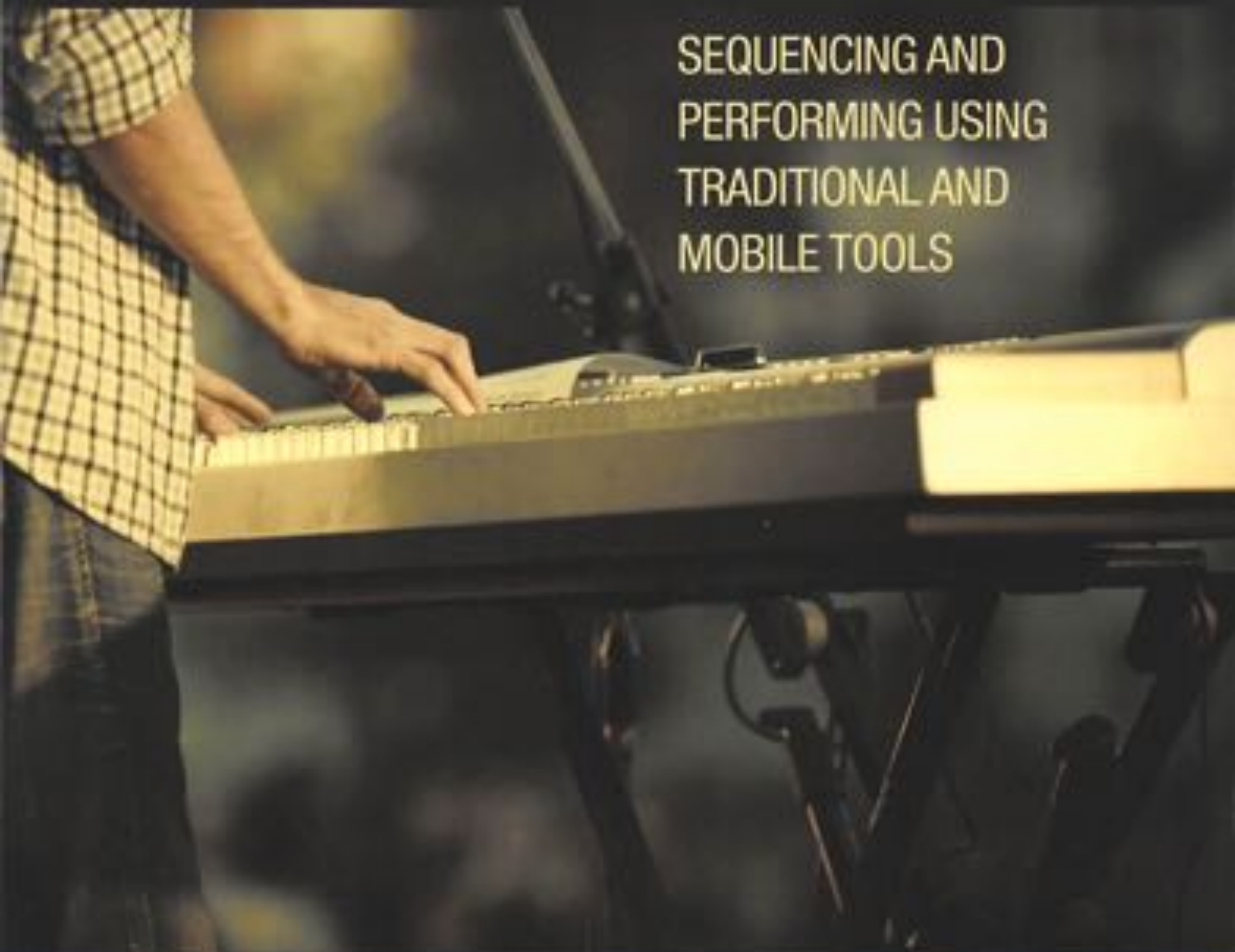
MODERN MIDI



SEQUENCING AND PERFORMING USING TRADITIONAL AND MOBILE TOOLS

SAM McGUIRE

ACKNOWLEDGMENTS

Thank you to Nathan van der Rest for your hard work; it wouldn't have happened without you! Thank you to Paul Musso and Adam Olson for your contributions. Thank you to Anais, Meagan, and everyone at Focal Press for all your hard work and efforts to get this book published! Thank you to all of my students over the years who have made my professional life a worthwhile endeavor.

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. The difference between this book and most other books 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 full orchestra, there is still the

Figure 1.2 FL Studio 11

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

An overused analogy for MIDI is the player piano. 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 20th 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 Studio. If you attach it to a MPC, then you have the potential to sound like a hit record.

Figure 1.3 Ampico Piano and Rolls

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 tells the sound 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 make an excellent virtual instrument sound good; both pieces need to be excellent in their level of quality.

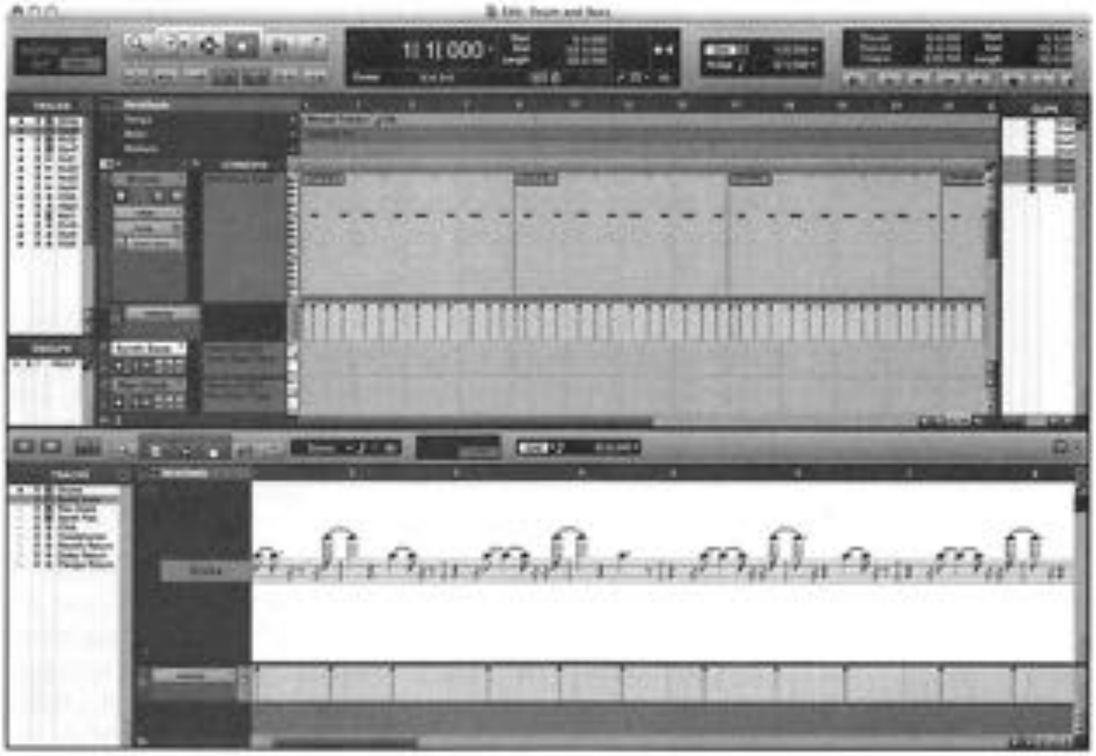


Figure 1.4 Editing MIDI in Pro Tools

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 his 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 that include 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, that have advanced options. Users have grown accustom to using these alternate view options instead of looking at the text-based views. In 10 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 that 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 that doesn't mean it doesn't do 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 non-musical 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

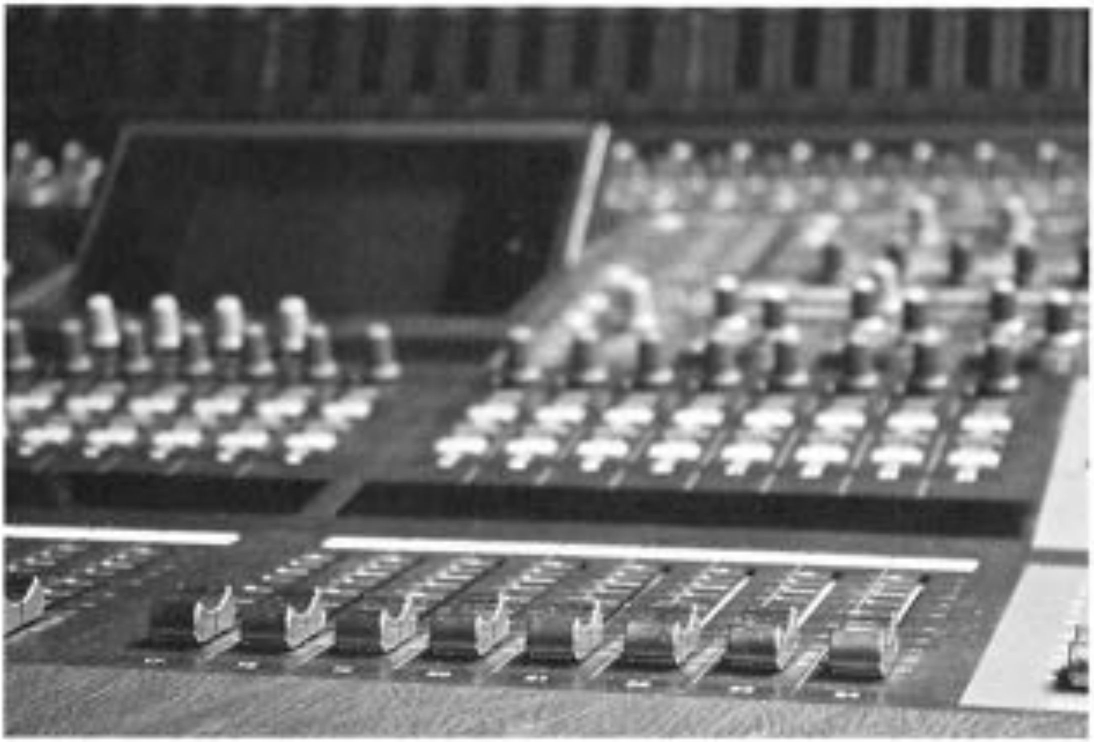


Figure 1.6 Yamaha DM2000 with Control Surface Functionality

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 often are 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 are typically proprietary solutions that work with other hardware/software that is 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 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

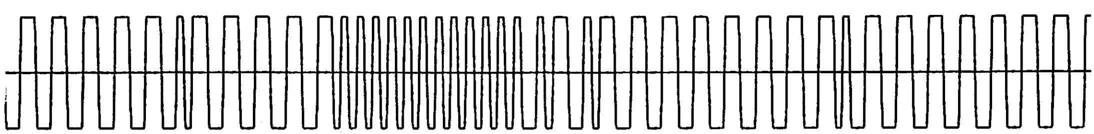


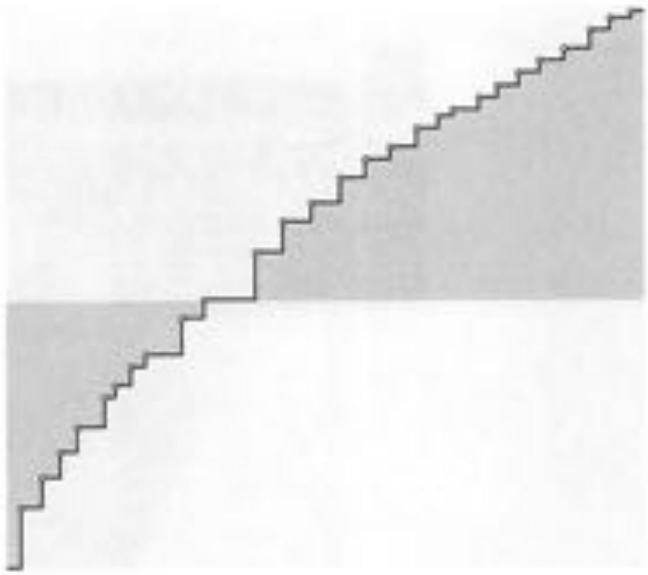
Figure 1.7 SMPTE Time Code

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.

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.

Figure 1.8 MIDI Data Zoomed to See Distinct Steps



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 training but are finding ways to make awesome music. Imagine what you can do once you have all of the tools before you 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

Figure 1.9 Data Entry with Track Pad

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 add additional instruments to, the director approved with great 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.

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 waste.

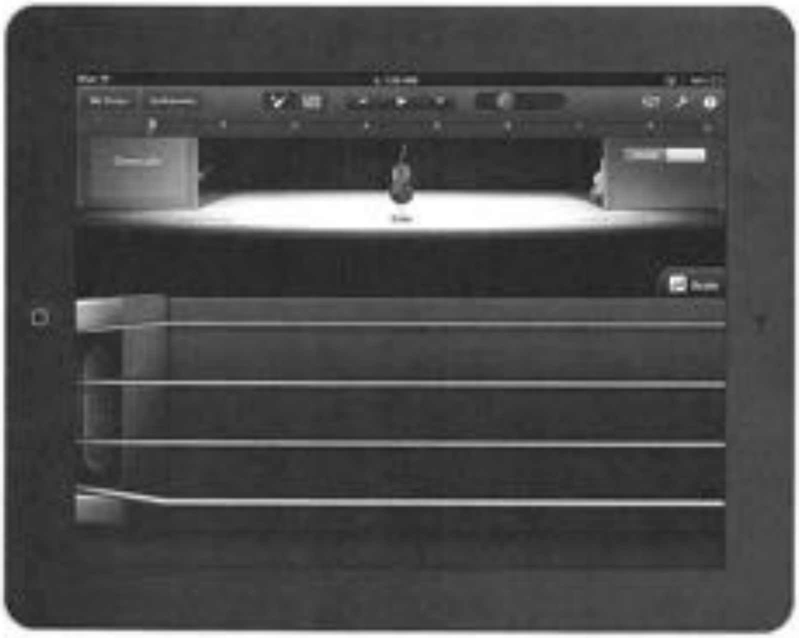


Figure 1.10 Orchestration Using Garageband

MIDI can be a real time and energy sucker for the composer that is still learning the technology. A seasoned expert uses his or her 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 him or her compose. Between session templates and an intimate knowledge of their sound banks, a seasoned composer can plow through cues as if he or she 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 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.

MIDI Data

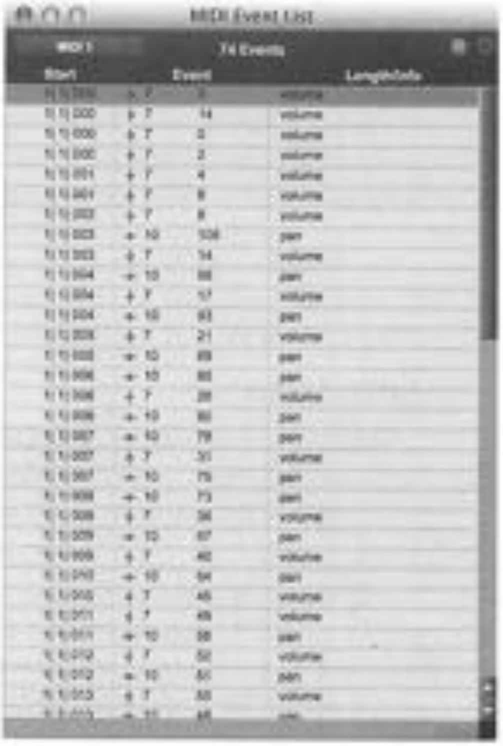
Understanding the details of MIDI data is important because it helps us understand the limitations and challenges while offering insight in how to achieve the most from a limited system.

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 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.

Figure 1.11 Pro Tools Event List



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 messages 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 "non-local" 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 setup 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 retune 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 programed 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 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 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. 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

As you can see in a 4-digit binary number, there are 16 possible values. If it were base 10, there would be a total of 10,000 (0,000–9,999).

Every time we add a number in binary we get a doubling of the previous max position:

- 1 binary number can represent two numbers: 0 and 1
- 2 binary numbers can represent 4 numbers: 00, 01, 10, and 11
- 3 binary numbers can represent 8 numbers: 000, 001, 010, 011, 100, 101, 110, and 111

The pattern of doubling continues as you add more binary numbers in the sequence.

So now let's think of MIDI in relation to a piano keyboard. If we wanted to have a full-size piano with all 88 keys, how many binary places would we need as a minimum to get 88? The answer is 7.

1 binary number = 2
2 binary numbers = 4
3 binary numbers = 8
4 binary numbers = 16
5 binary numbers = 32
6 binary numbers = 64
7 binary numbers = 128

One hundred twenty-eight is the value that we see represented over and over again with MIDI. You may have started to recognize that pattern in relation to other things you have often seen in computers with such things as RAM 128, 256, 512, 1024, 2048, etc. Computers are making the slow transition from 32-bit processing to 64-bit processing, which would be 2 to the 64th power or 1.84467440737096 E19 (a really, really big number).

The extra 40 (128 MIDI values minus 88 keys) can be used to trigger an extended range of notes, but in many cases the extra 40 numbers aren't used. Extra notes are also used for additional trigger information such as key switches, which allows you to play one note and cause other samples to be performed in a different style. These alternate performance styles might include staccato, legato, arco, and ponticello.

While 128 is a common number in MIDI, there are 24 numbers in a MIDI message broken up into groups of 8, which has a max value of 256 for each word. The reason for this discrepancy is that the start of each byte is reserved for indicating positional information. The Status byte always starts with a 1, and the start of Data byte 1 and 2 always begins with 0. As a result, the number is cut in half for the values of each that we are able to use because we have 7 free bits from every 8-bit word.

Status Byte	Data Byte 1	Data Byte 2
1 001 0000	0 1000000	0 1111111

The Status Message has bytes and nibbles (half of a byte.) The first nibble of the status byte has a maximum value of 8 (3 bits) because the first bit is always reserved for a 1 to indicate the start of a MIDI message. This number is the most significant bit (MSB) because it is the bit which has the most value. These status messages and their corresponding data byte function are listed in the appendix.

Reading Binary

Reading binary beyond 16 is really not all that difficult, though it's generally not needful to read since most event lists will show these data often as text for their values where

Table 1.1

Number to Add	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*		
Place Holder	8	7	6	5	4	3	2	1		
	1	0	0	0	0	0	0	0	=	128
	0	1	0	0	0	0	0	0	=	64
	0	0	1	0	0	0	0	0	=	32
	0	0	1	0	0	0	0	1	=	33
	0	0	1	0	0	0	1	0	=	34
	0	0	1	1	0	1	0	0	=	52
	1	0	0	0	0	0	1	0	=	130

possible. See the following table to see how you read the numbers simply by remembering their placeholder values and then adding them up based on whether or not they have a value of 1 in their place holder.

In the first example, the first 1 is in the 128 slot and all the rest are 0, so 128 + 0 + 0 + 0 + 0 + 0 + 0 = 128. Follow that pattern and you are all set to convert a binary number into a base 10 value.

MIDI Files

MIDI can be stored and passed around as small files and are much smaller in size than digital audio files would be. As you know by now, MIDI is not the audio itself but information much like music on a page. MIDI files can be found all over the Internet, many of which are free. There are several companies that sell performance-based MIDI files to be used in songs that

Table 1.2

*Channel Numbers	
0	1
1	2
10	3
11	4
100	5
101	6
110	7
111	8
1000	9
1001	10
1010	11
1011	12
1100	13
1101	14
1110	15
1111	16

have the time and feel of real musicians because they are played by real musicians. A few companies worth noting are Twiddly Bits, ToonTrack, BFD, and MusicSoft. Many companies' MIDI files are meant to be used inside proprietary players but at the foundation are MIDI files that trigger the performance.

These MIDI files have the advantage of using real musicians and in many cases using their performance to trigger real sampled or recorded audio rather than step entering or programing in music as you might read it on a page, which would feel stiff and robotic. This gives the ability to both transpose and change the tempo while still maintaining the feel of the performance.

Standard MIDI files (SMF) with the extension of .mid* are time stamped and stored in a simple file format. Before files formats became available, MIDI was purely a real-time protocol and were played sequentially as they were performed.

Standard MIDI files today are stored in two file types: Type 0 and Type 1. All major digital audio workstations on the market support both formats and can also export these formats. Type 0 stores all sequenced track as a single track, meaning that if you exported 12 tracks of MIDI or instruments in your notation software and chose File Type 0, it will be combined into a single track. There was also a third format that never really caught on, which was Type 2 and it allowed for a combination of Type 1 and Type 2.

When exporting Type 1 files, all of the tracks retain their individual/separated track information with all of their notes, controller information, and program changes. Standard MIDI files also support track names though these may not be exported with your file when saved since it is up to the DAW how the names are retained in the Type 1 files.

Tempo maps and other controller information are stored in the SMF, and when exporting these files you may choose to also send out initial messages at the start of each channel to insure that your sounds and performance will be played back as expected. Importing these files into many sequencers will likely still play back with the wrong triggered sounds despite your efforts to record proper patch change and other such messages at the start of the sequence. As a result these have to be assigned on the receiving device/virtual instrument.

Pitch bend range is adjusted on the sound module or virtual instrument; though there is an MMA standard RPN (Registered Parameter Number) for setting pitch bend range, this is often not adhered to. If the pitch bend information is intended to bend over a range of a whole step and you have your instrument set to bend over an octave, things might sound a bit strange. This information is not stored in the MIDI file as intervals but rather as relative range values based on your sound modules settings.

Messages that you may want to consider including at the start of your MIDI files to ensure accuracy in performance in your sequencer are the following:

Program Change # (status message)
Modulation (CC 1)
Initial Volume (CC 7)
Pan (CC 10–Center = 64)
Expression (CC 11)
Hold pedal (CC 64–0 = off)

Reset All Controllers (CC 121) (not all devices may recognize this command so you may prefer to zero out or reset individual controllers).

General MIDI

In 1992, General MIDI (GM) expanded on the MIDI 1.0 spec and further defined what could be expected from compatible devices. The reason behind the GM standard is that though there were standard ways of sending messages such as notes, pitch bend, etc., there was nothing in the specification that dictated which instruments would sound when a note was played. Sending a program change to one sound module may trigger an overdrive guitar, while the same program change on another module might trigger a vibraphone. Worse yet, you could also end up triggering drums or sound effects when you were expecting a pitched-based instrument instead. See the appendix for a list of GM instrument assignments.

In addition to defining what sounds will play, it also assigned a dedicated channel to percussion instruments where each note is assigned to an instrument such as a kick drum, snare drum, or ride cymbal.

Table 1.3

Note	*Drum Sound*
B0	Acoustic Bass Drum
C1	Bass Drum 1
C#1	Side Stick
D1	Acoustic Snare
Eb1	Hand Clap
E1	Electric Snare
F1	Low Floor Tom
F#1	Closed Hi Hat
G1	High Floor Tom
Ab1	Pedal Hi-Hat
A1	Low Tom
Bb1	Open Hi-Hat
B1	Low-Mid Tom
C2	Hi Mid Tom
C#2	Crash Cymbal 1
D2	High Tom
Eb2	Ride Cymbal 1
E2	Chinese Cymbal
F2	Ride Bell
F#2	Tambourine
G2	Splash Cymbal
Ab2	Cowbell
A2	Crash Cymbal 2

(*Continued*)

Table 1.3 (Continued)

Note	*Drum Sound*
Bb2	Vibraslap
B2	Ride Cymbal 2
C3	Hi Bongo
C#3	Low Bongo
D3	Mute Hi Conga
Eb3	Open Hi Conga
E3	Low Conga
F3	High Timbale
F#3	Low Timbale
G3	High Agogo
Ab3	Low Agogo
A3	Cabasa
Bb3	Maracas
B3	Short Whistle
C4	Long Whistle
C#4	Short Guiro
D4	Long Guiro
Eb4	Claves
E4	Hi Wood Block
F4	Low Wood Block
F#4	Mute Cuica
G4	Open Cuica
Ab4	Mute Triangle
A4	Open Triangle

In order to be considered compliant with the GM standard, a device has to meet a list of requirements:

1. Voices: It is required to be able to produce 16 simultaneous voices (notes) on pitched-based instruments. Every voice must be able to individually respond to its received velocity message. Eight voices are required for percussion instruments.
2. Instruments: A minimum of 16 instruments, all with their own unique sounds based on the GM instrument table (see appendix) need to be able to sound simultaneously.

Percussion sounds also need to map to the correct instruments based on the MIDI note number received.

3. Channels: All 16 MIDI channels must be supported and if a message is sent on channel 15 then there needs to be a sound that can respond to that message. Every channel also needs to be able to change to the assigned program instrument defined by GM on channels 1–9 and 11–16. Drums are reserved for channel 10.
4. Channel Voice Status Messages: Must use Note On, Channel Pressure, Program and Pitch Bend.
5. Control Change Messages:
   - 1 for modulation;
   - 7 for volume;
   - 10 for panning or placement of the sound in the speakers;
   - 11 for expression (this is generally used to change the volume and can be used as a trim control);
   - 64 pedal on or off, and is almost always used for sustain;
   - 121 and 123 RPN (registered parameter number) numbers 0 for pitch bend range, 1 for master fine tuning in cents (100 cents in a half-step), 2 for master course tuning in half-steps;
6. Other Messages: Respond to the data entry controller 6 for coarse adjustment when using the RPNs, and data entry 38 for fine adjustments for pitch bend range and master tuning, as well as all General MIDI Level 1 System Messages.

GS and XG

Both Roland and Yamaha came up with their own standards to further unify compatibility across their products. Roland came out with General Standard (GS) in 1991 and Yamaha has three versions of XG (doesn't stand for anything but is an extension of the GM format) known as "levels" that started in 1994—each with higher standards of voices, drum kits, and channels required to be XG 1, 2, or 3 compliant. They also defined how continuous controllers would be used on products that were listed as GS or XG. Keep in mind that this did not in any way alter the original MIDI spec, but rather further defined requirements of the hardware or software that bore these labels. Some of the messages that hadn't been defined were now required to follow a standard of what they were controlling. This serves much the same purpose as General MIDI does.

GM2

As a result of this in 1999, General MIDI Level 2 was passed by the MMA, which like the various Yamaha XG levels introduces further requirements while maintaining the requirements of GM 1. Thirty-two simultaneous voices became required rather than the 24 previously required. It also raised the number to 16 simultaneous voices vs. the 8 in GM1 for percussion instruments. Also similar to the GS spec that allowed more than 1 channel for percussion (channel 16 in GS), channel 10 and 11 became dedicated for use of percussion tracks. Several other control change and Universal SysEx messages were standardized.

Running Status

Running status is the process of sending MIDI messages without the Status Message to be more efficient. This relies on the most recently sent Status Message and keeps it stored in memory and applies all subsequent Data Bytes to the same message type. The advantage of running status is that since all MIDI messages are transferred serially, it means every note or message has to wait its turn to be played and omitting repetitive data can keep things efficient. MIDI is transmitted at a data rate of 31.25 kilobits per second with 10 bits total transmitted for every 8-bit message or, in other words, 30 bits for a single MIDI message. There is a start bit that has a value of 0, 8 data bits, and 1 stop bit that has a value of 1. This means that a single MIDI message takes slightly less than 1 millisecond (a thousandth of a second) to be transmitted.

This all is not much of a problem until you start getting into many chords all played at once. Having upward of 30 notes that are supposed to all start at the same time would cause almost 30 ms of difference between the first note and the last note, and this causes a smearing in the sound, especially when those notes are to be played in unison. To help alleviate this problem Running Status only transmits the data byte messages until the status changes. What is even more exciting is that a Note On message of zero can be sent in place of a Note Off message. As you would expect, there are many Note Off messages that are expected to be sent to stop the notes from playing, which would mix and mingle with Note On messages requiring the status byte to be sent again and again. This reduces the effectiveness of running status by sending Note On messages with a value of zero in place of Note Off messages; there is no need to send a new Status Message and all of the notes are played with less delay than would otherwise be possible.

Summary

Some of you may have gotten exactly what you wanted from this chapter and others may still be scratching your head about why MIDI Messages are important to understand. The truth is that understanding the basics of MIDI will not make your music better, but it can help you when troubleshooting and getting the most out of your sequencers. Let's move to the next chapter and explore MIDI hardware.

Section 2—GM Assignments

1. Acosutic Grand Piano	50. String Ensemble 2	99. FX 3 (Crystal)
2. Bright Acoustic Piano	51. SynthStrings 1	100. FX 4 (Atmosphere)
3. Electric Grand Piano	52. SynthStrings 2	101. FX 5 (Brightness)
4. Honky-Tonk Piano	53. Choir Aahs	102. FX 6 (Metallic)
5. Electric Piano 1	54. Voice Oohs	103. FX 7 (Echoes)
6. Electric Piano 2	55. Synth Voice	104. FX 8 (Sci-Fi)
7. Harpsichord	56. Orchestra Hit	105. Sitar
8. Clavichord	57. Trumpet	106. Banjo
9. Celesta	58. Trombone	107. Shamisen
10. Glockenspiel	59. Tuba	108. Koto
11. Music Box	60. Muted trumpet	109. Kalimba
12. Vibraphone	61. French Horn	110. Bag Pipe
13. Marimba	62. Brass Section	111. Fiddle
14. Xylophone	63. Synth Brass 1	112. Shanai
15. Tubular Bells	64. Synth Brass 2	113. Tinkle Bell
16. Dulcimer	65. Soprano Sax	114. Agogo
17. Drawbar Organ	66. Alto Sax	115. Steel Drums
18. Percussive Organ	67. Tenor Sax	116. Woodblock
19. Rock Organ	68. Baritone Sax	117. Taiko Drum
20. Church Organ	69. Oboe	118. Melodic Tom
21. Reed Organ	70. English Horn	119. Synth Drum
22. Accordian	71. Bassoon	120. Reverse Cymbal
23. Harmonica	72. Clarinet	121. Guitar Fret Noise
24. Tango Accordian	73. Piccolo	122. Breath Noise
25. Acoustic Guitar (Nylon)	74. Flute	123. Seashore
26. Acoustic Guitar (Steel)	75. Recorder	124. Bird Tweet
27. Electric Guitar (Jazz)	76. Pan Flute	125. Telephone Ring
28. Electric Guitar (Clean)	77. Blown Bottle	126. Helicopter
29. Electric Guitar (Muted)	78. Shakuhachi	127. Applause
30. Overdriven Guitar	79. Whistle	128. Gun Shot
31. Distortin Guitar	80. Ocarina	
32. Guitar Harmonics	81. Lead 1 (Square)	
33. Acoustic Bass	82. Lead 2 (Sawtooth)	
34. Electric Bass (Finger)	83. Lead 3 (Calliope)	
35. Electric Bass (Pick)	84. Lead 4 (Chiff)	
36. Fretless Bass	85. Lead 5 (Charang)	
37. Slap Bass 1	86. Lead 6 (Voice)	
38. Slap Bass 2	87. Lead 7 (Fifths)	
39. Synth Bass 1	88. Lead 8 (Bass + Lead)	
40. Synth Bass 2	89. Pad 1 (New Age)	
41. Violin	90. Pad 2 (Warm)	
42. Viola	91. Pad 3 (Polysynth)	
43. Cello	92. Pad 4 (Choir)	
44. Contrabass	93. Pad 5 (Bowed)	
45. Tremolo Strings	94. Pad 6 (Metallic)	
46. Pizzicato Strings	95. Pad 7 (Halo)	
47. Orchestral Harp	96. Pad 8 (Sweep)	
48. Timpani	97. FX 1 (Rain)	
49. String Ensemble	98. FX 2 (Soundtrack)	

Figure 12.8 GM Instruments

Figure 12.9 GM Percussion

35. Acoustic Bass Drum
36. Bass Drum 1
37. Side Stick
38. Acoustic Snare
39. Hand Clap
40. Electric Snare
41. Low Floor Tom
42. Closed Hi-Hat
43. High Floor Tom
44. Pedal Hi-Hat
45. Low Tom
46. Open Hi-Hat
47. Low-Mid Tom
48. Hi-Mid Tom
49. Crash Cymbal 1
50. High Tom
51. Ride Cymbal 1
52. Chinese Cymbal
53. Ride Bell
54. Tambourine
55. Splash Cymbal
56. Cowbell
57. Crash Cymabl 2
58. Vibraslap
59. Ride Cymbal 2
60. Hi Bongo
61. Low Bongo
62. Mute Hi Conga
63. Open Hi Conga
64. Low Conga
65. High Timbale
66. Low Timbale
67. High Agogo
68. Low Agogo
69. Cabasa
70. Maracas
71. Short Whistle
72. Long Whistle
73. Short Guiro
74. Long Guiro
75. Claves
76. Hi Wood Block
77. Low Wood Block
78. Mute Cuica
79. Open Cuica
80. Mute Triangle
81. Open Triangle

Section 3—MIDI Implementation Chart

MIDI IMPLEMENTATION CHART V2 INSTRUCTIONS

Introduction

IMPORTANT: MMA recommends manufacturers of MIDI devices and software ship a MIDI Implementation Chart with the device, or make the chart available online. The Version 2 format described in this RP has three pages and is the preferred format. Manufacturers who prefer a one-page chart may continue to use the original format described in the MIDI 1.0 Specification.

This revised version of the standard MIDI Implementation Chart is designed as a quick reference guide that allows users to identify at a glance which MIDI messages and functions are implemented by the device. In this document, the term "device" is defined as a hardware device or software program that (a) transmits and/or receives MIDI messages, and/or (b) reads and/or writes MMA-defined file formats. Use of the V2 MIDI Implementation Chart is optional. The standardization of this chart enables a user to judge the compatibility between two devices to be connected, simply by comparing the "Transmit/Export" column of one device with the "Recognize/Import" column of the other. For this reason, each chart should be the same size and should have the same number of lines if at all possible. This chart has been designed to fit both standard A4 and 8½ inch x 11 inch paper. If a smaller page size is required for a particular product, page breaks may be inserted as necessary, but it is strongly recommended to maintain the row height of the original chart, in order to facilitate comparisons.

IMPORTANT: The MMA Technical Standard Board will review the MIDI Implementation Chart annually, and will update the chart template and these instructions as necessary to reflect newly standardized MIDI features.

All Pages

- Use the header at the top of each page of the chart to enter the manufacturer's name, model name/number of the device, version number, and date of chart preparation.
- On all pages, if the manufacturer wishes to present additional information that will not physically fit in the "Remarks" column, this must be done by inserting a reference to the appropriate page or section number in the user manual where the information can be found. If the number of banks the device supports does not fit in the "Comments" section, the manufacturer should continue the list on a separate sheet of paper.

Page 1: Basic Information, MIDI Timing and Synchronization, and Extensions Compatibility?

General?

The body of page 1 of the chart is divided into four columns. The first column lists the specific function or item, the next two columns give information about whether the specified function is transmitted or exported and/or received or imported (and, if so, may contain information about the range of data)/. The fourth column is used for remarks about anything unique to this implementation. For functions involving files, the 2nd and 3rd columns give information on whether the files can be saved (exported) or opened (imported), and, if so, what degree of compatibility is provided.

Functions Description

MIDI channels	The range of MIDI channels that the device transmits, exports, responds to, and/or imports. Devices using extended channel systems via multiple cables or input/output ports should list the total number of channels in the appropriate "Transmitted" or "Recognized"

	columns and should use the "Remarks" column to indicate the terminology used by the device to identify the extra channels (i.e., "A1–A16, B1–B16").
Note numbers	The total range of transmitted or recognized notes.
Program Change	Indicate the range of Program Change numbers that are transmitted and/or recognized. If not implemented, enter a "No" in the appropriate column.
Bank Select response	Use a "Yes" or "No" to indicate whether or not the device correctly responds to Bank Select messages as per the MIDI 1.0 Specification. Devices that respond only to Bank Select MSB (cc #0) but not to the LSB (cc #32) should place a "No" in the "Recognized" column and should indicate this in the "Remarks" column. If the device does correctly respond to Bank Select messages, use the "Remarks" column to indicate what banks or ranges of banks are available in the device. If certain banks are accessible only by MIDI (and not by front panel user control), these should be listed in the "Remarks" column.
Modes supported	Use a "Yes" or "No" to indicate whether or not the device supports each of the five listed modes of reception.
Note-On Velocity	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports Note-On Velocity.

Note-Off Velocity	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports Note-Off Velocity.
Channel Aftertouch	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports Channel Aftertouch.
Poly (Key) Aftertouch	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports Poly (Key) Aftertouch.
Pitch Bend	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports Pitch Bend.
Active Sensing?	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports Active Sensing.
System Reset?	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports System Reset.
Tune Request?	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports Tune Request.
Universal System Exclusive	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports the various Universal System Exclusive messages described. If the device supports additional Universal System Exclusive messages that are not listed, for example, the SP-MIDI MIP message or

	Global Parameter Control, use the "Other" category and, in the Remarks column, enter the name(s) of the message(s) supported.
Manufacturer or Non-Commercial System Exclusive	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports any Manufacturer System Exclusive messages or Non-Commercial System Exclusive messages. In the Remarks column, enter the name(s) of the message(s) supported, and either the words "Non-Commercial" or the manufacturer name(s) and MMA Manufacturer ID(s) for the message(s) supported.
NRPNs	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports NRPNs. Manufacturers may wish to list the NRPNs the device uses in the "Remarks" column (if this information will not physically fit in the "Remarks" column, provide a reference to the page or section number in the user manual where the information can be found).
RPNs	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports each of the specified RPNs.

MIDI Timing and Synchronization

MIDI Clock?	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports MIDI Time Code (MTC).

Song Position Pointer	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports MIDI Machine Control (MMC). If yes, indicate in the Remarks column whether the device transmits and/or responds in Open or Closed Loop mode. Manufacturers of devices utilizing MIDI Machine Control may wish to attach a separate chart indicating the specific MMC messages transmitted and/or recognized by the device. If so, indicate the presence of this "sub-chart" in the Remarks column.
Song Select	Indicate whether or not the device transmits, exports, responds to, and/or imports MIDI Show Control (MSC). If not, indicate "No". If yes, indicate the Level of MIDI Show Control supported. Manufacturers of devices utilizing MIDI Show Control may wish to attach a separate chart indicating the specific MSC messages transmitted and/or recognized by the device. If so, indicate the presence of this "sub-chart" in the Remarks column.
Start/ Continue/ Stop	Indicate whether or not the device has a mode of operation that complies with any of the General MIDI specifications: General MIDI System Level 1 (GM), General MIDI System Level 2 (GM2), and/or General MIDI Lite (GM Lite). If not, indicate "No." If yes, indicate the GM Level(s) supported. Also, if GM is the default power-up mode, indicate GM Lite, GM Level 1 or GM Level 2. If not, indicate "No."

MIDI Time Code	Indicate whether or not the device has a mode of operation that complies with any of the Downloadable Sounds specifications: DLS Level 1 (DLS), DLS Level 2 (DLS2, including DLS 2.1 and DLS 2.2), and/or Mobile DLS. If not, indicate "No." If yes, indicate the DLS Level(s) supported. Also, indicate whether or not the device can import and/or export DLS files. If not, indicate "No." If yes, indicate what types. It is recommended that manufacturers indicate in the Remarks column the means of receiving DLS data (i.e., specific physical format, device interface, or transport protocol, etc.) and, if a file system media is used, indicate in the Remarks column the exact format(s) supported (i.e., Windows, Mac OS, or Linux file system version, etc.).
MIDI Machine Control	Use a "Yes" or "No" to indicate whether or not the device has a mode of operation that can play, import, and/or export any of the Standard MIDI File formats, and, if so, the formats(s) supported: format 0 (single track), format 1 (multi-track), and/or format 2 (multiple independent single-track patterns). If yes, it is also recommended that manufacturers indicate in the Remarks column the means of receiving SMF data (i.e., specific physical format, device interface, or transport protocol, etc.) and, if a file system media is used, indicate in the Remarks column the exact format(s) supported (i.e., Windows, Mac OS, or Linux file system version, etc.).

MIDI Show Control	Use a "Yes" or "No" to indicate whether or not the device transmits, exports, responds to, and/or imports MIDI Time Code (MTC).

Extensions Compatibility

General MIDI	Indicate whether or not the device transmits, exports, responds to, and/or imports MIDI Show Control (MSC). If not, indicate "No." If yes, indicate the Level of MIDI Show Control supported. Manufacturers of devices utilizing MIDI Show Control may wish to attach a separate chart indicating the specific MSC messages transmitted and/or recognized by the device. If so, indicate the presence of this "sub-chart" in the Remarks column.
DLS	Indicate whether or not the device has a mode of operation that complies with any of the General MIDI specifications: General MIDI System Level 1 (GM), General MIDI System Level 2 (GM2) and/or General MIDI Lite (GM Lite). If not, indicate "No." If yes, indicate the GM Level(s) supported. Also, if GM is the default power-up mode, indicate GM Lite, GM Level 1 or GM Level 2. If not, indicate "No."
Standard MIDI Files	Indicate whether or not the device has a mode of operation that complies with any of the Downloadable Sounds specifications: DLS Level 1 (DLS), DLS Level 2 (DLS2, including DLS 2.1 and DLS 2.2), and/or Mobile DLS. If not, indicate

	"No." If yes, indicate the DLS Level(s) supported. Also, indicate whether or not the device can import and/or export DLS files. If not, indicate "No." If yes, indicate what types. It is recommended that manufacturers indicate in the Remarks column the means of receiving DLS data (i.e., specific physical format, device interface, or transport protocol, etc.) and, if a file system media is used, indicate in the Remarks column the exact format(s) supported (i.e., Windows, Mac OS, or Linux file system version, etc.).
XMF	Indicate whether or not the device has a mode of operation that can play, import, and/or export any of the officially defined XMF File Types: XMF Type 0, XMF Type 1, or Mobile XMF
(XMF Type 2).	If the device uses the XMF Meta File Format in a manner that does not conform to any of the XMF File Type specifications, indicate this in the Remarks column.
SP-MIDI	Indicate whether or not the device has a mode of operation that can play, import, and/or export Scalable Polyphony MIDI (SP-MIDI) data. If yes, indicate which SP-MIDI profile specification(s) that the device conforms to, for example, SP-MIDI 5-24 Voice Profile for 3GPP.

Pages 2 and 3: Control Number Information

General

Pages 2 and 3 of the chart are used to describe how the device implements the 128 MIDI Control Change messages (including those reserved for Channel Mode messages). IMPORTANT: The use of pages 2 and 3 is optional for devices that do not transmit, export, respond to, and/or import any Control Change messages. The first 120 Control Change messages are controller numbers, and the last 8 (cc# 120–127) reserved for Channel Mode messages. These pages are divided into five columns, with the first column listing the control number in decimal. The second column lists the defined function from the MIDI 1.0 Specification for that control number if one exists, or is blank if undefined in the MIDI 1.0 Specification. Manufacturers using these undefined controller numbers should enter in the title of the assigned function in this column and should make an entry in the fifth, "Remarks" column noting this proprietary usage. The third and fourth columns are used to indicate whether the specified controller number is transmitted, exported, responded to, and/or imported.

Functions Description

The inclusion of these two pages in a MIDI device's owner's manual is optional. Use a "Yes" or "No" to indicate whether or not the device transmits and/or responds to each of the listed control numbers. Use the "Remarks" column to indicate whether a particular controller number is assignable or if the controller is being used in a non-standard way (i.e., if the device is capable of receiving the controller message but routes it in an unusual way). If using any undefined controller number, enter the title of the assigned function in the second, "Function" column and make an entry in the fifth, "Remarks" column noting this proprietary usage.

MIDI Implementation Chart v. 2.0 (Page 1 of 3)			
Manufacturer: Model:	Version: Date:		
	Transmit/Export	Recognize/Import	Remarks
1. Basic Information			
MIDI channels			
Note numbers			
Program change			
Bank Select response? (Yes/No) If yes, list banks utilized in remarks column			
Modes supported: Mode 1: Omni-On, Poly (Yes/No) Mode 2: Omni-On, Mono (Yes/No) Mode 3: Omni-Off, Poly (Yes/No) Mode 4: Omni-Off, Mono (Yes/No) Multi Mode (Yes/No)			
Note-On Velocity (Yes/No)			
Note-Off Velocity (Yes/No)			
Channel Aftertouch (Yes/No)			
Poly (Key) Aftertouch (Yes/No)			
Pitch Bend (Yes/No)			
Active Sensing (Yes/No)			
System Reset (Yes/No)			
Tune Request (Yes/No)			
Universal System Exclusive: Sample Dump Standard (Yes/No) Device Inquiry (Yes/No) File Dump (Yes/No) MIDI Tuning (Yes/No) Master Volume (Yes/No) Master Balance (Yes/No) Notation Information (Yes/No) Turn GM1 System On (Yes/No) Turn GM2 System On (Yes/No) Turn GM System Off (Yes/No) DLS-1 (Yes/No) File Reference (Yes/No) Controller Destination (Yes/No) Key-based Instrument Ctrl (Yes/No) Master Fine/Coarse Tune (Yes/No) Other Universal System Exclusive			
Manufacturer or Non-Commercial System Exclusive			
NRPNs (Yes/No)			
RPN 00 (Pitch Bend Sensitivity) (Yes/No) RPN 01 (Channel Fine Tune) (Yes/No) RPN 02 (Channel Coarse Tune) (Yes/No) RPN 03 (Tuning Program Select) (Yes/No) RPN 04 (Tuning Bank Select) (Yes/No) RPN 05 (Modulation Depth Range) (Yes/No)			
2. MIDI Timing and Synchronization			
MIDI Clock (Yes/No)			
Song Position Pointer (Yes/No)			
Song Select (Yes/No)			
Start (Yes/No) Continue (Yes/No) Stop (Yes/No)			
MIDI Time Code (Yes/No)			
MIDI Machine Control (Yes/No)			
MIDI Show Control (Yes/No) If yes, MSC Level supported			
3. Extensions Compatibility			
General MIDI compatible? (Level(s)/No) Is GM default power-up mode? (Level/No)			
DLS compatible? (Levels(s)/No) (DLS File Type(s)/No)			
Standard MIDI Files (Type(s)/No)			
XMF Files (Type(s)/No)			
SP-MIDI compatible? (Yes/No)			

MIDI Implementation Chart v. 2.0 Control Number Information (Page 2 of 3)				
Manufacturer:	Model:	Version:	Date:	
Control #	Function	Transmitted (Y/N)	Recognized (Y/N)	Remarks
0	Bank Select (MSB)			
1	Modulation Wheel (MSB)			
2	Breath Controller (MSB)			
3				
4	Foot Controller (MSB)			
5	Portamento Time (MSB)			
6	Data Entry (MSB)			
7	Channel Volume (MSB)			
8	Balance (MSB)			
9				
10	Pan (MSB)			
11	Expression (MSB)			
12	Effect Control 1 (MSB)			
13	Effect Control 2 (MSB)			
14				
15				
16	General Purpose Controller 1 (MSB)			
17	General Purpose Controller 2 (MSB)			
18	General Purpose Controller 3 (MSB)			
19	General Purpose Controller 4 (MSB)			
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32	Bank Select (LSB)			
33	Modulation Wheel (LSB)			
34	Breath Controller (LSB)			
35				
36	Foot Controller (LSB)			
37	Portamento Time (LSB)			
38	Data Entry (LSB)			
39	Channel Volume (LSB)			
40	Balance (LSB)			
41				
42	Pan (LSB)			
43	Expression (LSB)			
44	Effect Control 1 (LSB)			
45	Effect Control 2 (LSB)			
46				
47				
48	General Purpose Controller 1 (LSB)			
49	General Purpose Controller 2 (LSB)			
50	General Purpose Controller 3 (LSB)			
51	General Purpose Controller 4 (LSB)			
52				
53				
54				
55				
56				
57				
58				
59				
60				
61				
62				
63				

MIDI Implementation Chart v 2.0 Control Number Information (Page 2 of 3)

Manufacturer: Model: Version: Date:

Control #	Function	Transmitted (Y/N)	Recognized (Y/N)	Remarks
64	Sustain Pedal			
65	Portamento On/Off			
66	Sostenuto			
67	Soft Pedal			
68	Legato Footswitch			
69	Hold 2			
70	Sound Controller 1 (default: Sound Variation)			
71	Sound Controller 2 (default: Timbre / Harmonic Quality)			
72	Sound Controller 3 (default: Release Time)			
73	Sound Controller 4 (default: Attack Time)			
74	Sound Controller 5 (default: Brightness)			
75	Sound Controller 6 (GM2 default: Decay Time)			
76	Sound Controller 7 (GM2 default: Vibrato Rate)			
77	Sound Controller 8 (GM2 default: Vibrato Depth)			
78	Sound Controller 9 (GM2 default: Vibrato Delay)			
79	Sound Controller 10 (GM2 default: Undefined)			
80	General Purpose Controller 5			
81	General Purpose Controller 6			
82	General Purpose Controller 7			
83	General Purpose Controller 8			
84	Portamento Control			
85				
86				
87				
88				
89				
90				
91	Effects 1 Depth (default: Reverb Send)			
92	Effects 2 Depth (default: Tremolo Depth)			
93	Effects 3 Depth (default: Chorus Send)			
94	Effects 4 Depth (default: Celeste [Detune] Depth)			
95	Effects 5 Depth (default: 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				
103				
104				
105				
106				
107				
108				
109				
110				
111				
112				
113				
114				
115				
116				
117				
118				
119				
120	All Sound Off			
121	Reset All Controllers			
122	Local Control On/Off			
123	All Notes Off			
124	Omni Mode Off			
125	Omni Mode On			
126	Poly Mode Off			
127	Poly Mode On			

Section 4—MIDI Messages

MIDI Messages

Table 1—Summary of MIDI Messages

The following table lists many of the major MIDI messages in numerical (binary) order. This table is intended as an overview of MIDI, and is by no means complete. Additional messages are listed in the printed documentation available from the MMA.

WARNING! Details about implementing these messages can dramatically impact compatibility with other products. We strongly recommend consulting the official *MMA Detailed MIDI Specification* **(**http://www.midi.org/techspecs/midispec.php**) for additional information.**

Table 12.1 MIDI 1.0 Specification Message Summary

Status D7–D0	*Data Byte(s) D7–D0*	*Description*
Channel Voice Messages [nnnn = 0–15 (MIDI Channel Number 1–16)]		
1000nnnn	0kkkkkkk 0vvvvvvv	Note Off event. This message is sent when a note is released (ended). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity.
1001nnnn	0kkkkkkk 0vvvvvvv	Note On event. This message is sent when a note is depressed (start). (kkkkkkk) is the key (note) number. (vvvvvvv) is the velocity.
1010nnnn	0kkkkkkk 0vvvvvvv	Polyphonic Key Pressure (Aftertouch). This message is most often sent by pressing down on the key after it "bottoms out". (kkkkkkk) is the key (note) number. (vvvvvvv) is the pressure value.
1011nnnn	0ccccccc 0vvvvvvv	Control Change. This message is sent when a controller value changes. Controllers include devices such as pedals and levers. Controller numbers 120–127 are reserved as "Channel Mode Messages" (below). (ccccccc) is the controller number (0–119). (vvvvvvv) is the controller value (0–127).

1100nnnn	0ppppppp	Program Change. This message sent when the patch number changes. (ppppppp) is the new program number.
1101nnnn	0vvvvvvv	Channel Pressure (After-touch). This message is most often sent by pressing down on the key after it "bottoms out". This message is different from polyphonic after-touch. Use this message to send the single greatest pressure value (of all the current depressed keys). (vvvvvvv) is the pressure value.
1110nnnn	0lllllll 0mmmmmmm	Pitch Wheel Change. This message is sent to indicate a change in the pitch wheel. The pitch wheel is measured by a fourteen bit value. Center (no pitch change) is 2000H. Sensitivity is a function of the transmitter. (lllllll) are the least significant 7 bits. (mmmmmm) are the most significant 7 bits.
Channel Mode Messages (See also Control Change, above)		
1011nnnn	0ccccccc 0vvvvvvv	Channel Mode Messages. This the same code as the Control Change (above), but implements Mode control and special message by using reserved controller numbers 120–127. The commands are: All Sound Off. When All Sound Off is received all oscillators will turn off, and their volume envelopes are set to zero as soon as possible. c = 120, v = 0: All Sound Off Reset All Controllers. When Reset All Controllers is received, all controller values are reset to their default values. (See specific Recommended Practices for defaults). c = 121, v = x: Value must only be zero unless otherwise allowed in a specific Recommended Practice. Local Control. When Local Control is Off, all devices on a given channel will respond only to data received over MIDI. Played data, etc. will be ignored. Local Control On restores the functions of the normal controllers. c = 122, v = 0: Local Control Off c = 122, v = 127: Local Control On All Notes Off. When an All Notes Off is received, all oscillators will turn off. c = 123, v = 0: All Notes Off (See text for description of actual mode commands.) c = 124, v = 0: Omni Mode Off c = 125, v = 0: Omni Mode On c = 126, v = M: Mono Mode On (Poly Off) where M is the number of channels (Omni Off) or 0 (Omni On) c = 127, v = 0: Poly Mode On (Mono Off) (Note: These four messages also cause All Notes Off)

(Continued)

Table 12.1 (Continued)

Status D7—D0	Data Byte(s) D7—D0	Description
System Common Messages		
11110000	0iiiiiii [0iiiiiii 0iiiiiii] 0ddddddd — — 0ddddddd 11110111	System Exclusive. This message type allows manufacturers to create their own messages (such as bulk dumps, patch parameters, and other non-spec data) and provides a mechanism for creating additional MIDI Specification messages. The Manufacturer's ID code (assigned by MMA or AMEI) is either 1 byte (0iiiiiii) or 3 bytes (0iiiiiii 0iiiiiii 0iiiiiii). Two of the 1 Byte IDs are reserved for extensions called Universal Exclusive Messages, which are not manufacturer-specific. 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 (0ddddddd). Otherwise, the message will be ignored. (Note: Real-Time messages ONLY may be interleaved with a System Exclusive.)
11110001	0nnndddd	MIDI Time Code Quarter Frame. nnn = Message Type dddd = Values
11110010	0lllllll 0mmmmmmm	Song Position Pointer. This is an internal 14 bit register that holds the number of MIDI beats (1 beat = six MIDI clocks) since the start of the song. l is the LSB, m the MSB.
11110011	0sssssss	Song Select. The Song Select specifies which sequence or song is to be played.
11110100		Undefined. (Reserved)
11110101		Undefined. (Reserved)
11110110		Tune Request. Upon receiving a Tune Request, all analog synthesizers should tune their oscillators.
11110111		End of Exclusive. Used to terminate a System Exclusive dump (see above).
System Real-Time Messages		
11111000		Timing Clock. Sent 24 times per quarter note when synchronization is required (see text).
11111001		Undefined. (Reserved)
11111010		Start. Start the current sequence playing. (This message will be followed with Timing Clocks).
11111011		Continue. Continue at the point the sequence was Stopped.
11111100		Stop. Stop the current sequence.
11111101		Undefined. (Reserved)

11111110	Active Sensing. This message is intended to be sent repeatedly to tell the receiver that a connection is alive. Use of this message is optional. When initially received, the receiver will expect to receive another Active Sensing message each 300ms (max), and if it does not then it will assume that the connection has been terminated. At termination, the receiver will turn off all voices and return to normal (non- active sensing) operation.
11111111	Reset. Reset all receivers in the system to power-up status. This should be used sparingly, preferably under manual control. In particular, it should not be sent on power-up.

Table 12.2 Expanded Status Bytes List

Status Byte		*Data Bytes*	
1st Byte Value Binary / Hex / Dec	*Function*	*2nd Byte*	*3rd Byte*
10000000 = 80 = 128	Chan 1 Note off	Note Number (0–127)	Note Velocity (0–127)
10000001 = 81 = 129	Chan 2 Note off	Note Number (0–127)	Note Velocity (0–127)
10000010 = 82 = 130	Chan 3 Note off	Note Number (0–127)	Note Velocity (0–127)
10000011 = 83 = 131	Chan 4 Note off	Note Number (0–127)	Note Velocity (0–127)
10000100 = 84 = 132	Chan 5 Note off	Note Number (0–127)	Note Velocity (0–127)
10000101 = 85 = 133	Chan 6 Note off	Note Number (0–127)	Note Velocity (0–127)
10000110 = 86 = 134	Chan 7 Note off	Note Number (0–127)	Note Velocity (0–127)
10000111 = 87 = 135	Chan 8 Note off	Note Number (0–127)	Note Velocity (0–127)
10001000 = 88 = 136	Chan 9 Note off	Note Number (0–127)	Note Velocity (0–127)
10001001 = 89 = 137	Chan 10 Note off	Note Number (0–127)	Note Velocity (0–127)
10001010 = 8A = 138	Chan 11 Note off	Note Number (0–127)	Note Velocity (0–127)
10001011 = 8B = 139	Chan 12 Note off	Note Number (0–127)	Note Velocity (0–127)
10001100 = 8C = 140	Chan 13 Note off	Note Number (0–127)	Note Velocity (0–127)
10001101 = 8D = 141	Chan 14 Note off	Note Number (0–127)	Note Velocity (0–127)
10001110 = 8E = 142	Chan 15 Note off	Note Number (0–127)	Note Velocity (0–127)
10001111 = 8F = 143	Chan 16 Note off	Note Number (0–127)	Note Velocity (0–127)
10010000 = 90 = 144	Chan 1 Note on	Note Number (0–127)	Note Velocity (0–127)
10010001 = 91 = 145	Chan 2 Note on	Note Number (0–127)	Note Velocity (0–127)

(*Continued*)

Table 12.2 (Continued)

Status Byte		Data Bytes	
1st Byte Value *Binary /Hex/ Dec*	*Function*	*2nd Byte*	*3rd Byte*
10010010 = 92 = 146	Chan 3 Note on	Note Number (0–127)	Note Velocity (0–127)
10010011 = 93 = 147	Chan 4 Note on	Note Number (0–127)	Note Velocity (0–127)
10010100 = 94 = 148	Chan 5 Note on	Note Number (0–127)	Note Velocity (0–127)
10010101 = 95 = 149	Chan 6 Note on	Note Number (0–127)	Note Velocity (0–127)
10010110 = 96 = 150	Chan 7 Note on	Note Number (0–127)	Note Velocity (0–127)
10010111 = 97 = 151	Chan 8 Note on	Note Number (0–127)	Note Velocity (0–127)
10011000 = 98 = 152	Chan 9 Note on	Note Number (0–127)	Note Velocity (0–127)
10011001 = 99 = 153	Chan 10 Note on	Note Number (0–127)	Note Velocity (0–127)
10011010 = 9A = 154	Chan 11 Note on	Note Number (0–127)	Note Velocity (0–127)
10011011 = 9B = 155	Chan 12 Note on	Note Number (0–127)	Note Velocity (0–127)
10011100 = 9C = 156	Chan 13 Note on	Note Number (0–127)	Note Velocity (0–127)
10011101 = 9D = 157	Chan 14 Note on	Note Number (0–127)	Note Velocity (0–127)
10011110 = 9E = 158	Chan 15 Note on	Note Number (0–127)	Note Velocity (0–127)
10011111 = 9F = 159	Chan 16 Note on	Note Number (0–127)	Note Velocity (0–127)
10100000 = A0 = 160	Chan 1 Polyphonic Aftertouch	Note Number (0–127)	Pressure (0–127)
10100001 = A1 = 161	Chan 2 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10100010 = A2 = 162	Chan 3 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10100011 = A3 = 163	Chan 4 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10100100 = A4 = 164	Chan 5 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10100101 = A5 = 165	Chan 6 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10100110 = A6 = 166	Chan 7 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10100111 = A7 = 167	Chan 8 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10101000 = A8 = 168	Chan 9 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10101001 = A9 = 169	Chan 10 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10101010 = AA = 170	Chan 11 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10101011 = AB = 171	Chan 12 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10101100 = AC = 172	Chan 13 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10101101 = AD = 173	Chan 14 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10101110 = AE = 174	Chan 15 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)

10101111 = AF = 175	Chan 16 Polyphonic Aftertouch	Note Number (0–127	Pressure (0–127)
10110000 = B0 = 176	Chan 1 Control/Mode Change	see Table 3	see Table 3
10110001 = B1 = 177	Chan 2 Control/Mode Change	see Table 3	see Table 3
10110010 = B2 = 178	Chan 3 Control/Mode Change	see Table 3	see Table 3
10110011 = B3 = 179	Chan 4 Control/Mode Change	see Table 3	see Table 3
10110100 = B4 = 180	Chan 5 Control/Mode Change	see Table 3	see Table 3
10110101 = B5 = 181	Chan 6 Control/Mode Change	see Table 3	see Table 3
10110110 = B6 = 182	Chan 7 Control/Mode Change	see Table 3	see Table 3
10110111 = B7 = 183	Chan 8 Control/Mode Change	see Table 3	see Table 3
10111000 = B8 = 184	Chan 9 Control/Mode Change	see Table 3	see Table 3
10111001 = B9 = 185	Chan 10 Control/Mode Change	see Table 3	see Table 3
10111010 = BA = 186	Chan 11 Control/Mode Change	see Table 3	see Table 3
10111011 = BB = 187	Chan 12 Control/Mode Change	see Table 3	see Table 3
10111100 = BC = 188	Chan 13 Control/Mode Change	see Table 3	see Table 3
10111101 = BD = 189	Chan 14 Control/Mode Change	see Table 3	see Table 3
10111110 = BE = 190	Chan 15 Control/Mode Change	see Table 3	see Table 3
10111111 = BF = 191	Chan 16 Control/Mode Change	see Table 3	see Table 3
11000000 = C0 = 192	Chan 1 Program Change	Program # (0–127)	none
11000001 = C1 = 193	Chan 2 Program Change	Program # (0–127)	none
11000010 = C2 = 194	Chan 3 Program Change	Program # (0–127)	none
11000011 = C3 = 195	Chan 4 Program Change	Program # (0–127)	none
11000100 = C4 = 196	Chan 5 Program Change	Program # (0–127)	none
11000101 = C5 = 197	Chan 6 Program Change	Program # (0–127)	none
11000110 = C6 = 198	Chan 7 Program Change	Program # (0–127)	none
11000111 = C7 = 199	Chan 8 Program Change	Program # (0–127)	none
11001000 = C8 = 200	Chan 9 Program Change	Program # (0–127)	none
11001001 = C9 = 201	Chan 10 Program Change	Program # (0–127)	none
11001010 = CA = 202	Chan 11 Program Change	Program # (0–127)	none
11001011 = CB = 203	Chan 12 Program Change	Program # (0–127)	none
11001100 = CC = 204	Chan 13 Program Change	Program # (0–127)	none
11001101 = CD = 205	Chan 14 Program Change	Program # (0–127)	none
11001110 = CE = 206	Chan 15 Program Change	Program # (0–127)	none
11001111 = CF = 207	Chan 16 Program Change	Program # (0–127)	none

(Continued)

Table 12.2 (Continued)

Status Byte		Data Bytes	
1st Byte Value *Binary / Hex / Dec*	*Function*	*2nd Byte*	*3rd Byte*
11010000 = D0 = 208	Chan 1 Channel Aftertouch	Pressure (0–127)	none
11010001 = D1 = 209	Chan 2 Channel Aftertouch	Pressure (0–127)	none
11010010 = D2 = 210	Chan 3 Channel Aftertouch	Pressure (0–127)	none
11010011 = D3 = 211	Chan 4 Channel Aftertouch	Pressure (0–127)	none
11010100 = D4 = 212	Chan 5 Channel Aftertouch	Pressure (0–127)	none
11010101 = D5 = 213	Chan 6 Channel Aftertouch	Pressure (0–127)	none
11010110 = D6 = 214	Chan 7 Channel Aftertouch	Pressure (0–127)	none
11010111 = D7 = 215	Chan 8 Channel Aftertouch	Pressure (0–127)	none
11011000 = D8 = 216	Chan 9 Channel Aftertouch	Pressure (0–127)	none
11011001 = D9 = 217	Chan 10 Channel Aftertouch	Pressure (0–127)	none
11011010 = DA = 218	Chan 11 Channel Aftertouch	Pressure (0–127)	none
11011011 = DB = 219	Chan 12 Channel Aftertouch	Pressure (0–127)	none
11011100 = DC = 220	Chan 13 Channel Aftertouch	Pressure (0–127)	none
11011101 = DD = 221	Chan 14 Channel Aftertouch	Pressure (0–127)	none
11011110 = DE = 222	Chan 15 Channel Aftertouch	Pressure (0–127)	none
11011111 = DF = 223	Chan 16 Channel Aftertouch	Pressure (0–127)	none
11100000 = E0 = 224	Chan 1 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11100001 = E1 = 225	Chan 2 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11100010 = E2 = 226	Chan 3 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11100011 = E3 = 227	Chan 4 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11100100 = E4 = 228	Chan 5 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11100101 = E5 = 229	Chan 6 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11100110 = E6 = 230	Chan 7 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11100111 = E7 = 231	Chan 8 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11101000 = E8 = 232	Chan 9 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11101001 = E9 = 233	Chan 10 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11101010 = EA = 234	Chan 11 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11101011 = EB = 235	Chan 12 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11101100 = EC = 236	Chan 13 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)

11101101 = ED = 237	Chan 14 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11101110 = EE = 238	Chan 15 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11101111 = EF = 239	Chan 16 Pitch Wheel Control	Pitch Wheel LSB (0–127)	Pitch Wheel MSB (0–127)
11110000 = F0 = 240	System Exclusive	**	**
11110001 = F1 = 241	MIDI Time Code Qtr. Frame	–see spec–	–see spec–
11110010 = F2 = 242	Song Position Pointer	LSB	MSB
11110011 = F3 = 243	Song Select (Song #)	(0–127)	none
11110100 = F4 = 244	Undefined (Reserved)	—	—
11110101 = F5 = 245	Undefined (Reserved)	—	—
11110110 = F6 = 246	Tune request	none	none
11110111 = F7 = 247	End of SysEx (EOX)	none	none
11111000 = F8 = 248	Timing clock	none	none
11111001 = F9 = 249	Undefined (Reserved)	—	—
11111010 = FA = 250	Start	none	none
11111011 = FB = 251	Continue	none	none
11111100 = FC = 252	Stop	none	none
11111101 = FD = 253	Undefined (Reserved)	—	—
11111110 = FE = 254	Active Sensing	none	none
11111111 = FF = 255	System Reset	none	none

** Note: System Exclusive (data dump) 2nd byte = Vendor ID (or Universal Exclusive) followed by more data bytes and ending with EOX.

Table 12.3 Control Changes and Mode Changes

(Status Bytes 176–191)

Control Number 2nd Byte Value			*Control Function*	*3rd Byte Value*	
Decimal	*Binary*	*Hex*		*Value*	*Used As*
0	00000000	00	Bank Select	0–127	MSB
1	00000001	01	Modulation Wheel or Lever	0–127	MSB
2	00000010	02	Breath Controller	0–127	MSB
3	00000011	03	Undefined	0–127	MSB

(Continued)

Table 12.3 (Continued)

Control Number 2nd Byte Value			Control Function	3rd Byte Value	
Decimal	*Binary*	*Hex*		*Value*	*Used As*
4	00000100	04	Foot Controller	0–127	MSB
5	00000101	05	Portamento Time	0–127	MSB
6	00000110	06	Data Entry MSB	0–127	MSB
7	00000111	07	Channel Volume (formerly Main Volume)	0–127	MSB
8	00001000	08	Balance	0–127	MSB
9	00001001	09	Undefined	0–127	MSB
10	00001010	0A	Pan	0–127	MSB
11	00001011	0B	Expression Controller	0–127	MSB
12	00001100	0C	Effect Control 1	0–127	MSB
13	00001101	0D	Effect Control 2	0–127	MSB
14	00001110	0E	Undefined	0–127	MSB
15	00001111	0F	Undefined	0–127	MSB
16	00010000	10	General Purpose Controller 1	0–127	MSB
17	00010001	11	General Purpose Controller 2	0–127	MSB
18	00010010	12	General Purpose Controller 3	0–127	MSB
19	00010011	13	General Purpose Controller 4	0–127	MSB
20	00010100	14	Undefined	0–127	MSB
21	00010101	15	Undefined	0–127	MSB
22	00010110	16	Undefined	0–127	MSB
23	00010111	17	Undefined	0–127	MSB
24	00011000	18	Undefined	0–127	MSB
25	00011001	19	Undefined	0–127	MSB
26	00011010	1A	Undefined	0–127	MSB
27	00011011	1B	Undefined	0–127	MSB
28	00011100	1C	Undefined	0–127	MSB
29	00011101	1D	Undefined	0–127	MSB
30	00011110	1E	Undefined	0–127	MSB
31	00011111	1F	Undefined	0–127	MSB

32	00100000	20	LSB for Control 0 (Bank Select)	0–127	LSB
33	00100001	21	LSB for Control 1 (Modulation Wheel or Lever)	0–127	LSB
34	00100010	22	LSB for Control 2 (Breath Controller)	0–127	LSB
35	00100011	23	LSB for Control 3 (Undefined)	0–127	LSB
36	00100100	24	LSB for Control 4 (Foot Controller)	0–127	LSB
37	00100101	25	LSB for Control 5 (Portamento Time)	0–127	LSB
38	00100110	26	LSB for Control 6 (Data Entry)	0–127	LSB
39	00100111	27	LSB for Control 7 (Channel Volume, formerly Main Volume)	0–127	LSB
40	00101000	28	LSB for Control 8 (Balance)	0–127	LSB
41	00101001	29	LSB for Control 9 (Undefined)	0–127	LSB
42	00101010	2A	LSB for Control 10 (Pan)	0–127	LSB
43	00101011	2B	LSB for Control 11 (Expression Controller)	0–127	LSB
44	00101100	2C	LSB for Control 12 (Effect control 1)	0–127	LSB
45	00101101	2D	LSB for Control 13 (Effect control 2)	0–127	LSB
46	00101110	2E	LSB for Control 14 (Undefined)	0–127	LSB
47	00101111	2F	LSB for Control 15 (Undefined)	0–127	LSB
48	00110000	30	LSB for Control 16 (General Purpose Controller 1)	0–127	LSB
49	00110001	31	LSB for Control 17 (General Purpose Controller 2)	0–127	LSB
50	00110010	32	LSB for Control 18 (General Purpose Controller 3)	0–127	LSB
51	00110011	33	LSB for Control 19 (General Purpose Controller 4)	0–127	LSB
52	00110100	34	LSB for Control 20 (Undefined)	0–127	LSB
53	00110101	35	LSB for Control 21 (Undefined)	0–127	LSB
54	00110110	36	LSB for Control 22 (Undefined)	0–127	LSB
55	00110111	37	LSB for Control 23 (Undefined)	0–127	LSB
56	00111000	38	LSB for Control 24 (Undefined)	0–127	LSB
57	00111001	39	LSB for Control 25 (Undefined)	0–127	LSB
58	00111010	3A	LSB for Control 26 (Undefined)	0–127	LSB
59	00111011	3B	LSB for Control 27 (Undefined)	0–127	LSB
60	00111100	3C	LSB for Control 28 (Undefined)	0–127	LSB
61	00111101	3D	LSB for Control 29 (Undefined)	0–127	LSB
62	00111110	3E	LSB for Control 30 (Undefined)	0–127	LSB
63	00111111	3F	LSB for Control 31 (Undefined)	0–127	LSB

(*Continued*)

Table 12.3 (Continued)

Control Number 2nd Byte Value			*Control Function*	*3rd Byte Value*	
Decimal	*Binary*	*Hex*		*Value*	*Used As*
64	01000000	40	Damper Pedal on/off (Sustain)	≤63 off, ≥64 on	–
65	01000001	41	Portamento On/Off	≤63 off, ≥64 on	–
66	01000010	42	Sostenuto On/Off	≤63 off, ≥64 on	–
67	01000011	43	Soft Pedal On/Off	≤63 off, ≥64 on	–
68	01000100	44	Legato Footswitch	≤63 Normal, ≥64 Legato	–
69	01000101	45	Hold 2	≤63 off, ≥64 on	–
70	01000110	46	Sound Controller 1 (default: Sound Variation)	0–127	LSB
71	01000111	47	Sound Controller 2 (default: Timbre/Harmonic Intens.)	0–127	LSB
72	01001000	48	Sound Controller 3 (default: Release Time)	0–127	LSB
73	01001001	49	Sound Controller 4 (default: Attack Time)	0–127	LSB
74	01001010	4A	Sound Controller 5 (default: Brightness)	0–127	LSB
75	01001011	4B	Sound Controller 6 (default: Decay Time—see MMA RP-021)	0–127	LSB
76	01001100	4C	Sound Controller 7 (default: Vibrato Rate—see MMA RP-021)	0–127	LSB
77	01001101	4D	Sound Controller 8 (default: Vibrato Depth—see MMA RP-021)	0–127	LSB
78	01001110	4E	Sound Controller 9 (default: Vibrato Delay—see MMA RP-021)	0–127	LSB
79	01001111	4F	Sound Controller 10 (default undefined—see MMA RP-021)	0–127	LSB
80	01010000	50	General Purpose Controller 5	0–127	LSB
81	01010001	51	General Purpose Controller 6	0–127	LSB
82	01010010	52	General Purpose Controller 7	0–127	LSB
83	01010011	53	General Purpose Controller 8	0–127	LSB
84	01010100	54	Portamento Control	0–127	LSB
85	01010101	55	Undefined	–	–
86	01010110	56	Undefined	–	–
87	01010111	57	Undefined	–	–

88	01011000	58	High Resolution Velocity Prefix	0–127	LSB
89	01011001	59	Undefined	—	—
90	01011010	5A	Undefined	—	—
91	01011011	5B	Effects 1 Depth	0–127	—
			(default: Reverb Send Level—see MMA RP-023) (formerly External Effects Depth)		
92	01011100	5C	Effects 2 Depth (formerly Tremolo Depth)	0–127	—
93	01011101	5D	Effects 3 Depth	0–127	—
			(default: Chorus Send Level—see MMA RP-023) (formerly Chorus Depth)		
94	01011110	5E	Effects 4 Depth (formerly Celeste [Detune] Depth)	0–127	—
95	01011111	5F	Effects 5 Depth (formerly Phaser Depth)	0–127	—
96	01100000	60	Data Increment (Data Entry +1) (see MMA RP-018)	N/A	—
97	01100001	61	Data Decrement (Data Entry −1) (see MMA RP-018)	N/A	—
98	01100010	62	Non-Registered Parameter Number (NRPN)—LSB	0–127	LSB
99	01100011	63	Non-Registered Parameter Number (NRPN)—MSB	0–127	MSB
100	01100100	64	Registered Parameter Number (RPN)—LSB*	0–127	LSB
101	01100101	65	Registered Parameter Number (RPN)—MSB*	0–127	MSB
102	01100110	66	Undefined	—	—
103	01100111	67	Undefined	—	—
104	01101000	68	Undefined	—	—
105	01101001	69	Undefined	—	—
106	01101010	6A	Undefined	—	—
107	01101011	6B	Undefined	—	—
108	01101100	6C	Undefined	—	—
109	01101101	6D	Undefined	—	—
110	01101110	6E	Undefined	—	—
111	01101111	6F	Undefined	—	—

(Continued)

Table 12.3 (Continued)

Control Number 2nd Byte Value			Control Function	3rd Byte Value	
Decimal	*Binary*	*Hex*		*Value*	*Used As*
112	01110000	70	Undefined	–	–
113	01110001	71	Undefined	–	–
114	01110010	72	Undefined	–	–
115	01110011	73	Undefined	–	–
116	01110100	74	Undefined	–	–
117	01110101	75	Undefined	–	–
118	01110110	76	Undefined	–	–
119	01110111	77	Undefined	–	–
Note: Controller numbers 120–127 are reserved for Channel Mode Messages, which rather than controlling sound parameters, affect the channel's operating mode. (See also Table 12.1.)					
120	01111000	78	[Channel Mode Message] All Sound Off	0	–
121	01111001	79	[Channel Mode Message] Reset All Controllers (See MMA RP-015)	0	–
122	01111010	7A	[Channel Mode Message] Local Control On/Off	0 off, 127 on	–
123	01111011	7B	[Channel Mode Message] All Notes Off	0	–
124	01111100	7C	[Channel Mode Message] Omni Mode Off (+ all notes off)	0	–
125	01111101	7D	[Channel Mode Message] Omni Mode On (+ all notes off)	0	–
126	01111110	7E	[Channel Mode Message] Mono Mode On (+ poly off, + all notes off)	0	–
Note: This equals the number of channels, or zero if the number of channels equals the number of voices in the receiver.					
127	01111111	7F	[Channel Mode Message] Poly Mode On (+ mono off, + all notes off)	0	–

Table 12.3a: Registered Parameter Numbers

To set or change the value of a Registered Parameter:

1. Send two Control Change messages using Control Numbers 101 (65H) and 100 (64H) to select the desired Registered Parameter Number, as per the following table.
2. To set the selected Registered Parameter to a specific value, send a Control Change messages to the Data Entry MSB controller (Control Number 6). If the selected Registered Parameter requires the LSB to be set, send another Control Change message to the Data Entry LSB controller (Control Number 38).
3. To make a relative adjustment to the selected Registered Parameter's current value, use the Data Increment or Data Decrement controllers (Control Numbers 96 and 97).

Parameter Number		*Parameter Function*	*Data Entry Value*
MSB: Control 101 (65H) Value	*LSB: Control 100 (64H) Value*		
00H	00H	Pitch Bend Sensitivity	MSB = +/− semitones LSB = +/− cents
	01H	Channel Fine Tuning (formerly Fine Tuning—see MMA RP-022)	Resolution 100/8192 cents 00H 00H = −100 cents 40H 00H = A440 7FH 7FH = +100 cents
	02H	Channel Coarse Tuning (formerly Coarse Tuning—see MMA RP-022)	Only MSB used Resolution 100 cents 00H = −6400 cents 40H = A440 7FH = +6300 cents
	03H	Tuning Program Change	Tuning Program Number
	04H	Tuning Bank Select	Tuning Bank Number
	05H	Modulation Depth Range (see MMA General MIDI Level 2 Specification)	For GM2, defined in GM2 Specification. For other systems, defined by manufacturer
. . .	. . .	All RESERVED for future MMA Definition	. . .
		Three Dimensional Sound Controllers	
3DH (61)	00H	AZIMUTH ANGLE	See RP-049
	01H	ELEVATION ANGLE	See RP-049
	02H	GAIN	See RP-049
	03H	DISTANCE RATIO	See RP-049

(*Continued*)

Table 12.3 (Continued)

Parameter Number		Parameter Function	Data Entry Value
MSB: Control 101 (65H) Value	*LSB: Control 100 (64H) Value*		
	04H	MAXIMUM DISTANCE	See RP-049
	05H	GAIN AT MAXIMUM DISTANCE	See RP-049
	06H	REFERENCE DISTANCE RATIO	See RP-049
	07H	PAN SPREAD ANGLE	See RP-049
	08H	ROLL ANGLE	See RP-049
. . .	. . .	All RESERVED for future MMA Definition	. . .
7FH	7FH	Null Function Number for RPN/NRPN	Setting RPN to 7FH, 7FH will disable the data entry, data increment, and data decrement controllers until a new RPN or NRPN is selected.

Table 12.4 Defined Universal System Exclusive Messages

Non-Real Time (7EH)		Description
Sub-ID #1	*Sub-ID #2*	
00		Unused
01		Sample Dump Header
02		Sample Data Packet
03		Sample Dump Request
04	nn	MIDI Time Code
	00	Special
	01	Punch In Points
	02	Punch Out Points
	03	Delete Punch In Point
	04	Delete Punch Out Point
	05	Event Start Point
	06	Event Stop Point
	07	Event Start Points with additional info.

	08	Event Stop Points with additional info.
	09	Delete Event Start Point
	0A	Delete Event Stop Point
	0B	Cue Points
	0C	Cue Points with additional info.
	0D	Delete Cue Point
	0E	Event Name in additional info.
05	nn	Sample Dump Extensions
	01	Loop Points Transmission
	02	Loop Points Request
	03	Sample Name Transmission
	04	Sample Name Request
	05	Extended Dump Header
	06	Extended Loop Points Transmission
	07	Extended Loop Points Request
06	nn	General Information
	01	Identity Request
	02	Identity Reply
07	nn	File Dump
	01	Header
	02	Data Packet
	03	Request
08	nn	MIDI Tuning Standard (Non-Real Time)
	00	Bulk Dump Request
	01	Bulk Dump Reply
	03	Tuning Dump Request
	04	Key-Based Tuning Dump
	05	Scale/Octave Tuning Dump, 1 byte format
	06	Scale/Octave Tuning Dump, 2 byte format
	07	Single Note Tuning Change with Bank Select
	08	Scale/Octave Tuning, 1 byte format
	09	Scale/Octave Tuning, 2 byte format
09	nn	General MIDI
	01	General MIDI 1 System On
	02	General MIDI System Off
	03	General MIDI 2 System On

(*Continued*)

Table 12.4 (Continued)

Non-Real Time (7EH)		*Description*
Sub-ID #1	*Sub-ID #2*	
0A	nn	Downloadable Sounds
	01	Turn DLS On
	02	Turn DLS Off
	03	Turn DLS Voice Allocation Off
	04	Turn DLS Voice Allocation On
0B	nn	File Reference Message
	00	reserved (do not use)
	01	Open File
	02	Select or Reselect Contents
	03	Open File and Select Contents
	04	Close File
	05-7F	reserved (do not use)
0C	nn	MIDI Visual Control
	00-7F	MVC Commands (*See MVC Documentation*)
7B	–	End of File
7C	–	Wait
7D	–	Cancel
7E	–	NAK
7F	–	ACK

Real Time (7FH)		*Description*
Sub-ID #1	*Sub-ID #2*	
00	–	Unused
01	nn	MIDI Time Code
	01	Full Message
	02	User Bits
02	nn	MIDI Show Control
	00	MSC Extensions
	01-7F	MSC Commands (*see MSC Documentation*)

03	nn	Notation Information
	01	Bar Number
	02	Time Signature (Immediate)
	42	Time Signature (Delayed)
04	nn	Device Control
	01	Master Volume
	02	Master Balance
	03	Master Fine Tuning
	04	Master Course Tuning
	05	Global Parameter Control
05	nn	Real Time MTC Cueing
	00	Special
	01	Punch In Points
	02	Punch Out Points
	03	(Reserved)
	04	(Reserved)
	05	Event Start points
	06	Event Stop points
	07	Event Start points with additional info.
	08	Event Stop points with additional info.
	09	(Reserved)
	0A	(Reserved)
	0B	Cue points
	0C	Cue points with additional info.
	0D	(Reserved)
	0E	Event Name in additional info.
06	nn	MIDI Machine Control Commands
	00-7F	MMC Commands (*See MMC Documentation*)
07	nn	MIDI Machine Control Responses
	00-7F	MMC Responses (*See MMC Documentation*)
08	nn	MIDI Tuning Standard (Real Time)
	02	Single Note Tuning Change
	07	Single Note Tuning Change with Bank Select
	08	Scale/Octave Tuning, 1 byte format
	09	Scale/Octave Tuning, 2 byte format

(*Continued*)

Table 12.4 (Continued)

Real Time (7FH)		*Description*
Sub-ID #1	*Sub-ID #2*	
09	nn	Controller Destination Setting (*See GM2 Documentation*)
	01	Channel Pressure (Aftertouch)
	02	Polyphonic Key Pressure (Aftertouch)
	03	Controller (Control Change)
0A	01	Key-based Instrument Control
0B	01	Scalable Polyphony MIDI MIP Message
0C	00	Mobile Phone Control Message

Section 5—MIDI Specification Summaries

General MIDI 1, 2, and Lite Specifications

General MIDI (GM1)

"General MIDI" is not the same as MIDI. Where MIDI is a language, a file format, and a connector specification, the "General MIDI System Level 1" specification—also known as "GM," "General MIDI 1" and "GM1"—defines specific features of a MIDI instrument. Without General MIDI, playback of MIDI files created on one MIDI instrument might sound totally different on a different MIDI instrument, because the only sound definition in MIDI is a sound number (Program Number), not the actual characteristics of the sound. GM helped establish a consistent level of performance compatibility among MIDI instruments by establishing that specific sounds should be selected by certain Program Numbers, paving the way for MIDI data to be used in computer games and cell phones.

Note: The General MIDI (GM1) specification was superceded in 1999 by General MIDI 2, which supports additional features and capabilities commonly available. However, GM1 remains a popular format for lower-cost synthesizers and is commonly used for music distributed as Standard MIDI Files. General MIDI Lite is an even lower-cost version of GM developed specifically for use in low-power portable devices such as cell phones.

GM1 Features

To be GM1 compatible, a GM1 sound-generating device (keyboard, sound module, sound card, IC, software program, or other product) must meet the General MIDI System Level 1 performance requirements outlined below, instantaneously upon demand, and without additional modification or adjustment/configuration by the user.

- Voices: A minimum of either 24 fully dynamically allocated voices are available simultaneously for both melodic and

percussive sounds, or 16 dynamically allocated voices are available for melody plus 8 for percussion. All voices respond to velocity.
- Channels: All 16 MIDI Channels are supported. Each Channel can play a variable number of voices (polyphony). Each Channel can play a different instrument (sound/patch/timbre). Key-based percussion is always on MIDI Channel 10.
- Instruments: A minimum of 16 simultaneous and different timbres playing various instruments. A minimum of 128 preset instruments (MIDI program numbers) conforming to the GM1 Instrument Patch Map and 47 percussion sounds which conform to the GM1 Percussion Key Map.
- Channel Messages: Support for continuous controllers 1, 7, 10, 11, 64, 121, and 123; RPN #s 0, 1, 2; Channel Pressure, Pitch Bend.
- Other Messages: Respond to the data entry controller and the RPNs for fine and course tuning and Pitch Bend range, as well as all General MIDI Level 1 System Messages.

GM1 Developer Information

The MMA's GM Developer Guidelines document describes additional recommendations and clarifications of the GM Specification for content producers and device makers, to ensure improved compatibility among GM products.

The GM1 Logo was created to ensure consumer recognition for products that meet the General MIDI Level 1 Specification. The GM Logo is the property of the MMA and AMEI and must be used in accordance with guidelines established to ensure the value of the GM Logo for our members and for the consumer.

General MIDI 2 (GM2)

General MIDI 1 made great strides in the music industry by providing a platform for compatibility between device manufacturers and content providers. Still, by 1999 many manufacturers felt there needed to be additional functionality. General MIDI 2 (GM2) is a group of extensions made to General MIDI 1, which increases both the number of available sounds and the amount of control available for sound editing and musical performance. All GM2 devices are also fully compatible with General MIDI 1.

New MIDI Messages

To support new features in GM2 devices, the MIDI specification was also extended with numerous new control messages, include MIDI Tuning, Controllers, RPNs, and Universal System Exclusive Messages. Of particular significance are the new Universal System Exclusive Messages, including Controller Destination Setting, Key-Based Instrument Controllers, Global Parameter Control, and Master Fine/Coarse Tuning.

- Controller Destination SysEx Message (.pdf)
- Key-Based Instrument Controller SysEx Message (.pdf)
- Global Parameter Control SysEx Message (.pdf)
- Master Fine/Course Tuning SysEx Message (.pdf)
- Redefinition of RPN01 and RPN02 (Channel Fine/Course Tuning)
- RPN05 Modulation Depth Range (.pdf)

GM2 Specification Update 1.1

In September 2003 a new version of the General MIDI 2 Specification document was made available, reflecting changes to the specification mandated by two new Recommended Practices:

- RP-036: Sets a default Pan Curve for future AMEI/MMA specifications (equivalent to the Pan Curve defined in GML and DLS) and amends GM2 to include this curve.
- RP-037: Adds a recommendation that GM 2 devices support the MIDI Tuning

Extension "Scale/Octave Tuning Real Time One-Byte form" message.

GM2 Specification Update 1.2

In 2007 Recommend Practice RP-044 was adopted for implementing the Mod Depth Range RPN message on GM2 devices. This recommendation does not apply to other device specifications (i.e. SP-MIDI) that refer to GM 2. For those devices, the recommended response to Mod Depth Range RPN is either undefined or should be defined in those specifications.

GM2 Features

- Requirements
- Number of Notes: 32 simultaneous notes
- MIDI Channels: 16
    - Simultaneous Melodic Instruments = up to 16 (all Channels)
    - Simultaneous Percussion Kits = up to 2 (Channel 10/11)
- Control Change Messages (Some Optional)
    - Bank Select (cc#0/32)
    - Modulation Depth (cc#1)
    - Portamento Time (cc#5)
    - Channel Volume (cc#7)
    - Pan (cc#10)
    - Expression (cc#11)
    - Hold1 (Damper) (cc#64)
    - Portamento ON/OFF (cc#65)
    - Sostenuto (cc#66)
    - Soft (cc#67)
    - Filter Resonance (Timbre/Harmonic Intensity) (cc#71)
    - Release Time (cc#72)
    - Attack time (cc#73)
    - Brightness (cc#74)
    - Decay Time (cc#75) (new message)
    - Vibrato Rate (cc#76) (new message)
    - Vibrato Depth (cc#77) (new message)
    - Vibrato Delay (cc#78) (new message)
    - Reverb Send Level (cc#91)
    - Chorus Send Level (cc#93)
    - Data Entry (cc#6/38)
    - RPN LSB/MSB (cc#100/101)
- Registered Parameter Numbers
    - Pitch Bend Sensitivity
    - Channel Fine Tune
    - Channel Coarse Tune

- Modulation Depth Range (Vibrato Depth Range)
- RPN NULL
- Universal System Exclusive Messages
  - Master Volume
  - Master Fine Tuning
  - Master Coarse Tuning
  - Reverb Type
  - Reverb Time
  - Chorus Type
  - Chorus Mod Rate
  - Chorus Mod Depth
  - Chorus Feedback
  - Chorus Send to Reverb
  - Controller Destination Setting
  - Scale/Octave Tuning Adjust
  - Key-Based Instrument Controllers
  - GM2 System On
- GM2 Instrument Sound Set
- GM2 Percussion Sound Set

GM2 Developer Information

Developers of GM2 compatible devices or content are urged to consult the GM1 Developer Guidelines (included in the Complete MIDI 1.0 Specification) that describe recommendations for content producers and device makers to ensure improved compatibility among GM products.

The GM2 Logo was created to ensure consumer recognition for products that meet the General MIDI 2 Specification. The GM Logos are the property of the MMA and AMEI and must be used in accordance with guidelines established to ensure the value of the GM Logos for our members and for the consumer.

General MIDI "Lite" (GML)

plus Guidelines for Mobile Applications

The General MIDI Lite Specification describes one of two platforms for mobile MIDI communication that have been approved and adopted as a standard by the MIDI industry. This document has three primary components:

- A specification called General MIDI Lite (GM Lite), which defines a new level of tone generation (sound module) device;
- Authoring guidelines for music data in SMF (Standard MIDI File) format that is intended for playback on GM Lite devices;
- Implementation guidelines for GM Lite file players.

GM Lite vs. SP-MIDI

The General MIDI Lite Specification defines a fixed-polyphony MIDI device, intended to meet a particular set of current and future market needs. The Scalable Polyphony MIDI (SP-MIDI) Specification complements GML by defining flexible polyphony MIDI devices and content. Developers of GM Lite players are strongly advised to keep as much flexibility as possible in how their players handle channel priorities, drum channels, and other System messages. This will make it far easier for their products to be compatible with song data authored for the Scalable Polyphony MIDI specification.

GM Lite vs. GM1

The General MIDI Lite device specification is intended for equipment that does not have the capability to support the full feature set defined in General MIDI 1.0, on the assumption that the reduced performance may be acceptable (and even required) in some mobile applications. GM Lite represents just one standardized set of performance capabilities for portable applications—other performance levels are likely to be standardized in the future.

GM Lite Features

- Requirements
- Number of Notes: 16 simultaneous notes
- MIDI Channels: 16
    - Simultaneous Melodic Instruments = up to 15
    - Simultaneous Percussion Kits = 1 (Channel 10)

- Control Change Messages
  - Modulation Depth (cc#1)
  - Channel Volume (cc#7)
  - Pan (cc#10)
  - Expression (cc#11)
  - Data Entry (cc#6/38)
  - Hold1 (Damper) (cc#64)
  - RPN LSB/MSB (cc#100/101)
  - Pitch Bend
  - All Sound Off, All Notes Off, Reset All Controllers
- Registered Parameter Numbers
  - Pitch Bend Sensitivity
- Universal System Exclusive Messages
  - GM1 System On
- GM Lite Instrument Sound Set
- GM Lite Percussion Sound Set

For complete details on GML features and MIDI message syntax, please consult the General MIDI Lite Specification (see below to order).

GM Lite Developer Information

Recommended guidelines for Using GM Lite in Mobile Applications are included in the GM Lite Specification document. Developers are also urged to consult the GM1 Developer Guidelines (included in the Complete MIDI 1.0 Specification).

The GML Logo was created to ensure consumer recognition for products that meet the General MIDI Lite Specification. The GM Logos are the property of the MMA and AMEI and must be used in accordance with guidelines established to ensure the value of the GM Logos for our members and for the consumer.

INDEX

Page numbers in italic format indicate figures and tables.

AUDIO

In the last five years the environment in which the Musical Instrument Digital Interface (MIDI) specification works and the tools that communicate via MIDI have changed dramatically. *Modern MIDI: Sequencing and Performing Using Traditional and Mobile Tools* gives you all the tools you need to properly and effectively use MIDI in a modern setting, while still incorporating vintage MIDI gear. Exploring typical workflows and techniques for both the studio and the performing environment, this book helps you navigate the changes that mobile computing has made to the way that music producers and engineers work with MIDI.

If you're a MIDI user seeking to increase your efficiency and productivity while still gaining an understanding of the fundamentals of MIDI, or a music professional looking to incorporate your mobile devices into your creative process, this is the book for you. *Modern MIDI* shows you how to implement the necessary components to use MIDI on your iPad, Android phone, or laptop.

- The first book on MIDI covering today's mobile application and usage
- A revamped approach to explaining commonly used music technology, covering mobile technology in the studio and on stage
- Written by a professor at one of the most established music technology programs in the USA with an approach that has been forged in the classroom and refined through the use of MIDI on professional projects

SAM McGUIRE, an active audio engineer, composer, and faculty member of the University of Colorado Denver, is the primary author of two audio technology books and has worked on multiple audio software instructional videos for VTC.com and CreativeCow.com. Sam has also scored and mixed multiple award-winning documentary and feature films.

Cover image: © Getty

Focal Press
Taylor & Francis Group
www.focalpress.com

COMPANION WEBSITE
http://modernmidi.com

an informa business
ISBN 978-0-415-83927-3
9 780415 839273

Focal titles are available as eBook editions in a range of digital formats