COSC 4P98 Lecture notes: **MIDI**
Feb 12, 2014
B. Ross

- MIDI: Musical Instrument Digital Interface
    - specification is available online at www.midi.org
- invented in 1983 by consortium of instrument manufacturers
- basically an asynchronous serial network protocol
- signals are musical events: note on/off messages ("piano roll")
- midi is to audio waves, as a book is to a movie
- there is no sound data in a MIDI file
    - this explains why midi files of songs are small and compact compared to MP3/WAVE files!
- Uses:
    - separate the input device (keyboard) from the sound generator
    - one keyboard can play many MIDI instruments
    - novel controllers: different interfaces
        - eg. keyboard, breath controller, microphone, jello,…
    - software on computers can generate MIDI to control instruments
    - keyboard playing can be recorded via MIDI software
    - device-independent software and compositional environments
    - can use MIDI to communicate with particular hardware (device dependent librarians, controllers)
    - can control other audio devices: effects boxes, mixers, lights,...
    - can control computer animation!
    - transfer music between devices and software
- Physical characteristics:
    - 31.25 baud (31,250 bits/sec = 0.03 Mbits/sec = 0.00003 Gb/sec)
        - compare: USB 1: 12 Mbits/ sec
        - USB 2: 480 Mbits/sec
        - Firewire: 400 Mbits/sec
        - USB 3.1: 10 Gb/sec
        - Thunderbold 2.20: 20 Gbits/sec
        - sampling rates: 44k samples/sec at 16 bits/sample = 704 kbits/sec
        - → MIDI rate is more than adequate for keyboard playing, but can't handle real-time sampling!
    - 5-pin DIN plug, 50 foot cable maximum (otherwise delay or latency, signal "skew")
    - However, USB is now used as physical medium, with MIDI software protocol is communication language
        - overcomes latency problems, plus USB is everywhere
    - plugs in traditional MIDI instruments:
        - IN = control instrument
        - OUT = record (or control other instruments)

        - THRU = pass IN to other instruments
    - Idea is to daisy-chain instruments together
        - but # instruments, and length of total connection, are important!
- Software characteristics
    - asynchronous serial protocol
    - 16 channels: 16 instruments, or 16 logical data streams in 1 instrument (or combination of both)
    - each channel is a stream of data, labeled to go on that channel number
    - if you need more than 16 channels (and most instruments do), you then need to use multiple midi ports (or modes)
    - Messages format:
        - 1 byte data words, with start and stop bits: 10 bits

    eg:

NOTE	CHAN	KEY #	Velocity
ON	1	60 (mid. C)	116

        - pitch is therefore a key #, mapped by instrument to keyboard key
        - pitch is NOT a frequency!
        - velocity: value between 0 and 255
        - but instrument can interpret velocity any way it wants… loudness? brightness? timbre? etc.

    - Other messages:
        - OFF 1 60 40
        - ALL NOTES OFF (turns off all “on” notes on the instrument)
        - aftertouch: finger pressure on key while note is being played
        - continuous controller change: pitch bend wheel (expression)
        - song: select position in a MIDI stream (resident in memory)
        - MIDI clock: pulse to sequence (multiple) instruments with each other, and computer
        - System exclusive (SYSEX): protocol for manufacturers to send device-specific messages
            - a “hack” or catch-all for miscellaneous communications
            - can be used to send sample data to/from samplers (but not in real-time)
    - Note that MIDI is used for real-time communication, as well as file storage
        - very similar, except that file is static, and hence timing information must be explicitly denoted (duration between events)
        - see “midi file specification” at www.midi.org
- MIDI application software
    - DAW: digital audio workstation
        - MIDI, plus audio sampling/playback
    - commercial sequencers: Cubase, Ableton, Cakewalk, Fruityloops,…

    - "sequencer": sequence the notes (music events)
    - many automatically transcribe into music notation as well
    - early programs were exclusively MIDI
    - but advent of affordable, high-quality digital audio cards meant that audio processing was introduced
    - most programs mix audio and MIDI
        - however, both are still separate data streams!
    - programming language interfaces exist: C, Lisp, Java,…
    - Specialized applications: pitch detection, algorithmic composition environments, others
    - Software has essentially replaced hardware in terms of functionality; but many musicians still prefer hardware interfaces, plus stand-alone instruments.
        - software is still more powerful, and way cheaper! Easier to update too.
- VST: Steinberg's Virtual Studio protocol
    - a programming interface that lets 3rd party developers extend commercial software visaviz "plug-ins"
    - Some plug-ins are as commercially successful and expensive as the host sequencers!
    - a whole industry exists around VST (and Digidesign's RTAS, which is similar)
    - more on VST soon!
- Future: MIDI 2.0 (aka HD-MIDI)
    - still in the works. It has been discussed for at least a decade.
    - a high definition HD-MIDI
        - Midi 1 has 127 levels... too coarse!
    - Rumoured support for
        - wireless
        - audio-video bridging
    - hopefully it will also include audio data information
    - But it probably doesn't need to address hardware issues (besides wireless and networking); USB2 and Firewire are already accepted standards.
    - It might be a software protocol only.
- A related technology: DMX512 (or DMX)
    - digital protocol for controlling stage lighting and FX
    - Since FX are often synchronized to music, there are MIDI-to-DMX converters.
    - Like midi, there is a HW and SW specification.

Bibliography:

http://www.askaudiomag.com/articles/midi-20-next-steps-and-midi-alternatives-for-musicians