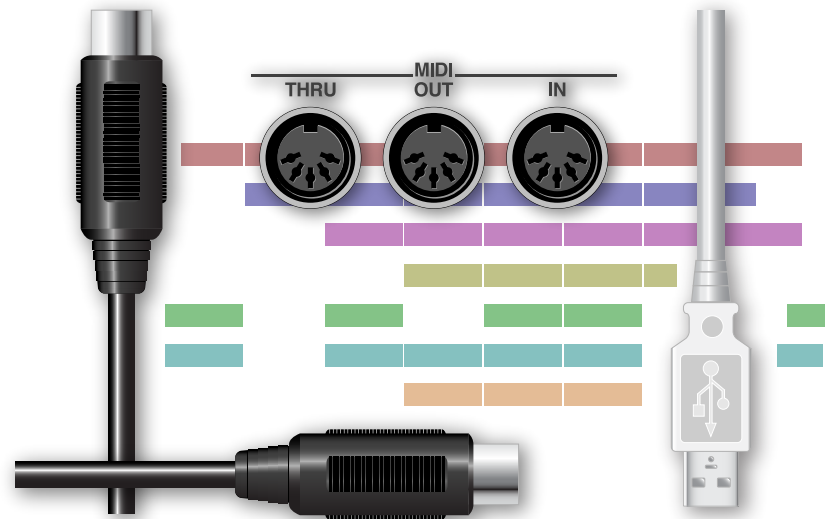


An Introduction to MIDI



© 2006, 2009 Roland Corporation U.S.,
© 2009 MIDI Manufacturers Association

All rights reserved. No part of this publication may be reproduced in any form without the written permission of Roland Corporation U.S. and the MIDI Manufacturers Association.

About This Booklet

This booklet provides important basic background information that can help you get the most out of your MIDI products.

While there's always plenty to say about any one instrument, there are some topics that apply to a wide range of instruments.

For information about a specific product, please check out the websites of the individual companies that make MIDI equipment.

Electronic musical instruments offer an incredible array of sounds to professional and amateur musicians. They place at a musician's fingertips everything from traditional, real-world sounds to noises that have never been heard before.

They come in all shapes and sizes, too, from solid, physical instruments such as keyboards, beat boxes, and electronic drum kits, to computer-based "virtual" instruments, or "soft synths."



All of these devices rely on something called "MIDI." You really can't get too far with an electronic instrument without bumping into MIDI. This is actually a good thing, though, since MIDI allows you to do all sorts of exciting musical things.

Of course, it helps to know what MIDI is—that's what this booklet's for. Once you understand the basics, you can consult the documentation for your device to learn more about how it uses MIDI.

Understanding the Symbols in This Booklet

Throughout this booklet, you'll come across information that deserves special attention—that's the reason it's labeled with one of the following symbols.



A note is something that adds information about the topic at hand.



A tip offers suggestions for using the feature being discussed.



Warnings contain important information that can help you avoid possible damage to your equipment, your data, or yourself.

Understanding MIDI

What MIDI Isn't

Yes, it's an odd place to start, but one of the things that confuses people new to MIDI is that they assume MIDI is some type of sound, or audio, that travels through MIDI cables. It's not.

What MIDI Is

MIDI, which stands for "Musical Instrument Digital Interface," is a system that allows electronic musical instruments and computers to send instructions to each other. It sounds simple, but MIDI provides some profound creative opportunities.

Why MIDI Matters

Here are just some of the things MIDI makes possible:

- You can use a MIDI instrument with which you're comfortable to play the sounds belonging to any other MIDI device, including soft synths.
- Create rich musical textures by layering sounds from multiple MIDI devices, or assign different sounds to play in different pitch ranges.
- When you play a MIDI instrument, it produces data that can be captured by a MIDI "sequencer." When the sequencer plays back the data, your original performance is recreated. Sequencers aren't just MIDI recorders, though—they let you fix mistakes, change the pitches of your notes, fix their timing, the way they play, the sounds they use, and more. In addition, most sequencers can capture multiple performances on separate "tracks," allowing you to build up full musical arrangements, one track at a time.
- The General MIDI sound set allows you to instantly play back pre-recorded music on many MIDI instruments, or on a computer.
- One MIDI device can control another, letting you use the most convenient tool for any job. Edit MIDI instrument sounds on your computer, operate an audio recorder from a MIDI beat box, and on and on.

We'll discuss all of these things later in this booklet.

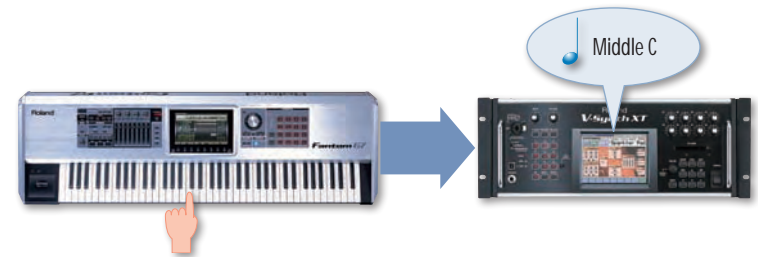
What MIDI Does

When one MIDI device communicates with another, it's all about instructions: It's just one MIDI device telling another one what to do. Here's how playing a note—the most common MIDI activity—happens.

Suppose you've got a MIDI keyboard connected to a MIDI synthesizer box, or "module," and you want to play the module from the keyboard.



When you hold down the Middle C on the keyboard, the keyboard sends out a Middle C "key down" MIDI message to the module. The module says to itself, "Ah, right, Middle C," and plays Middle C.



When you let go of Middle C, the keyboard sends a Middle C "note off" message to the module, and the module stops playing its Middle C note.

Note-on and note-off messages are just the beginning, though. Most MIDI controllers can also sense how hard you play a note, transmitting a MIDI "velocity" value to the receiving device so the volume or brightness of its note responds to your touch.

And, of course, you'll usually play more than one note at a time.

We'll discuss other types of MIDI messages a bit later, and explain how they're put together.

A World of MIDI Controllers



A MIDI device that controls another is referred to as a MIDI "controller." Though most people think of MIDI as being just for keyboard players, a wide range of MIDI controllers and control surfaces make it easy for all sorts of players and engineers to harness the power of MIDI.

Let's talk about some of the available controllers just to give you some idea of the possibilities.

Guitar Controllers

A steel-stringed guitar with a MIDI pickup or a dedicated MIDI guitar synthesizer can turn a guitar into a full-fledged MIDI controller of exceptional expressiveness.



Whether you're playing the guitar synth's own sounds or external MIDI sounds, standard guitar techniques can make MIDI sounds come alive.

Drum Controllers



Electronic drum sets offer a totally natural playing experience along with incredible sounds, all without the hassles of a traditional kit. You can practice on headphones all night if you want, or record drum tracks straight into a MIDI sequencer from a drum kit's pad and cymbal controllers.

There are smaller drum controllers you can play with sticks, or even with your hands. You can also control MIDI devices from traditional acoustic drums by installing MIDI drum triggers.



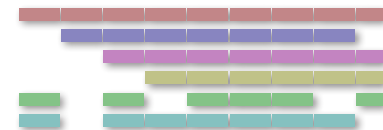
Other Controllers

There are a number of other instruments and devices that use MIDI. There are specially made MIDI wind controllers, mallet controllers, and even MIDI accordions. Since MIDI was primarily designed for keyboard players, these devices are often referred to as "alternate controllers".

Another category of alternate MIDI controllers are those that don't mirror traditional musical instruments. These are simple switches and controls that can be used in items as various as clothing (gloves, suits, mats), to objects that can be hit, strummed, blown, or even controlled by a light beam.

MIDI devices can not only be used to trigger note information, but they can also send control information to other devices, such as lighting controllers.

The Power of MIDI Sequencers



A MIDI sequencer can provide a tremendous amount of fun for the amateur musician, and it's a powerful tool in the hands of a professional. In fact, sequencers play an important role in many modern commercial recordings.

During recording, the sequencer captures and plays back live MIDI performances. Performances can also be constructed slowly, note-by-note, using a variety of methods that may include "step sequencing" and onscreen pencil tools that let you "draw" the notes you want.

Each recorded performance is typically stored on its own track, and a sequencer may offer anywhere from 16 tracks to over a hundred.



Remember, too, that a captured MIDI performance is just a set of instructions for recreating the performance. Since it doesn't contain any actual sound, you can choose the best sound for a performance even *after* it's been recorded.

A sequencer may let you view notes in a variety of ways, from a list of MIDI events, to a piano-roll-type view, to onscreen notation. (Many sequencers also allow you to print the notation you see onscreen.)



The capturing of MIDI notes is just the beginning, since a sequencer allows you to do all sorts of things to perfect your music. Some of the most commonly used sequencer tools are:

- *quantization*—that corrects the timing of notes.
- *transposition*—that moves notes to new musical keys.
- *scaling*—that changes the feel of recorded musical phrases by adjusting recorded velocity values, note lengths, and more.

Some instruments contain a built-in sequencer, giving you sounds and sequencing in a single “workstation.” There are also numerous computer-based sequencing programs and standalone hardware sequencers.



Some sequencers can record audio in addition to MIDI, allowing you to work on all of the elements in a song at the same time. Some MIDI instruments also provide this capability, as do computer-based sequencing programs called “DAWs,” an acronym for “Digital Audio Workstations.”

Standard MIDI Files, or “SMFs”

Whether a sequencer’s in a workstation, on a computer, or a standalone device, each song’s data has to be stored as a file somewhere, somehow. Each manufacturer has their own type of sequencer file that works for their

products’ features, resulting in files that only their sequencers can play. This presents a roadblock for musicians who want to collaborate or otherwise share sequencer files.

The solution is the Standard MIDI File, or “SMF,” format. This standardized sequence file format allows music to travel from one sequencer to another, since most modern sequencers can play SMFs, and store songs as SMFs. An SMF may not contain all of the bells and whistles that make any one sequencer unique, but it supports the features that are common to all sequencers, and this makes them extremely handy.



Whatsit.mid

All SMF files have “.mid” at the end of their name.

Two Types of SMFs

SMFs come in two flavors:

- *Type 0 SMFs*—contain a single sequencer track that holds recorded performances for up to 16 different sounds.
- *Type 1 SMFs*—can contain multiple sequencer tracks.

MIDI and Video

MIDI doesn’t have to be simply about music. Some MIDI devices allow performers—called “VJs”—to manipulate video images onstage, creating exciting visuals. Special software on a laptop computer along with DJ-style physical controls open up a world of video possibilities, letting VJs remotely select clips and control how they behave.



Also, some keyboards have a button that, when pressed, instantly reconfigures the keyboard’s knobs, sliders, and buttons as controllers for an attached video playback system.

General MIDI and General MIDI 2

General MIDI, or “GM,” is very specific set of standards that allows MIDI composers and arrangers to create music that always plays correctly on any device that supports General MIDI.



Why Do We Need GM?

MIDI instruments store their sounds in numbered memory locations, and each manufacturer can use any location for any sound. However, if a composer’s MIDI recording is played by someone who has different equipment, there’s no way for the composer to know where the required sounds are.

The GM Solution

General MIDI solves this since all General MIDI devices contain the same set of 128 standard sounds and drum kit sounds, stored in a specified order. Each product creates these sounds using its own unique capabilities, but the goal is to have them all sound similar enough when playing back GM data.



The General MIDI sounds originally came from Roland’s SoundCanvas sound module. These sounds did such a good job of meeting musicians’ needs they became the foundation of the General MIDI standard.



Instruments that support GM may also have lots of other sounds stored elsewhere in their memories, but these sounds can’t be used in General MIDI recordings.

General MIDI also specifies certain MIDI messages for controlling the behavior of sounds and effects.

GM recordings are always SMF files.

What GM Makes Possible

There are lots of GM recordings on the Internet, and you can also purchase them from a variety of sources. These easy-to-use files can be great for singing or playing along with, allowing you to enjoy MIDI music without having to deal with any of its complexities.

Most computers ship with a set of General MIDI sounds, and some computer games use them. Often, all you need to do is double-click a GM file and hit Play on a computer. GM-enabled MIDI instruments may also require little or no effort on the user’s part— you often simply select a GM song and press PLAY.

General MIDI 2

General MIDI 2 includes everything in General MIDI 1, adding more sounds, standards for sound editing, and some other niceties. If a device supports GM 2, it sports the GM 2 logo.



MIDI Connections

MIDI Cables



MIDI devices are typically connected to each other using MIDI cables. All MIDI cables use the same wiring and have the same type of 5-pin-DIN-type connector on each end.



A MIDI cable’s connector



Some instruments offer MIDI communication with a computer using a standard USB cable instead of a MIDI cable, as we’ll see.



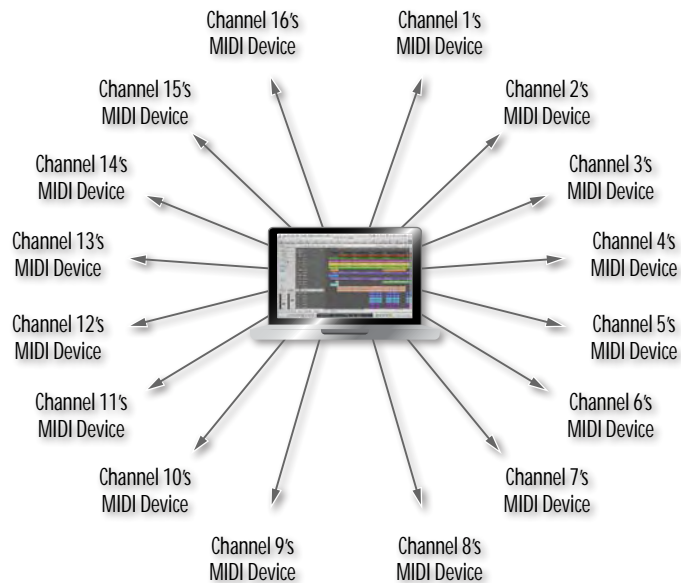
Many instruments also use MIDI on the inside as a way for one internal component to talk to another. These private conversations happen under-the-hood, using internal MIDI wiring you obviously don’t see.

MIDI Channels

A single MIDI connection can carry 16 independent streams, or “channels,” of MIDI information. The ability to have 16 separate MIDI conversations going on at the same time allows a couple of important things.

One MIDI Device Can Control All of Your Other MIDI Devices

By setting up each MIDI device to listen only to MIDI instructions on a particular MIDI channel, you can control a whole roomful of MIDI devices from a single central MIDI device, such as a controller or a computer.



You Can Play Multiple Sounds in a Single MIDI Device

“Multitimbral” MIDI devices can play multiple sounds at the same time. Each sound responds to MIDI instructions received on a specified MIDI channel, whether they’re coming live from a controller or being played back by a sequencer track.



<i>MIDI Channel 1</i> Music Bells	<i>MIDI Channel 5</i> UltimatGrand	<i>MIDI Channel 9</i> Stage EP	<i>MIDI Channel 13</i> Purple Organ
<i>MIDI Channel 2</i> Full Strings	<i>MIDI Channel 6</i> Strat Gtr	<i>MIDI Channel 10</i> Rock Kit 1	<i>MIDI Channel 14</i> Epic Lead
<i>MIDI Channel 3</i> MC-404 Bass	<i>MIDI Channel 7</i> Wired Synth	<i>MIDI Channel 11</i> Dark Trumpet	<i>MIDI Channel 15</i> Good Old Hit
<i>MIDI Channel 4</i> Elec Perc	<i>MIDI Channel 8</i> DynamicBrass	<i>MIDI Channel 12</i> Motion Pad	<i>MIDI Channel 16</i> Choral Sweep

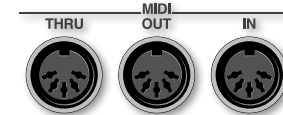


MIDI keyboards typically allow you to split the keyboard into different areas with their own MIDI channels so you can play different sounds from different areas of the keyboard.



In General MIDI music, MIDI Channels 1-9 and 11-16 play standard sounds. Each note on MIDI Channel 10 plays a different drum kit sound.

MIDI Jacks



A typical MIDI device provides MIDI jacks, or “ports,” into which you can plug a MIDI cable’s connector. There are three types of MIDI ports, and a device may offer one, two, or all three, depending on what the device does:

- *MIDI IN*—A MIDI IN jack receives MIDI data from some other MIDI device.
- *MIDI OUT*—A MIDI OUT jack sends MIDI data produced by the device out to another MIDI device.
- *MIDI THRU*—A device’s MIDI IN port receives MIDI data and passes it back out unchanged through the MIDI THRU port.



Some devices have a “soft THRU.” These devices have no actual MIDI THRU port, but their MIDI OUT port can be set to act as one.



MIDI devices that communicate via USB may not provide MIDI ports.

How MIDI Devices Are Connected

MIDI vs Audio, Part Two

In this section, we’ll talk about the connections that allow MIDI devices to communicate. Once MIDI communication is established, you can connect the audio outputs of the devices you’re using to amps or to a mixing console in order to hear the sounds they produce.

One-to-One MIDI Setups

The simplest type of MIDI setup is a connection between two MIDI devices where one device controls the other. The controlling device’s MIDI OUT port is connected using a single MIDI cable to the MIDI IN port of the device being controlled.



Some situations call for two-way MIDI communication, such as:

- *Recording into an external sequencer*—You record into a sequencer from a MIDI instrument, and the sequencer then plays the MIDI instrument’s sounds.
- *Storing both devices’ data in one place*—It can be convenient to store patch data and other settings for both devices in the controlling device. This frees you from having to remember where all your data is later on.

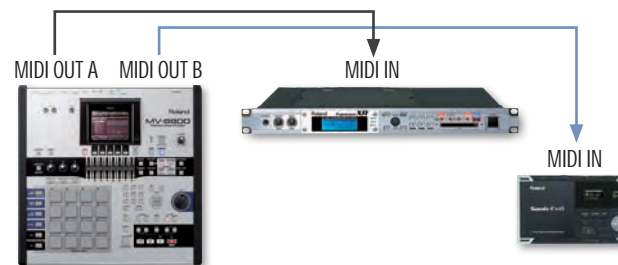
For two-way MIDI communication, a MIDI cable connects the MIDI OUT port of each of the two devices to the MIDI IN port of the other device.



MIDI instruments may also allow you to establish two-way MIDI communication with a computer using a single USB cable. We’ll discuss this type of connection later.

MIDI Devices with Multiple MIDI OUT Ports

Some instruments have two MIDI OUT ports, allowing the instrument to act as a controller for two devices, with 32 independent MIDI channels.



An instrument with two MIDI OUT ports and one MIDI IN port can have two-way communication with one of the MIDI devices to which it’s connected.

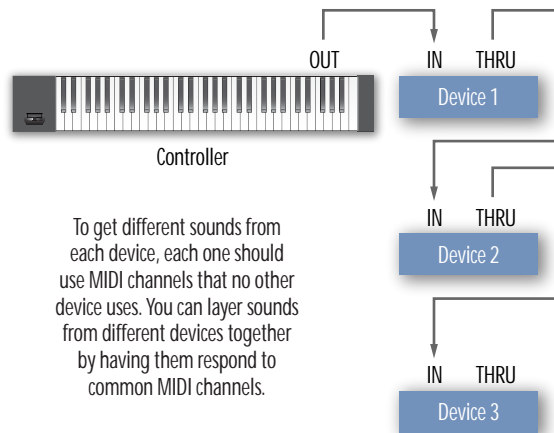
Larger MIDI Setups

You can set up MIDI communication between larger numbers of MIDI devices with “daisy chaining,” or by using a MIDI patchbay.

Daisy chaining

With daisy chaining, MIDI devices are connected in a series. Here's how a daisychain is constructed:

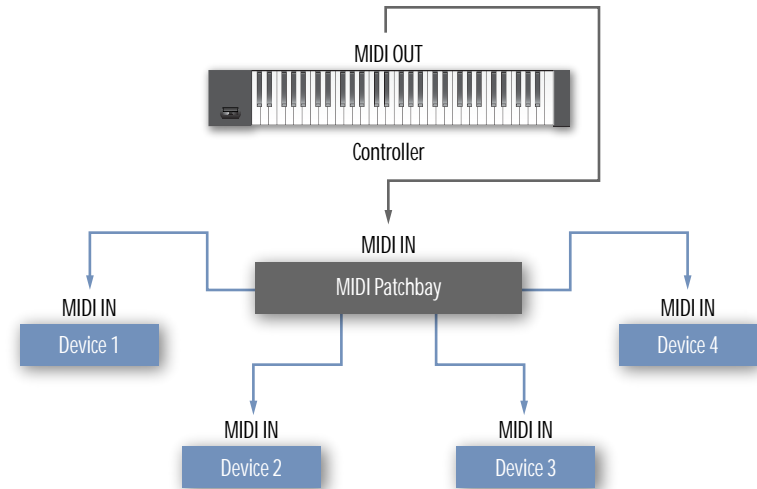
- The MIDI OUT of the controlling MIDI device is connected to a second device's MIDI IN.
- The second device's MIDI THRU port passes the controller device's data to a third device's MIDI IN port.
- Each remaining device's MIDI THRU port passes the controller device's data to the following device's MIDI IN port.



The advantage of a daisy chain is that it requires no additional hardware other than MIDI cables. The disadvantage is that each device in the daisy chain receives the controlling device's data a little later than the device before it, causing its sounds to play a little later, too. This can sometimes cause noticeable timing issues, especially with longer daisy chains. If you experience timing issues with a daisy chain, you can remove devices until it goes away, or use shorter MIDI cables.

Using a MIDI Patchbay

The best way to connect a large number of MIDI devices is by using a MIDI patchbay. The patchbay receives MIDI data from a controlling device and passes it to the rest of your devices at the same time for much tighter musical timing.



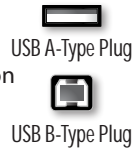
MIDI patchbays of varying degrees of complexity are available. The simplest are "MIDI THRU" boxes—like the one above—that receive MIDI data from a single controlling device and distribute it simultaneously to all other connected MIDI devices. Other patchbays allow you to designate one or more masters from among your connected devices, and then send the MIDI data to MIDI devices of your choice.

About Computers and MIDI Connections

Computers rarely have built-in MIDI ports. MIDI connectivity can be added to a computer with a:

- *MIDI interface*—A MIDI interface is usually a rack-mounted or tabletop box that connects to the computer via USB or FireWire, and provides MIDI ports you can connect to your other MIDI devices. An interface may simply provide MIDI ports for the computer, or it may be more complex, acting as a central MIDI patchbay for your entire MIDI setup. You may even be able to stack multiple MIDI interfaces for lots of MIDI connections.
- *sound card*—Sound cards can offer direct MIDI connectivity or a game port to which you connect a special cable that adds MIDI ports to the computer.

- *USB connection*—Some MIDI devices, can communicate with a computer through a USB connection. This form of two-way MIDI communication is especially easy to set up since it requires only a standard A-to-B USB cable and no additional interface.



You may need to install driver software on your computer for USB MIDI communication.



You can connect multiple USB MIDI devices to your computer at the same time. (Mixing USB MIDI devices from different manufacturers may require an understanding of how to configure your computer.)

MIDI Messages

In order for MIDI devices from different manufacturers to understand each other, MIDI uses its own language that all MIDI devices understand. This language consists of an agreed-upon set of messages. The messages cover just about anything you might want to do with a MIDI device.



Not all MIDI devices respond to all MIDI messages. The documentation for most MIDI devices contains a MIDI implementation chart that details the MIDI messages the device understands.

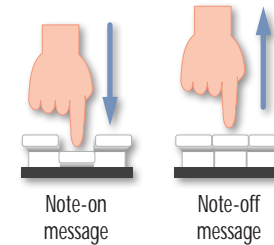


MIDI messages typically have 128 possible values. Most manufacturers, list them as “0-127.” Some use “1-128.” Again, check a device’s documentation to learn the numbering scheme it uses.

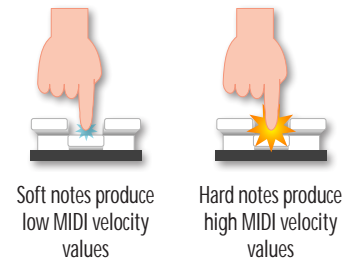
Basic Performance Messages

We’ve already touched on the MIDI messages that represent the most significant elements of a performance when you’re playing a MIDI device’s sounds:

- *Note-on and note-off messages*—When you begin to play a note on a MIDI controller, a note-on message is sent to the receiving MIDI device, which starts playing the note. When you stop playing the note on the controller, a note-off message is transmitted and the receiving device stops sounding the note.



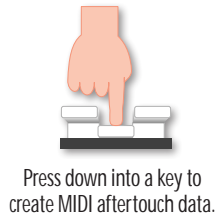
- *Velocity*—The force with which you play a note on the controller is measured as the speed, or “velocity,” with which you play the note. When you play a note, a velocity value representing the force with which you played the note is transmitted to the receiving device along with the note’s note-on message. Settings in the receiving device’s sound determine if and how the sound of the note is to change in response to velocity. Typically, notes get louder or quieter as you play harder or softer, respectively. They may also get brighter with higher velocity values. Sounds can be programmed to respond in all sorts of ways to velocity.



- *Pitch bend*—These messages instruct a MIDI device to raise or lower the played note's pitch. Pitch bend messages are usually transmitted by moving a pitch bend/modulation lever or pitch bend wheel at the left of a keyboard controller, though they can be transmitted from any knob, button, or joystick programmed to do so. String-bending on a guitar controller can also generate pitch bend messages.



- *Channel and polyphonic aftertouch*—"Aftertouch" is most often generated by keyboard controllers. You can generate aftertouch messages by pressing down on a key you've played. The aftertouch value reflects how hard you press. On the receiving device, a sound can be programmed to respond to aftertouch in any number of ways. With "channel" aftertouch, pressing down on any one key affects all currently sounding notes. "Polyphonic" aftertouch affects only the note actually being pressed down.



Program Changes

A MIDI device can select sounds in another MIDI device. This can be handy onstage, letting you to change sounds in all of your connected MIDI devices without your hands leaving the instrument you're playing. It also allows a sequencer to remember the sounds you want it to play and automatically select them for you in your receiving devices.

Program Change values run from 0-127, since MIDI-device sounds are arranged in "banks" containing up to 128 sounds each. (You can change banks using Bank Select Control Change messages, as we'll discuss on Page 12.)

Control Change Messages

What Control Change Messages Are

There's an separate category of MIDI messages called Control Change, or "CC," messages. Control Change messages cover a wide range of behaviors that can be controlled via MIDI.

Control Change messages are numbered from 0-127, like so many things in MIDI. There's a complete list of MIDI Control Change messages later in this book. For now, let's discuss how they work.



Some instruments allow you to easily send any Control Change message to another device.



In the following sections, we'll explain how to properly construct Control Change messages. While most instruments take care of this for you, you can manually build your own in some MIDI sequencers.

How Control Change Messages Work

About MIDI Device Sounds

Each sound-producing MIDI device contains an engine of some sort for producing its sounds. This engine has a collection of settings, or "parameters," that control its behavior. These parameters, and their values, allow the device to produce a range of different sounds. A sound, often called a "patch," is really nothing more than one set of values for the engine's parameters.

Control Change Messages and Sound Parameters

Many sound/patch parameters correspond to a Control Change number. This allows a MIDI controller to set the parameters remotely. A properly constructed Control Change message has two elements:

- First, the desired parameter's Control Change number is transmitted to the receiving device to let it know the parameter to be set.
- Second, the desired value for the parameter is transmitted.

For example:

- 1 First, you send a CC #7 message—CC #7 is the Volume Control Change message—to tell the receiving unit you want to adjust its loudness.
- 2 Next, you send a value that sets the unit's loudness as desired.

A Control Change message can act as an:

- *override*—that sets the parameter to the transmitted value.
- *offset*—that adjusts the parameter up or down by the value's amount.

MSB and LSB

"MSB" and "LSB" stand for "Most Significant Byte" and "Least Significant Byte," respectively. MSB Control Change messages typically act as coarse controls, while LSB messages generally allow fine adjustments.

A number of the Control Change messages have both MSB and LSB versions. MIDI devices that contain sounds/patches typically respond to both Bank Select MSB and LSB Control Change messages. Beyond that, MIDI devices may respond only to Control Change MSB messages, or to both MSB and LSB—you can check a device's documentation for details.

Bank Selects

As we noted earlier, Program Changes allow you to select sounds within the current sound bank. Many instruments, however, have more than 128 sounds, and therefore offer multiple sound banks. Bank Select Control Change messages, or simply "Bank Selects," allow you to choose different banks within the receiving MIDI device.

In fact, a full Program Change message includes the Bank Select MSB and LSB values that choose the desired sound's bank, and then the Program Change value for the sound itself within the bank. The message therefore has five elements, sent in the following order:

- 1 *CC #0*—the Bank Select MSB Control Change message
- 2 *the MSB value*—for the sound bank you want
- 3 *CC #32*—the Bank Select LSB Control Change message
- 4 *the LSB value*—for the sound bank you want
- 5 *the Program Change number*—for the desired sound within the now-selected bank.

Other Common Control Change Messages

Some of the other most commonly used messages are:

- *CC #1 Modulation*—Modulation can add vibrato or other changes to a sound. Modulation messages are usually produced by pushing a pitch bend/modulation lever or modulation wheel forward.
- *CC #7 Volume*—This message can control the overall level of the receiving device's sound.
- *CC #64 Sustain*—This message instructs the receiving unit's sound that a sustain pedal is being pressed, or not, causing notes to hold, or not.
- *CC #74 Brightness*—Brightness adjusts a sound's filter cutoff, allowing you to create filter "sweeps" in the sound.

Undefined CC Numbers

Some of the 128 Control Change numbers aren't assigned to MIDI device behaviors. These "undefined" Control Change numbers can be used by MIDI device manufacturers for unique purposes within their own products.

RPNs and NRPNs

"Registered Parameter Numbers," or "RPNs," and "Non-Registered Parameter Numbers," or "NRPNs" are special Control Change subgroups that provide control over other significant MIDI-device behaviors.

- *RPNs*—allow for the MIDI control of important global settings, including things like coarse and fine tuning. As the "Registered" in their name suggests, RPNs control a standardized set of parameters, regardless of manufacturer.
- *NRPNs*—allow manufacturers to control parameters in their products that are otherwise not controlled via MIDI.

Properly constructed RPN and NRPN messages contain the appropriate RPN or NRPN MSB and LSB values—to identify the parameter being set—along with a Data Entry MSB (and sometimes LSB) value to set the targeted parameter as desired.



As always, consult a device's documentation to learn if and how it responds to RPNs and/or NRPNs.

System Exclusive Messages

While MIDI is usually about standardized messages that every MIDI device can understand, MIDI can also be used for communicating things specific to particular devices, such as:

- the device's global settings
- the contents of a device's memory
- messages that can remotely operate its controls

These types of data and more can be transmitted and received as “System Exclusive,” or “SysEx,” MIDI data. At the beginning of each SysEx message is information that only the intended receiving device can understand. It's manufacturer-specific, product-specific, and even individual-device specific, since a SysEx message always contains a SysEx ID number that tags the data as being for a particular device—this is important in situations where there are more than one of the same type of device.

Bulk Dumps

You may be able to back up all of a device's settings by performing a SysEx “bulk dump” to an external storage device, such as a computer's hard drive. The charm of a bulk dump is that the storage device doesn't need to have any idea what the data does or means—it simply directs the data back to its originating device for re-loading.

Synchronization

MIDI can synchronize the timing of multiple MIDI devices that contain sequencers, from full-blown production suites to simple beat boxes.

- *MIDI clocks*—provide timing information from the controlling device to other connected devices. This keeps them all running at the same speed.
- *Song Position Pointer Messages*—or “SPP” messages, accompany MIDI clocks, and pass tempo and time signature information from one MIDI device to another.
- *MTC*—or “MIDI Time Code,” keeps the time counters of connected devices in sync.



When synchronizing MIDI devices, the controlling device is called the “master,” and any device being controlled is called a “slave.”

MIDI Control Change List

CC #	Control Function Value	Value	Used As
0	Bank Select MSB	0-127	MSB
1	Modulation Wheel or Lever	0-127	MSB
2	Breath Controller	0-127	MSB
3	Undefined	0-127	MSB
4	Foot Controller	0-127	MSB
5	Portamento Time	0-127	MSB
6	Data Entry MSB	0-127	MSB
7	Channel Volume	0-127	MSB
8	Balance	0-127	MSB
9	Undefined	0-127	MSB
10	Pan	0-127	MSB
11	Expression Controller	0-127	MSB
12	Effect Control 1	0-127	MSB
13	Effect Control 2	0-127	MSB
14	Undefined	0-127	MSB
15	Undefined	0-127	MSB
16	General Purpose Controller 1	0-127	MSB
17	General Purpose Controller 2	0-127	MSB
18	General Purpose Controller 3	0-127	MSB
19	General Purpose Controller 4	0-127	MSB
20	Undefined	0-127	MSB
21	Undefined	0-127	MSB
22	Undefined	0-127	MSB
23	Undefined	0-127	MSB
24	Undefined	0-127	MSB
25	Undefined	0-127	MSB
26	Undefined	0-127	MSB
27	Undefined	0-127	MSB
28	Undefined	0-127	MSB

CC #	Control Function Value	Value	Used As
29	Undefined	0-127	MSB
30	Undefined	0-127	MSB
31	Undefined	0-127	MSB
32	Bank Select LSB	0-127	LSB
33	LSB for Control 1 (Modulation Wheel or Lever)	0-127	LSB
34	LSB for Control 2 (Breath Controller)	0-127	LSB
35	LSB for Control 3 (Undefined)	0-127	LSB
36	LSB for Control 4 (Foot Controller)	0-127	LSB
37	LSB for Control 5 (Portamento Time)	0-127	LSB
38	LSB for Control 6 (Data Entry)	0-127	LSB
39	LSB for Control 7 (Channel Volume, formerly Main Volume)	0-127	LSB
40	LSB for Control 8 (Balance)	0-127	LSB
41	LSB for Control 9 (Undefined)	0-127	LSB
42	LSB for Control 10 (Pan)	0-127	LSB
43	LSB for Control 11 (Expression Controller)	0-127	LSB
44	LSB for Control 12 (Effect control 1)	0-127	LSB
45	LSB for Control 13 (Effect control 2)	0-127	LSB
46	LSB for Control 14 (Undefined)	0-127	LSB
47	LSB for Control 15 (Undefined)	0-127	LSB
48	LSB for Control 16 (General Purpose Controller 1)	0-127	LSB
49	LSB for Control 17 (General Purpose Controller 2)	0-127	LSB
50	LSB for Control 18 (General Purpose Controller 3)	0-127	LSB
51	LSB for Control 19 (General Purpose Controller 4)	0-127	LSB

CC #	Control Function Value	Value	Used As
52	LSB for Control 20 (Undefined)	0-127	LSB
53	LSB for Control 21 (Undefined)	0-127	LSB
54	LSB for Control 22 (Undefined)	0-127	LSB
55	LSB for Control 23 (Undefined)	0-127	LSB
56	LSB for Control 24 (Undefined)	0-127	LSB
57	LSB for Control 25 (Undefined)	0-127	LSB
58	LSB for Control 26 (Undefined)	0-127	LSB
59	LSB for Control 27 (Undefined)	0-127	LSB
60	LSB for Control 28 (Undefined)	0-127	LSB
61	LSB for Control 29 (Undefined)	0-127	LSB
62	LSB for Control 30 (Undefined)	0-127	LSB
63	LSB for Control 31 (Undefined)	0-127	LSB
64	Damper Pedal on/off (Sustain)	<63 off, >64 on	--
65	Portamento On/Off	<63 off, >64 on	
66	Sostenuto On/Off	<63 off, >64 on	--
67	Soft Pedal On/Off	<63 off, >64 on	--
68	Legato Footswitch	<63 Normal, >64 Legato	--
69	Hold 2	<63 off, >64 on	
70	Sound Controller 1 (Sound Variation)	0-127	LSB
71	Sound Controller 2 (Timbre)	0-127	LSB
72	Sound Controller 3 (Release Time)	0-127	LSB
73	Sound Controller 4 (Attack Time)	0-127	LSB
74	Sound Controller 5 (Brightness)	0-127	LSB
75	Sound Controller 6 (Decay Time)	0-127	LSB
76	Sound Controller 7 (Vibrato Rate)	0-127	LSB
77	Sound Controller 8 (Vibrato Depth)	0-127	LSB
78	Sound Controller 9 (Vibrato Delay)	0-127	LSB
79	Sound Controller 10 (undefined)	0-127	LSB
80	General Purpose Controller 5	0-127	LSB
81	General Purpose Controller 6	0-127	LSB
82	General Purpose Controller 7	0-127	LSB

CC #	Control Function Value	Value	Used As
83	General Purpose Controller 8	0-127	LSB
84	Portamento Control	0-127	
85	Undefined	---	LSB
86	Undefined	---	LSB
87	Undefined	---	LSB
88	Undefined	---	LSB
89	Undefined	---	LSB
90	Undefined	---	LSB
91	Effects 1 Depth		LSB
92	Effects 2 Depth	0-127	LSB
93	Effects 3 Depth		LSB
94	Effects 4 Depth	0-127	LSB
95	Effects 5 Depth	0-127	LSB
96	Data Increment (Data Entry +1)	N/A	LSB
97	Data Decrement (Data Entry -1)	N/A	LSB
98	Non-Registered Parameter Number (NRPN) LSB	0-127	LSB
99	Non-Registered Parameter Number (NRPN) MSB	0-127	MSB
100	Registered Parameter Number (RPN) LSB	0-127	LSB
101	Registered Parameter Number (RPN) MSB	0-127	MSB
102	Undefined	---	
103	Undefined	---	
104	Undefined	---	
105	Undefined	---	
106	Undefined	---	
107	Undefined	---	
108	Undefined	---	
109	Undefined	---	
110	Undefined	---	

CC #	Control Function Value	Value	Used As
111	Undefined	---	
112	Undefined	---	
113	Undefined	---	
114	Undefined	---	
115	Undefined	---	
116	Undefined	---	
117	Undefined	---	
118	Undefined	---	
119	Undefined	---	
120	All Sound Off	0	
121	Reset All Controllers		
122	Local Control On/Off	0 off, 127 on	
123	All Notes Off	0	
124	Omni Mode Off (+ all notes off)	0	
125	Omni Mode On (+ all notes off)	0	
126	Poly Mode On/Off (+ all notes off)	**	
127	Poly Mode On (+ mono off +all notes off)	0	

Other uses of MIDI

Games:

Computer and video game music, now generally referred to as game audio, began in the early days of MIDI and to some extent, its history exemplifies both the creativity and the limitations of 'MIDI music.' Today, game music delivers startling levels of 'reality'. Both recorded music and MIDI, are at the heart of most game audio developers' systems.

Interactivity, however, is a current hot topic in the game industry and MIDI's inherent flexibility coupled with microscopic file sizes, makes it a natural choice for a new generation of both hardware and software that puts control of game music in the hands of the players.

Mobile devices:

Limited storage and delivery bandwidth on cell phones dictated that ring-tones and audio alerts were almost always generated by MIDI.

Although the trend in recent years has been towards 'real' (compressed) audio versions of popular hits, MIDI ring-tones are still very popular for their ability to be changed in any way imaginable, and more and more phones now support downloadable games that benefit from the compact size of MIDI data.

The End



For complete MIDI technical documentation, visit the MIDI Manufacturers Association Web site at <http://midi.org>.