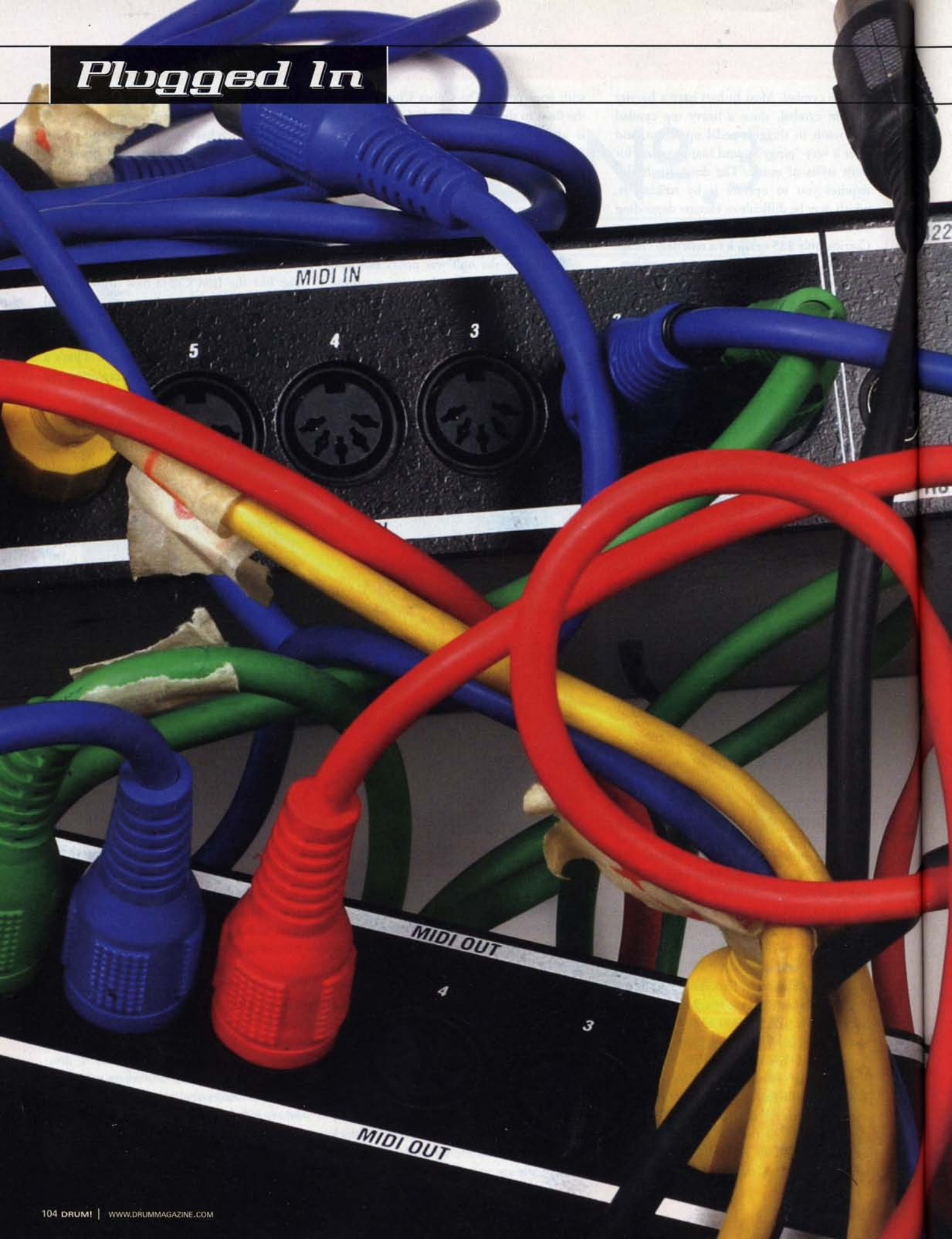
Plugged In



HOW TO MAKE IT HAPPEN

THE BUILDING BLOCKS OF MIDI

by Norman Weinberg

Photo by Ernie Rideout

MIDI (an acronym for Musical Instrument Digital Interface) first came onto the scene in 1984. There's no doubt that the advent of MIDI has changed the way that musicians make music. From the aspiring singer/songwriter to the high-budget soundtrack composer, MIDI makes it happen. Drummers and percussionists have embraced MIDI, computers, and electronic percussion instruments to enhance their art, exercise their creativity, and widen their musical potential. Today, it's not uncommon to find drummers who are using the tools of technology for everything from their home studio to the arena.

Controlling electronic instruments requires a different skill set than controlling a pair of sticks or putting hands to skin. Learning new skills often means learning new terminology and wrapping your brain around concepts that might at first seem confusing. But like everything else, knowledge is power and practice makes perfect. We're not going to dive into all of the gritty details – the bits, bytes, zeros and ones, or the Universal Non-Real-Time System Exclusive Messages – that may take place behind the scenes, but if you want to take control, you'll need to know what's going on. With that in mind, this article will cover the most common MIDI messages and how they might be incorporated in an electronic percussion system.

FIRST THINGS FIRST. MIDI is a communication protocol that acts as a command language between two computer-controlled devices that have been programmed to create and/or respond to the language. While MIDI messages can be incorporated into many different musical activities, its most common use is to have one electronic musical instrument control another. If you're playing a self-contained device, such as the Yamaha DTX kit or the Roland HandSonic, the controller (the surfaces you play) and sound producing electronics are all contained in the same physical box, and you many not need to deal with MIDI messages. But, whenever you wish to connect two different pieces of gear – perhaps firing the sounds from inside the HandSonic while you're playing on the DTX – you'll need to know your MIDI commands. As with any language, it takes two to communicate. One of the devices will act as the master controller, and one will act as the slave. In the example above, the DTX kit is the master device and the HandSonic is the slave. The DTX will generate the MIDI messages and the HandSonic will respond.

To get the messages from one device to another, you'll need to connect a MIDI cable between the two instruments. The cable required for MIDI is a five-pin DIN plug. All MIDI cables have two male ends (the ones with the pins), so that either end can be plugged into any MIDI instrument. So, where does the cable go? When you want to route MIDI messages *out from* your controller to some other device, you need to use the MIDI-Out jack on the controller. MIDI-In jacks are used to *bring in* MIDI messages to a device. So, MIDI-In messages are read by the microprocessor inside the device, but actually originate in another machine. To

return to our previous example, connecting a MIDI cable from the DTX's MIDI-Out to the HandSonic's MIDI-In would get the job done.

MIDI CHANNELS. The MIDI specification enables messages to be transmitted and received over 16 discrete channels. These aren't hard-wired channels like the channels at the back of a mixing console. Instead, they are soft-channels, similar to the channels on a television set. If you've got a cable-TV connection, then a single coax cable runs into the back of your television. This cable carries the signals from a large number of different channels, and you select which channel you want to view from the front-end of the TV (or from the remote control).

Whenever a master device sends messages to a slave, their MIDI channels must correspond. Let's say that you're using a drumKAT to create a movie-like, atmospheric composition with drum loops, drones, rhythmic orchestral hits, and ethnic vocal riffs all fired from a computer running Reason software. In a case like this, you'd program the KAT's pads to send the messages for each sound over a different channel – for example, you might program drum loop messages over channel 1, drones on channels 2-4, orchestral hits on 5, and vocal riffs on channel 6. You'd then set Reason's soft instruments to "listen" to channel 1 for the loops, channels 2-4 for drones, and … well, you get the idea. As this example illustrates, MIDI makes it possible to control up to 16 different timbres.

Even though the MIDI specification has a maximum of 16 channels, it's not too difficult to go beyond this limit. You'll need a master controller with more than one discrete MIDI output and a slave device that is capable of reading more than one MIDI data stream at a time. For instance, the drumKAT has four MIDI output jacks; two designated as "left" and two designated as "right." When programming the drumKAT, you have the option of routing MIDI messages to the left side MIDI-Outs, the right side MIDI-Outs, or both. Computer programs that are designed as multi-timbre soft synths can often read more than one MIDI input, depending on the MIDI interface connected to the computer. The software program Reason can read up to four MIDI data streams allowing for 64 unique MIDI channels. The well known soft-sampler GigaStudio can read as many as eight unique data streams, resulting in 128 channels.

CHANNEL VOICE MESSAGES. As an electronic drummer, MIDI's "channel voice messages" are the ones you'll be dealing with most often. These are the messages that transmit most performance information. Channel voice messages contain a designation that specifies one of the 16 MIDI channels. When a MIDI instrument sends channel voice messages over the MIDI cable, any slave device listening to that specific MIDI channel will react to these messages.

Note-On. This command is the most important MIDI message. It's the message used to tell a slave device to play a note. A note-on message consists of three pieces of information that form the complete MIDI action. The first piece of information is the command that tells the slave to turn on a note. Next comes a piece of information that tells the slave device *which* note to turn on. Since MIDI is a language of numbers, the value of the number determines the pitch. MIDI note numbers range from 0-127, with 0 being the lowest and 127 being the highest. Using this numbering system, "middle C" is note number 60. The distance between one number and another is a measurement of half steps. So, note number 61 would be the C# above middle C, note number 59 would be

the B natural below middle C, and note number 72 is the C an octave above middle C. With 127 available pitches, the total MIDI range is 10.5 octaves.

The last piece of information that creates a complete note-on message is the value for the note's velocity. With electronic percussion instruments, the MIDI velocity is an indication of how hard the surface was struck. If you lightly tap an electronic pad, the MIDI value will be lower than if you strike that pad with more force. MIDI velocities also range from 0-127, with 0 being no velocity and 127 being the maximum velocity.

Just because your controller sends a note-on message over channel 3 to play the F# above middle C, that doesn't mean you're going to actually hear a pitch. The master controller may be in charge of sending the MIDI command, but the sound module actually creates the audio. The sound module is simply going to play note 66 on channel 3. If the sound module has a set of Latin percussion samples assigned to channel 3, and there's a conga slap assigned to note 66, then that's what you'll hear. If a group of military sound effects are assigned to channel 3, you might hear an F-16 whooshing across the stereo field. If instead, there is a set of Balinese tuned gongs assigned to channel 3, you may, in fact, hear the F# pitch.

Note-Off. The note-off message does pretty much what you think it does. It is the command that tells a sound module to stop playing a note. A complete note-off command is just like the note-on, and requires three pieces of information: the actual instruction to turn off a note, the note number that is being turned off, and the off velocity. In many electronic percussion controllers, the note-off command is programmed to take place after a certain amount of time has passed. Since drummers can't hold down a key or keep air moving through a mouthpiece, we use something called "gate-time" to determine a note's length. If you've programmed your bass drum pad to have a gate time of .5 seconds, then a note-off message will be generated one-half second after the note-on message. Using gate time to manage note-off messages is a key factor in successfully playing melodic lines from a percussion controller.

Program Change. When used properly, the program change message can be one of the most powerful MIDI commands available. Program change is used to call up different sounds from the instruments in a MIDI rig. Depending on the machine, programs may be called "kits," "patches," "presets," "instru-

ments," or even "tones." The bottom line is that the program change message is sent whenever you request that a slave device change from, say, a timpani sound to a trombone sound. Several of today's electronic drum kits come complete with a general MIDI sound module built inside the brain. In such cases, you may use the program change message to determine which sound the onboard sound generator will play when you hit one of the pads, or send the kit commands from a software sequencer. Most electronic drum sets also use the program change message to call up the factory and user "kits" that have been programmed into the instrument's brain.

Program change messages can select between any of 128 different programs. A fairly recent expansion to the MIDI specification – called "bank select" – added the ability to select between any of 16,384 different banks, each containing 128 different programs. This is of course a theoretical limit, as no sound module currently available supports that many programs.

Pitch Bend. Pitch bend messages tell a receiving device to alter the pitch, either up or down. With some electronic drum kits, you can program the brain to send pitch bend messages when the hi-hat pedal is moved from open to closed position. Other percussion controllers, such as the drumKAT and the HandSonic, can be programmed to send pitch bend messages either from a foot pedal or from sensing the amount of pressure against one of the pads. How far the pitch changes, and the direction of pitch change (up or down) is a result of controller commands and the slave device's settings. For example, a sound module might be programmed to lower the pitch by a third, a fifth, or even an octave. Pitch bend messages can be an effective technique for bending tom sounds and for giving a more natural performance impression to melodic instruments.

Channel Pressure. Channel pressure is sometimes called "aftertouch." For keyboard synth players, it is a measurement of the amount of pressure given to a key between the note-on and note-off messages. Channel pressure is a sort of average pressure of all notes being held down. This pressure measurement will affect all the notes on the given channel to the same degree. While keyboard controllers are more adept at sending channel pressure, some electronic kits and multipads can route the signals from hi-hats, pedals, and pads to channel-pressure MIDI messages. Another type of MIDI aftertouch is called "polyphonic key pressure." The primary feature of polyphonic key pressure is that each key (or each pad and pedal) can send its own aftertouch amount. In a six-piece kit, all six pads could be sending different levels of aftertouch.

Control Change Messages. These controls are the knobs, switches, buttons, sliders, wheels, and pedals that are used to alter the tone of the sound module in some manner. They include controls that you move with your hands, your feet, or even your breath. To make matters a little more inclusive, the term can also be used to designate controls that don't actually change the sound, such as data entry buttons, on and off switches, and any other controls that the slave device can respond to.

The control change message is sort of a catchall for many different types of MIDI commands. Basically, they can be broken up into three different types of information:

1. *Continuous Controllers* – These can perform a continuously variable change from one value to another. The modulation wheel on a synthesizer is an example of this type of control.

2. *Switch Controllers* – These controllers really only have two possible positions: on and off (or to put it another way, a closed switch which will make contact and an open switch which will not make contact). An example of this type of controller is the sustain pedal. An example of this type of control on a software mixer is a channel's mute button.

3. *Registered and Non-Registered Controllers* – Registered controllers are those additional controllers that have been defined by the MIDI specification chart. Non-registered controllers are controller designations that will remain open and unassigned. These can be used by different manufacturers in a myriad of ways. They can do whatever the designer of the equipment wants them to do.

The intelligent use of these controls can turn a stiff and stale electronic performance into something expressive and meaningful. Since kit players generally have both feet and both hands actively involved in the performance process, it's often difficult to work these control change messages into a live performance. However, if you're working with a com-

puter sequencer when preparing demos of your songs, triggering loops during a live performance, or backing a DJ at a dance club, you should have more "free-limb" time to work these controls. Depending on how the slave is programmed, control change messages can be routed to a number of different destinations: modulation wheel, main volume, pan position, attack time, chorus depth, filter frequency, and more. Many of the latest software DAW programs use control change messages to turn your computer into a fully automated recording studio.

CONTROLLING OUTBOARD GEAR WITH MIDI. Even though MIDI was originally designed to control musical instruments, just about any electronic device can be designed to react to MIDI messages. For example, an effects processor that responds to MIDI commands might use program change messages to select between different types of effect presets. It might use a number of different continuous controllers to determine the values of the wet/dry mix, the size of the room, gate time setting, or the EQ's center frequency and boost/cut amount.

MIDI-controlled lighting boards might use note-on and note-off messages to control a large number of spots and background lights. You can perform by programming the board to respond to continuous controller messages. Video software such as Livid's Union uses MIDI Note-On messages to fire video, still images, and even live video images from a digital camera. Control change messages are in charge of effects, color, layers, transparency, outputs, and more, which can effectively turn a drumKAT or malletKAT into the perfect live VJ tool.

FINAL THOUGHTS. There is a genuine symbiotic relationship between the devices in your electronic rig, and getting them to communicate with each other is the "key to the kingdom." Understanding how channel voice messages operate both in your percussion controller and in your sound module will avoid frustration and enable you to truly express yourself through the electronic medium.

MORE ON MIDI

For more information on the MIDI Manufacturers Association (a.k.a. MMA), where companies work together to create the standards that assure compatibility among MIDI products, surf over to midi.org.