The MIDI Protocol: Digital Music for a Digital Age

By Gregory Bento, ECE '16

The introduction of the Musical Instrument Digital Interface [MIDI] protocol in the 1980's was a huge advancement in the electronic music field and is still used today. The MIDI protocol standard has fairly simple hardware and software requirements to be used and is now a universal standard in industry. This protocol was used in the Rust Team's project to produce the sounds created by the system.

Introduction

For over 30 years, Musical Instrument Digital Interface [MIDI] has been the standard protocol for almost all electronic musical instruments. MIDI specifies both the hardware and the data format used between multiple devices that allows for communication. When MIDI was first designed, it was intended to just be used for synthesizers. However, because of its low cost and simplistic design, it can now been seen in almost all electronic music devices and even in other applications such as theme parks and the fountains outside of Vegas casinos (Yelton, 2015). Before MIDI was invented, many electronic musicians struggled to find a simplistic way to enter the mainstream music industry because there was no standard and it was hard to connect devices together since they were not directly compatible. However MIDI now serves as a bridge between computers and musical instruments. Although there are limitations in the MIDI protocol, its usefulness as a standard far outweigh any limitations that are present.

How MIDI Came to Be

The origins of MIDI can be traced back to the 1981 Audio Engineering Society [AES]. The issues that was brought up, is that different components of electronic musical systems were not compatible because they were made by different manufacturers. At the convention, Dave Smith, president of Sequential Circuits, proposed the idea of making a standard for electronic music systems so that all these components would be able to work together. A few months later at the at the 1982 National Association of Music Merchants [NAMM] conference, Smith was able to meet with other synthesizer makers and proposed the basic elements of a system that would eventually become MIDI (Yelton, 2015). The next year, MIDI Specification 1.0 was published by Japanese developers and the Roland Corporation released the MPU-401 [MIDI Processing Unit 401] which allowed for computers to now access and process MIDI data (Doan, 2012).

How MIDI Works

MIDI specifications are broken down into two main categories of specifications: hardware and software. When the MIDI specification was specified, the hardware for MIDI was also standardized. All MIDI devices use MIDI cables which consist of 5 pins arranged in a 180 degree loop (Figure 1). Of the five

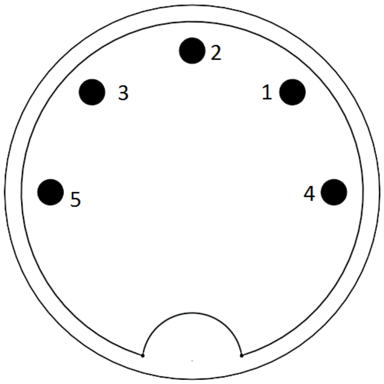

Figure 1. The layout of a MIDI cable. Pins 1 and 3 run the information across the cable while pin 2 is ground. Pins 4 and 5 are left open and not used.

pins, only three are used. While MIDI 1.0 was being written, two of the lines were to be used for XLR connections but the use was dropped (Loy, 1985). Most MIDI devices have three ports for the MIDI cables, MIDI IN, MIDI OUT, and MIDI THROU. The MIDI THROU port serves as a buffer and passes the same information coming into the MIDI IN port. All midi devices will have a microprocessor on the inside of the device. According to the MIDI specification, the microprocessor exports data at rate of 31250 bits per second. The only main restriction with the MIDI hardware is that the cables are mostly unidirectional, therefor the data can only move in one direction between two connected devices if only connected by one cable (Loy, 1985).

All this hardware is used to transmit a very specific defined software and data that is eventually used to make music. Over a MIDI channel, no actual sound information is transmitted, only information containing commands for the system to process. These commands include, note on, note off, change pitch and other similar commands. These commands are represented in 8 bits, where the first four bits of the byte are used to determine the command of the message. The second four bits are used to determine the channel on which the command will take place. Because there are only four bits for the channel, a MIDI system can only handle up to 16 different channels at any given time. (Doan, 2002). These commands may also be accompanied by one or two 7-bit data bytes. These data bytes often determine values for the different commands such as velocity of a note and which note is being played (McComb, 2013). (Figure 2) To get music and sounds from this information, the system will pull a sample from a sound bank based on the command received and play it out through a speaker.

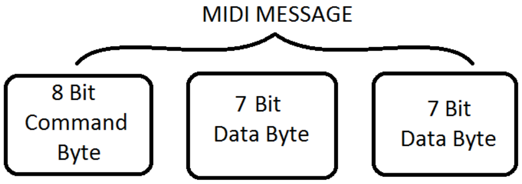

Figure 2. The breakdown of a MIDI message into its different bytes.

MIDI's Applications and Flaws

The main application for MIDI is the use for MIDI instruments such as a standard electronic key board. Most other electronic instruments (besides electronic string instruments), use MIDI to process the sounds they produce. Other applications include software based real time performances. Because MIDI data can be generated in software without the need of any hardware, it can be used to create music without actually having an instrument (Yuan, 2010). However with all these applications, there are still many flaws present in the MIDI protocol. This same protocol has been being used since the 1980s and because of that, the system is not as fast as some uses would require. There exists latency problems in some of the applications.

Another issue with MIDI is that it can only perform up to 16 notes at a single time, causing some musicians to need multiple MIDI devices in order to play certain musical pieces. Musicians also complain that, because MIDI is sample based, the quality of the sounds is not as good as other electronic music formats (Kim, 1999). A new protocol MIDI 2.0 or MIDI HD was proposed back in 2006 to not only update the system but to have a higher quality sound produced. However the new system has not been ratified by the industry yet and therefore is still not available (Yelton, 2015). Even with the old system starting to be out of date, it still provides a very reliable option for electronic musicians.

References

Doan, T. (2002). Understanding MIDI. The key to creating and conducting the music to your own MTV video. *IEEE Potentials,* 10-11. doi:10.1109/45.283853

K.N., Kim. (1999). Conversion from CD-DA format to MIDI format maintaining a sound quality. *Science and Technology, 1999. KORUS '99. Proceedings. The Third Russian-Korean International Symposium, Volume 1*, 300-303. doi:10.1109/KORUS.1999.875930

Loy, G. (1985). *Computer Music Journal*, v 9, n 4, 8-26, doi:10.2307/3679619

McComb, G. (2013). The Amazing Tunebot. In *Arduino Robot Bonanza* (pp. 225-264). New York: McGraw-Hill. Retrieved from Access Engineering database.

Yelton, G. (2015). The Story of MIDI. *Electronic Musician, 31*(11), 36-41.

Yuan, S. (2010). MIDI-Based Software for Real-Time Network Performances. *Cryptography and Network Security, Data Mining and Knowledge Discovery, E-Commerce & Its Applications and Embedded Systems (CDEE), 2010 First ACIS International Symposium, 1*, 226-230. doi:10.1109/CDEE.2010.51