8-bit byte of data is framed by a start bit and a stop bit…each data byte actually requires 10 bits to travel down the circuit, one after the other…As another cost-saving measure, the MIDI communications bus is essentially unidirectional, so that it does not promote two-way conversations between devices."

Charles Dodge and Thomas Jerse, *Computer Music*

"When you go beyond the realm of pedals, keyboards, and faders, MIDI's rigid way of defining music performance can get awkward fast…the data bandwidth of MIDI is painfully narrow [and] MIDI uses absurdly small, rigid chunks of data."

Peter Kirn, *Real World Digital Audio*