Taking the Mystery Out of MIDI

by
Howard Massey

Endorsed by

IMA International MIDI Association

MMA MIDI MANUFACTURERS ASSOCIATION

Published by

NAMM

The National Association of Music Merchants

Written by Howard Massey.
Graphic design, illustrations and cover art by Linda E. Law.

Produced for the National Association of Music Merchants by
On The Right Wavelength.

Special thanks to: Susan Davies, Jim Kleeman, David Snow,
Delilah Snow, and Lachlan Westfall

INTRODUCTION

Imagine being able to place the power of an entire orchestra at your fingertips. Picture yourself playing a piece of music on a keyboard, guitar, or wind instrument while a computer records your performance and prints out a score of it moments later—in fact, suppose your personal computer could actually write songs with you! And wouldn't it be great to purchase any electronic musical instrument at all and, with the connection of just one cable, have it control or be controlled by any other electronic instrument? MIDI—the Musical Instrument Digital Interface—is the magical ingredient that allows all of these things—and more—to be accomplished. It is the glue that binds together electronic and computer equipment made by all sorts of different manufacturers and it is the common ground upon which most of today's music is created.

For many people, though, MIDI is a nebulous, vaguely intimidating "thing" which seems difficult if not impossible to understand. The word "MIDI" itself brings to mind technicians and computer operators pushing buttons, typing obscure commands, and doing strange things with complex electronic devices. In short, it's a word that has become surrounded in a cloud of mystery. And this mystery has made a lot of people who aren't programmers or engineers feel that MIDI is just too complicated to deal with.

Baloney.

Like so many other things in life, MIDI isn't nearly as involved as it first appears. Once you get past the jargon, you'll find that it is easy to understand and even easier to use. What's more, you really don't need to know all the intricacies of MIDI in order to use it effectively—just as you don't need to understand the theory of combustion engines in order to drive a car.

As the title says, the aim of this booklet is to take the mystery out of MIDI. Once that mystery is gone, you'll discover a powerful tool that will enhance the musical creativity within you.

Howard Massey
March, 1992

WHAT IS MIDI?

In a word, MIDI is a *language*. More specifically, it is a computer language "spoken" by virtually every electronic musical instrument manufactured since the early 1980s. How do you know if an instrument has MIDI capability? Look for one or more five-pin MIDI connectors (called *ports*), usually found on the rear panel:

If these ports are present, the instrument is capable of communicating with the outside world by sending and receiving digital data to or from other MIDI devices. And it isn't just electronic musical instruments that fall into the category of "MIDI devices"—in fact, most personal computers are capable of speaking the MIDI language (including all IBM PC-compatibles, all members of the Macintosh family, the Apple II, the Atari ST, and the Commodore Amiga and C-64 / C-128).

The purpose of MIDI is to allow all kinds of electronic musical instruments—even if they're made by different manufacturers—to work with one another and with computers. As a result, one of the real strengths of buying a MIDI instrument is that it can never become obsolete—since it can "interface" (that is, communicate) with any other MIDI instrument (and with your computer), you can always expand your system by adding new MIDI devices.

Before we get into the specifics of what MIDI is and what it can do, let's talk about what we mean by the word "language." The main purpose of any language is to convey information; however, language can also be used to ask questions, give orders, and (in the case of languages spoken by humans) stimulate the imagination. These words, for example, are written in the English language; they will make sense to you if you are capable of reading English, but probably won't have much meaning if the only language you understand is, say, Swahili. In their most basic form, human languages such as English consist of letters, words, and sentences. For example, there are 26 letters in our alphabet and there are literally hundreds of thousands of ways these letters can be strung together meaningfully into English words. But words on their own often have little significance; that's why we almost always place them in context by arranging them in a particular order into a sentence.

In order to determine the way words are placed together in sentences, every language also contains a set of rules. For example, the English language specifies that the first word of every written sentence should have its first letter capitalized. Similarly, there is a rule that causes every sentence to end with either a period, a question mark, or an exclamation point. These kinds of rules are not limited to the written language, by the way—there are also a set of rules that govern the spoken language. For example, in English, we pause briefly when we come to a comma and pause a bit longer when we arrive at the end of a sentence. Also, the pitch of our voice tends to change at the end of sentences (try asking a question and notice how your voice goes up slightly at the end; then try giving an order and notice how your voice goes down slightly at the end).

Similarly, the MIDI language consists of letters, words, and sentences—and it also contains a set of rules that determine how sentences are to be constructed and communicated. Remember, however, that MIDI is a **computer** language, not a human one. Computers are only capable of thinking in terms of numbers; more specifically, they always reduce everything down to either the number one or the number zero. Thus, the MIDI alphabet consists of only two "letters"—one and zero. In computerese, each "letter" is called a *bit*.

Although "human language" words can be of any length, all MIDI words are eight "letters" long (thus they are said to be eight *bits*). For example, the MIDI "word" that says "someone's just played a note on my keyboard" is 10010000*, whereas the MIDI "word" that says "someone's just stopped playing a note" is 10000000*. But don't worry—you won't need to memorize the "spelling" of these or any other digital words in order to understand and use MIDI.

The set of rules that determines how MIDI words and sentences are constructed is, in computerese, called a *protocol*. The MIDI protocol is spelled out in a published document called the *MIDI Specification*—and every device that implements MIDI must follow this protocol closely. The MIDI Specification is periodically updated as new features are added—but the basic rules governing MIDI always stay the same.

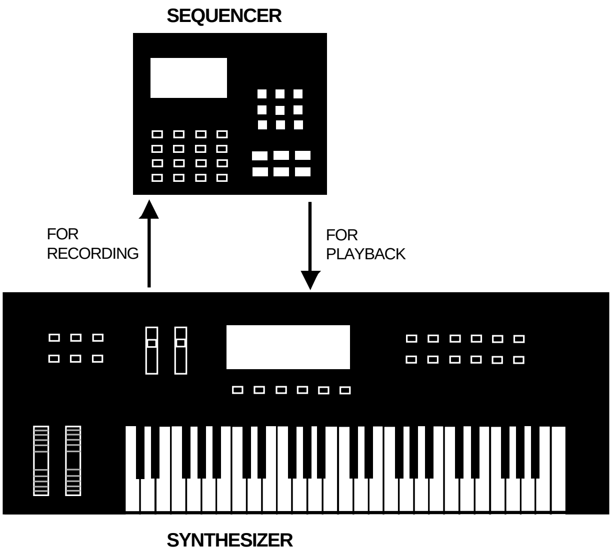
** Actually, the last four bits in these words don't have to necessarily be all zeroes (they vary according to the MIDI channel being used), but we'll talk more about that later.*

A BRIEF HISTORY OF MIDI

Why was MIDI was developed in the first place? The story is fascinating and can help you understand the importance of MIDI in today's music. In the early days of electronic musical instruments (the 1920s through the mid-1970s), *analog* synthesizers generated sounds by the direct manipulation of electrical signals (just as most music was heard through the analog medium of record players). By the late 1970s, however, computer technology had evolved to the point where it was possible (and cost-effective) for synthesizers to create sounds *digitally*—that is, with the aid of microchips and other computer components (just as today's CDs use digital processes to produce much higher-quality sound than vinyl records).

What's more, an important add-on digital component called a *sequencer* had been developed by that time. Sequencers are essentially tapeless recorders which use computer memory in order to store information about a performance. Unlike a tape recorder, a sequencer does not record the sound itself—instead, it records digital information that describes (among other things) which notes were played and the timing relationships between those notes. A sequencer requires a synthesizer or other electronic instrument in order for that performance to then be recreated.

For this to work, there must be a pathway through which the synthesizer can "speak" to the sequencer for recording and another pathway through which the sequencer can "speak" to the synthesizer for playback. In other words, there needs to be two-way communication between a sequencer and a synthesizer:



The manufacturers of early digital synthesizers and sequencers accomplished this by developing proprietary computer languages which their instruments used for communication. The problem, however, was that each manufacturer independently developed a unique language which no other manufacturer's instrument could speak! The end result was that if you bought a sequencer from manufacturer A, it could only work with a synthesizer from manufacturer A. Similarly, manufacturer B's synth could only operate with a sequencer made by manufacturer B.

These difficulties were compounded further still when specialized instruments such as drum machines and samplers began appearing in the early 1980s. The musician of 1980 who wished to use electronic sounds was pretty much forced to purchase all of his or her instruments from the same manufacturer; there was simply no efficient means of having digital instruments made by different manufacturers work with one another.

Fortunately, this problem was recognized by the manufacturing community. Industry leaders saw that the development of a standardized language—one that would allow the use of *any* manufacturer's instrument with any other—would give the musician greater creative freedom and would therefore help make electronic instruments even more popular. A formal proposal to create MIDI was first introduced at the 1981 Audio Engineering Society convention in New York City. In a universal spirit of cooperation rarely seen in any industry, most manufacturers joined forces to make this idea a reality. Overseeing organizations were soon formed in the United States (the MIDI Manufacturer's Association, or MMA for short) and Japan (the Japan MIDI Standards Committee, or JMSC)—and MIDI was born.

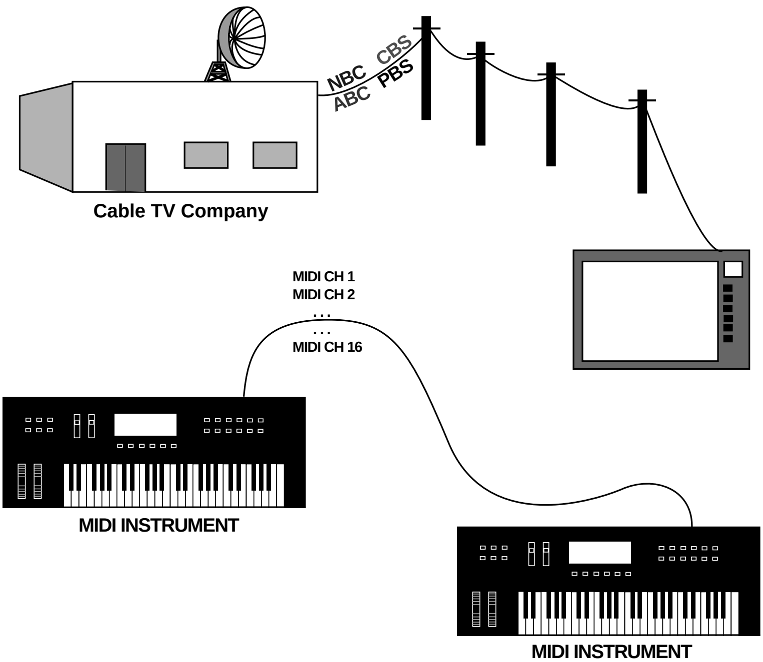
By 1983, virtually every new electronic instrument "spoke" the MIDI language and came equipped with MIDI ports. Shortly thereafter, specialized hardware devices called *interfaces* appeared for most major personal computers, allowing them to "speak" MIDI and so have the ability to work with synthesizers, sequencers, samplers, and drum machines. In the mid-1980s, alternate MIDI *controllers*—devices that resembled guitars, drum pads, and wind instruments—began making an appearance, for the first time making it possible for non-keyboardists to enter the world of MIDI. By the early 1990s, MIDI had evolved to the point where it was being used for everything from mix automation to computer-assisted composition; from remote control of tape recorders and stage lighting to the transcription of musical scores. We'll take a closer look at these and a number of other MIDI applications later in this booklet; for now, let's return to our discussion of language and actually learn a few of the words in the MIDI vocabulary.

TYPES OF MIDI MESSAGES

MIDI words are usually known as *messages* and all MIDI messages are either *channel* messages or *system* messages. Channel messages address general features which are found in most MIDI instruments, while system messages take on more specific functions, such as timing information. The main difference is that channel messages are always labeled ("flagged") with a channel number from 1 to 16; this allows particular instruments (or particular sounds within an instrument) to respond selectively.

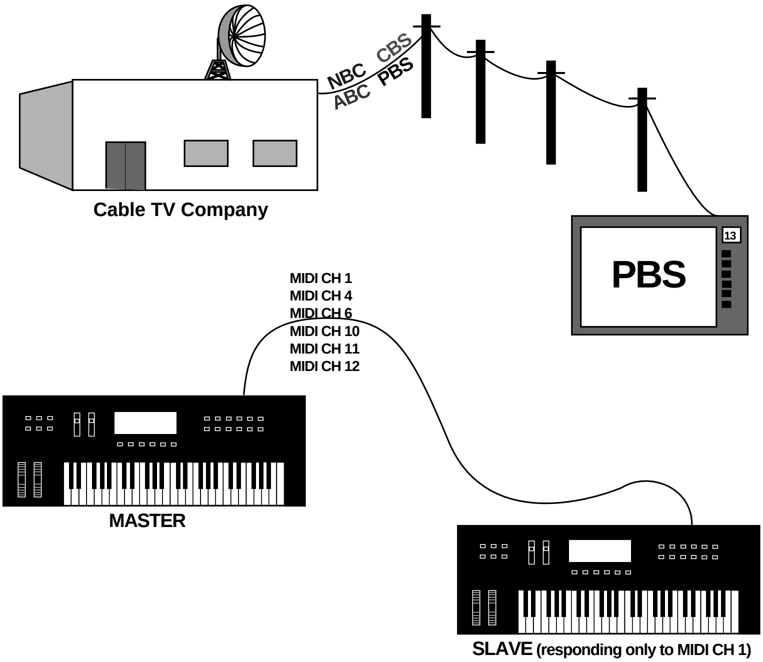
The best way to understand how MIDI channels work is to think of the way TV stations always broadcast on their own specific channel. If you want to watch CBS, for example, you will have to tune your TV set to receive on the same channel that your local CBS station uses to transmit on. Similarly, for two MIDI devices to communicate, the one that is receiving messages (usually called the "slave") must be set to the same MIDI channel being used by the one that is transmitting messages (called the "master"). The selection of a MIDI instrument's transmit and receive channels is usually made from its front panel—see the instrument's owners manual for more information.

Just as a cable TV signal (a composite of many different TV channels) are sent through a single wire, so too are MIDI messages (from all sixteen MIDI channels) sent through one wire, in this case called a *MIDI cable*.

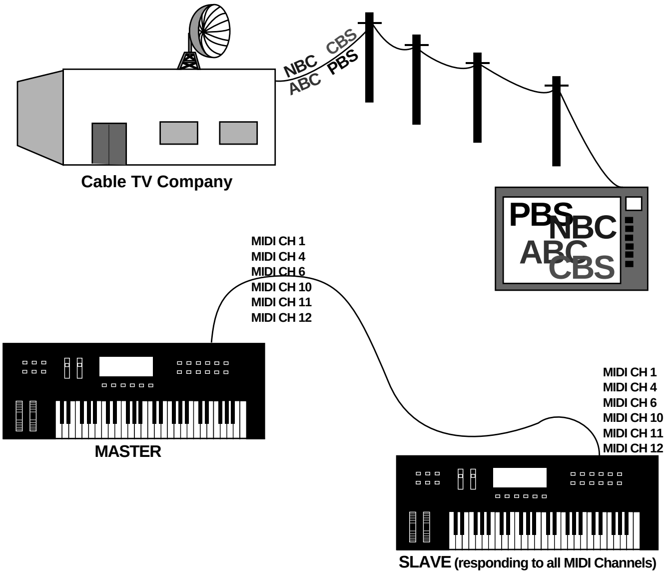


MIDI RECEIVE MODES

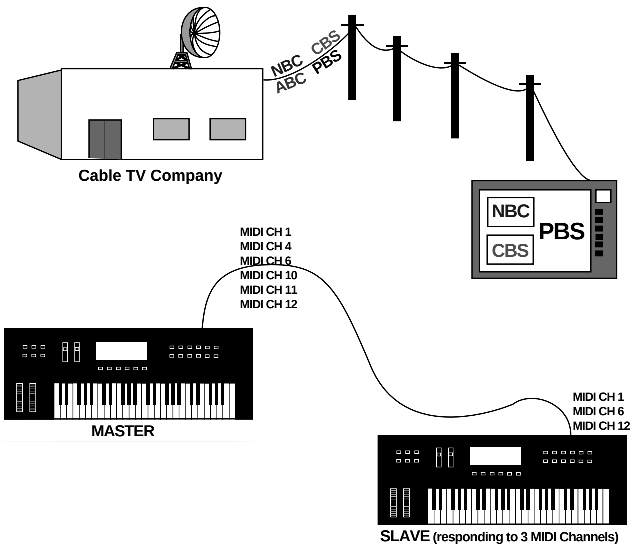
MIDI instruments can be set to receive channel messages in one of three different ways; these are known as *receive modes.* The first option is to have the instrument resond to only one MIDI channel. This is called *poly* mode and is equivalent to the normal operation of a TV set, where it receives only one broadcast station at a time.



The second alternative is to have your MIDI instrument receive messages in *omni* mode. Here, it will "listen" and respond to messages being transmitted on **all** MIDI channels, providing a fast and easy means for determining that the instrument is correctly connected into a MIDI system. As you can see on the next page, there's no earthly reason why you'd ever want to do this on your TV set, since your screen would show a jumble of all stations being broadcast simultaneously!



Many current MIDI instruments have the capability of playing more than one sound at the same time (they are said to be *multitimbral)*. These instruments are effectively acting like several different synthesizers in one box and so can use a third receive option, which is usually called *multi* mode.* This mode allows the instrument to "listen" to several specific MIDI channels at the same time so that individual sounds within that one instrument can respond selectively to messages being transmitted on different MIDI channels. This is roughly equivalent to the operation of some newer models of TV sets which can receive two or more channels at the same time and place each image in a small window on the screen.



** Multi mode is not a "true" MIDI mode in that it is not described in the MIDI Specification. However, many manufacturers of multitimbral instruments have adopted this terminology.*

Setting the receive mode of your MIDI instrument is usually as simple as setting its receive channel. This is typically accomplished from the device's front panel, although the receive mode can also be changed remotely with the use of a special MIDI message (appropriately enough called a *channel mode* message).

Since MIDI uses sixteen channels, it allows up to sixteen different musical parts to be played simultaneously. But if that isn't enough for you, bear in mind that many dedicated "master" MIDI controllers and computer MIDI interfaces have two or more MIDI outputs, often with the ability to direct various MIDI messages to any or all of their outputs. With each output jack, then, you add another potential sixteen channels.

CHANNEL MESSAGES

Here are some of the MIDI channel messages you are most likely to use in everyday applications:

• Note On - This is a three-part message which is transmitted whenever a note is played on a MIDI controller (typically a keyboard, although alternate controllers, such as MIDI guitars, drum pads, or wind devices can be used). After the Note On word itself is transmitted, a note number (from 0 to 127) is sent—every musical note is assigned a unique number. Bear in mind that MIDI allows for 128 notes in total—that's 40 more than the 88 notes offered by a full piano keyboard!

The third part of the Note On message contains a Velocity value, ranging from 0 to 127. Velocity is a means by which a MIDI controller can specify the dynamics of each note—whether it was played with a heavy touch, a light touch, or somewhere in-between. If the controller is a keyboard, velocity will be a measurement of the speed with which a key is moved downward from its (horizontal) resting position. Depending upon the settings of the receiving MIDI device, velocity may be used to alter the volume, brightness, or other aspect of the sound.

• Note Off - The opposite to Note On, this signifies that a particular note is no longer being played. This MIDI message is rarely used, however—instead, a Note On message with a velocity value of 0 is used by most instruments as a substitute for a Note Off message since it is a more efficient way of "saying" the same thing.

A few MIDI keyboards use a Note Off velocity value to create a control called *release velocity*. Instead of measuring the speed with which a key is moved downward from its resting position, this measures the speed with which the key **returns** to its resting position, giving the MIDI musician an additional degree of control. Because this is a technique which requires a special skill to develop, release velocity is usually an option that can be turned on or off.

• Control Change - Expressivity is a vital component of all musical sound. Acoustic musicians can use various techniques to add vibrato to sustained notes, for example, or can bow harder, change reed pressure, or bend a string in order to alter the volume, tone, or pitch of the sound. It is equally important that MIDI musicians be able to add expressivity, so virtually all MIDI instruments have a number of sliders, wheels, switches, and / or footpedals that can be used to alter the sound as a note is being played. These very important peripherals are called *real-time controllers*, and the MIDI Specification allows for up to 121 of these to be used. A Control Change message consists of two parts: a Control Number (from 0 to 120) followed by a value from 0 to 127. This value shows the position of the controller at the time of transmission; if you are, for example, moving a modulation wheel while holding a note, your instrument will transmit a whole stream of Control Change messages showing the position of the wheel during each fraction of a second. If the controller is a switch-type (such as a sustain pedal), the value will instantly jump from minimum (0) to maximum (127) whenever the switch is pressed.

The MIDI Specification leaves many “undefined” control numbers, but also assigns several specific numbers to often-used controllers. A few examples are:

Controller # 1 = Modulation wheel
Controller # 2 = Breath control
Controller # 4 = Foot controller (sweep footpedal)
Controller # 7 = Volume
Controller # 10 = Pan
Controller # 64 = Sustain pedal

• Pitch Bend - Most MIDI instruments also provide some kind of device (typically a center-indented wheel or joystick) that enables the musician to alter the pitch of a sound as it is being played (the effect is similar to a guitarist bending a string, hence the term “pitch bend”). The MIDI Specification treats Pitch Bend as a special kind of controller because the human ear is very sensitive to changes in pitch—for this reason, the 0 - 127 range used by most other controllers is too coarse for this purpose. Instead, Pitch Bend messages use data values that range from 0 to

16,384. These values are usually displayed as both negative and positive numbers (that is, from -8,192 to +8,192), so that the receiving instrument knows whether the pitch is to be raised or lowered (the center value of 0 indicates no pitch bend).

• After Touch (sometimes called Key Pressure) - This is another "special case" real-time controller which at first glance may appear somewhat similar to velocity. However, it is in fact quite different, since it is a measurement not of the speed with which a key is depressed but of actual pressure placed on the key **after** it is depressed (wind controllers often use breath pressure after the initial attack to generate after touch messages). Depending upon the setting of the receiving instrument, this may be used to alter the volume, tone, and / or pitch of the sound.

Like velocity, there are two kinds of after touch. Most instruments use what is known as *channel key pressure* while others use a more sophisticated version known as *polyphonic key pressure*. An instrument that uses channel key pressure will apply the same after touch value (the maximum one being produced) to all notes transmitting or receiving on the same MIDI channel, while one utilizing polyphonic key pressure generates a separate after touch value for each note played (even to individual notes within a chord).

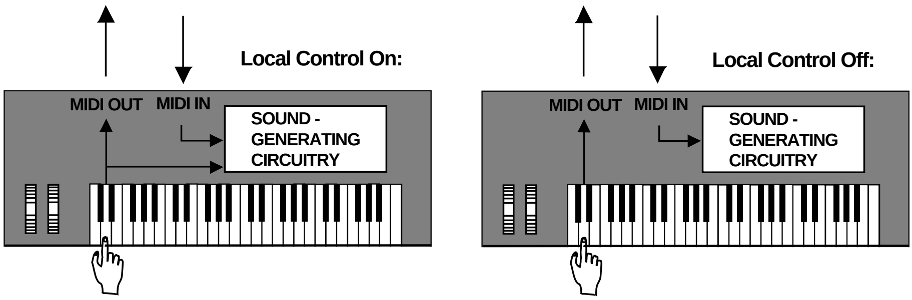
• Program Change - This is a MIDI message which specifies the number of the sound (called a "voice," "patch," "performance," or "program," depending upon the manufacturer) currently being used by your instrument. All electronic instruments made since the development of MIDI have a memory area into which sound data can be stored; most instruments allow you to store dozens, if not hundreds of sounds for immediate recall. The Program Change message (which is usually transmitted when you call up a different sound) simply tells any receiving MIDI instruments the **number** of the sound you have called up—it doesn't tell them anything about the sound itself (there is a whole category of MIDI messages that **can** impart specific information about the sound, however—see "System Exclusive," below).

Some newer MIDI instruments precede Program Change messages with a controller #0 message (this control number is used for a function called Bank Select) in order to specify the area of memory (internal, card, preset, etc.) that the program is stored in.

• All Notes Off - This is a kind of "panic button"—a MIDI message which tells the receiving instrument to stop playing any notes which are currently sounding. Stuck notes can occur if a Note On message is not followed eventually by an equivalent Note Off (or Note On with Velocity of 0) message; for this reason, some sequencers transmit an All Notes Off message whenever playback is stopped.

• Reset All Controllers - Another "panic" type of message, this causes the receiving instrument to reset all of its controllers (including pitch bend) to an inactive role. As with All Notes Off, some sequencers transmit a Reset All Controllers message when playback is halted.

• Local Control - This MIDI message affects an instrument's connection between its keyboard and sound-generating circuitry. When Local Control is on, playing the keyboard causes internal sounds to be heard and also causes data to be transmitted via the MIDI Out port. When Local Control is off, the keyboard continues to transmit MIDI data from the Out port, but the internal path is disconnected so that no sound is heard unless data is received from the MIDI In port:



Local Control should usually be turned off in a MIDI controller when it is also receiving data from a MIDI sequencer. This ensures that the notes you play on the keyboard do not interfere with the playback performance of the sequencer. As an alternative to using this MIDI message, Local Control can usually be turned on or off from the instrument's front panel. For more information, see the "Setting Up A MIDI System" section below.

SYSTEM MESSAGES

Here's a summary of some of the most frequently used MIDI system messages:

• Sequencer Start, Stop, and Continue - As their names imply, these messages cause a MIDI sequencer to start, stop, or continue its playback. The difference between the Start and Continue commands is that Start always causes a sequencer to commence playback from the beginning of the first measure whereas Continue causes it to play from its current location (this may be where it was last stopped or any point you specify). The value of this command is not only that it allows the musician to operate a sequencer remotely but that it allows two or more MIDI sequencers to begin or end playback simultaneously.

• Song Select - Many MIDI sequencers and drum machines are capable of storing several songs in memory simultaneously, making it easy to quickly go from song to song in live performance. This MIDI command allows the user to remotely select the song to be played.

• MIDI Clock - This is a timing command that allows all sequencers and drum machines in a system to operate at the same tempo. During playback (and sometimes even when idling), MIDI sequencers and drum machines generate 24 of these MIDI Clocks during each quarter note.

• Song Position Pointer - This is another MIDI timing command. However, while the MIDI clock message ensures that all instruments in a system play at the same tempo, MIDI song position pointer messages ensure that all sequencers and drum machines in a system begin playback at the same point. The sequencer or drum machine which is acting as the "master" device continually counts the number of sixteenth notes which have occurred since the start of the song and uses this message to "point" to the running total. If playback is started from a point other than the beginning of the song, the master sequencer or drum machine quickly transmits its current song position before commencing playback; the slave machines then go to that identical position in order to begin playback from the same point. A MIDI Start message (which causes playback from the start of the song) is always followed by a song position pointer value of 0.

• MIDI Time Code (MTC) - This is a sophisticated set of MIDI timing commands that is very similar to the standard SMPTE (Society of Motion Picture and Television Engineers) time code used in video production. It uses a running clock to assign each instant in time a unique "address," thus allowing for much finer synchronization (to within 1/120 of a second) than Song Position Pointer messages. MTC is most often used in professional film or video scoring applications, where musical events must be timed to precisely coincide with visual images.

• System Exclusive (often called "Sysex" for short) - Every MIDI device has an area of memory in which information about its internal programming is stored. In the case of a synthesizer, this information would typically describe sound edit parameters; in the case of a drum machine, this might describe its user-programmed rhythmic patterns. MIDI System Exclusive messages provide a means for MIDI devices to transmit ("save") and receive ("load") the entire or partial contents of their memory. This allows their memory contents (called *bulk data*) to be offloaded to computers and other data storage devices (the process of transmitting bulk data is often called a *bulk dump*). Because the internal data of one MIDI instrument will usually be meaningless to any other instrument (exept for one which is the same make and model), all System Exclusive messages are “flagged” with a unique code (assigned by the MIDI Manufacturers Association) that identifies the manufacturer’s name as well as the make and model of instrument.

- Sample Dump Standard (SDS) - This special group of MIDI messages allow samplers to send and receive sample data to and from computers or other data storage devices. SDS also enables the free exchange of sounds between samplers.

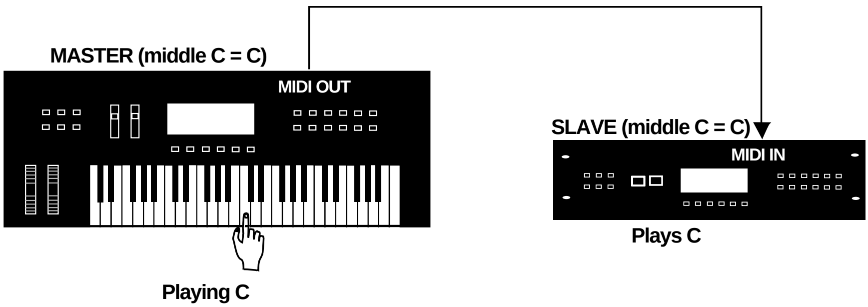
By the way, just as not everyone who speaks English uses the entire English vocabulary, not every MIDI device transmits or responds to all MIDI messages. For example, it’s usually just sequencers and drum machines that need to work with timing signals such as MIDI clock, song position pointer, or MTC—many synthesizers and samplers ignore these messages entirely. Similarly, only MIDI controllers that generate after touch will transmit that message, although most sound-producing instruments (including keyboardless rack-mounted or tabletop tone generators) are capable of responding to it. To find out precisely which words are in your instrument’s MIDI “vocabulary,” refer to the *MIDI Implementation Chart* provided in most owner manuals.

MIDI APPLICATIONS

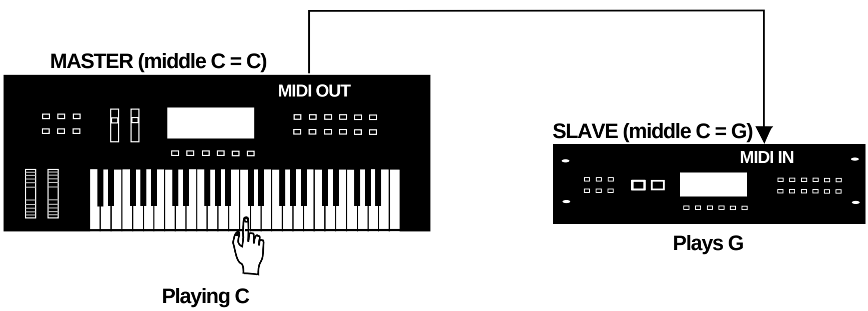
Now that we understand what MIDI is, how it came to be developed, and what some of the words in the MIDI language are, the obvious question is: What can all this be used for?

LAYERING SOUNDS

The first and most basic application is to allow the layering of two or more electronic sounds. If you aren't satisfied with the richness of the sound you get from one electronic instrument, MIDI gives you the option of adding the sound from other instruments. The end result is a composite sound of the two (or more) instruments. This allows the MIDI musician to quickly create complex tonalities which are far richer than those which could be generated from any single instrument. The way this works is simple—via MIDI, Note On, Note Off and Velocity messages are sent from one instrument to another (the receiving instrument must be set to the same channel as the transmitting instrument). When you then play a note on the "master," you will hear the same note come from the "slave":



What's more, most electronic musical instruments have editing features which allow you to adjust their tuning transposition. By transposing slave instruments to a different key than that used by the master controller, the MIDI musician can play block chords from a single note!



SEQUENCING

Probably the most widely used MIDI application is that of *sequencing*. As described earlier in this booklet (in the “Brief History of MIDI” section), sequencers are essentially tapeless recorders that store information about a performance in memory. Like a tape recorder, most MIDI sequencers (which can exist as dedicated hardware devices or as computer programs) offer multiple tracks for the storage of data over all sixteen MIDI channels; as discussed previously, many provide multiple MIDI outputs so that even more channels can be utilized. But MIDI sequencers are much more powerful than tape recorders. For one thing, all data stored in a sequencer can be freely edited, typically by using standard computer cut, copy, and paste techniques. This allows you to easily reorchestrate your music—add a verse here, shorten a bridge there, double the length of a chorus, whatever. MIDI sequencers also allow you to do “micro-surgery” on individual events so that mistakes can be corrected, individual notes moved forward or backward in time, passages transposed, tempos adjusted, etc. Also, because it is the performance and not the sound itself that is stored, you can speed up or slow down the tempo of a sequence without changing pitch. You can even freely change the sounds of the different musical parts well after the piece was recorded! It’s easy to see why MIDI sequencers have become an absolute boon to composers, orchestrators, and arrangers.

MIDI sequencers are capable of storing all kinds of MIDI messages. In addition to note on, note off, and note numbers, they can also record information about dynamics (in the form of velocity), expressivity (in the form of after touch, pitch bend, and controller messages), and can even be used to automate mixing (by recording MIDI volume, pan, and other controller data as well as program change messages).

PATCH LIBRARIANS AND DATA FILERS

Computer programs known as *patch librarians* and hardware *bulk storage* devices (sometimes called *MIDI data filers*) carry out another vital MIDI function—that of allowing the musician to build extensive libraries of sounds easily and inexpensively. These programs and devices are capable of storing large amounts of MIDI System Exclusive messages so that hundreds or even thousands of sounds can be archived onto standard floppy disks. The stored data can then be reloaded back into the instrument in a matter of seconds.

NOTATION

MIDI can also be used by *notation* programs to print out standard musical scores derived from a live performance. You can literally play your part on a MIDI keyboard (or alternate controller, such as a MIDI guitar or wind device) and have your computer print out the score moments later! And your performance doesn't have to be perfect, either—any mistakes you make can easily be corrected on the computer screen prior to printout.

FILM AND VIDEO SCORING

In film and video scoring, MIDI is a powerful tool. A number of computer interfaces and dedicated hardware devices are capable of correlating SMPTE time code (used in video production) with song position pointer messages or MIDI Time Code (MTC). This makes it easy to synchronize a MIDI sequencer with video or film playback and also assists in the process of *spotting*, where particular musical events are made to coincide with specific visual images.

COMPUTER-ASSISTED COMPOSITION

Those of you who occasionally wish you had a songwriting partner will appreciate the MIDI application of computer-assisted composition. There are a number of software packages (often called *algorithmic composition* programs) that enable the computer to actually collaborate with you in the process of writing music! These use the powerful number-crunching capabilities of the computer to manipulate musical phrases played in from a MIDI controller.

OTHER APPLICATIONS

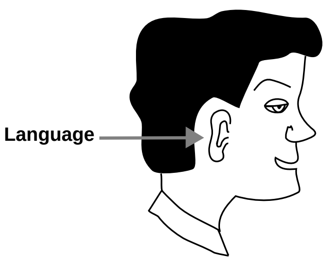
This list of MIDI applications is by no means exhaustive. For example, MIDI can be used to change sound parameters in audio signal processors (such as reverb units or digital delay lines); it can be used to control stage lighting or tape recorder transports; we're even starting to see CD players with MIDI outputs (allowing you to actually reorchestrate prerecorded music)! Innovative manufacturers and computer programmers are constantly coming up with new applications for MIDI. That's because the versatility and standardization of the MIDI language lends itself to development—and it is today's musician who is the direct beneficiary of all this hard work!

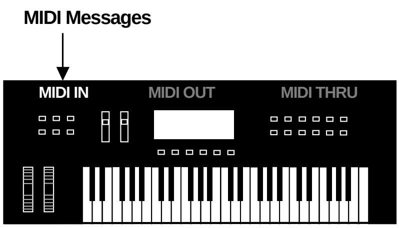
MIDI HARDWARE

The MIDI Specification not only defines the MIDI language but also lays down a set of rules (a “protocol”) for the hardware used by MIDI instruments. Clearly, the transmission speed of MIDI messages is critical, since the idea is to have an instrument respond more or less instantly to incoming messages. To accomplish this, MIDI messages are transmitted at a rate of 31,250 bits per second (that’s nearly 4,000 messages per second)—some thirteen to twenty-six times faster than standard computer modem speeds!

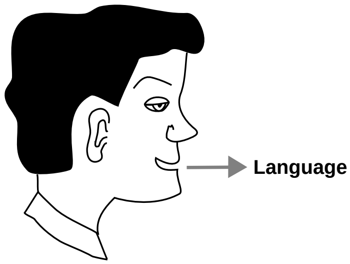
A device that “speaks” MIDI will typically contain one or more MIDI input and / or output ports. The physical jacks used accommodate a standard 5-pin DIN plug. MIDI cables (which have a 5-pin DIN plug at either end) should always be of the shortest possible length in order to minimize possible transmission errors; the MIDI Specification specifically states that they should be no more than fifty feet (fifteen meters) long. However, several manufacturers produce MIDI line amplifiers that allow you to use much longer lengths. Although many electronics stores sell “generic” DIN-to-DIN cables, it is recommended that you use cables specifically manufactured for MIDI usage; you’ll find a wide variety of MIDI cables in different lengths at your local music dealer.

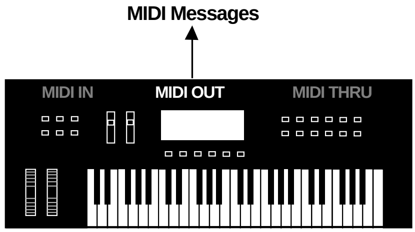
The MIDI input port is, naturally enough, where MIDI data is received by the instrument. It is usually simply labeled “In.” In terms of our analogy between the MIDI language and the English language, it is roughly equivalent to the human ear:



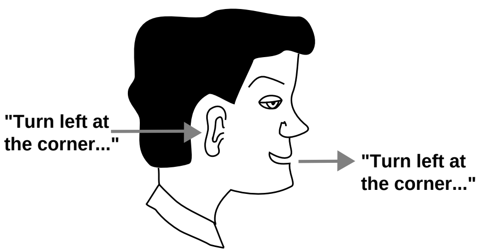


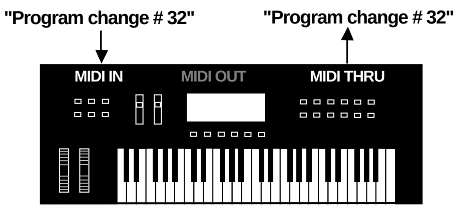
MIDI messages which are generated by the instrument are transmitted via the MIDI output port (usually labeled “Out”); this is roughly equivalent to the human voice:





Many instruments offer a third type of MIDI port, called “Thru”. This is actually a special kind of MIDI output. Like the “Out” port, MIDI data leaves the instrument from here. However, there is an important difference. While the messages transmitted via the Out port have been generated within the instrument itself, the messages transmitted via the Thru port are instead an exact copy of whatever is being received at the MIDI input. These messages are transmitted from the Thru port at virtually the same time they are received so that you can route the same MIDI signal to several instruments in a “daisy-chain” fashion.* A good human analogy might be that of a person receiving information in one ear and simultaneously repeating that information verbatim:





* *More about this shortly.*

Some devices (particularly sequencers) offer a function called "echo" or "thru," whereby their MIDI output acts as both a MIDI Out and a MIDI Thru. When this function is activated (it is usually a software option), the data leaving the MIDI output is a combination of the messages being received by the input and new data being generated by the device itself.

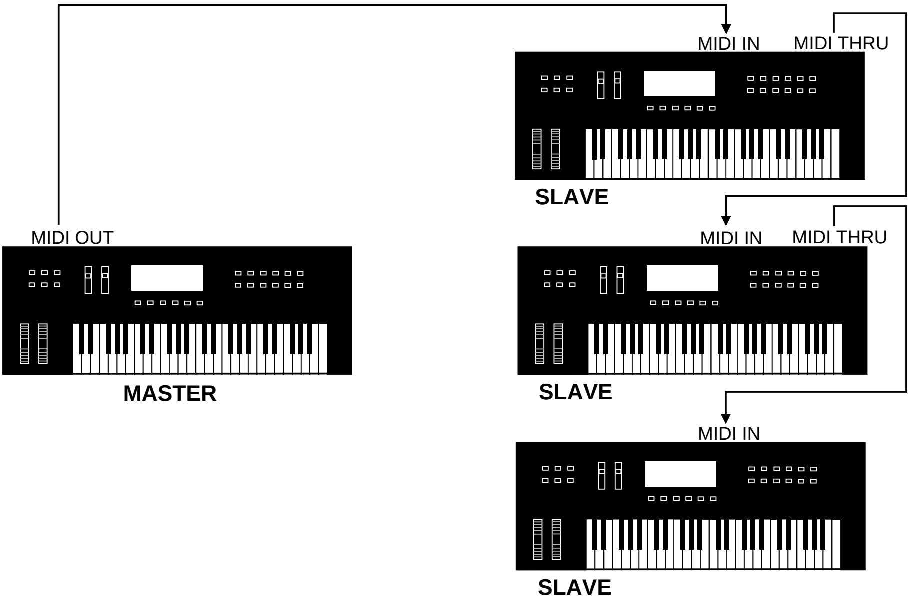
If you plan on using a personal computer with your MIDI equipment, you will in most cases need to purchase a MIDI *interface* (the exception to this rule is the Atari ST / STE family of computers, which have a MIDI interface built in). This is a hardware device which adds one or more MIDI In, Out, and (sometimes) Thru ports to your computer. For some computers (notably IBM-PC and PC-compatibles and the Apple II), the MIDI interface consists of a card which fits into a computer expansion slot. In other cases (notably the Macintosh family of computers), the MIDI interface is an external device that connects to the existing modem port. Bear in mind that the interface alone will not be enough to enable your computer to "speak" MIDI; you will also need appropriate software to complete the job. Your local music dealer can advise you on the types of interfaces and software packages available for your particular computer.

SETTING UP A MIDI SYSTEM

One of the joys of MIDI is that it allows you to expand your system as your budget and musical needs evolve. The main MIDI devices you'll probably be using will include synthesizers, samplers, drum machines, sequencers (or computer sequencing programs), computer interfaces, and other computer MIDI software (such as patch librarians or notation programs). There is also a type of instrument known as the *workstation*, which is kind of the Swiss Army knife of the MIDI world. These instruments usually combine a keyboard synthesizer (and / or sampler and / or drum machine) with an onboard sequencer—all in one cost-effective package. Are you better off buying a workstation than individual MIDI components? This will depend upon factors like features, sound quality, cost and your future musical plans—but bear in mind that whenever you buy **any** piece of equipment that "speaks" MIDI, you are buying a product that really cannot ever become obsolete. Consult with your local music dealer to find out about specific products and to identify those that are best for your needs.

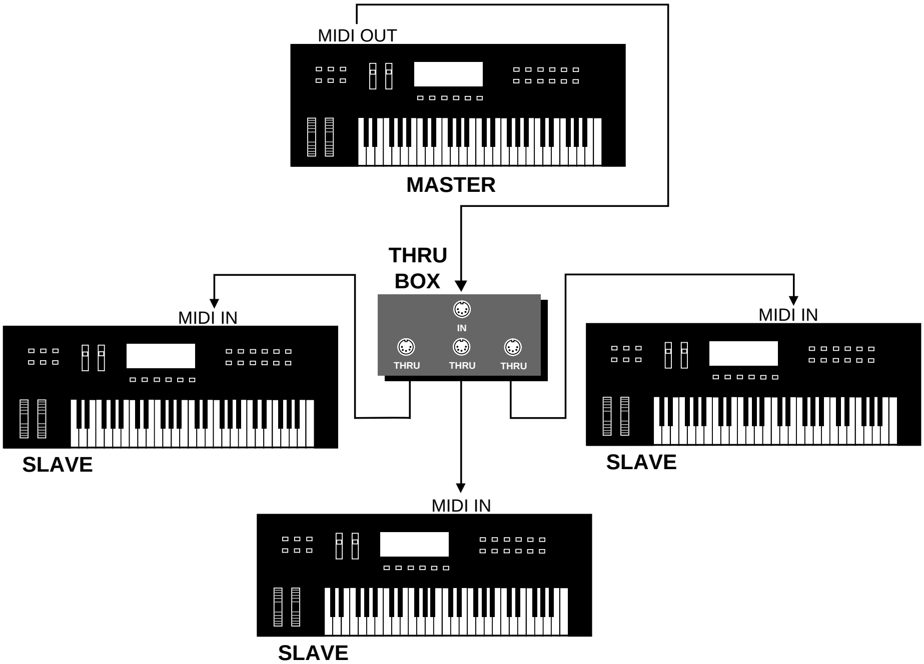
We've already described the most basic kind of MIDI system, where the MIDI output of a "master" transmitter is connected to the MIDI input of a "slave" receiver. The transmitter (usually called a *master controller*) is usually a keyboard, but can alternatively be a MIDI guitar, wind instrument, or drum pad. The slave device can be any device (keyboard, rack-mount, tabletop design, or computer interface) that has a MIDI input.

What if you have a master controller that you wish to hook up to two or three slave devices? This is where the MIDI thru port can be put to use, since it is an output that simply reflects whatever is coming into the MIDI input. It can therefore be used to direct the MIDI signal to the next instrument in the “chain”:

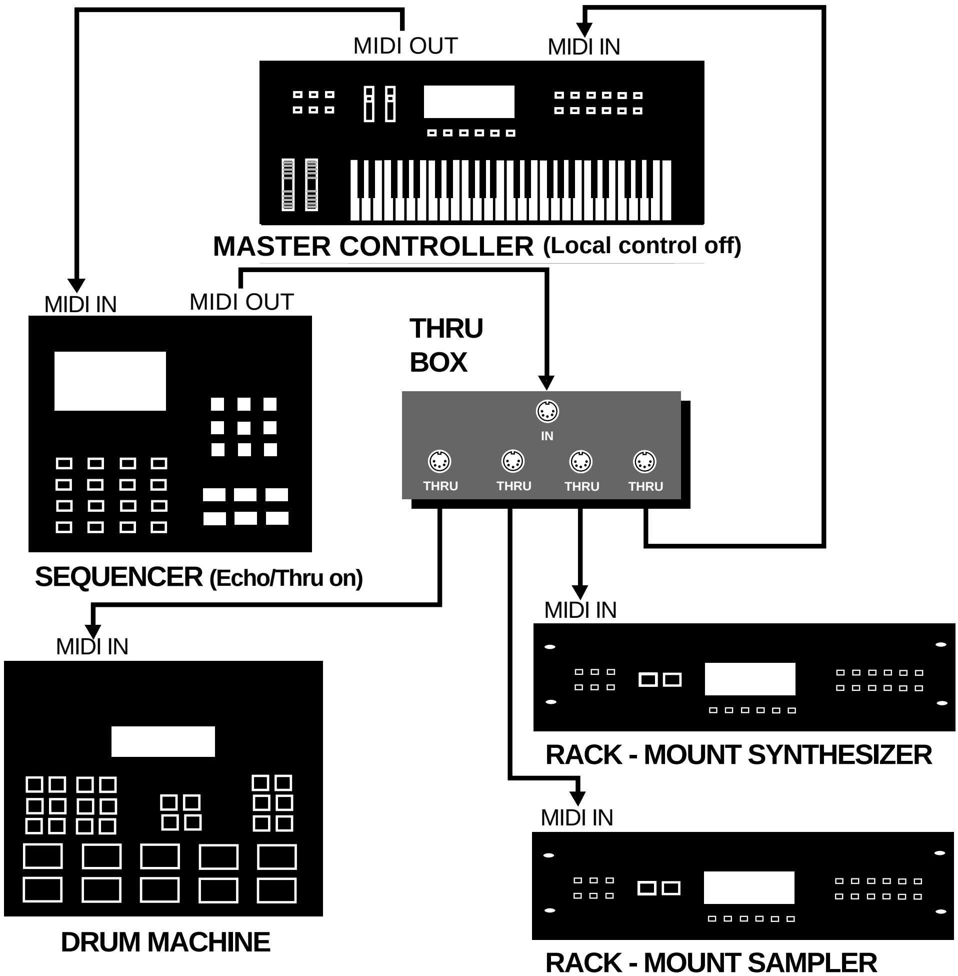


For layering, each slave device in the chain should be receiving on the same MIDI channel; for orchestration, each should be set to a different MIDI channel, with each instrument (or each “voice” within a multitimbral instrument) playing a different sound.

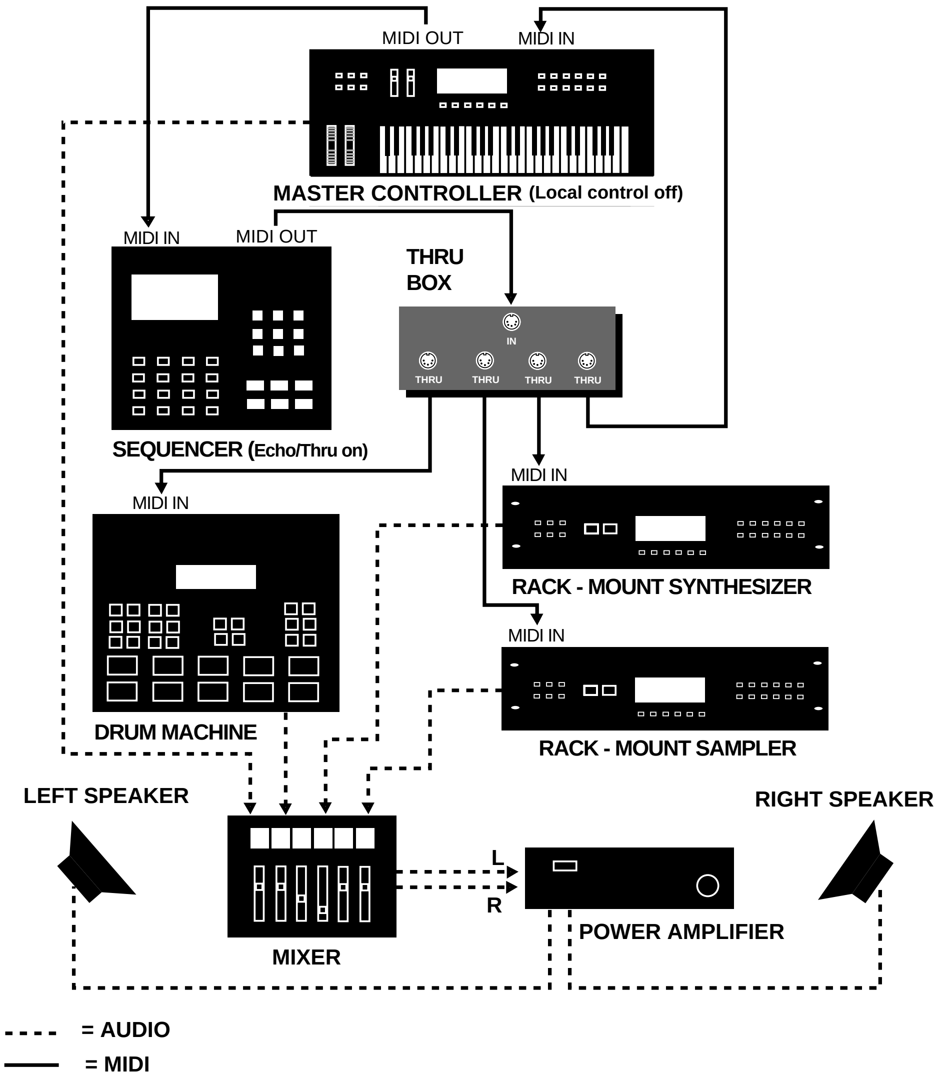
However, you shouldn’t use this kind of *daisy-chain* setup if you are using more than three MIDI slave devices, for the following important reason: By sending a digital signal through more than three instruments, you increase the possibility of data errors occurring. Therefore, if you want to link up more than three MIDI slaves, you will require a special piece of hardware called a *thru box*. This compact and inexpensive device is a MIDI signal distributor which accepts one (or more) MIDI input signals and routes them to four (or more) MIDI thru-puts (there are no true outputs since the thru box itself does not generate MIDI data). The illustration on the next page shows a typical usage of a thru box:



As we've seen, if you're using a MIDI sequencer (or a computer running a MIDI sequencing program), two-way communications are required for recording and playback. To accomplish this, connect the master controller output to the sequencer input and the sequencer output to the input of all sound-generating MIDI devices in your system. If your master controller is a synthesizer or sampler, make sure to turn its Local Control off. You'll also probably want to use the sequencer's "echo" or "thru" function, as described above. On the next page is shown a typical setup for a mid-sized system consisting of a keyboard synthesizer acting as master controller along with a sequencer, a drum machine, a rack-mount synthesizer, a rack-mount sampler, and a MIDI thru box:

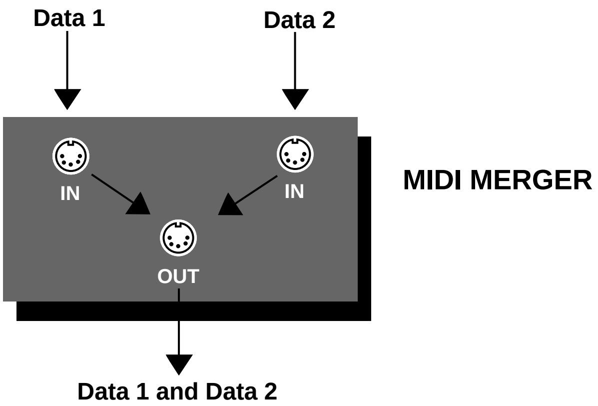


Bear in mind that each sound-generating device in your system (in this case, the master controller, drum machine, synthesizer, and sampler) has its own audio outputs which are entirely separate from the MIDI routings. In order to hear all these audio signals over a single pair of stereo speakers, you will need a *mixer*. On the next page is an illustration of the same system, showing audio as well as MIDI routings:



If you have a large MIDI system with many devices, you might want to use a *MIDI switcher* instead of a thru box. This provides a more sophisticated way to distribute MIDI signals; typically, it offers eight or more MIDI inputs along with eight or more MIDI thru-puts. Most switchers allow the simultaneous routing of various inputs to thru-puts and also utilize computer memory so that you can store and recall often-used setups.

Another peripheral that you may need to add is a *MIDI merger*, which blends two separate MIDI inputs into a single output.* MIDI mergers are usually required when there is a need to combine channel messages from one device with timing signals (clocks, song position pointers or MTC) from another device. A MIDI merger also permits live “jam” sessions between two MIDI musicians to be recorded into a sequencer.



* *Mixing two MIDI signals together is a complicated process. Don’t try to use a Y-cord to mix two MIDI signals together because data errors will almost certainly occur!*

GENERAL MIDI

In 1991, the MIDI Manufacturers Association and Japan MIDI Standards Committee approved an optional set of guidelines called *General MIDI*. The idea behind General MIDI is that it provides a standard set of rules for a new category of MIDI instruments (appropriately enough called “General MIDI instruments”) so as to guarantee an even greater degree of compatibility.

For example, General MIDI instruments are always multitimbral (they must be able to play at least sixteen sounds simultaneously) and they all have a memory capacity of at least 128 sounds. The organization of their memory is also standardized. For example, piano sounds are always stored in memory slots 1 - 8 and percussion sounds are always stored in memory slots 9 - 16, etc. There are also conventions for the reception of MIDI channel data; for instance, General MIDI instruments always use channel 10 for drum and percussion parts. The rules of General MIDI ensure that you can create a sequence with one instrument and successfully play that sequence back with any other instrument—secure in the knowledge that the sounds you hear will at least be similar to those used in the original recording, if not exactly the same.

General MIDI was designed with the hobbyist or casual user in mind. Your local music dealer can show you the latest line of General MIDI instruments, made by a variety of manufacturers. If you're just getting into MIDI, you'll probably find these instruments to be the ideal "starters." Most importantly, because they speak the universal language of MIDI, they can be integrated with any other MIDI devices as you build your system in the future.

FOR MORE INFORMATION ABOUT MIDI

The best source for information about MIDI is your local music dealer, who will be happy to answer any further questions you may have and aid you in making the best MIDI equipment purchases for your needs. Bear in mind that even though mail-order houses may sell some MIDI products for a few dollars less, they won't be able to help you if and when you run into a problem. You can't place a dollar value on the importance of friendly advice and assistance—so support your local service-oriented music dealer!

Another excellent source for information is the MIDI Manufacturers Association (MMA). In addition to disseminating information about the latest developments in MIDI, the MMA also publishes the official MIDI Specification and other related documents. Check out their Web site at www.midi.org or contact them at:

P.O. Box 3173
La Habra, CA 90632 USA

There are also a number of noteworthy books that deal with various aspects of MIDI and the process of setting up a home MIDI studio. You'll find these at your local music dealer, along with a full range of MIDI instruments and peripherals. Now go create some great music!