A Focal Press Book

Modern MIDI

2nd edition

Sequencing and Performing Using Traditional and Mobile Tools

Sam McGuire

ROUTLEDGE

Modern MIDI

Modern MIDI equips you with everything you need to use MIDI in your music productions. With a particular focus on practical application and step-by-step explanations, this book does far more than tell you about how MIDI works. Simple explanations and real-world scenarios encourage you to test out the techniques for yourself.

The second edition of *Modern MIDI* builds on the first, making sequencing approachable for absolute beginners, reinforcing the fundamentals of MIDI and expanding the coverage into new mobile technology and vintage gear.

Modern MIDI is essential reading for beginners from all backgrounds, including students and professionals, as well as experienced users looking to develop their existing setups.

Supplementary videos are available on the Routledge website.

Sam McGuire is an active audio engineer, composer, and faculty member of the University of Colorado, Denver. He is the author of multiple audio technology books and numerous software instructional videos for VTC.com. Sam's research into 360° audio and impulse response technology has taken him to Europe and Asia; recording balloon pops in castles, cathedrals, and concert halls.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

MODERN MIDI

Sequencing and Performing Using Traditional and Mobile Tools

2nd edition

Sam McGuire

Routledge
Taylor & Francis Group
NEW YORK AND LONDON

Second edition published 2020
by Routledge
52 Vanderbilt Avenue, New York, NY 10017

and by Routledge
2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

Routledge is an imprint of the Taylor & Francis Group, an informa business

© 2020 Taylor & Francis

The right of Sam McGuire to be identified as author of this work has been asserted by him
in accordance with sections 77 and 78 of the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this book may be reprinted or reproduced or utilized
in any form or by any electronic, mechanical, or other means, now known or
hereafter invented, including photocopying and recording, or in any information
storage or retrieval system, without permission in writing from the publishers.

Trademark notice: Product or corporate names may be trademarks or registered trademarks,
and are used only for identification and explanation without intent to infringe.

First edition published by Focal Press 2014

Library of Congress Cataloging-in-Publication Data
Names: McGuire, Sam, author.
Title: Modern MIDI : sequencing and performing using
traditional and mobile tools / Sam McGuire.
Description: Second edition. |
New York, NY : Routledge, 2019. | Includes index.
Identifiers: LCCN 2019016716 (print) | LCCN 2019017301 (ebook) |
ISBN 9781351263849 (master ebk) | ISBN 9781351263832 (adobe pdf) |
ISBN 9781351263818 (kindle mobi) | ISBN 9781351263825 (epub3) |
ISBN 9781138578746 (hbk : alk. paper) | ISBN 9781138578777 (pbk : alk. paper)
Subjects: LCSH: MIDI (Standard) | Sequencer (Musical instrument)–Instruction
and study. | Software sequencers.
Classification: LCC MT723 (ebook) |
LCC MT723 .M35 2019 (print) | DDC 784.190285/46–dc23
LC record available at https://lccn.loc.gov/2019016716

ISBN: 978-1-138-57874-6 (hbk)
ISBN: 978-1-138-57877-7 (pbk)
ISBN: 978-1-351-26384-9 (ebk)

Typeset in Palatino
by Newgen Publishing UK

eResource link: https://www.routledge.com/9781138578777

To my family, for their support during long hours of using and writing about MIDI.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

CONTENTS

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

INTRODUCTION TO MIDI 1

Chapter co-authored by Adam Olson, Shenandoah Conservatory

MIDI (Musical Instrument Digital Interface) has been around for over 30 years and shows no signs of slowing down or being replaced. MIDI is a control protocol that sends performance data between multiple musical instruments and sequencers. Instead of starting with the history of MIDI in this chapter, let's get right into what it is and how you are going to be using it. One significant advantage of this book on the subject of MIDI is that we are going to take a very practical approach to using this tool and focus a lot less on much of the underlying data.

Ten Things You Need to Know about MIDI:

1. MIDI is everywhere.
2. MIDI is incapable of making sound by itself.
3. It is possible to successfully use MIDI without knowing anything about it.
4. Famous people use MIDI.
5. MIDI isn't just for music.
6. MIDI's "magic" number is 128.
7. The key to using MIDI is being a skilled musician.
8. MIDI is a huge time saver/waster.
9. The differences between digital audio and MIDI are fewer and fewer.
10. MIDI has changed very little in the past 30 years.

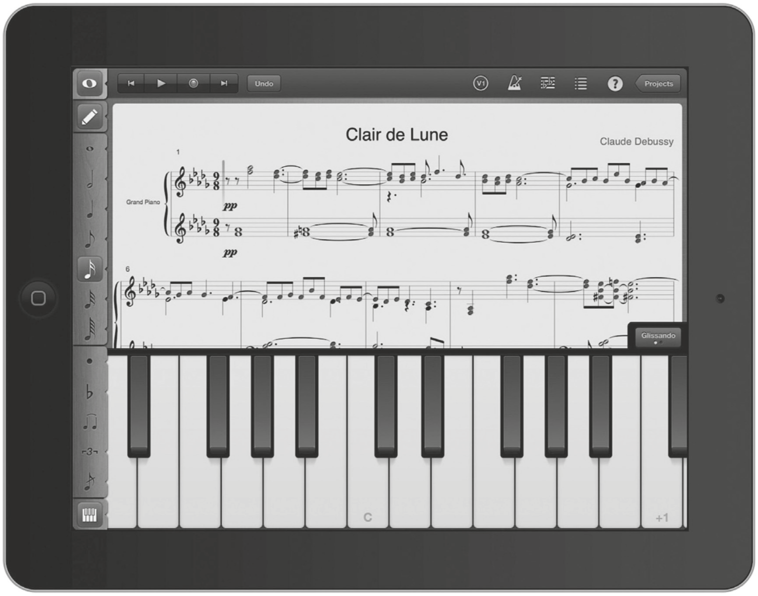


Figure 1.1 Notation in Symphony Pro

1. MIDI Is Everywhere

Chances are that you have held a MIDI device in the past 20 minutes, heard a MIDI performance in the last 12 hours, and possibly even used MIDI without knowing it. From cell phones to computers and from television to the movie theater to the stage, MIDI is used in nearly every situation. It is so pervasive because it is cheap, reliable, and universal. The most exciting chapter of this book covers the mobile uses of MIDI and looks at what the future is bringing.

There has been a really healthy resurgence of television music being performed by live musicians. A few years ago only a handful of shows used an actual orchestra, with the Simpsons and Lost heading the pack. The majority of shows use a MIDI-based soundtrack and the viewing audience is none the wiser. The 30-year-old MIDI protocol can sound so good because of the programmer in charge and the sounds she controls, and not because of MIDI itself. Even when a feature film uses a

Figure 1.2 FL Studio for iOS

full orchestra, there is still the likelihood that MIDI was used to layer additional instruments and sounds into the mix.

If you turn on the radio (not taking into account the oldies or classical stations), there is an overwhelming chance that you'll hear the effects of MIDI multiple times each hour. It might be the on-air talent triggering a sound effect or the latest mega hit that was crafted by producers in the studio, but you can't escape MIDI. The next time you play a musical app on your iPhone or Galaxy you might be using some iteration of MIDI. If you go out to a club there is an excellent chance that the DJ will be using Ableton or FL Studio, which means he is using MIDI. It is literally all around us.

The sheer number of implementations of MIDI means that it is a format that is here to stay. The expense involved with a move away from MIDI is prohibitive to say the least, and it works well enough for nearly all situations. This is discussed further in the history chapter, and the ramifications of competing formats are explored when we look at the future.

2. MIDI Is Incapable of Making Sound by Itself

A common analogy for MIDI is the player piano, which thanks to HBO's *Westworld* has come back into the public eye. The paper roll is covered in holes that indicate where the piano should play notes. While a player piano often conjures thoughts of the Wild West and smoky saloons, there are some extremely elegant player pianos. Ampico created a concert grand piano that is controlled by a roll of paper. These pianos were in circulation in the early twentieth century and were used by the likes of Rachmaninoff and Debussy. You can still find piano rolls of actual performances of Rachmaninoff, which when played through a grand piano are just like having him in the room. The paper is just a roll of paper until you attach it to a beautiful piano, in the very same way that MIDI is only as good as the sounds you attach it to. If you use the default sounds in FL Studio then it will sound like FL

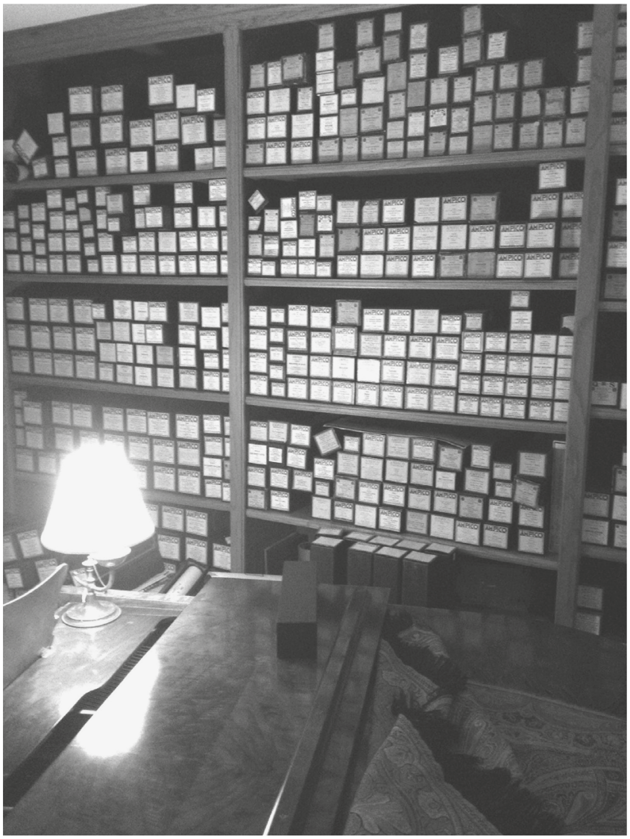
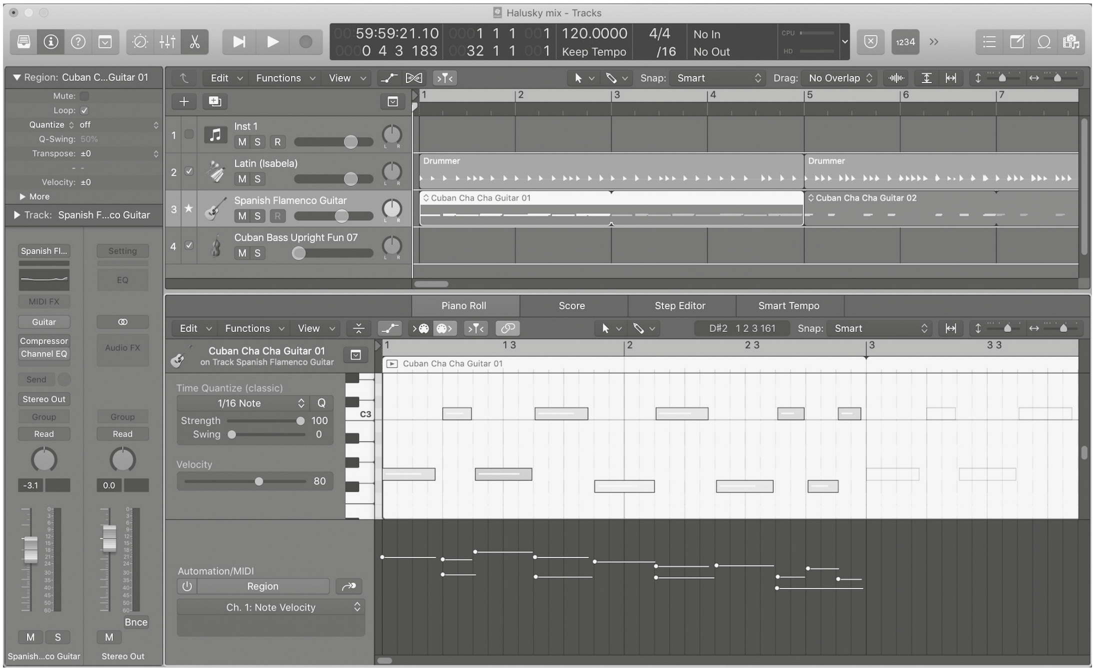


Figure 1.3 Ampico Piano and Rolls



Halusky mix - Tracks

59:59:21.10 1 1 1 1 120.0000 4/4 No In
0 4 3 183 32 1 1 1 Keep Tempo /16 No Out

Region: Cuban C...Guitar 01

Mute:
Loop: ✓
Quantize: off
Q-Swing: 50%
Transpose: ±0
Velocity: ±0

Snap: Smart
Drag: No Overlap

	Track
1	Inst 1
2	Latin (Isabela)
3	Spanish Flamenco Guitar
4	Cuban Bass Upright Fun 07

Cuban Cha Cha Guitar 01
on Track Spanish Flamenco Guitar

Time Quantize (classic)
1/16 Note
Strength 100
Swing 0
Velocity 80

Automation/MIDI
Region
Ch. 1: Note Velocity

D#2 1 2 3 161 Snap: Smart

Figure 1.4 Editing MIDI in Logic Pro X

Studio. If you attach it to a MPC, then you have the access to the same sounds as hundreds of hit records.

MIDI data include a number of pieces of information. When you press a key on a MIDI keyboard, the velocity is tracked along with the note being pressed and any continuing pressure caused by your finger on the key. Once you release the key there is a message transmitted that forces the instrument to stop playing. All of this information is transmitted in the MIDI stream and controls the destination instrument. Not only can MIDI not make a sound by itself, but the quality of any sound triggered by MIDI is completely dependent on the quality of the instrument. In a related discussion, a poorly programmed MIDI track cannot possibly make an excellent virtual instrument sound good; both pieces need to be excellent in their level of quality.

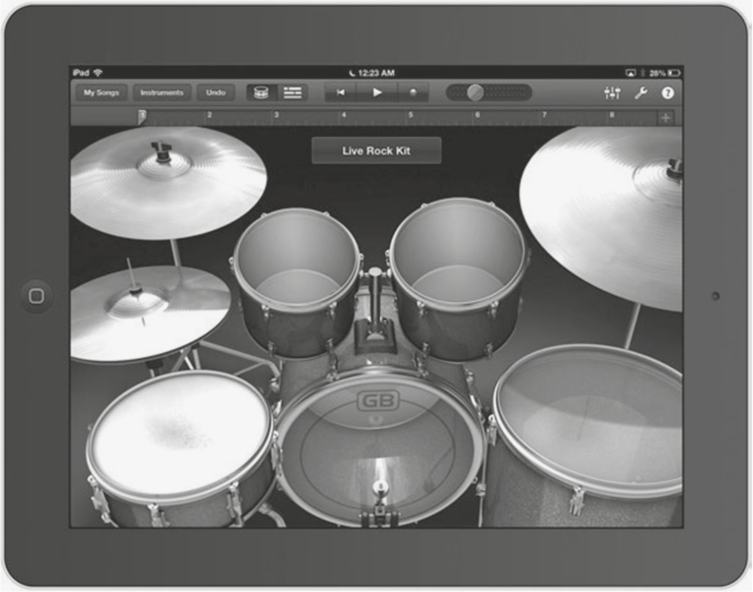
3. It Is Possible to Successfully Use MIDI without Knowing Anything about It

Let's look at a user group who needs to know very little, if anything at all, about MIDI in relation to daily usage. For a number of years, GarageBand was released bundled with every new Mac computer. Instead of using much of the traditional recording studio terminology, GarageBand simplified things and barely even used the word MIDI. A musician could hook up her musical keyboard and things would just work. In many cases there was zero setup and everything simply worked. By establishing an interface where digital audio and MIDI essentially looked and functioned in the same way, the end user needed to learn very little about the MIDI spec.

The simplified interface for sequencers is not the only indication that the way we use MIDI is continually changing. Sifting through lists of MIDI data used to be a common occurrence but is not nearly as common anymore. Several things have evolved to help us move in this direction, including more reliable sequencer technology, an exodus away from using external outboard gear, and DAWs that offer much more than basic sequencing.

The reliability of DAWs (Digital Audio Workstations) has decreased the need to be constantly editing and sifting through MIDI data in a numerical format. The more you can trust the visual representations of your data, the less troubleshooting you will have to do. The more frequent use of virtual instruments and the decrease of external sound modules is an example of a fundamental signal flow change that has affected how MIDI is used. Inside Logic Pro you can send three MIDI notes simultaneously to a synthesizer, and all three notes will arrive simultaneously. When sending three notes to a hardware module, the notes are sent as a series of notes and they are not received at the exact same time. Even when you set three notes to start at the same time, they are still received one after another.

Figure 1.5 Drums in GarageBand



In certain situations when using external hardware and with a large number of notes being played together, you might want to make sure certain notes are played earlier than others and so looking directly into the data stream is an important option. The last change is due to the diverse offerings of DAWs. Instead of a simple sequencer, there are now a number of additional expected features, such as a notation editors and piano roll views, which have advanced options. Users have grown accustom to using these alternate view options instead of looking at the text-based views. In 13 years of teaching, I have seen students move away from the actual MIDI data as they rely more heavily on the visual representations. Neither way is right or wrong.

The process for recording and editing MIDI is still the same as it ever was, and you can certainly manipulate the data in a lot of different ways. The point here is that DAWs are now designed to make the process easier and more transparent. As long as your sequencer of choice still has the advanced functionality, simplifying the workflow is probably never going to be a bad thing.

4. Famous People Use MIDI

MIDI is not just a format that is used by hobbyists or by basement musicians. MIDI is not some consumer gimmick that has very few professional applications. MIDI is used in every recording studio in the world and by some of the most influential musicians over the years. Later in this book we feature a number of musicians and composers who use MIDI and discuss their individual setups. The prominent theme is that MIDI is accessible to everyone who wants to use it.

Anyone who is willing to work hard and be persistent can program beats and create records that sound amazing. The tools are accessible and quite affordable, but you'll still need a good idea and a catchy hook. In the end, it doesn't really matter who uses MIDI except that it is nice to know that MIDI is a tool that has enabled a few generations of musicians to create the music they love.

5. MIDI Isn't Just for Music

Our primary focus is using MIDI in the music production process but it is capable of so much more. MIDI plays many different roles: instrument control, control surface communication, time code sync, and lighting control. Each of these uses the basic MIDI format but with varying individual characteristics. The strength of MIDI is that it has a diversity of abilities and is reliable in each situation. The weakness is that MIDI is not the best at any of them.

When considering the nonmusical options, MIDI is the older and slower protocol. Control surfaces with newer control implementations, such as EuCon from AVID, can control faders with over 8x resolution and 250x the speed. This type of control is used to control various parameters in your DAW.

You can use a physical fader to control the virtual mixer in your software. You can set panning and control effects. You can also control the project transport and various other tasks through key commands and menu options. However, MIDI is still a viable option and one that is working for thousands of engineers. If you have to make a choice between MIDI-based control and a shinier new format, I would go with the one you can reasonably afford. The new formats are often attached to more costly equipment and that is okay too.

When coordinating sync between gear and/or software you have multiple options. Traditional choices include SMPTE time code and MIDI time code. Newer options include protocol such as Propellerhead's Rewire or Steinberg's VST connection. Timecode is a transmitted positional reference that two or more pieces of equipment can use to sync. Again, MIDI has less resolution than SMPTE but in many cases is enough to provide accurate sync. The biggest issue with timecode is that a "master" has to send the code and then all of the "slaves" have to chase it, which creates a less efficient situation and potential inaccuracies. One of my favorite stories to tell about timecode happened when working on an independent film that started in Nuendo and then an additional team started working in Pro Tools; both sessions were at different sample rates and file formats. Combining the sessions is not as easy as pushing export and then import because Nuendo and Pro Tools don't like to share files. The solution was to sync them together using MIDI timecode, which allowed the sessions to remain separate but it created doubts about sync. It was far from the ideal solution but demonstrates the reliability of MIDI.

Other sync formats include proprietary solutions that work with similar hardware/software that are designed specifically to work using the same protocol. A solution like Rewire is a much tighter sync option that still requires a master and a slave but which doesn't rely on chasing code and instead

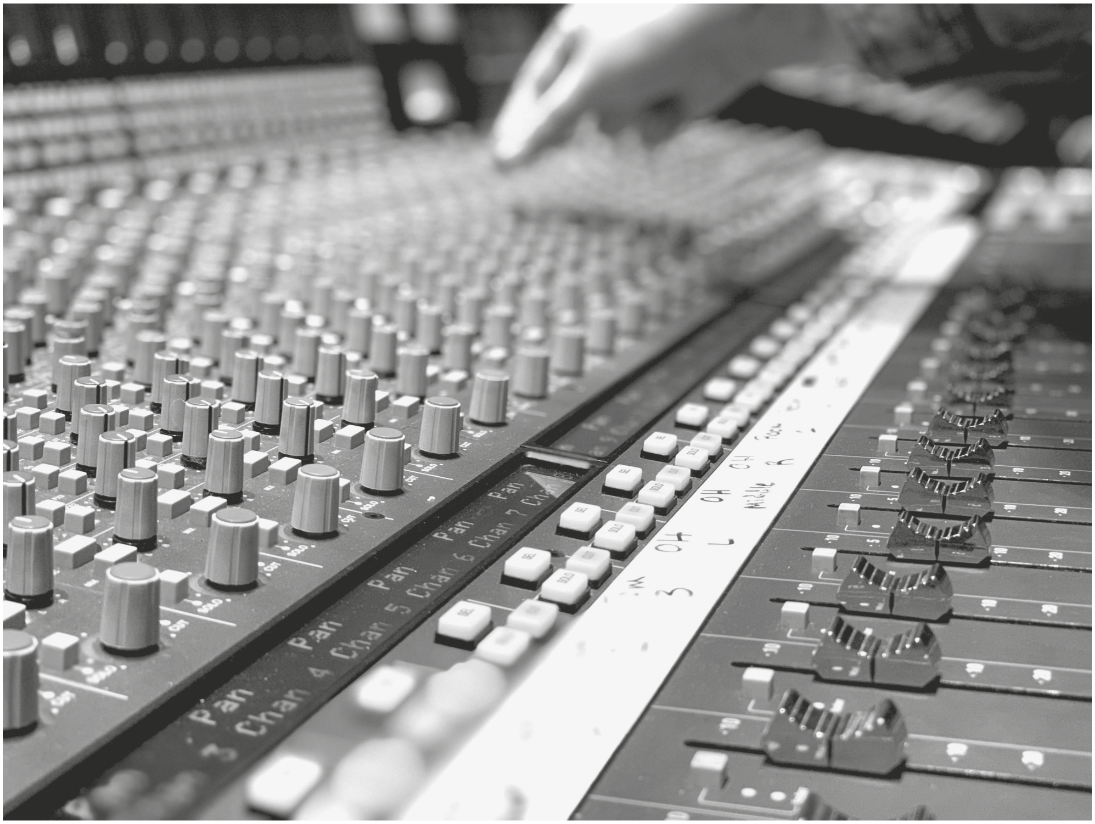


Figure 1.6 SSL AWS 924 with Control Surface Functionality

locks the transports together for very tight integration. Most sequencers simply do not sync together very easily.

Lighting control is one area where MIDI excels but it doesn't work alone. Lights and stage effects are typically controlled by a DMX capable unit that works well with MIDI. Many performers use MIDI output from their instruments to control various lights, or the control boxes are fed from a sequencer timed to the performance. If you have the right equipment it is quite easy to set up a fully interactive music and light show.

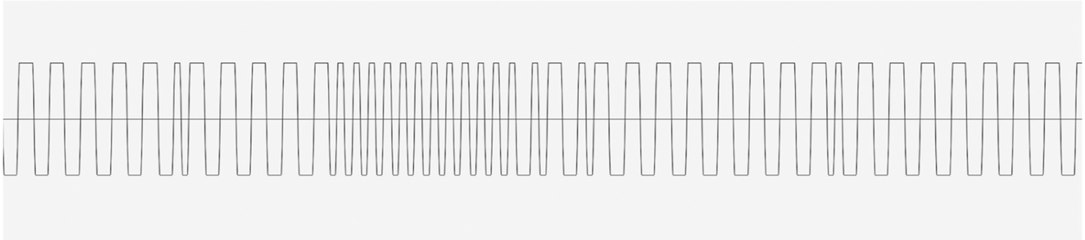


Figure 1.7 SMPTE Time Code

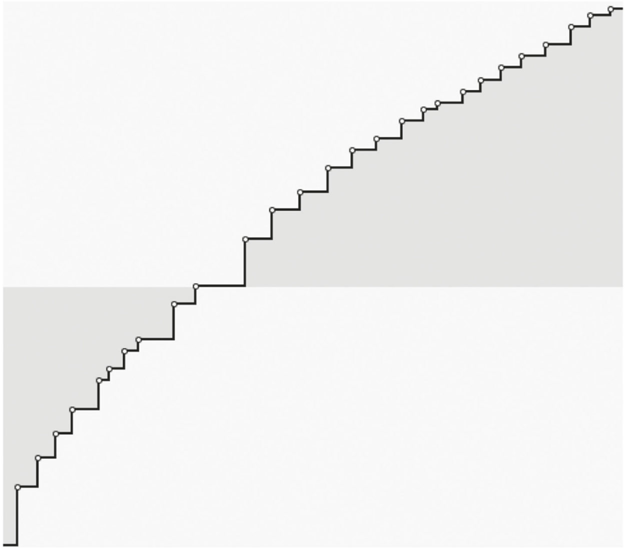
6. MIDI's "Magic" Number Is 128

MIDI was invented under a pretty severe set of limitations. In order to keep the bandwidth at acceptable limits and to create an interface that was both efficient and safe at lower costs, MIDI was designed to operate using modest operating parameters. All MIDI data are transmitted serially over a single cable, which means that each piece of datum is sent one piece at a time. The interesting thing about this is that when MIDI is used for relatively small performance needs then it can handle everything really well but when you push it to its limits it definitely starts to fall apart. More of the details of the MIDI specification are described later in this chapter and we look at the math behind the pervasive 128.

7. The Key to Using MIDI Is Being a Skilled Musician

MIDI works well enough that once it is set up you can simply perform. If that hadn't been possible then MIDI would never have survived. While it is possible to create performances without being able to play them, it is highly recommended that you study music and music performance to get the most out of MIDI. Later in the book there is an entire chapter on the basics of music theory, which is a really critical topic for MIDI users to understand. It is not assumed that you already have that knowledge because more and more people are using MIDI without having official musical

Figure 1.8 MIDI Data Zoomed to See Distinct Steps



training but are finding ways to make awesome music. Imagine what you can do once you have all of the tools in your toolkit and also have an understanding of the building blocks of music.

Being able to perform on a MIDI controller instead of entering notes with your mouse is something that can make your production sound more natural. Consider for a moment the data that are collected in a performance that would have to be entered manually without a performance: tempo variations, musical dynamics, additional performance controllers such as sustain pedals or modulation wheels, and the human element. It is possible to re-create all of these by hand but it is not efficient. Working on a score for a film, I worked hours on a single cue, trying to get it just right. I did it all by hand because I had a concept of how I wanted it to sound and thought I could put everything in just the right place. After three to four versions were rejected by the director and under a new deadline, I put away my mouse and just played. After a single take that I never needed to edit but only added additional instruments to, the director approved with great

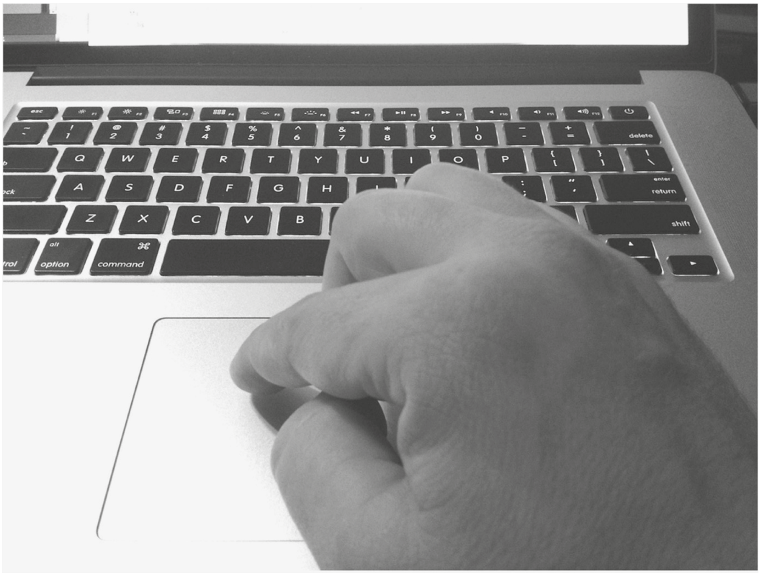


Figure 1.9 Data Entry with Track Pad

pleasure. The hours I had put into the cue had been eclipsed by a three-minute performance that was "perfect" almost immediately. My ability to play piano and understand theory created an environment where I could compose without MIDI getting in the way.

8. MIDI Is a Huge Time Saver/Waster

Continuing on the previous point, MIDI has the ability to be an enormous saver of resources or a huge waster of them. People get hired because they can create an entire orchestra in their home studio with amazing results, but using MIDI doesn't guarantee a speedier process and can often be a huge drain on the composer who does it all. Using a professional studio orchestra will likely cost more, but you will walk away with an amazing product that isn't in need of another couple days of tweaking. It is also still arguable that an orchestra controlled by MIDI can't sound as good as the real thing, but it is closer now than ever and virtual instruments can sound amazing.

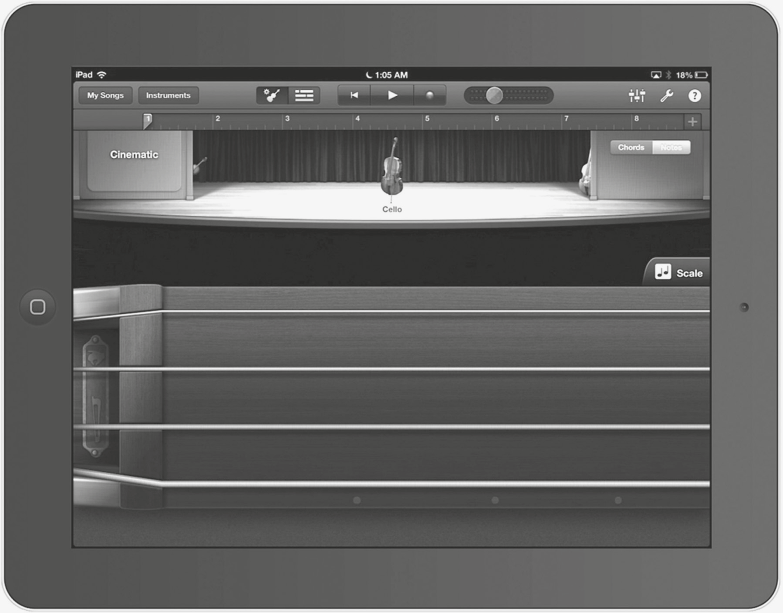


Figure 1.10 Orchestration Using GarageBand

For large-budget projects, the score is going to be produced via MIDI first as a mockup to gain approval before the money is spent on musicians. Some of these mockups could likely pass as final scores by most accounts but serve as a gateway for later recordings. In these cases, having a good MIDI mockup can function as a huge efficiency because if the music is recorded and then everyone finds out that they don't like it, then a lot of money will be wasted rerecording it. A mockup hopefully prevents such a situation.

MIDI can be a real time and energy sucker for the composer who is still learning the technology. A seasoned expert uses her or his knowledge to move quickly and avoid potholes in the process. If you want to see MIDI used in an amazing fashion, then ask a television composer if you can watch her or him compose. Between session templates and an intimate knowledge of their sound banks, a seasoned composer can plow through cues as if she or he is on a tight deadline, probably because they are.

9. The Differences between Digital Audio and MIDI Are Fewer and Fewer

One of the things that has preserved MIDI is the ability of DAWs to treat audio and MIDI in the same way. When you change your project tempo, then both the audio and MIDI change their speed. When using loops, it barely makes a difference whether it is audio or MIDI because they both react the same way. Before audio caught up to the flexibility of MIDI, and MIDI caught up to the sonic quality and reliability of audio, most production systems kept them relatively separate. Early sequencers didn't have any audio capabilities, and audio software didn't have any MIDI features. The unification of both technologies has finally reached congruence and DAWs integrate them equally. New technology will continue to make the differences between audio and MIDI fewer and fewer in the way we edit and manipulate our music.

10. MIDI Has Changed Very Little in the Past 30 Years

The more you learn about MIDI, the more you'll begin to understand that MIDI has a substantial amount of history and "baggage." You can use MIDI hardware that was designed at the very beginning of MIDI with a certain amount of success because a lot of the basics of MIDI haven't changed but be prepared to run into issues. MIDI is surprisingly robust yet implemented MIDI designs are riddled with issues and inconsistencies. MIDI is still a valuable tool that has reached maturity over the years but the more you use it, the more likely you are going to be aware of the issues. The lesson to be learned is that you need to develop patience when using MIDI and you'll need to use your MIDI tools enough to get to know them just as you would get to know any musical instrument. Not every piece of MIDI equipment is designed with the same level of quality and precision and so you should purchase equipment that has reviewed well and has a proven track record.

Updates in 2018

The MIDI Association announced some of the biggest changes to the MIDI protocol in 2018 which moves forward the ability of MIDI to work with future hardware/software options. The following sections explain the MIDI protocol as it was first created and as it still functions, which is critical to understanding its continuing development.

MIDI Capability Inquiry (MIDI-CI)

The key to backwards compatibility is having all parts of the system efficiently communicate with each other. The newest innovation with MIDI is the ability for different components to start by determining what they are capable of and to adjust their abilities accordingly. If you have a MIDI keyboard with all of the latest features and a workstation which is compatible, then they will both use the features. If one or more of the components has limited features, then all of the other elements will revert to basic MIDI when communicating with them. The power of this system is that in the future all elements will work together at the highest possible level, without having to compromise due to legacy equipment.

What to Expect

The three queries of MIDI-CI are:

1. Protocol Negotiation
2. Profile Configuration
3. Property Exchange

These are the stages of communicating between hardware/software in MIDI-CI and documentation from the MIDI Association is available at MIDI.org. The system is an impressive attempt at maintaining true backwards compatibility without ruining future development. However, as of the submission of this manuscript there are no widespread implementations of MIDI-CI. None of the major DAWs has been updated with support and very few development options have come from the Association.

Shortly after the announcement of MIDI-CI, the Association also released RP53 and RP54. RP53 describes the protocol for MIDI Polyphonic Expression (MPE) and RP54 describes the ability for MIDI to be transferred over TRS cables as opposed to 5-pin DIN cables. It is unclear which manufacturers will adopt these new specifications in the future, and until they do it is difficult to write about their true capabilities. The point here is to be aware that MIDI continues to slowly move into the future and to know of the changes even though in nine out of ten cases you will not be using the new features in the near future.

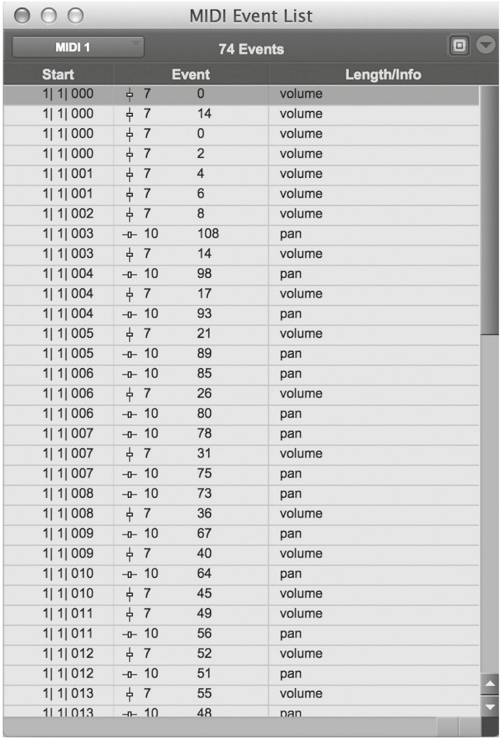
MIDI Data

Understanding the details of MIDI data is important because it helps us understand the limitations and challenges while offering insight in how to achieve the most from a limited system.

MIDI data are composed of binary bits, which are combined in groups of eight to form bytes. Bytes are the basic building blocks of MIDI Messages and are organized into a standard format to communicate between devices. MIDI Messages are made up of three essential pieces of data. Each portion is assigned a specific role such as triggering notes, setting velocity, and adjusting other controllers.

The first piece of datum (byte) is known as the Status Message, which communicates the controlling parameter to the receiving device. For example, to trigger a note you would use a Note On Status Message assigned to the corresponding MIDI Channel. The Status Message always begins with the number zero.

The second and third portions of the MIDI Message are Data Byte 1 and Data Byte 2. These are used to transmit a variety of



74 Events

Start	Event		Length/Info
1| 1| 000	ǂ 7	0	volume
1| 1| 000	ǂ 7	14	volume
1| 1| 000	ǂ 7	0	volume
1| 1| 000	ǂ 7	2	volume
1| 1| 001	ǂ 7	4	volume
1| 1| 001	ǂ 7	6	volume
1| 1| 002	ǂ 7	8	volume
1| 1| 003	-▫- 10	108	pan
1| 1| 003	ǂ 7	14	volume
1| 1| 004	-▫- 10	98	pan
1| 1| 004	ǂ 7	17	volume
1| 1| 004	-▫- 10	93	pan
1| 1| 005	ǂ 7	21	volume
1| 1| 005	-▫- 10	89	pan
1| 1| 006	-▫- 10	85	pan
1| 1| 006	ǂ 7	26	volume
1| 1| 006	-▫- 10	80	pan
1| 1| 007	-▫- 10	78	pan
1| 1| 007	ǂ 7	31	volume
1| 1| 007	-▫- 10	75	pan
1| 1| 008	-▫- 10	73	pan
1| 1| 008	ǂ 7	36	volume
1| 1| 009	-▫- 10	67	pan
1| 1| 009	ǂ 7	40	volume
1| 1| 010	-▫- 10	64	pan
1| 1| 010	ǂ 7	45	volume
1| 1| 011	ǂ 7	49	volume
1| 1| 011	-▫- 10	56	pan
1| 1| 012	ǂ 7	52	volume
1| 1| 012	-▫- 10	51	pan
1| 1| 013	ǂ 7	55	volume
1| 1| 013	-▫- 10	48	pan

Figure 1.11 Pro Tools Event List

information based on the individual Status Messages. When using a Note On message, Data Byte 1 is used to indicate the specific note being triggered and Data Byte 2 is used to indicate the velocity of the note. Different Status Messages use the Data Bytes in different ways, and some even combine the Data Bytes together.

Types of MIDI Messages

The Status Message at the beginning of each MIDI Message indicates which parameter is to be controlled. Here is a summary of the available Status Messages, and a chart listing each message and their corresponding Data Bytes is included in Appendix Table 12.2.

1. Note On and Off Messages

Note On and Off messages are used to trigger note playback. Not all controllers send Note Off messages and instead they substitute it with a Note On message with a Data Byte 2 value (velocity) of zero. This message is sent when the key is released and effectively turns the note off because the new note is set to silence.

Another consideration is that MIDI note number 60 is always middle C (C3 is the Yamaha common naming and C4 is the common Roland naming). While note number 60 is always middle C, most MIDI software allows you to change the visual preference for how MIDI On and Off message 60 is referenced. Changing this affects the way the rest of the notes are displayed.

2. Polyphonic Aftertouch and Channel Aftertouch

Aftertouch allows a controller to send additional control messages correlating to the pressure placed on the MIDI keyboard. These messages are created by varying the pressure of keys that are held down. Sound modules need to be configured to respond to these messages and can use these data to adjust the depth of vibrato, tremolo, or how bright a sound gets.

The difference between Polyphonic Aftertouch and Channel Aftertouch is that Polyphonic Aftertouch responds on a per-note basis, which means that it transmits a lot more data but it is dependent on how hard each individual note is pressed. Polyphonic Aftertouch is also more expensive to implement since individual sensors are needed for each note, and that makes this option cost prohibitive for many manufacturers.

On the other hand, Channel Aftertouch sends out MIDI messages based on the overall key pressure and only requires one sensor for all of the keys.

Many keyboard controllers don't offer Channel Aftertouch, and you'll want to consult the MIDI implementation chart in the back of the keyboard manual to see if your device sends and responds to these messages. Channel Aftertouch only uses Data Byte 1 for this information and leaves the second byte empty.

3. Program Change

This message is used to change what sounds are being triggered on the sound module. This is another example of a message that only uses Data Byte 1, which equals 128 different program changes that can be sent per channel. Had both data bytes been implemented for Program Change, then the total possible program changes would have been 16,384. Sound modules are not limited to only 128 sounds, and it is possible to combine two Control Change messages (CC 0 and CC 32) to give us what is referred to as banks of sounds for a total possibility of 2,097,152 different sounds or 16,384 banks with 128 sounds each. The math behind this is described more below.

There are foot switches on the market that are dedicated to sending out Program Change messages. These are very helpful in situations requiring a switch between loaded sounds, changing settings on an amp, adjusting lighting cues, and switching between preset levels on a digitally controlled mixing console.

4. Pitch Wheel Change

Most controllers include a pitch wheel or a joystick to control pitch changes and are used to adjust global pitch. Very few controllers have the ability to control pitch for individual notes and only a select group of software sequencers can do so either. One way of bypassing this is to map Polyphonic Aftertouch to pitch control, which allows pitch control of every note by adjusting the pressure applied to each key.

Data Byte 1 and 2 are combined to provide the resolution of 16,384 steps, but smooth pitch bend is reliant on both hardware and software design. Even in a limited format such as MIDI, high-quality bending can be extrapolated through intelligent design.

5. Control Change or Continuous Controllers (CC)

These are often referred to as performance controllers and are used to communicate with the knobs and sliders on many MIDI controllers. CC messages usually win the "cool" factor when it comes to MIDI because of the wide range of parameters that can be adjusted by them.

The controllers listed in the chart below are parameters that have been declared over the years by the MIDI Manufactures Association (MMA) for specific functions in controlling sound modules. In practice what this means is that if you have a control knob assigned to send CC 73, under the General MIDI 2 specification it adjusts the attack time of a sound module (the length of a fade from zero to full volume at the beginning of each note). Alternatively, this could also be mapped to adjust other parameters such as the threshold on a compressor, the send level to headphones, or the volume level of channels in your DAW. This is why CC messages are so popular: they give you tactile control over hardware and software parameters that otherwise might take a lot of scrolling around to find and adjust. See the appendix for a full list of CC messages.

Channel Mode Messages (Subset of Control Change Messages)

Channel Mode messages are used to change how a device responds to MIDI messages that are received. These messages are part of the Control Message spec, are limited to numbers 120–127, and are used to control global parameters on a device instead of channel specific parameters. These messages will only be received on the device's "basic channel" (one of the 16 channels) that is sometimes predetermined by the device or otherwise programmable. This means that if you simply send one of these messages, it may not change the way the module behaves if not sent on the specific device's "basic" channel number. You should consult the devices manual to find out what channel this is and if it can be changed. A description of each of these messages follows.

All Sound Off (CC 120)

When the All Sound Off message is received, all oscillators are turned off and their volume envelopes are set to zero.

Reset All Controllers (CC 121)

When this message is received by a device, all values are reset to the default values set by the manufacture. The Data Byte 2 value should always be set to zero unless other values are supported by the device.

Local Control (CC 122)

Local control is used for devices that have a controller that is also a sound generator. The local control parameter switches the destination of MIDI output from the keyboard between the "local" or built-in sounds to "nonlocal" or external MIDI destinations. When used in conjunction with a computer sequencer, it is usually the best practice to turn local control off so that the hardware doesn't trigger the internal sounds and also send out MIDI simultaneously. This keeps notes from being double triggered and causing a hollowness to the sound due to the close phase relationship to the double triggered notes.

While working this way, the user can edit in the computer sequencer and then press play and have the edited performance played on the sound module. This is very analogous to the way a word processor works versus using a typewriter. All the editing can be done in the computer before printing.

When using a keyboard in a concert setting, local control should be activated so that when the keys are played the sounds will be triggered. Data Byte 2 should be set to 0 for Local Control off and 127 for Local Control on. Usually this parameter is changed on the device itself rather than sending a Channel Mode message.

All Notes Off (CC 123)

Send a Data Byte 2 message of 0 to turn all oscillators and notes off.

Omni Mode Off (CC 124)

This parameter switches the MIDI Input to respond only to specific channels. Activating Omni Mode Off is used to set up a MIDI device chain and have each respond to only one channel. You could use this for key splitting notes on a keyboard by setting the lower keys to play out a different channel than the upper keys.

An example is setting one module to receive on channel 1 with an acoustic bass patch and a second module connected to the first module's Thru port triggered on channel 2 with a piano patch. There are many other similar uses for Omni Mode Off.

Omni Mode On (CC 125)

This sets a MIDI device to respond indiscriminately to all incoming MIDI channels that it receives. This is often helpful for troubleshooting, and most DAWs default to receiving from all channels (and, in many cases, all ports).

Data byte 2 should be set to 0. Note: The MMA could have used two values on CC 124 to switch between Omni Mode On and Off but instead decided to use two different CCs.

Mono Mode On (CC 126)

Mono sets the sound generator to play only one note at a time, which is useful when emulating monophonic instruments such as many wind instruments that can only play a single note at a time.

Pressing and holding a note while playing and releasing other notes on top of the first creates a trill-like transition between two notes (your sound module should be set to re-trigger for this to play the first held note). This is how many synth players are able to get lightning-fast lead lines between successive notes in conjunction with a short portamento time (glide or transition time between notes).

Keep in mind that you could have 16 notes played at the same time in Mono Mode if every note played is on a different channel. Data byte 2 should be set to 0.

Poly Mode On (CC 127)

This mode allows more than one note to be played at a time on a given channel. The amount of polyphony is up to the sound module, and if it is General MIDI compliant it must have up to 24 notes of polyphony simultaneously for all channels combined (not 24 per channel at the same time). Data byte 2 should be set to 0.

System Messages

The last status message is the System Message and is divided into three categories known as System Common Messages, System Real-Time Messages, and System Exclusive Messages (SysEx). These messages transmit other miscellaneous information such as synchronization, positional data, and information relating to setup. Sometimes all of these messages are incorrectly referred to as SysEx message when they are discussed, but System Common and System Real-Time Messages have various control parameters that are not "exclusive" to one manufacturer.

System Common Messages

System Common messages do not have MIDI channels assigned to them, and all devices in the MIDI chain should respond to these messages if they are capable and are properly configured.

1. MIDI Timecode (MTC) allows for location within a song by absolute position timing using hours, minutes, seconds, and frames. Frames are a further division of seconds based on how many pictures frames pass when syncing to picture and is determined by the project frame rate. Common frame rates are 23.976, 24, 25, 29.97, 29.97 drop frame, and 30 frames per second. When using MTC to sync two devices together, and when the picture is not involved, it is recommended to use the highest frame rate. When video is involved then you should be using the frame rate associated with the video file.
   Unlike its analog counterparts, SMPTE timecode (Society of Motion Picture & Television Engineers) and LTC (Longitudinal Time Code) MTC is not contiguous and is broken up with one message transmitted every quarter frame. This means that MTC is less accurate and prone to timing inconsistencies.
2. Song Position Pointer sends the song start position so that two sequencers can start playback at the same point. Positional data are measured using 16th notes with 0 being song start. So a value of 4 would be the second quarter note and 8 the third quarter note from the start and so forth. You have a total of 4,096 quarter notes or 1,024 measures in 4/4 as possible location information.
3. Song Select is used to jump to a particular song or sequence.
4. Tune Request is used to request that the receiving device tune its internal oscillators, is used mostly for analog oscillators, and is generally not needed for digital synthesizers.

System Real-Time Messages

1. System Real-Time messages support transport control (start, stop, and continue), also referred to as MIDI Machine Control (MMC).
2. Beat clock is 24 pulses per quarter note (PPQN) that allows two devices to sync in tempo. So as the tempo increases, so does the beat clock allowing the “slave” device or the device following the master to follow tempo variations. This is important because even if two devices have the same

tempo, they will eventually drift apart without continued synchronization.

3. Active Sensing allows a receiving device to know the sending device is still attached even when no MIDI data are sent by the user. This is helpful because unplugged cables stop the active sensing messages and the receiving device stops playback, which prevents stuck notes.

System Exclusive Messages

System Exclusive messages allow manufactures a proprietary way to be able to update files, firmware (software running on hardware), fix bugs and enhance usability, and control parameters. Each manufacture is provided a unique SysEx number assigned by the MMA that is used to identify messages designed specifically for each manufacturer's devices.

When a SysEx message is recognized on the receiving end, it will respond in whatever way the manufacture has programmed for it to respond. This may be a system restore, new samples in a sampler, or a change of software menus. If the manufacture's number is received by a device that is from another manufacture or another piece of hardware from the same manufacture, it will not respond to any of the data sent until a closing SysEx message is sent, otherwise known as an EoS or End of SysEx message. This allows for both a universal protocol and the ability for companies to still have a unique way to program and interact with their own equipment. SysEx messages are also used when there are not enough CC values to control all of the parameters within a device and gives the flexibility to change parameters with single proprietary SysEx messages.

MIDI Polyphonic Expression (MPE)

Earlier in this chapter MPE was mentioned briefly as a new feature introduced in 2018. Because of backwards compatibility the developers had to be really creative when implementing polyphonic expressiveness in MIDI. The trick is to assign each note to its own MIDI channel, which gives each note all the bandwidth needed for individual pitch control and other control. Released instruments which take advantage of MPE include the Continuum Fingerboard, Linnstrument, ROLI Seaboard, Equator, UVI Falcon, Sandman Pro, Modal Electronics 002, Futuresonus Parva, Modor NF-1, and Eigenharp. Each let performers control pitch and timbre for individual notes.

MIDI Controllers and Subprotocols

Subprotocols or standards such as HUI (Human User Interface), Mackie Control Universal, CS-10, and many others have been created to give further control over a DAW and standardize how MIDI messages are used to control a workstation. Keep in mind that the MIDI messages are not changing, but how these messages are being responded to is changing. For instance, pressing play on a control surface often sends a MIDI Note On message, but rather than playing a note it activates the sequencer transport. Sequencers set up to be controlled by these units will know not to use the messages from these controllers to play a note.

A fun experiment if you have one of these controllers is to assign them to a MIDI instrument and press buttons, move faders, and turn knobs and listen to the results. The standard MIDI messages are being sent but not being received as originally intended. It is important to make sure that these controllers are not set up to work as intended in the DAW's setup/preferences or this experiment will not work since the workstation will know not to route messages from these controllers to MIDI instruments.

A continued back and forth connection or "handshake" is required with many of these substandards in order for them to communicate with DAWs. An early controller called the CS-10 uses channel 16 when communicating with DAWs. The assignments for the CS-10 are listed below, and if you compare CC assignments you can see that they do not line up with the MMA's standards that would be used to control a MIDI instrument. For example CC 77 is used for pan instead of CC 10.

CS-10 Controller Assignments

All Continuous Controllers must be sent on channel 16. Controller # and Name

0. Mute 1
1. Mute 2
2. Mute 3
3. Mute 4
4. Mute 5
5. Mute 6
6. Mute 7
7. Mute 8
8. Mode
9. Shift
10. F1
11. F2
12. F3
13. F4
14. F5
15. F6
16. F7
17. F8
18. F9
19. REWIND
20. FAST FORWARD
21. STOP
22. PLAY
23. RECORD
24. LEFT WHEEL
25. RIGHT WHEEL
26. UP CURSOR
27. DOWN CURSOR

28. LEFT CURSOR
29. RIGHT CURSOR
30. FOOT-SWITCH
64. FADER 1
65. FADER 2
66. FADER 3
67. FADER 4
68. FADER 5
69. FADER 6
70. FADER 7
71. FADER 8
72. BOOST/CUT
73. FREQUENCY
74. BANDWIDTH
75. SEND 1
76. SEND 2
77. PAN
96. WHEEL

Understanding Parts per Quarter Note

There are 960 parts per quarter note (PPQN) represented in most digital audio workstations; some will even let you adjust what the base value is. PPQN are referred to as ticks, which is the smallest resolution you will work with in MIDI timing. Ninety-six PPQN or less will generally start to have a noticeable quantized feel (see Chapter 5, Section 2: Quantization). This, of course, will depend on the music played and the tempo it was played at. Keep in mind that the display resolution may be different from the actual internal resolution of the recorded MIDI in a sequencer.

As you speed up the tempo, the recorded resolution per second increases since the resolution of the ticks stays the same per quarter note. There is a trick that manipulates the resolution on sequencers by doubling the tempo but halving the speed of the parts providing more resolution in the quantization of the played part.

Some sequencers even limit the time that could be recording in them based on what the tempo of the music is. By lowering the tempo it is possible to get more time to record, since the limitation of the max PPQN is expanded.

Binary Numbers

Since binary numbers play a role in the fundamental functionality of MIDI, let's take a look at how to count with them. Binary is a base 2 counting system with only two numbers to deal with, which are often referred to as on or off being a 1 or a 0.

MIDI messages are 24 bits, which means there are 24 binary number slots that make up each message. Let's start with a 4-bit number to see how this works. Binary numbers count from right to left instead of left to right.

0 = 0000
1 = 0001
2 = 0010
3 = 0011
4 = 0100
5 = 0101
6 = 0110
7 = 0111

At this point you should be able to guess what the rest of the sequence is. See if you can write it down next to the numbers above and then check the answers that follow.

8 = 1000
9 = 1001
10 = 1010
11 = 1011
12 = 1100
13 = 1101
14 = 1110
15 = 1111