“MIDI 2018”

The Musical Instrument Digital Interface
Thirty-Five Years and Counting

Tom Moxon

- Product Design Consultant
  PatternAgents, LLC
  Halcyon Modular LLC
  (Audio, Video, Electronic Product Design)

- "HardLight VR" Suit HW/FW Design
  Full Body Motion Capture with Haptics

- Integrated Circuit & Chip Design consulting for
  Cray Research, SGI/MIPS/Nintendo, Hyundai, others

- Design Engineer, EMU Systems
  Applied Magic for the Arts
  Drumulator/E-Drum
  Emulator2 Sampler

Musical Instrument Digital Interface

- is an electronic musical instrument industry specification that allows a variety of digital musical instruments and other devices to connect and communicate with one another.

- The MIDI Standard is administered by the MIDI Manufacturers Association, on the web at : http://www.midi.org

MIDI MANUFACTURERS ASSOCIATION

Source : MMA/midi.org

Musical Instrument Digital Interface

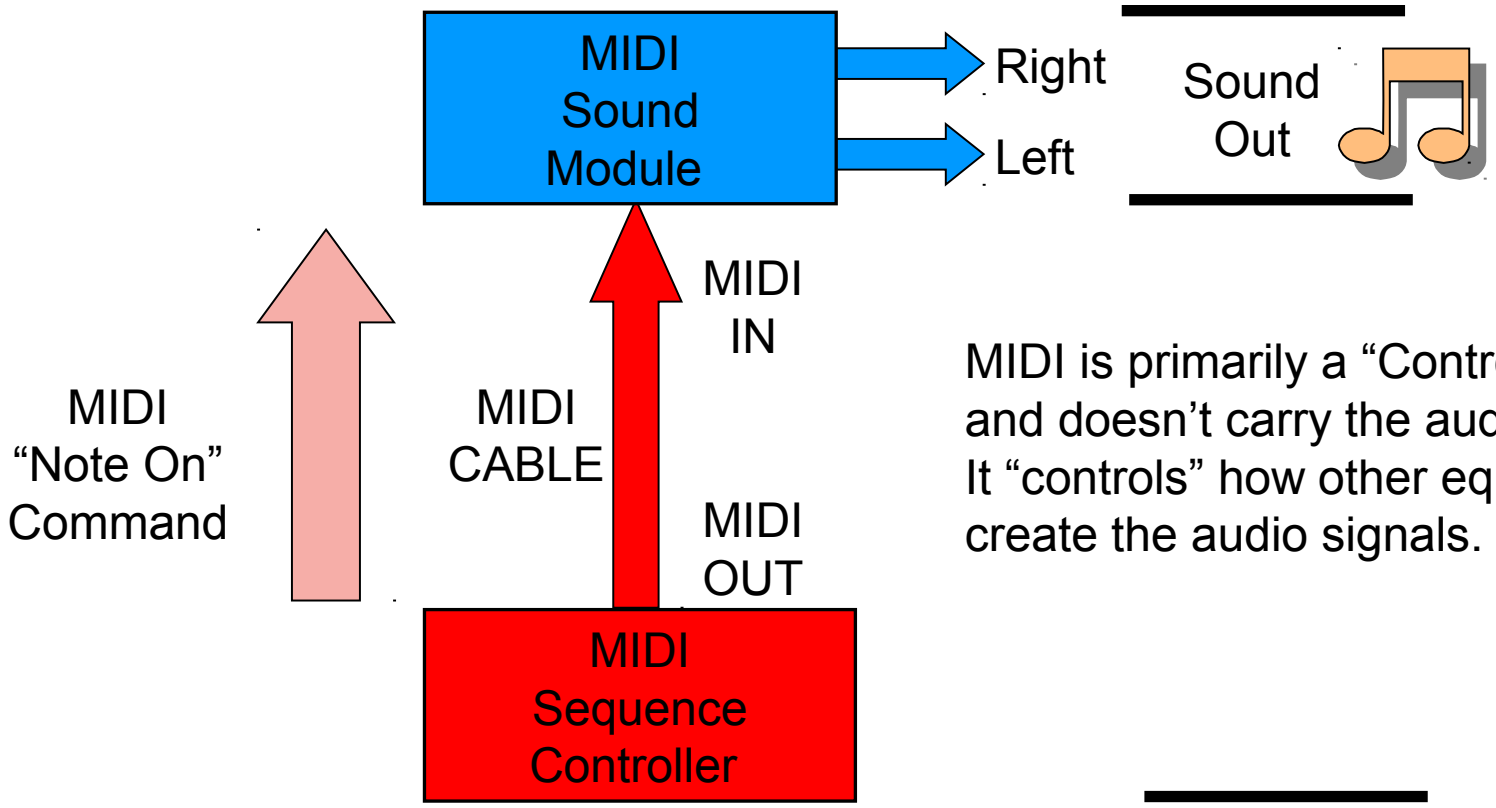
- is a set of standard commands that allow electronic musical instruments, performance controllers, computers and related devices to communicate, as well as a hardware standard that guarantees compatibility between them.

  Commands include messages as : note on, note off, key pressure, pitch bend, etc.

MIDI control of a Yamaha Piano by Disklavier Module

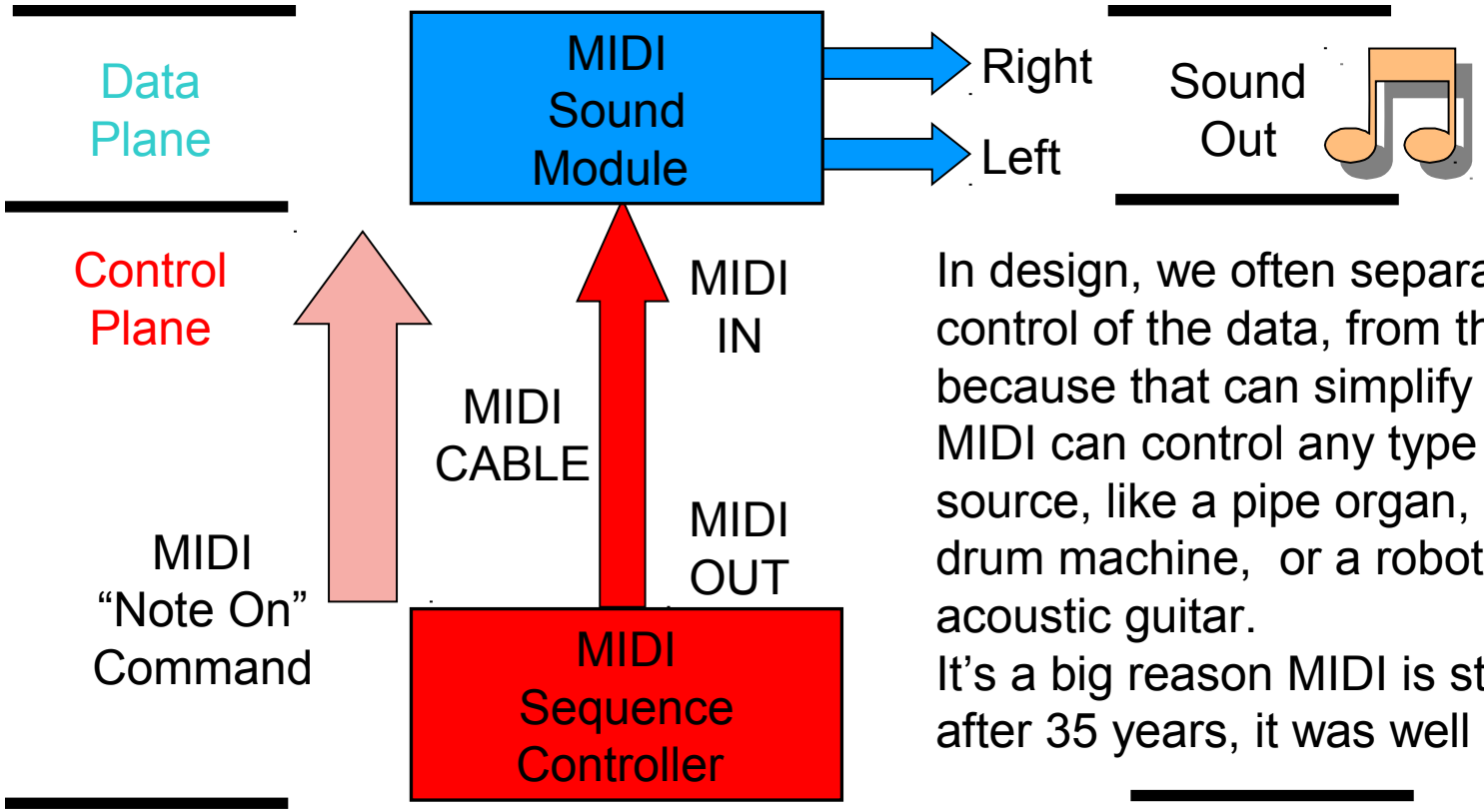
Source : midi,org

MIDI Control Example



MIDI is primarily a "Control" interface and doesn't carry the audio signals. It "controls" how other equipment create the audio signals.

Control & Data Planes

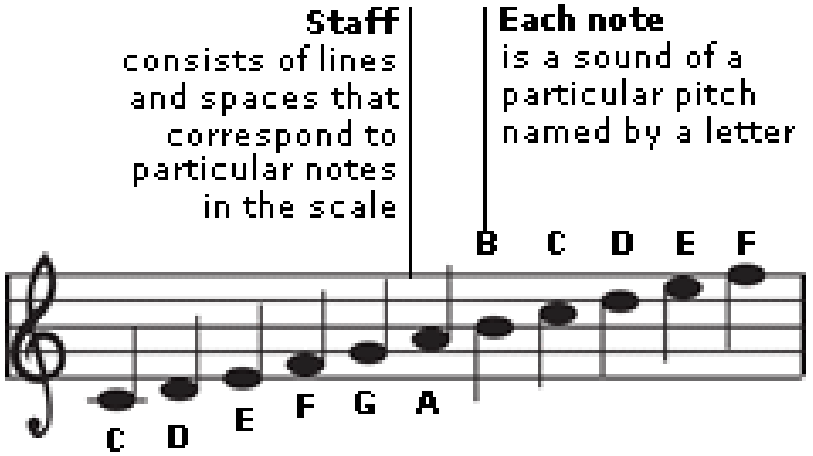


In design, we often separate the control of the data, from the data itself; because that can simplify the problem. MIDI can control any type of data source, like a pipe organ, synthesizer, drum machine, or a robot playing an acoustic guitar.

It's a big reason MIDI is still popular after 35 years, it was well engineered

The Dark Ages – Before MIDI

- Musicians realized early on that they needed a language to communicate, and the first machine printed music scores appeared around 1473, and this was the “manual control” standard until...



Source : wikipedia

Paper Roll Control

- The “Music Roll” appeared in 1877, and Player Pianos appeared in halls and bars

- Most had basic control : note on, note off an early equivalent of what MIDI does today…

Source : wikipedia

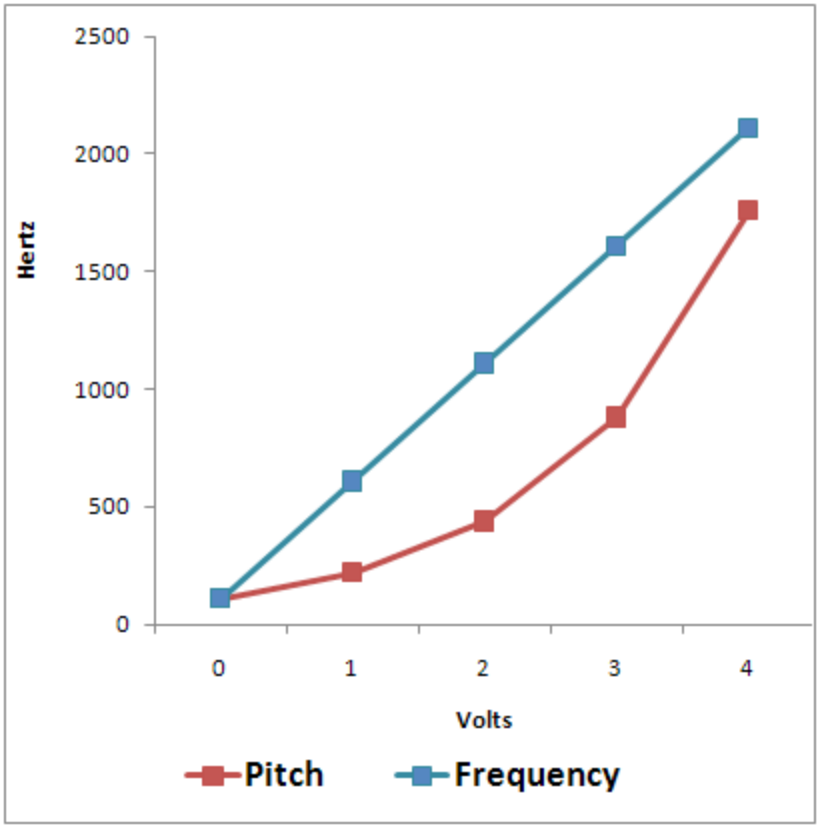
Voltage Control

- In the 1960's voltage control of audio became popular with the analog synthesizers designed by Bob Moog, Don Buchla, and others

Source : wikipedia

Volt per Octave Pitch Control

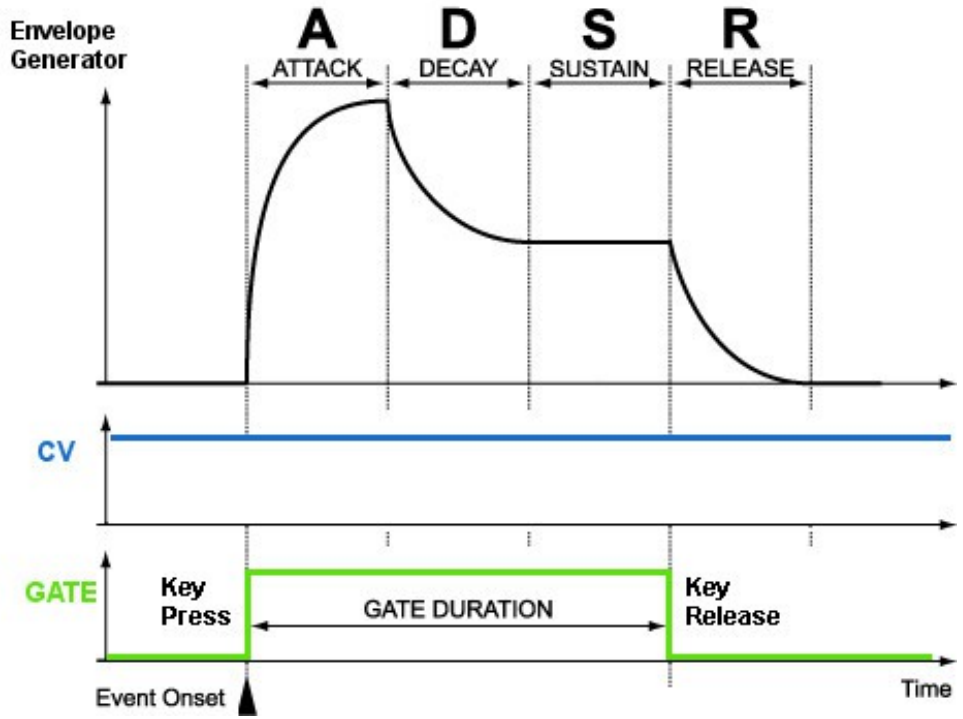
- Voltage Control, or “CV” was often used with Volt per Octave Pitch control.
- “Subjective Pitch” can be nonlinear, and is measured in Mels.

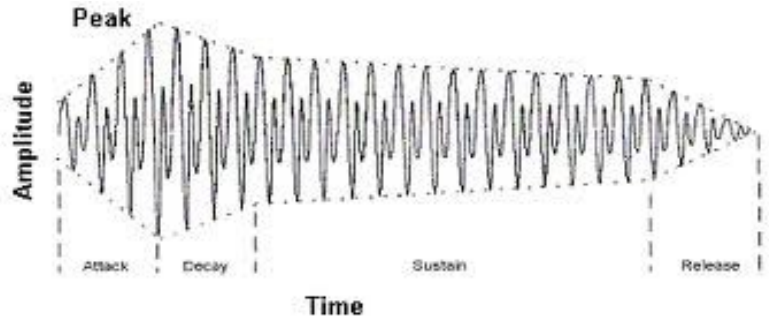


Source : emusician.com

CV/Gate

- "**Control Voltage/Gate**" is the term used when an analog ***Control Voltage*** signal is used in conjunction with a digital ***Gate*** signal





Source : moxon

Early CV/Gate Disadvantages

- Wiring : it took many patch cables
- Compatibility : different voltages
- Calibration : many voltage sources
- Drift : with temperature, tuning issue
- Reliability : too many connections to fail
- Portability : too many connections to rewire
- Patches : each song required rewiring
- Polyphony : difficult for more than a solo voice

Source : moxon

Digital/Computer Based

- Early computer music programs stated in the 1950's; and by the 1970's many universities and groups were using mini-computers for direct sound generation, the beginning of widely available Digital Signal Processing (a topic in itself…)

DEC PDP/11 Computer

Source : moxon

Computer Interfaces

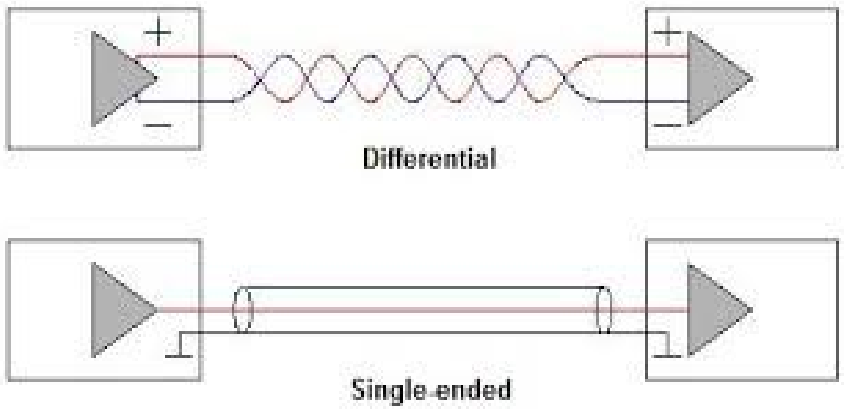
- Synthesizer makers started adding serial (RS-232) computer interfaces to their instruments; but there was no industry standard language or protocol for how to “talk” to each other, so interoperability was very limited between models

Babel and Confusion

- Before MIDI there were several competing interface standards from Roland (DCB), Emu, Yamaha, and others. Most were proprietary and worked only between their own gear
- There were physical incompatibility issues as well as software protocol incompatibility issues
- There was also concern about being able to retrofit older, or even recent gear to work together

Hardware Differences

- Differential Signals vs. Single Ended



- USB, AppleTalk, and DMX are differential examples MIDI, and Modular CV are single-ended examples
- Differential is generally faster, more noise tolerant and works over longer distance, but more $$$

Protocol Differences

- 1980's - Newer communications chips (like the Zilog Z-80 SCC) could do both synchronous and asynchronous protocols as well as hardware addressing, multidrop
- Some companies (Emu, Oberheim, etc.) wanted to make use of those newer capabilities
- Bigger companies (Roland, Yamaha,etc) were more concerned with retrofitting existing gear

Universal Synthesizer Interface

- Dave Smith and Chet Wood of Sequential Circuits devised a “Universal Synthesizer Interface”, and proposed it to the Acoustic Engineering Society (AES) in 1981

Source : Sequential Circuits

Standards Process

- It took another two years of negotiation between companies like Sequential Circuits, Roland, Korg, Kawai, Oberheim, Moog, Yamaha, and others to get all the makers to adopt a common standard, renamed **MIDI**, the Musical Instrument Digital Interface

Source : midi.org

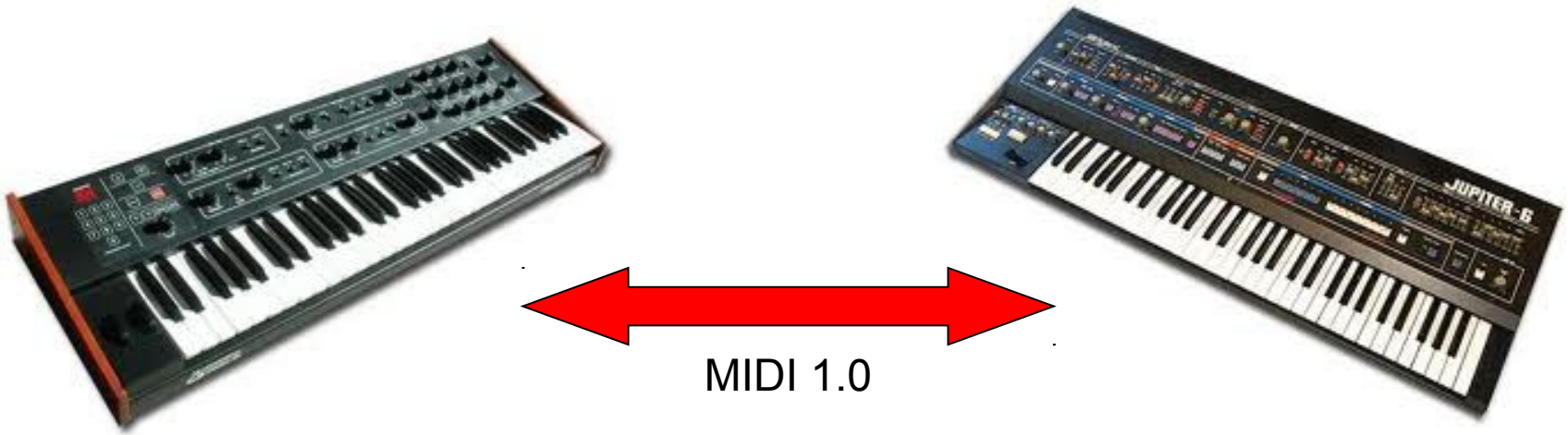
Enter MIDI…

- MIDI was announced to the public in the October 1982 edition of Keyboard Magazine in an article by Bob Moog

Source : Bob Moog Foundation

First MIDI Demonstrations

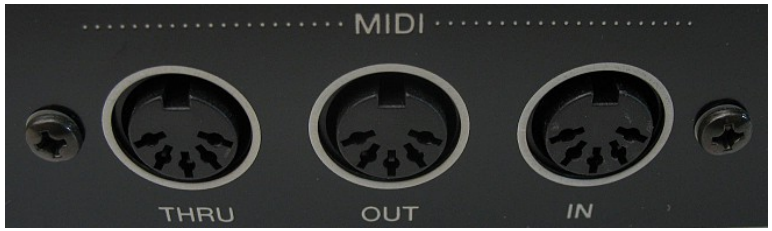
- At the January 1983 NAMM show, the MMA was able to demonstrate a Sequential Circuits Prophet 600 connected to Roland Jupiter-6 using the new MIDI 1.0 interface

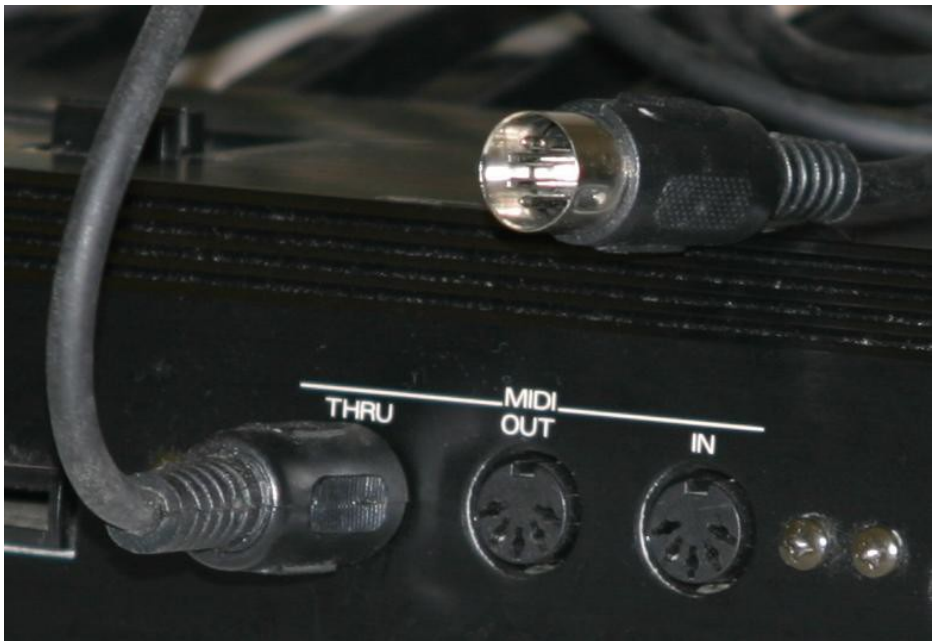


Source : midi.org

MIDI Cabling

- MIDI uses a 180 degree, five (5) pin DIN connector as cable termination

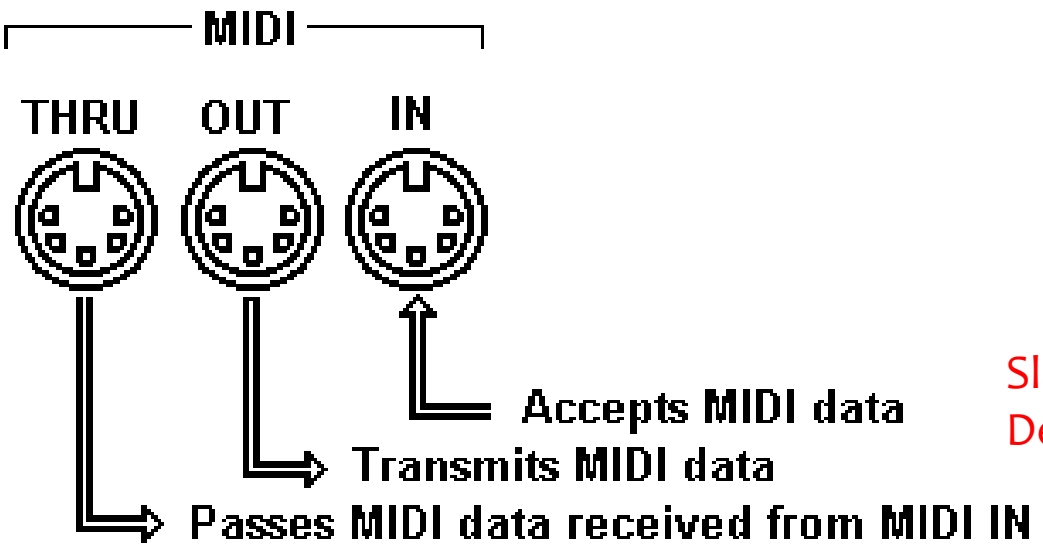


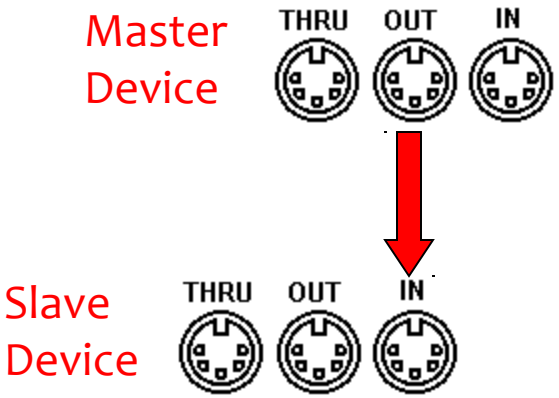


Source : midi.org

MIDI Wiring

- MIDI connects a master device (OUT) to a slave device (IN) as a minimum

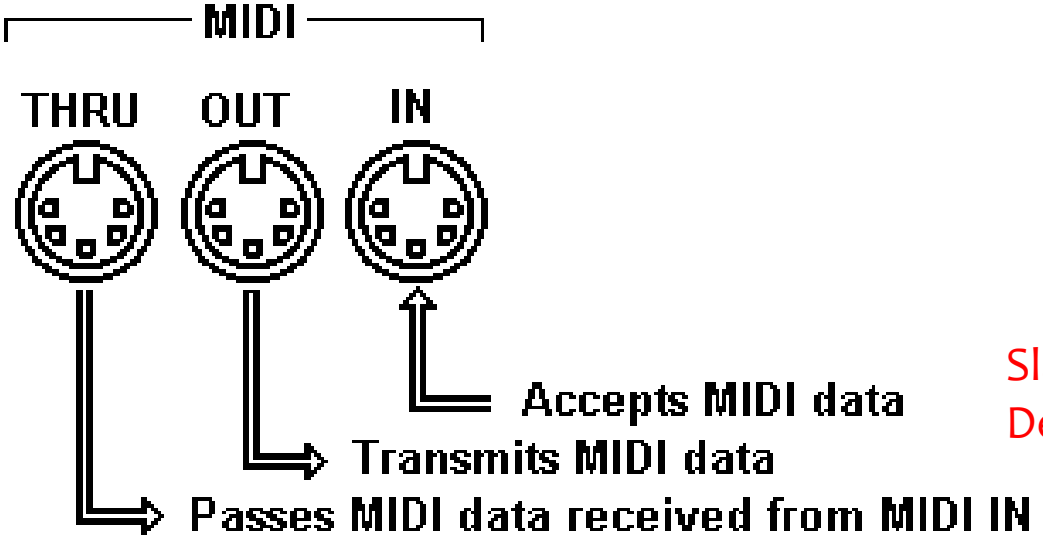


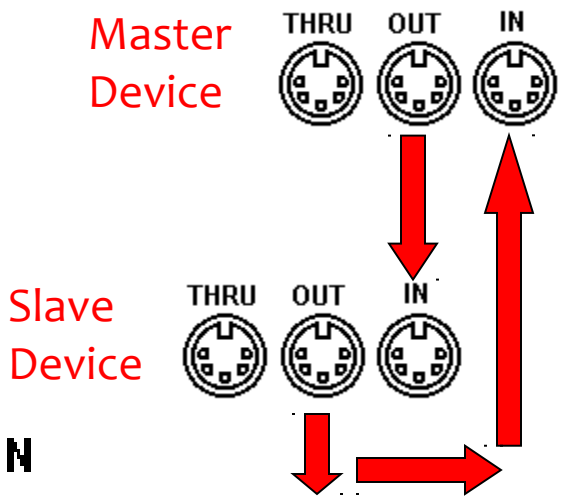


Source : midi.org

MIDI Signal Direction

- In order to receive data back from the Slave device, it's MIDI OUT needs to be wired back to the Master device MIDI IN

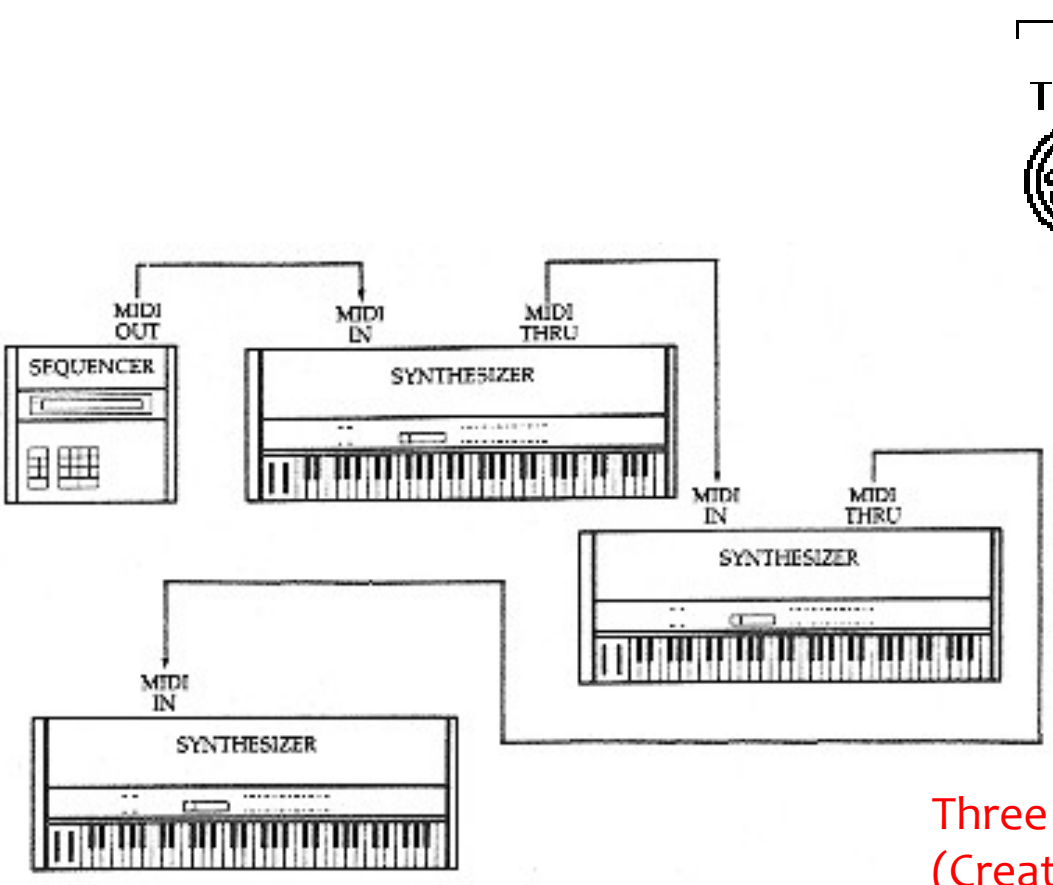


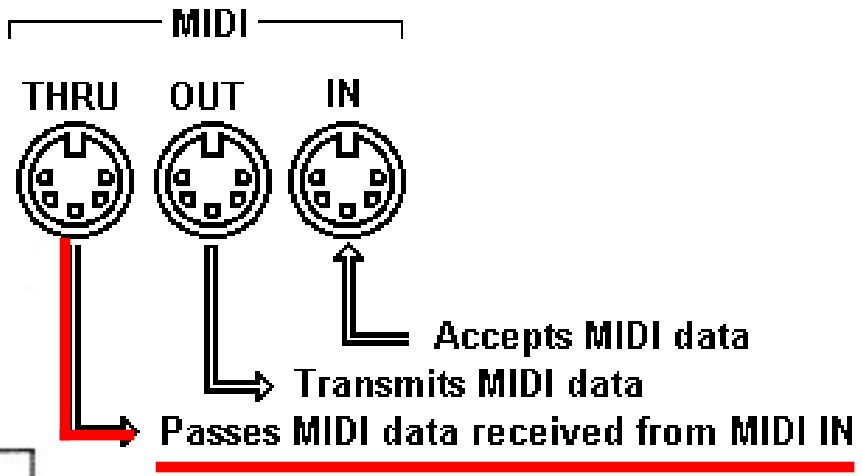


Source : midi.org

MIDI Daisy Chain

- MIDI “THRU” passes data from MIDI “IN”



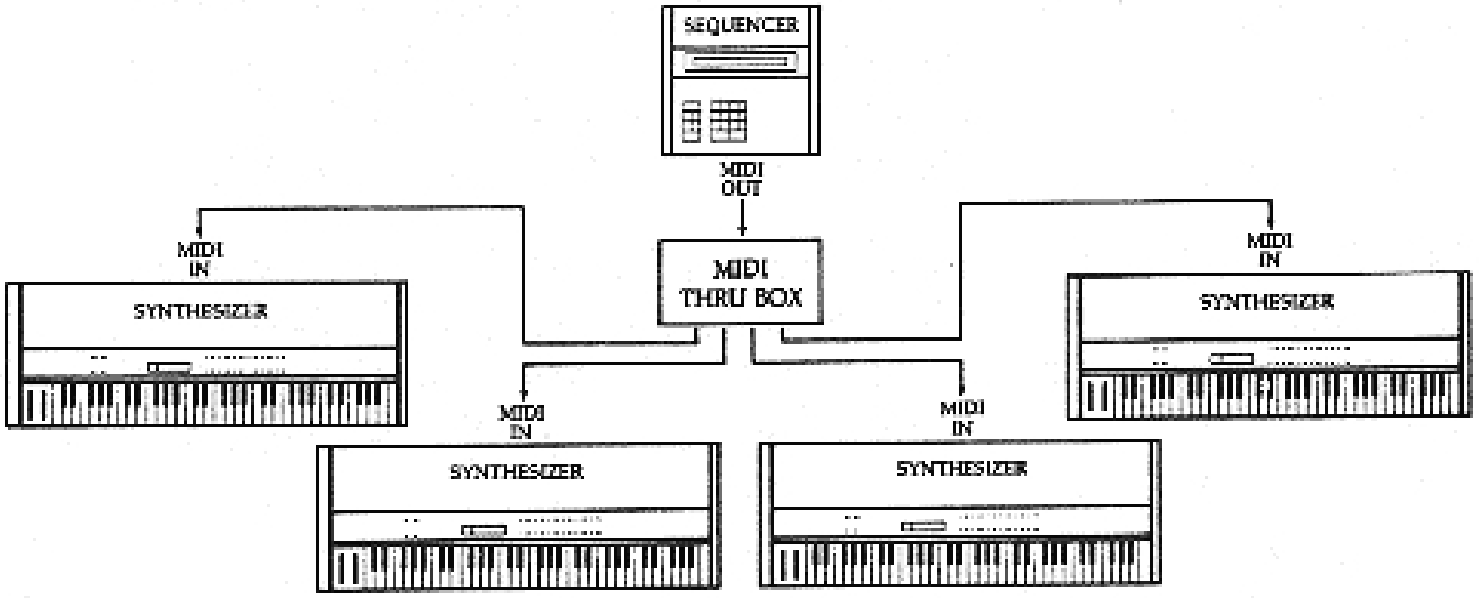


Three (3) Device Limit to Daisy Chain!
(Creates too much timing delay otherwise)

Source : midi.org

MIDI Distribution

- Better to use a “MIDI Thru Box” to distribute the MIDI output to many slave devices, this reduces timing jitter

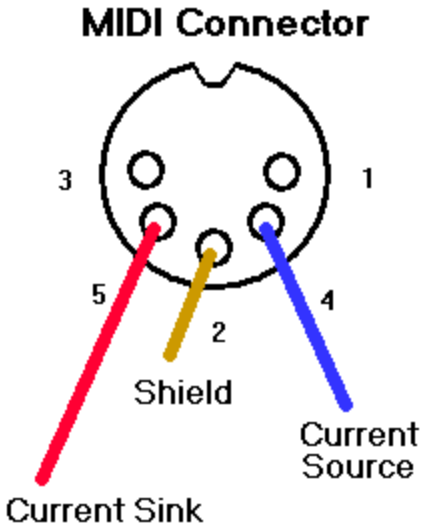


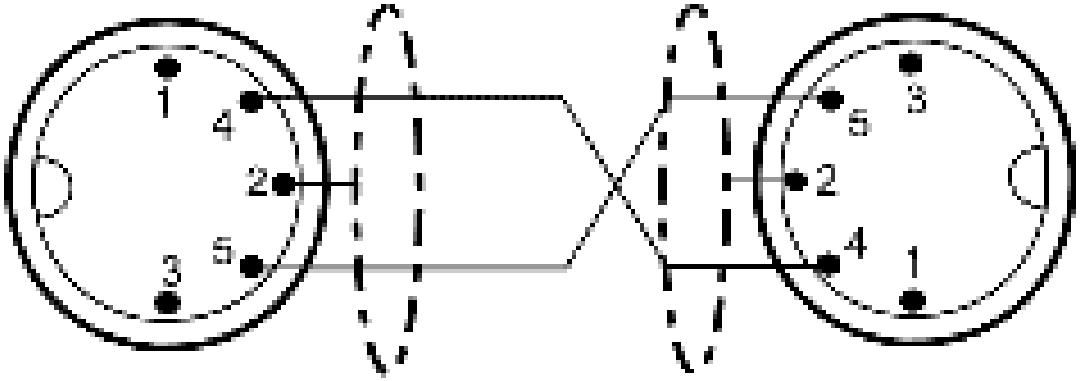
Maximum MIDI cable length is 50 feet!

Source : midi.org

MIDI – Under the hood…

- of the MIDI cable





Source : midi.org

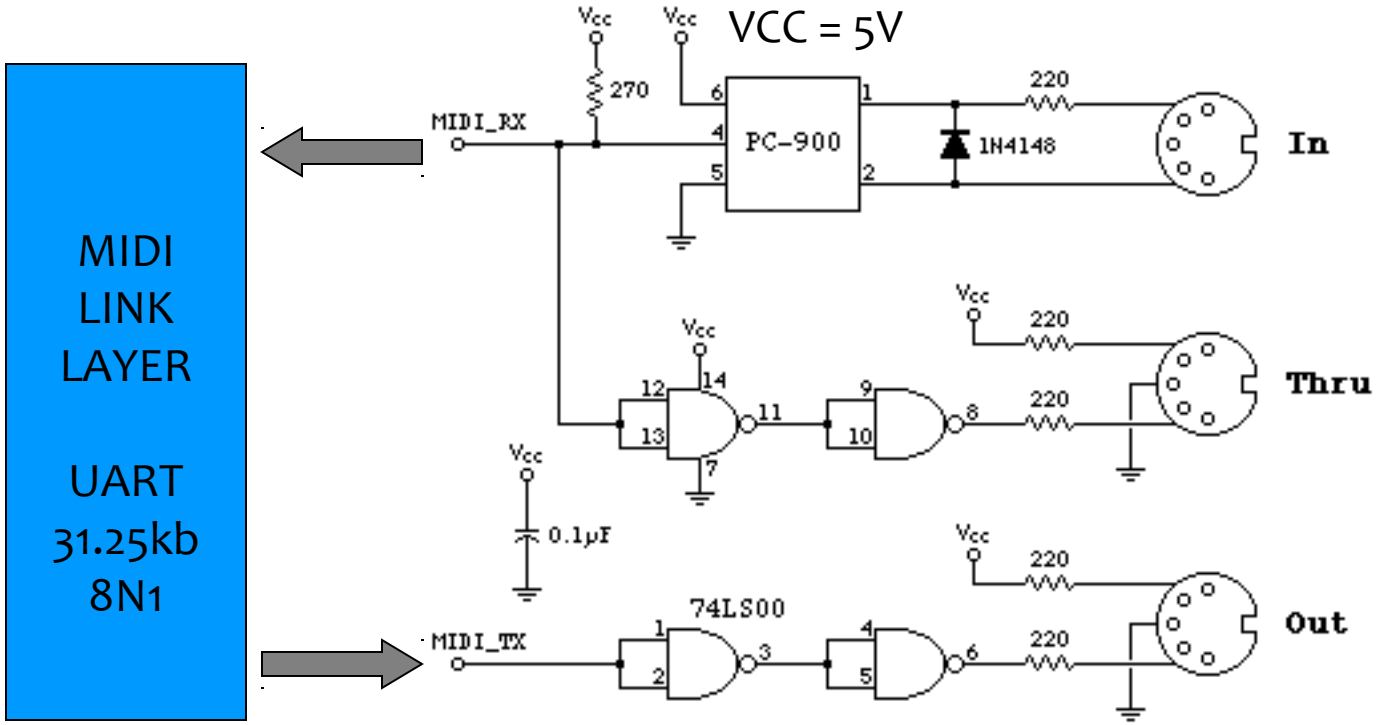
MIDI Physical Layer (PHY)

- PHY is the abbreviation for the PHYsical layer of the OSI model
- The MIDI physical circuit is a 5mA current loop, with logic Zero (0) as current on.
- MIDI is generally optically isolated in order to avoid ground loops that can cause “hum” and “static” in the audio signal

Source : midi.org

MIDI PHY

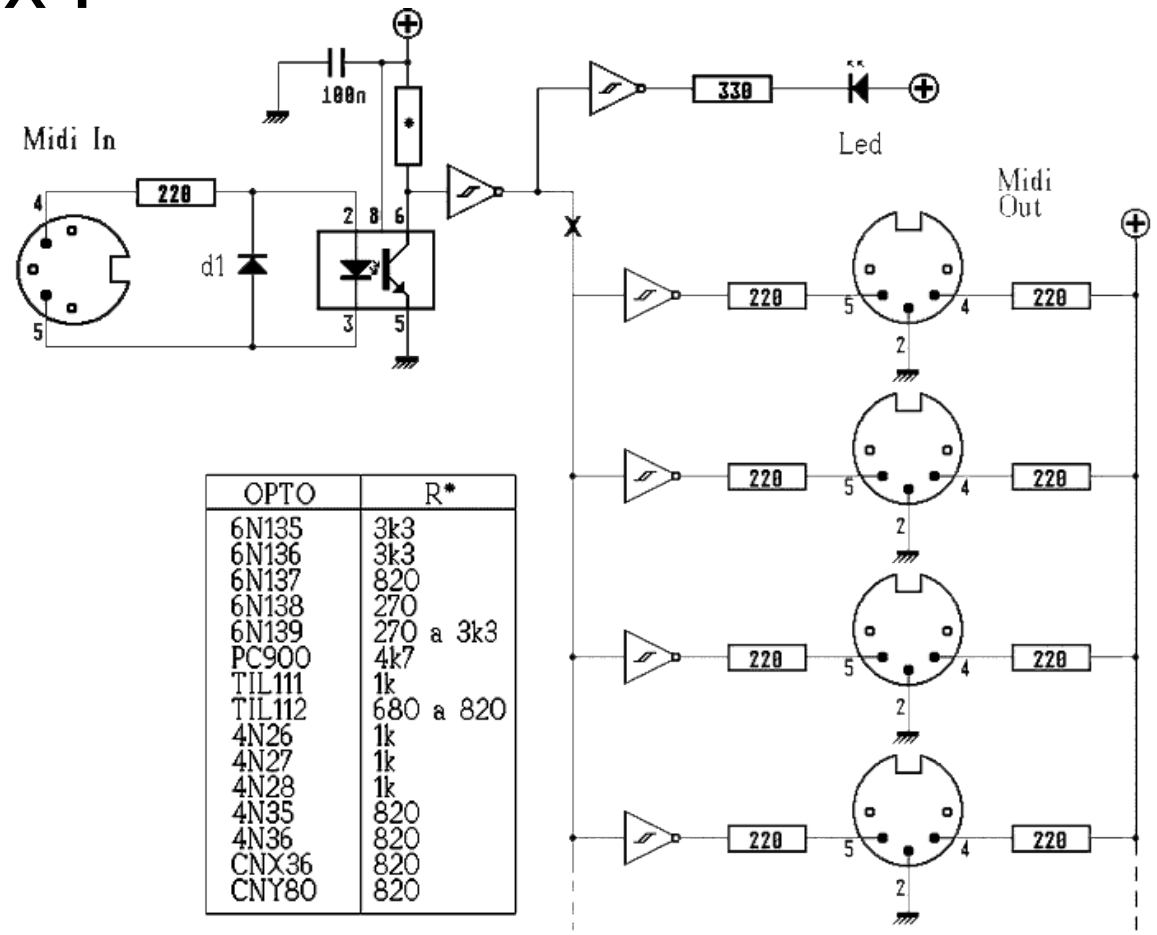
- Opto-Isolator is Sharp PC-900 or HP 6N138
- Traditionally, 7404 or 7400 drivers used



Source : midi.org

MIDI Thru Box

- MIDI Thru Box :



OPTO	R*
6N135	3k3
6N136	3k3
6N137	820
6N138	270
6N139	270 a 3k3
PC900	4k7
TIL111	1k
TIL112	680 a 820
4N26	1k
4N27	1k
4N28	1k
4N35	820
4N36	820
CNX36	820
CNY80	820

Source : http://m.bareille.free.fr/midithrubox/midithrubox.htm

MIDI Link Layer

- MIDI is a simple asynchronous serial interface, readily available on microprocessors of the early 1980's
- MIDI uses a baud rate of 31.25K baud (your old 56K baud modem was faster!)
- MIDI uses a 10 bit frame, with 1 start bit, 8 data bits, and 1 stop bit, no parity.

Source : midi.org

MIDI Protocol

- The MIDI protocol is made up of messages. A message consists of a series of bytes, from a single byte to many bytes.
  Most messages are from 1 to 3 bytes long.

- The first byte of every MIDI message is the ***Status*** byte, and it is special in that it is the only byte in a message to have Bit #7 set to a one, such as the Reset message

11111111 = oxFF = MIDI Reset Status Message

Source : midi.org

MIDI Status Messages

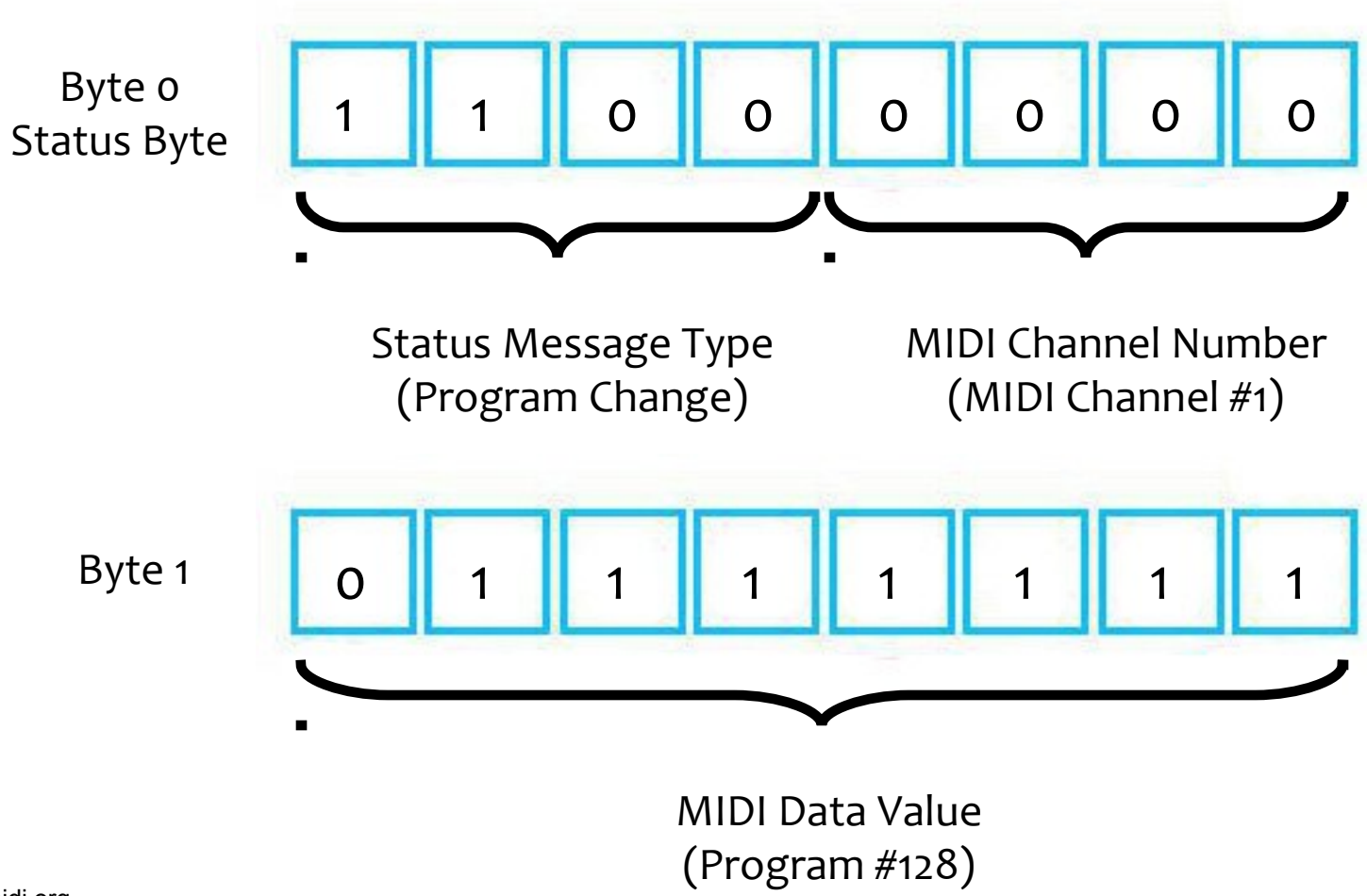
- Status Messages (bit 7 set) > 0x80 – 0xFF

Status	Explanation	Msg Size	Byte 1	Byte 2
0x8c	Note Off	2	pitch	velocity
0x9c	Note On	2	pitch	velocity
0xAc	Key Pressure	2	key	pressure
0xBc	Controller Change	2	controller	value
0xCc	Program Change	1	preset	
0xDc	Channel Pressure	1	pressure	
0xEc	Pitch Bend	2	bend LSB	bend MSB
0xF0	System Exclusive	*n*	vendor ID	anything
0xF2	Song Position	2	position LSB	position MSB
0xF3	Song Select	1	song number	
0xF5	*Unofficial Bus Select*	*1*	*bus number*	
0xF6	Tune Request	0		
0xF7	End of SysEx	0		
0xF8	Timing Tick	0		
0xFA	Start Song	0		
0xFB	Continue Song	0		
0xFC	Stop Song	0		
0xFE	Active Sensing	0		
0xFF	System Reset	0		

Source : midi.org

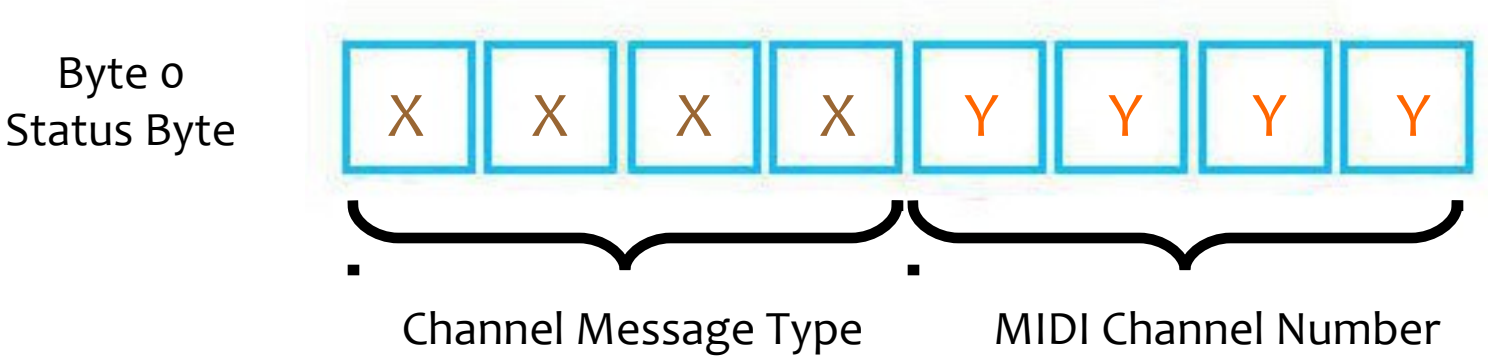
Decoding MIDI Messages

Command: Program Change to Program #128 on MIDI Channel #1



Source : midi.org

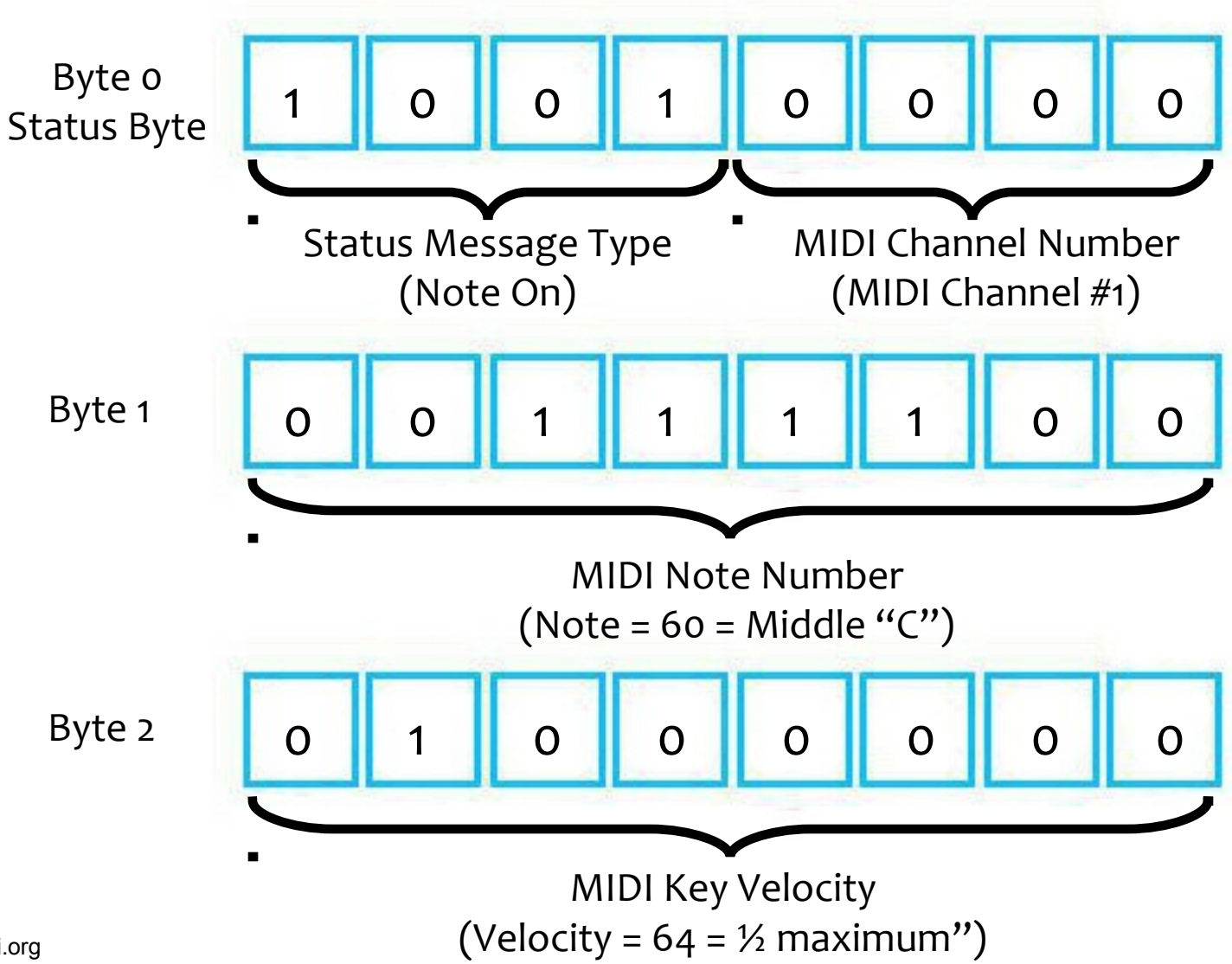
MIDI Channel Messages



- XXXX: 1000 = 0x8 = Note Off
  1001 = 0x9 = Note On
  1010 = 0xA = After Touch/Pressure
  1011 = 0xB = Controller Change
  1100 = 0xC = Program Change
  1101 = 0xD = Channel Pressure
  1110 = 0xE = Pitch Wheel
  1111 = 0xF = System Common/Real Time
- YYYY: MIDI Channel Number (1 – 16 channels)

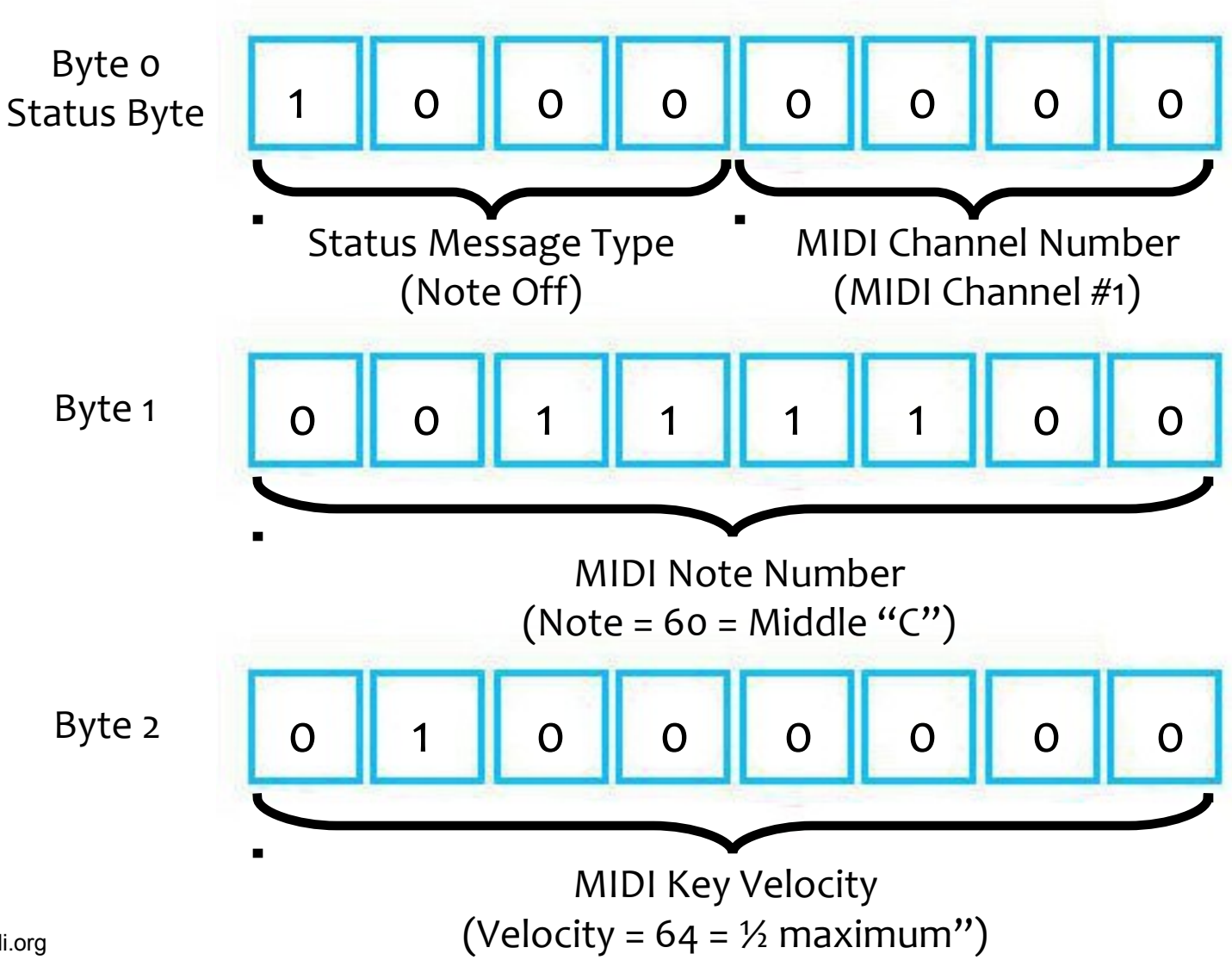
Source : midi.org

Note On Message



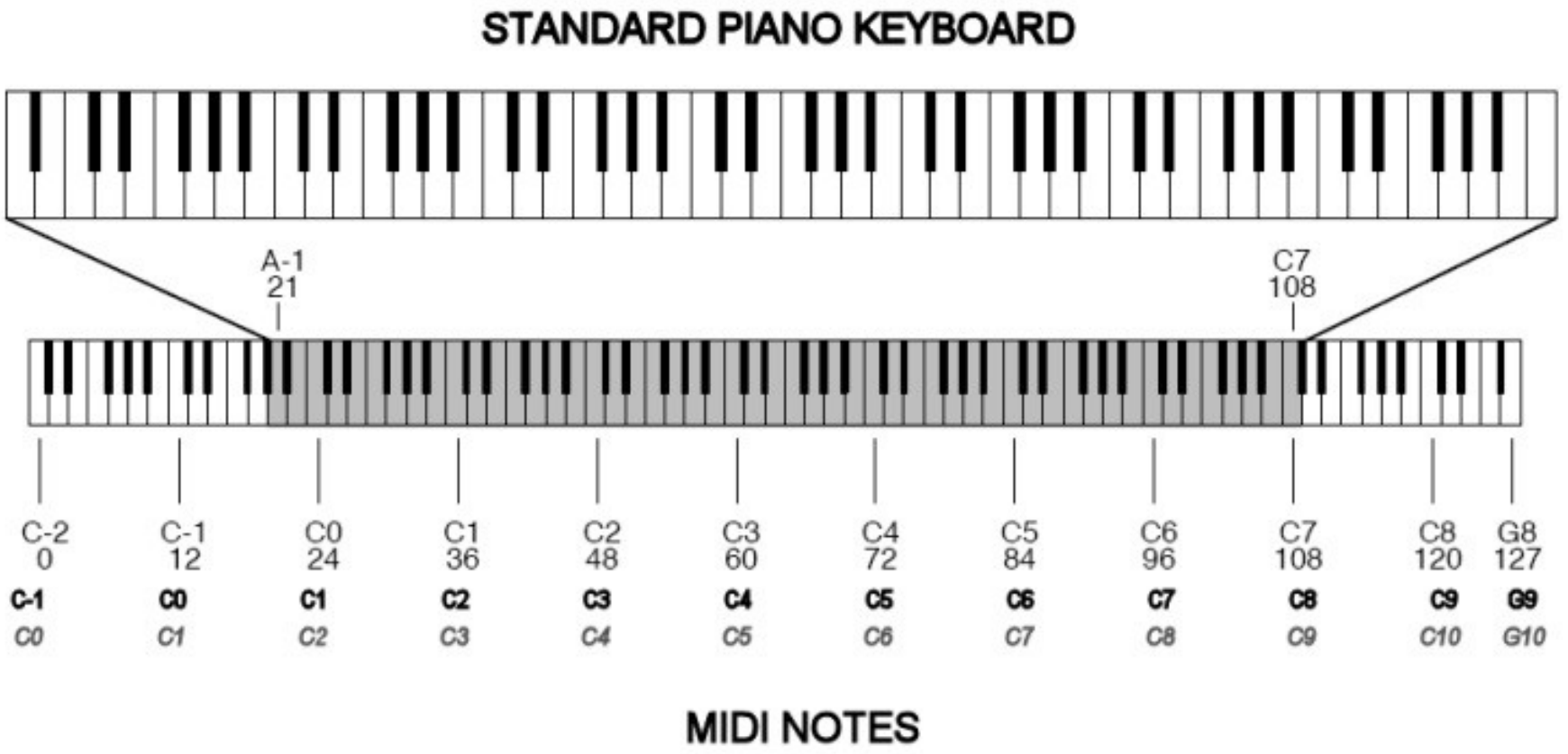
Source : midi.org

Note Off Message



Source : midi.org

MIDI Notes



- MIDI Note values outside the normal 88 key range

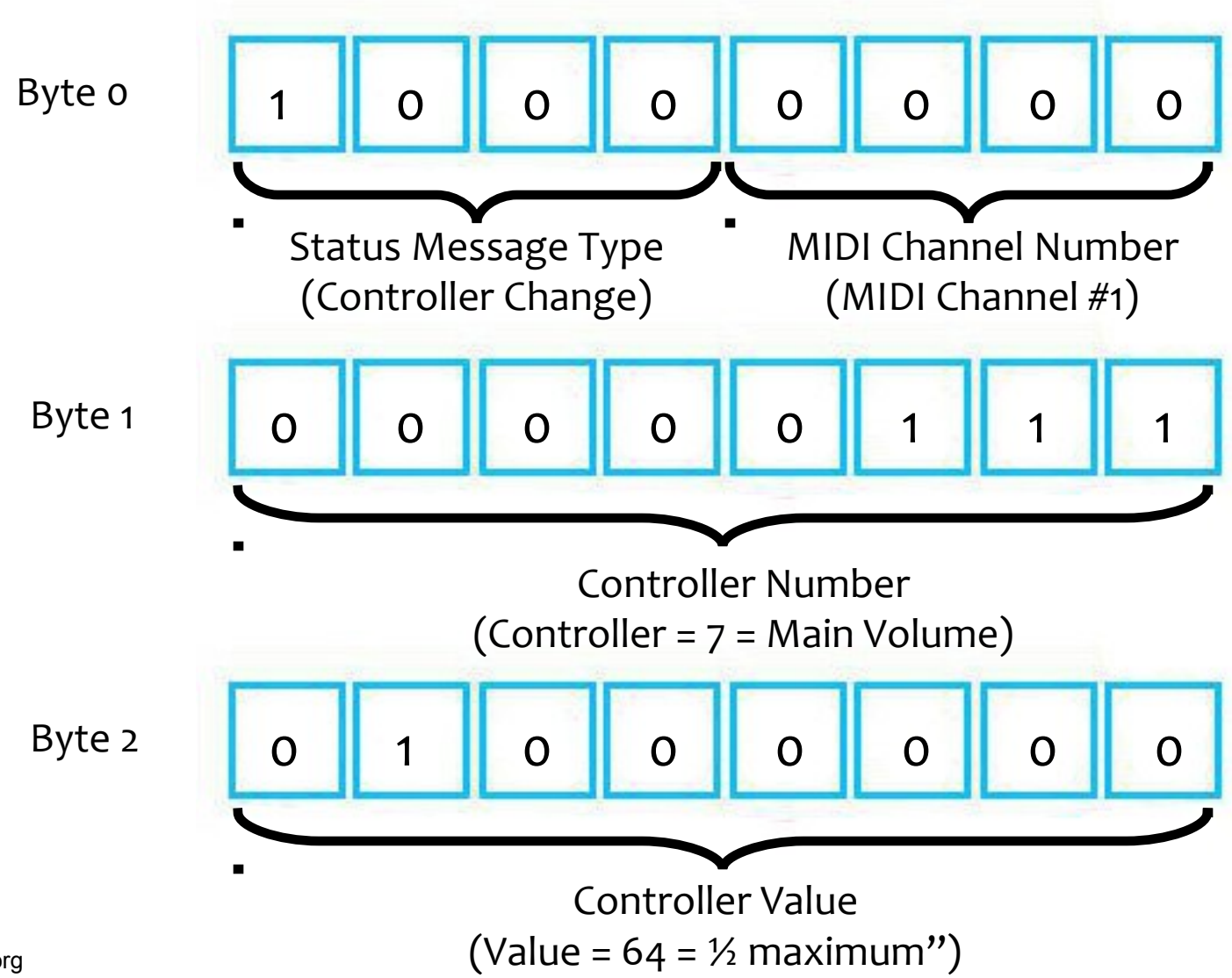
Source : midi.org

MIDI Controllers

- Pedals, Sliders, Knobs, etc. are referred to as “Controllers” or as “Continuous Controllers”
- The first 32 controllers have the ability to have two bytes of control information sent (14 bits)

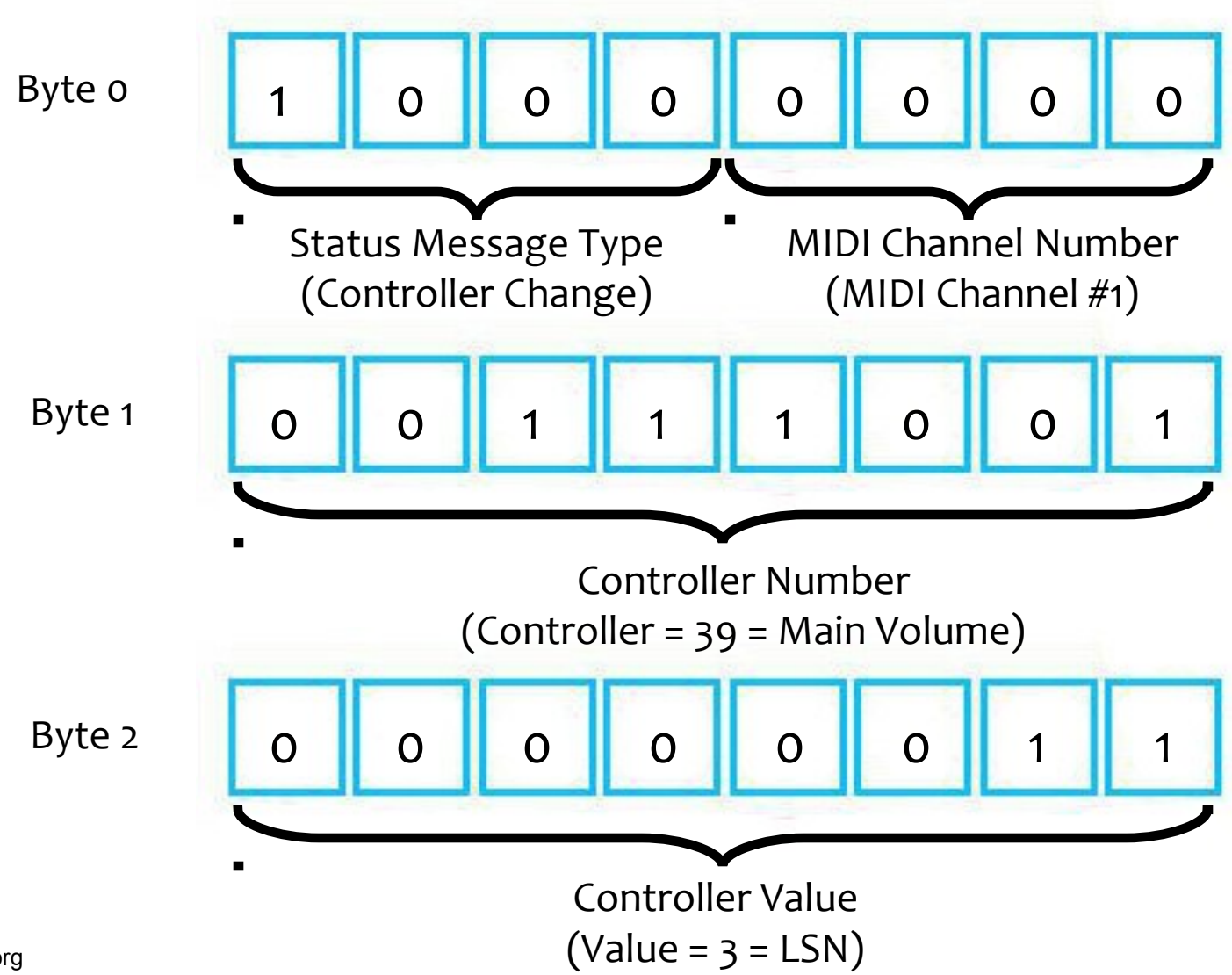
Controller Number	On/Off	Function
0		undefined
1		Modulation Wheel
2		Breath Controller
3		undefined
4		Foot Controller
5		Portamento Time
6		Data Entry Slider (uses Parameter Number)
7		Main Volume
8		Balance
9		undefined
10		Pan
11		Expression Controller
12-15		undefined
16-19		General Purpose Controllers 1-4
20-31		Undefined
32-63		LSB for controllers 0-31
64	✓	Damper Pedal
65	✓	Portamento
66	✓	Sostenuto
67	✓	Soft Pedal
68		undefined
69	✓	Hold 2
70-79		undefined
80-83		General Purpose Controllers 5-8
84-91		undefined
92		Tremolo Depth
93		Chorus Depth
94		Celeste (Detune) Depth
95		Phaser Depth
96		Data Increment
97		Data Decrement
98		Non-Registered Parameter Number LSB
99		Non-Registered Parameter Number MSB
100		Registered Parameter Number LSB
101		Registered Parameter Number MSB
102-121		undefined
122-127		Channel Mode Messages

Controller Message (Course)



Source : midi.org

Controller Message (Fine)



Source : midi.org

System Realtime Messages

- System Realtime messages are only a single byte long to reduce timing delay and overhead

- Realtime: 0xF8 = Timing Clock
  0xF9 = Reserved
  0xFA = Start Sequencer
  0xFB = Continue Sequencer
  0xFC = Stop Sequencer
  0xFD = Reserved
  0xFE = Active Sensing
  0xFF = System Reset
- Active Sensing handles runaway devices by silencing them if a new MIDI message is not received by the timeout time of 300 milliseconds

Source : midi.org

System Common Messages

- The System Exclusive Message can be any length in bytes and is terminated the “End of SYSEX” message.

- Realtime: 0xF0 = System Exclusive Message
  0xF1 = MIDI Time Code Quarter Frame
  0xF2 = Sequencer Song Position
  0xF3 = Sequencer Song Select
  0xF4 = Reserved
  0xF5 = Reserved
  0xF6 = Tune Request
  0xF7 = End of System Exclusive Message

Source : midi.org

System Exclusive Message

- The SYSEX message type allows manufacturers to create their own messages (such as bulk dumps, patch parameters, and other non-MIDI specified data)
- It uses both a Manufacturer's ID code and Model ID Code to exchange information specific to a particular instrument
- Two of the Manufacturer's IDs are reserved for extensions called Universal Exclusive Messages, which are not manufacturer-specific, and was a built-in way to expand the MIDI specification later.
- If a device recognizes the ID code as its own (or as a supported Universal message) it will listen to the rest of the message. Otherwise, the message will be ignored.
- Only Real-Time messages may be interleaved with a System Exclusive message

Source : midi.org

MIDI Protocol Examples

- MIDI Status is *stateful* , in that it only needs to be sent once (unless it changes)

- Note On, Middle C, Channel #1, ½ velocity
  0x90, 0x3C, 0x40
- Note, On, G above C, Channel #1, ½ velocity
  0x43, 0x40

- This is known as *running status*, and applies only to Channel messages

Source : midi.org

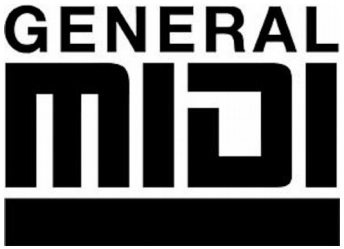
So, MIDI 1.0 solved everything?

- Hardly – most synthesizers had different internal formats and patch tables
- For example one synthesizer might have a piano sound on Program #1, while another might have a trumpet sound on Program #1
- Overuse of the SYSEX messages for custom purposes made it difficult for software design

Source : midi.org

Enter General MIDI…

- It took until 1991 for the first version of General MIDI to be published



- A basic list of requirements that synthesizers needed to meet to get the General MIDI stamp of approval.

  24 Voice Polyphony (16 melodic, 8 drum)
  Support 16 MIDI channels (Drums on #10)
  Support Note Velocity control
  Support 128 Standard Program Table

Source : midi.org

General MIDI

- Finally with General MIDI, you could issue a Program Change to Program #1 and expect to get an “Acoustic Grand Piano” sound on all of your General MIDI gear (or the closest sound to an “Acoustic Grand Piano” that particular gear could create…)

Source : midi.org

General MIDI Instruments

- A brief look at some of the supported voices

Piano

- 1 Acoustic Grand Piano
- 2 Bright Acoustic Piano
- 3 Electric Grand Piano
- 4 Honky-tonk Piano
- 5 Rhodes Piano
- 6 Chorused Piano
- 7 Harpsichord
- 8 Clavinet

Bass

- 33 Acoustic Fretless Bass
- 34 Electric Bass Fingered
- 35 Electric Bass Picked
- 36 Fretless Bass
- 37 Slap Bass 1
- 38 Slap Bass 2
- 39 Synth Bass 1
- 40 Synth Bass 2

Reed

- 65 Soprano Sax
- 66 Alto Sax
- 67 Tenor Sax
- 68 Baritone Sax
- 69 Oboe
- 70 English Horn
- 71 Bassoon
- 72 Clarinet

Synth Effect

- 97 Ice Rain
- 98 Sound Tracks
- 99 Crystal
- 100 Atmosphere
- 101 Brightness
- 102 Goblins
- 103 Echoes
- 104 Space

Chromatic Percussion

- 9 Celesta
- 10 Glockenspiel
- 11 Music Box
- 12 Vibraphone
- 13 Marimba
- 14 Xylophone
- 15 Tubular bells
- 16 Dulcimer

Strings and Orchestra

- 41 Violin
- 42 Viola
- 43 Cello
- 44 Contrabass
- 45 Tremolo Strings
- 46 Pizzicato Strings
- 47 Orchestral Harp
- 48 Timpani

Pipe

- 73 Piccolo
- 74 Flute
- 75 Recorder
- 76 Pan Flute
- 77 Bottle Blow
- 78 Shakuhachi
- 79 Whistle
- 80 Ocarina

Ethnic

- 105 Sitar
- 106 Banjo
- 107 Shamisen
- 108 Koto
- 109 Kalimba
- 110 Bagpipe
- 111 Fiddle
- 112 Shanai

Source : midi.org

SMF – Standard MIDI Files

- MIDI data can also be stored as files - The standard MIDI file (.smf, or .mid) is slightly different from the “live” MIDI data in that it contains *time-stamping* information

- There two main types of SMF, namely;
  Type-0 (all data in a single track)
  Type-1 (data is in multiple tracks)

  (Type-2 never caught on - and is obsolete)

Source : midi.org

MIDI File Structure

- Composed of “Chunks” of information

SMF =

<header_chunk>
+ <track_chunk>
[+ <track_chunk> ...]

Source : midi.org

SMF File Header

- header_chunk =

"MThd"	Literal Header
+ <header_length>	Header Length
+ <format>	Type-0/1
+ <n>	# of Track Chunks
+ <division>	Timing Units/Beat

Source : midi.org

SMF Track Chunk

- track_chunk =

"MTrk"	Literal Header
+ <length>	Bytes in Track Chunk
+ <track_event>	A Track Event
[+ <track_event> ...]	more track events…

Source : midi.org

SMF Track Event

- track_event =

<d_time>	Delta Time Stamp
+ <midi_event>	MIDI Message
| <meta_event>	SMF Meta Event
| <sysex_event>	SYSEX Event

- Meta Events are for non-MIDI data like track name, copyright, labels

Source : midi.org

USB MIDI Class

- MIDI over USB possible with MIDI USB Class USB has replaced older MIDI interfaces on computers for the most part, but the MIDI interface is still used on instruments, and low cost USB to MIDI adaptors are available

USB
UNIVERSAL SERIAL BUS

Source : midi.org

Is MIDI Obsolete now?

- No. The MIDI standard keeps evolving
- MIDI is now on several different transports such as USB, FireWire, Ethernet, others
- MIDI has IETF MIME mapping for internet
- MIDI has eXtensible Music Format (XMF)
- MIDI also has XML mappings now

Source : midi.org

MIDI – Thirty-Five Years and Counting

- It's been thirty-five (35) years since the MIDI specification was released, and it remains as relevant and useful today as when it was first released.
- MIDI continues to evolve and find new applications for controlling musical instruments as well as other devices.

Source : MMA/midi.org

MIDI – What's next?

- MIDI Polyphonic Expression (MPE)
  MPE is a method of using MIDI which enables multidimensional controllers to control multiple parameters of every note within compatible systems

- Normally MIDI Channel-wide messages (such as Pitch Bend) are applied to all notes being played on a single MIDI Channel. In MPE, each note is assigned its own MIDI Channel so that those messages can be applied to each note individually

Source : MMA/midi.org

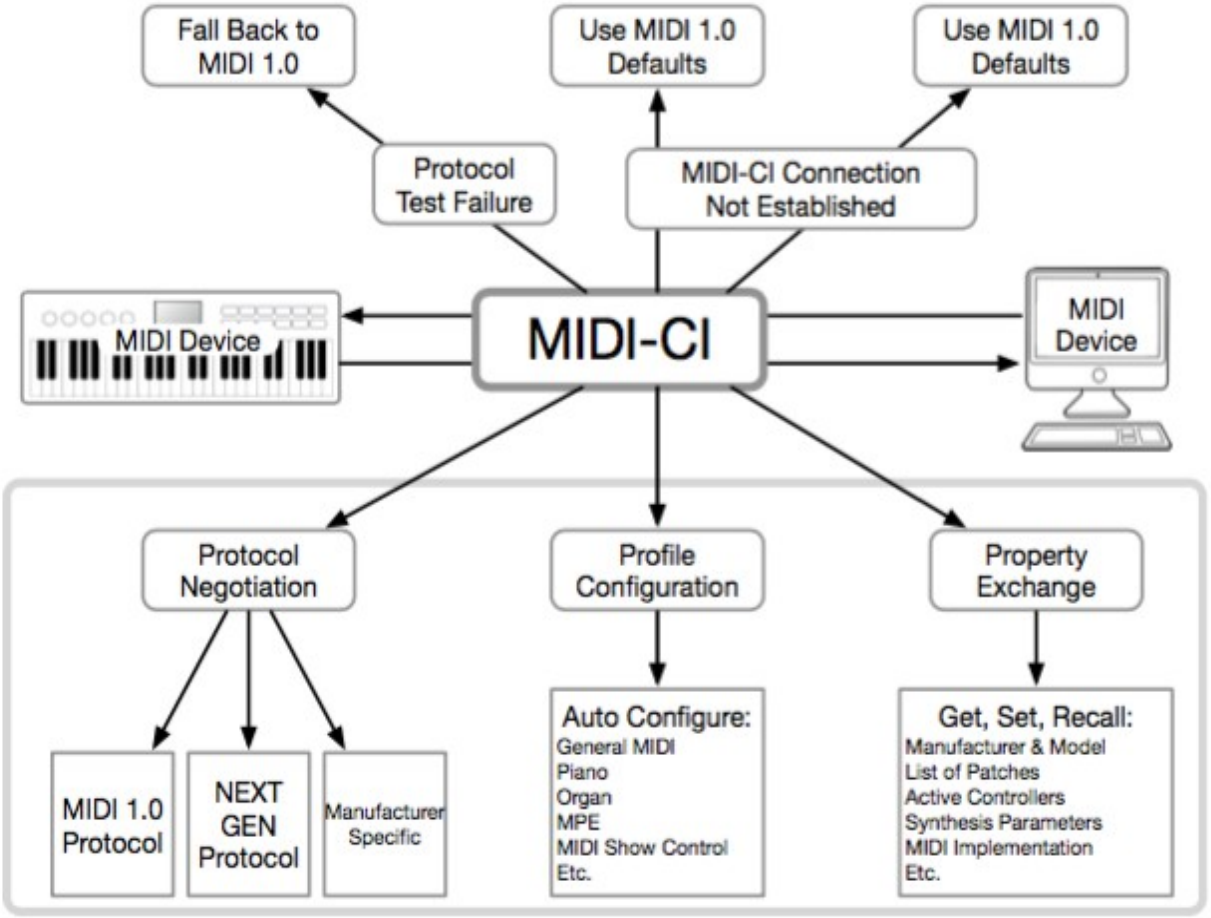
MIDI – What's next?

- MIDI-CI “Capability Inquiry”
  - Protocol Negotiation (fallback)
  - Profile Configuration (controller maps)
  - Property Exchange (ask for capabilities)

- Similar to what RDM-512 did for DMX-512, the electronic lighting standard

Source : MMA/midi.org

MIDI – What's next?



Source : MMA/midi.org

Sources and References :

- MIDI Manufacturer's Association : http://www.midi.org
  The MIDI logos and graphics are Copyrighted Material owned by the MMA, and used here with their kind permission.

- Wikipedia : en.wikipedia.org/wiki/MIDI/

- eMusician : http://www.emusician.com/news/0766/pitch-vs-frequency/146705

Thank You !

Questions?

Notes :