MIDI 2.0: Promises and Challenges

Paul D. Lehrman
Tufts University
paul.lehrman@tufts.edu

Abstract

MIDI, the musical instrument digital interface, is a highly successful protocol for conveying and, through the use of Standard MIDI Files, representing musical performance information. However, it lacks the ability to convey notation information. The newly approved MIDI 2.0 protocol gives us a chance to rectify that by including notation information in the next version of the MIDI File Specification.

Introduction

Outside of standard Western notation itself, MIDI is the longest-serving and most ubiquitous method of representing musical performance. Its advantage over standard notation is its finer resolution in many dimensions. Its disadvantage is that it is not readable and interpretable in real time by human performers. The recently adopted MIDI 2.0 specification improves its resolution by orders of magnitude, and that it is still a work in progress means we have a potential opportunity to align it with other encoding technologies so that it can be used to represent music in human-readable form.

MIDI and standard MIDI files

The original MIDI 1.0 Specification, which was adopted in 1982 [1], was designed to enable electronic instruments from different manufacturers to communicate with each other digitally. Computer programmers were quick to realize that the data stream created by a MIDI instrument could be digitally recorded, and multiple data streams could be combined in a software program similarly to a multitrack tape recorder, allowing the creation of computer-controlled digital orchestras [2]. These programs, called sequencers, stored the MIDI stream in proprietary file formats, but by 1988, an addition to the MIDI specification created the Standard MIDI File (SMF), an open-source format for storing MIDI sequences [3]. Almost all makers of MIDI software, including makers of notation-based programs, adopted SMF as an alternative means of storage, thereby allowing users to bring sequences across multiple platforms, with minimal loss of performance information.

But SMFs do not carry much information specific to notation. While the MIDI Specification itself has expanded greatly since its initial adoption, it is still very much oriented to performance. Most musical gestures are recordable and reproducible in a SMF, but notation elements are limited to time signatures, tempos, key signatures, and lyrics. Beams, stems, ties, clefs, bowings, articulations, expression marks, repeats, and many other features of standard notation are not part of the SMF specification, and thus the conversion of a notation file into SMF, although a feature of many popular notation programs, results in a significant loss of information that cannot be recovered.

Advantages of MIDI

On the other hand, MIDI has several distinct advantages over standard notation. For one thing, it is exquisitely precise. The timing or length of a note in a Standard MIDI File can be resolved to as little as 1/3000th of a second, or 0.33 milliseconds, which is the equivalent of a triplet 1/2048th note at MM=120. The dynamic level of the onset of a note, called “velocity” in MIDI, can be specified to be any of 127 discrete values. Expressive information, including volume changes, portamento, vibrato depth and speed, and timbral changes, can also be resolved to 127 values, with the same timing resolution of 0.33 ms. (Pitch bend resolution is even higher, with 16,383 values.) Over 120 different expressive parameters can be controlled on each instrument in a MIDI orchestra using “continuous controllers” and other commands.

Unlike performances of printed music, a MIDI performance from a computer sequencer will always come out exactly the same if the performer or programmer wishes it to—but although a MIDI file cannot be “read” and interpreted by a musician the way a printed score can, it can be manipulated offline or in real time in terms of tempo, instrumental balance, orchestration, mode, or many other aspects of performance.

MIDI 2.0

From its beginning nearly 40 years ago until this year, the MIDI Specification has been labelled “1.0”. Although there have been many additions to the Specification, MIDI instruments introduced at the beginning of the MIDI era are still 100% compatible with instruments and programs being developed today—that is, although such early instruments will not recognize (and in fact will specifically ignore) commands that were added to the Specification subsequent to their introduction, their original capabilities remain completely viable.

Earlier this year, however, after several years of negotiation among the industry groups responsible for supervising the MIDI Specification in North America, Europe, and Asia, a new set of protocols known as MIDI 2.0 was adopted. While care has been taken to preserve compatibility with MIDI 1.0 devices, the 2.0 Specification greatly expands MIDI’s capabilities for a new generation of hardware and software [4].

Resolution

Primary among MIDI 2.0’s features are a greatly expanded feature set and greatly expanded resolution of musical parameters. When MIDI 1.0 was introduced, 8-bit data paths and computer clock speeds of 1 MegaHertz or less were standard. Today 32- and 64-bit data paths are the rule, and clock speeds are several orders of magnitude faster in the multi-GigaHertz range. MIDI 2.0 takes advantage of these greater bandwidths by expanding the resolution of commands from 8 bits (actually 7, since the first bit is used to determine whether a byte is a command or a data point), to 16. This allows, for example, the possible value of a note’s velocity byte to be expanded from 127 points to over 65,000.

Continuous controllers

Another important feature involves the implementation of continuous controllers. In MIDI 1.0, controllers are “per-channel,” e.g., if an instrument is using a single MIDI channel to produce the sound of a brass ensemble, introducing vibrato or pitch bend affects all of the notes on the channel identically. MIDI 2.0 has the ability to apply controller or pitchbend information to each note individually. Rather than 127 controllers per channel, there are now 512 available controllers *per note*. The resolution of all of these controllers is now 32 bits: that’s over 4 billion separate values. The controller set is expandable and customizable, with the potential to have over 32,000 discrete controllers.

Note messages

The note messages themselves in MIDI 2.0 carry a lot more information. A note can have an “attribute” assigned to it, which can communicate articulation, like a string sforzando or pizzicato; position of a hit on a drum or cymbal; or pitch information totally independently of the note number, making it easy to construct non-tempered or real-time variable scales. Since pitch information and note number are now separate parameters, multiple notes with the same note number but with different attributes can be transmitted and understood.

Channels

MIDI 1.0 limited the number of MIDI channels addressable over a single cable to 16. This was in large measure because at the original data rate of 3,125 bytes per second, attempting to control more instruments than that would likely have resulted in delays or dropped commands. MIDI 2.0 does not use the extremely slow—by today's standards—MIDI cable defined in the MIDI 1.0 Specification, but instead is "transport independent," meaning it will potentially be able to use any common connection protocol. The first transport for the new protocol will be USB, but it is expected in the near future that other mechanisms including Thunderbolt, WiFi, and Bluetooth will be adopted. Freed from this speed restriction, MIDI 2.0 offers 16 "groups", each of which has 16 channels, for a total of 256 channels per "cable." And unlike MIDI 1.0, which has separate "In" and "Out" connections on each device, MIDI 2.0 is bidirectional.

Hardware communication

The other improvements in MIDI 2.0 are primarily on the hardware side. It introduces new technologies called "Property Exchange" and "Profiles," designed to take advantage of this two-way communication. They are part of a new set of commands called MIDI Capability Inquiry, or MIDI-CI. Devices will include MIDI-CI "profiles" built into their operating systems. If two connected devices use MIDI-CI, they will be able to exchange important information about each other: their profiles will announce whether each device supports per-note pitchbend and controllers, how many channels or streams it responds to, how it handles controller commands, and what kind of instrument or device it is: a synthesizer, a silent keyboard, a sequencer, an arpeggiator, a rhythm computer, a mixer, an effects device, a lighting board, a video switcher, or even a drone.

For example, in the world of electronic organs, although many instruments have the standard nine drawbars, different manufacturers map different MIDI continuous controllers to the drawbars; but if two instruments subscribed to an agreed-upon "Drawbar Organ" profile, files would have identical drawbar settings when transferred from one instrument to the other.

Standard MIDI files 2.0

What remains to be written into the MIDI 2.0 Specification is how Standard MIDI Files will be updated to handle the new commands and resolutions. The Technical Standards Board of the MIDI Manufacturers Association—the volunteer industry group that oversees the Specification—is in the initial stages of developing a specification provisionally known as "SM2F."

In addition to implementing the new features of MIDI 2.0, this early stage of SM2F development offers an opportunity to integrate information not strictly related to performance, and that includes notation data. Given the large bandwidth and open structure of MIDI 2.0, there is plenty of room for the exchange of notation data in all of its forms in both real time and as part of a file. While it is much too early to even speculate whether SM2F will address notation issues, it is worth noting that one member of the group working on SM2F is Michael Good, the inventor of MusicXML, the expansive and expandable music notation file format that is the equivalent of SMF (1.0) in the area of music notation software [5]. Good represents the intersection of the MIDI community with the notation community, two bodies that previously have had little in common.

Conclusion

MIDI 2.0 is a major update to a highly successful technology that brings digital music-making up to date and opens up new means of expression and precision. The new Standard MIDI File 2.0 specification, which is to follow, represents an opportunity to include many musical features not available in the current Standard MIDI Files. Perhaps the ability to transfer both performance and notation information between applications and platforms could be among them.

Acknowledgements

Special thanks to Rick Cohen, chairman of the MIDI Manufacturers Association Technical Standards Board, and former chair of the MIDI 2.0 Protocol Working Group; and Michael Good, Vice-president of MusicXML Technology at MakeMusic, Inc.

Works cited

[1] The Complete MIDI 1.0 Detailed Specification. Version 96.1. MIDI Manufacturers Association, 1996.

[2] Lehrman, Paul, and Tim Tully. MIDI For the Professional. Second edition. New York: Music Sales Corporation, 1992.

[3] Standard MIDI Files, (Recommended Practice 001). MIDI Manufacturers Association, 1996.

[4] MIDI 2.0 Specification Overview. Version 1.0. Association of Musical Electronics Industry/MIDI Manufacturers Association, 2020.

[5] musicXML FAQ, https://www.musicxml.com/tutorial/faq/ (retrieved 12 May 2020).