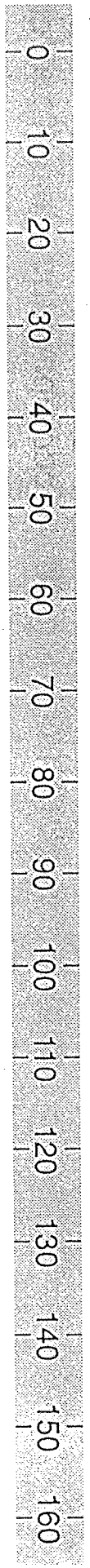


threshold of hearing p_0

quiet recording studio

quiet living room

quiet office

subdued conversation

average office

average conversation

small orchestra

busy street

average factory

heavy truck traffic

subway

$10 \log_{10} \frac{p}{p_0}$

loud rock music

power tools

thunder

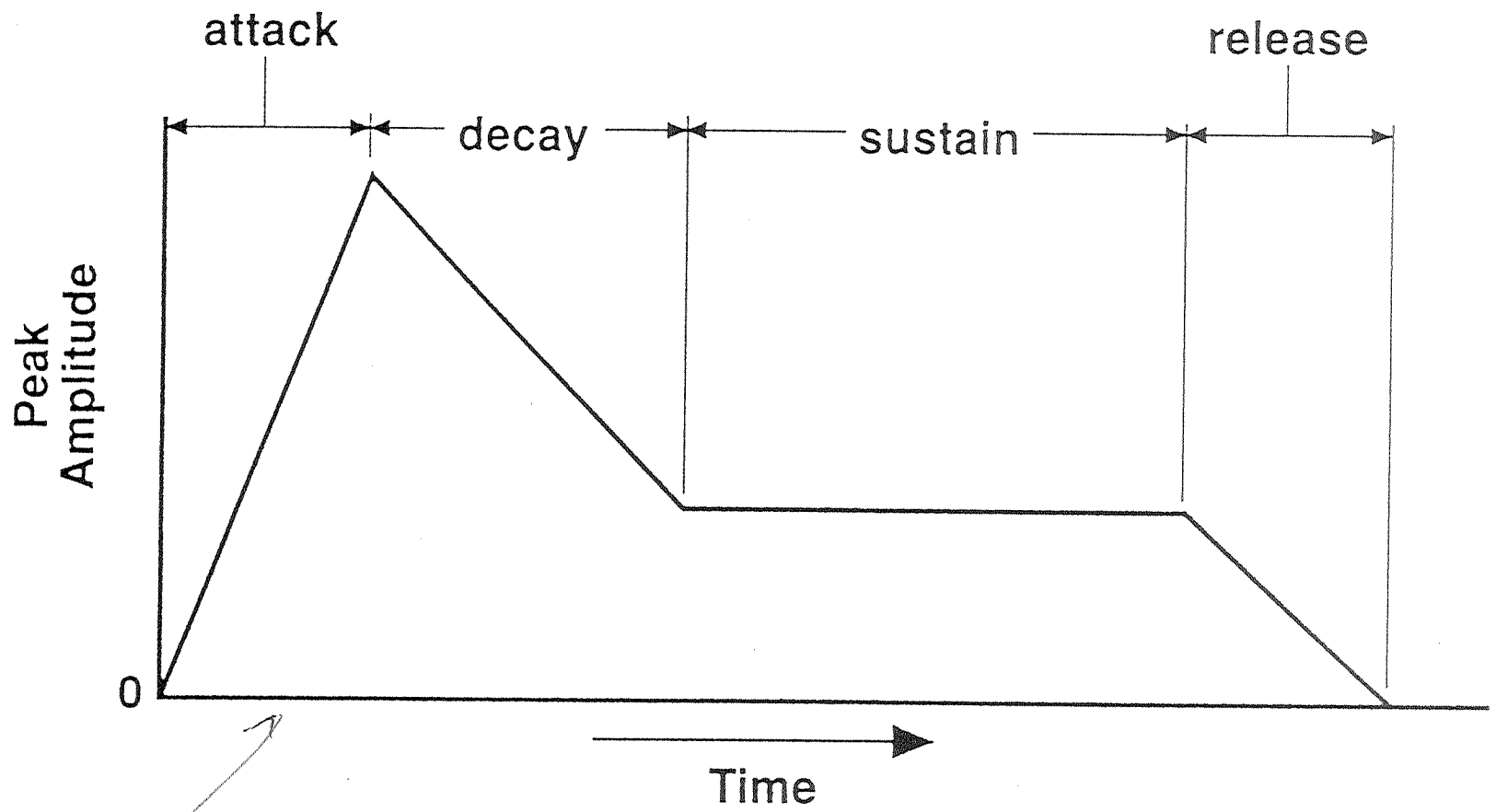
airport runway

jack hammer

pain threshold

1 bel = 10 dec.bel

jet engine in close proximity



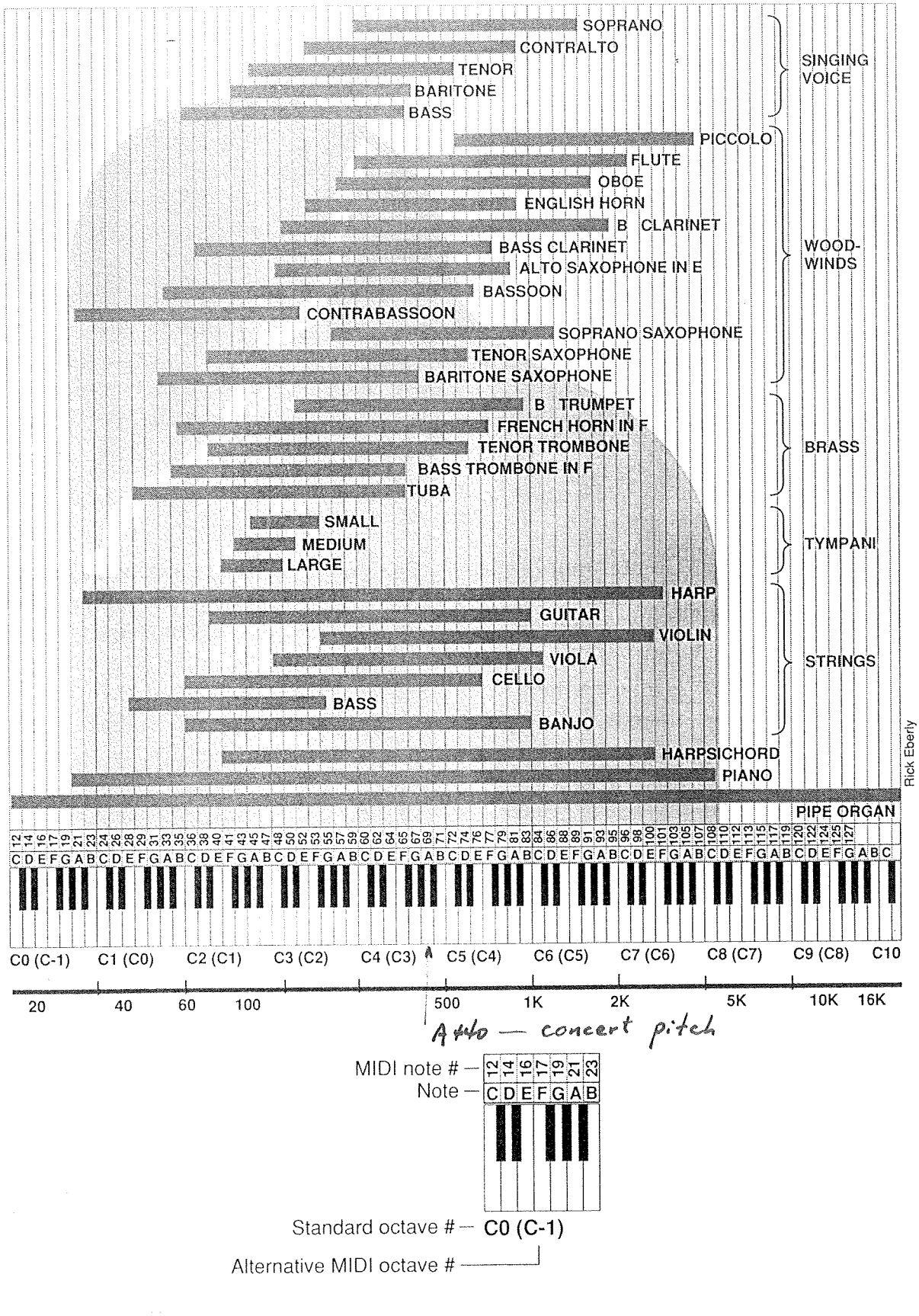
Electronic Organ — ① immediate attack, ② full sustain / no decay (when key depressed)
③ immedite release (when key released)

Wind instrument — longer attack & release, sustain determined by applied air pressu

piano — ① no sustain, ② longer release is possible when "sustain pedal" depressed

Figure 10.1

The pitch and fundamental frequency ranges for common instruments.



More on Timbre (Tone color)

Fourier's theory: any waveform may be decomposed into a series of sine waves.

Fundamental frequency:

e.g. 440 Hz in A-440

Harmonics

- integer multiples of fundamentals

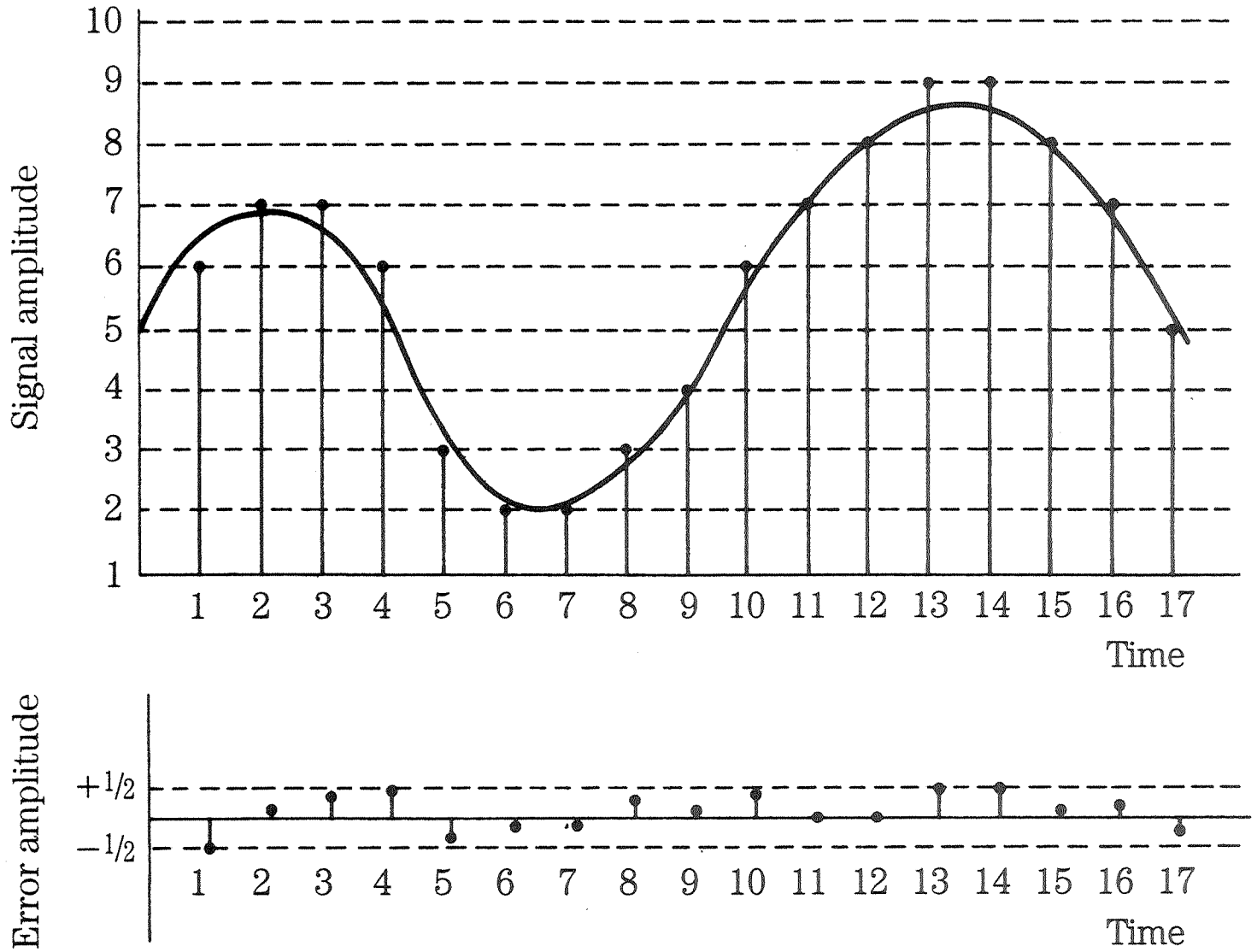
Partials

- other non-integer multiples of fundamentals.

Overtones

- often refers to both harmonics & partials.

- It is the number and relative strengths of harmonics and partials that determine the tone color (timbre).



2-7 Quantization error at sample times.

Introduction to MIDI (Musical Instrument Digital Interface)

Definition of MIDI: a protocol that enables computer, synthesizers, keyboards, and other musical devices to communicate with each other.

1. Terminologies:

Synthesizer:

- It is a sound generator (various pitch, loudness, duration, tone color).
- A good (musician's) synthesizer often has a microprocessor, keyboard, control panels, memory, etc.

Sequencer:

- It can be a stand-alone unit or a software program for a personal computer. (It used to be a storage server for MIDI data. Nowadays it is more a software *music editor* on the computer.)
- It has one or more MIDI INs and MIDI OUTs.

Track:

- Track in a sequencer is used to organize recordings.
- Tracks can be turned on or off on recording or playing back.

Channel:

- MIDI channels are used to separate information in a MIDI system.
- There are 16 MIDI channels in one cable.
- Channel numbers are coded into each MIDI message.

Timbre:

- The *tone quality* or *tone color*, e.g., flute sound, cello sound, etc.
- Multitimbral -- capable of playing many different sounds at the same time (e.g., piano, brass, drums, etc.)

Pitch:

- musical note that the instrument plays

Voice:

- Voice is the portion of the synthesizer that produces sound.
- Synthesizers can have many (e.g., 16, 20, 24, 32, 64, 128) voices.
- Voice work independently and simultaneously to produce sounds of different timbre and pitch.

Patch:

- The control settings that define a particular timbre.

2. Hardware Aspects of MIDI

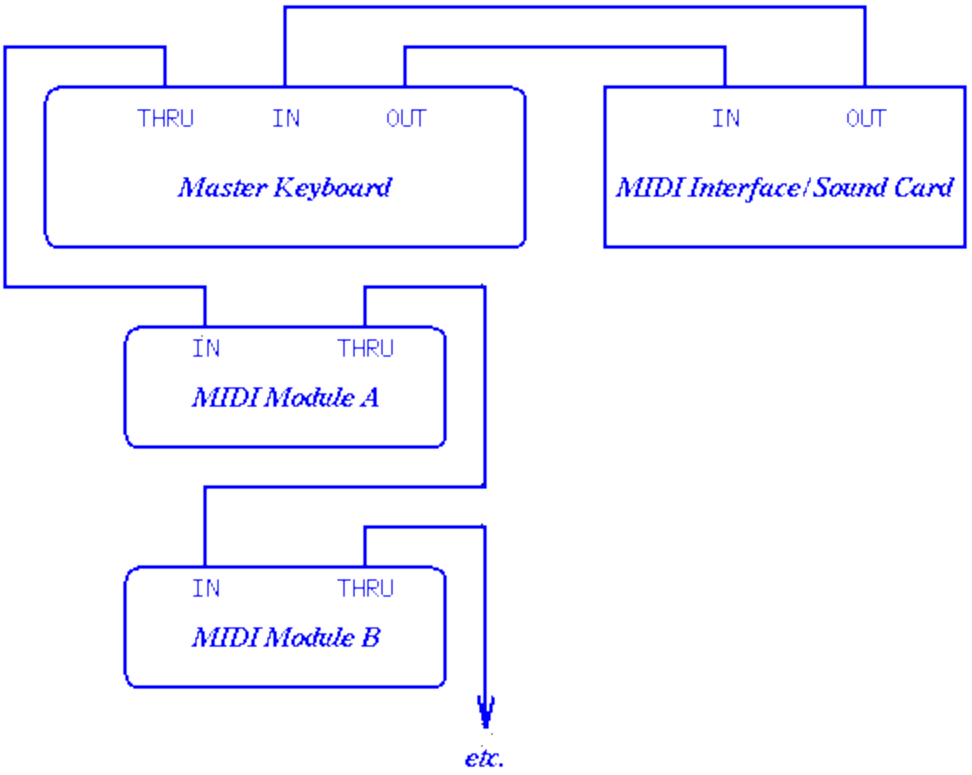
MIDI connectors:

-- three 5-pin ports found on the back of every MIDI unit

- MIDI IN: the connector via which the device receives all MIDI data.
- MIDI OUT: the connector through which the device transmits all the MIDI data it generates itself.
- MIDI THROUGH: the connector by which the device echoes the data receives from MIDI IN.

Note: It is only the MIDI IN data that is echoed by MIDI THROUGH. All the data generated by device itself is sent through MIDI OUT.

Example: A Typical MIDI Sequencer Setup:



- MIDI OUT of the synthesizer (the "Master Keyboard") is connected to MIDI IN of the sequencer.
- MIDI OUT of the sequencer is connected to MIDI IN of the synthesizer and "through" to each of the additional sound modules.
- During recording, the keyboard-equipped synthesizer is used to send MIDI message to the sequencer, which records them.
- During play back: messages are send out from the sequencer to the synthesizer and the sound modules which will play back the music.

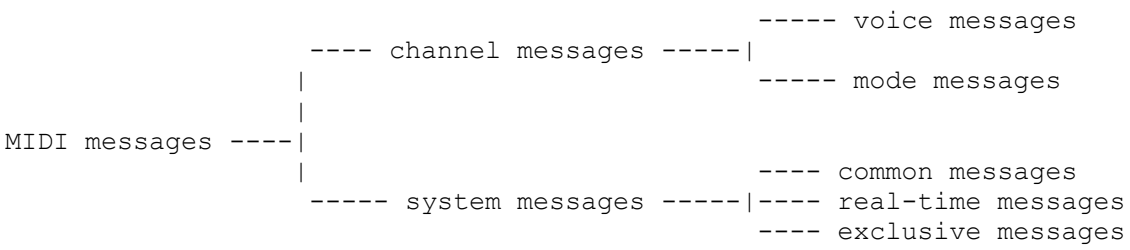
3. MIDI Messages

-- MIDI messages are used by MIDI devices to communicate with each other.

Structure of MIDI messages:

- MIDI message includes a *status byte* and up to two *data bytes*.
- Status byte
  - The most significant bit of status byte is set to 1.
  - The 4 low-order bits identify which channel it belongs to (four bits produce 16 possible channels).
  - The 3 remaining bits identify the message.
- The most significant bit of data byte is set to 0.

Classification of MIDI messages:



A. Channel messages:

-- messages that are transmitted on individual channels rather that globally to all devices in the MIDI network.

A.1. Channel voice messages:

- Instruct the receiving instrument to assign particular sounds to its voice
- Turn notes on and off
- Alter the sound of the currently active note or notes

Voice Message	Status Byte	Data Byte1	Data Byte2
Note off	&H8x	Key number	Note Off velocity
Note on	&H9x	Key number	Note on velocity
Polyphonic Key Pressure	&HAx	Key number	Amount of pressure
Control Change	&HBx	Controller number	Controller value
Program Change	&HCx	Program number	None
Channel Pressure	&HDx	Pressure value	None
Pitch Bend	&HEx	MSB	LSB

Notes: `x' in status byte hex value stands for a channel number.

Example: a Note On message is followed by two bytes, one to identify the note, and on to specify the velocity.
To play note number 80 with maximum velocity on channel 13, the MIDI device would send these three hexadecimal byte values: &H9C &H50 &H7F

A.2. Channel mode messages: -- Channel mode messages are a special case of the Control Change message (&HBx or 1011nnnn). The difference between a Control message and a Channel Mode message, which share the same status byte value, is in the first data byte. Data byte values 121 through 127 have been reserved in the Control Change message for the channel mode messages.

- Channel mode messages determine how an instrument will process MIDI voice messages.

1st Data Byte	Description	Meaning of 2nd Data Byte
&H79	Reset all controllers	None; set to 0
&H7A	Local control	0 = off; 127 = on
&H7B	All notes off	None; set to 0
&H7C	Omni mode off	None; set to 0
&H7D	Omni mode on	None; set to 0
&H7E	Mono mode on (Poly mode off)	**
&H7F	Poly mode on (Mono mode off)	None; set to 0

** if value = 0 then the number of channels used is determined by the receiver; all other values set a specific number of channels, beginning with the current basic channel.

B. System Messages:

- System messages carry information that is not channel specific, such as timing signal for synchronization, positioning information in pre-recorded MIDI sequences, and detailed setup information for the destination device.

B.1. System real-time messages:

- messages related to synchronization

System Real-Time Message	Status Byte
Timing Clock	&HF8
Start Sequence	&HFA
Continue Sequence	&HFB
Stop Sequence	&HFC
Active Sensing	&HFE
System Reset	&HFF

B.2. System common messages:

- contain the following unrelated messages

System Common Message	Status Byte	Number of Data Bytes
MIDI Timing Code	&HF1	1
Song Position Pointer	&HF2	2
Song Select	&HF3	1
Tune Request	&HF6	None

B.3. System exclusive message:

- (a) Messages related to things that cannot be standardized, (b) addition to the original MIDI specification.

- It is just a stream of bytes, all with their high bits set to 0, bracketed by a pair of system exclusive start and end messages (&HF0 and &HF7).

4. General MIDI

- MIDI + Instrument Patch Map + Percussion Key Map --> a piece of MIDI music sounds the same anywhere it is played
    - Instrument patch map is a standard program list consisting of 128 patch types.
    - Percussion map specifies 47 percussion sounds.
    - Key-based percussion is always transmitted on MIDI channel 10.
- Requirements for General MIDI Compatibility:
    - Support all 16 channels.
    - Each channel can play a different instrument/program (*multitimbral*).
    - Each channel can play many independent voices/melodies (*polyphony vs. monophony*).
    - Minimum of 24 fully dynamically allocated voices.

Appendix

A1. General MIDI Instrument Patch Map

A2. General MIDI Percussion Key Map

Top | CMPT 365 Home Page