MIDI *matters*

FEATURE

Photography *Alan Beeson*

In part two of our series aimed at demystifying MIDI for drummers, Rhythm puts you 'in the mode'.

Text by Norman Weinberg

DO YOU CARRY a truckload of gear to each gig, have to worry about changing heads for every important session, and arrive at the venue three hours early just to set up and make sure your drums are in tune? Well, wake up and smell the coffee! MIDI is it!

With MIDI, you simply play a small pad or even your regular acoustic drums, and the most marvelous sounds will come out of the speakers. You can have every sound in the world, and some sounds that are out of this world, right at your sticktips. Nothing up my sleeve, no mirrors or hidden cameras, just MIDI. You might be asking yourself, "Great, where can I get one?"

Well, the bad news is that you can't just go out to your local music store and buy a MIDI. MIDI is not an instrument, a drum or drum machine, a reverb unit, an audio mixer, or anything that you can physically get your hands on. MIDI is a language, nothing more and nothing less.

The pads of an electronic drumkit or the buttons on the front panel of a drum machine are really triggers. Triggers sense the vibration when you hit the pad, and send out a corresponding electrical spike. The electrical signal can then be read by a trigger-to-MIDI converter, which generates MIDI data when it receives a particular voltage. The MIDI data is then used to instruct the sound source to play the sounds that you hear. By playing a synthesizer, sampler, drum machine, or tone generator from your pads, you enter the world of MIDI.

WHEN TWO PEOPLE try to communicate with each other, they need a common language. If you speak French, while your friend speaks Russian, the two of you are not going to be able to communicate very well. But if you both spoke Italian, then you could make a mutual decision that when you talk to each other, you do so in Italian. This way, both of you know the letters, words, and sentence structure required to clearly state your thoughts.

The common language of all electronic instruments is computer language, and like it or not, computers have invaded our world. They are visible in every facet of our life. There are computers inside your digital watch, your calculator (where did all the slide rules go anyway?), your microwave, your stereo, and even inside your car. Unlike the invaders in a B movie however, computers can be quite friendly and help us in many ways. Since we use computers inside electronic musical instruments, the MIDI language had to be developed using computer jargon.

MIDI was officially born in August of 1983, as a common language which would allow different types of musical devices to communicate with each other and with other computers. This language (called an interface in computer terms) was developed by a group of people from many different facets of the music business. They all felt that the time had come for a universal standard for electronic musical instruments.

In the English language, we have twenty-six letters that we combine together to express words. Some words use only one letter while others use ten or more. By using different combinations of letters, we can express several hundred thousand different words. In computer language, the language of MIDI, there are only two letters, called 'bits' – which stands for binary digits. We talk of 'digits' because the language is expressed in numbers, and 'binary' because there are only two numbers. At the most basic level, computers do one, and only one thing; They can read whether an electrical switch is turned on or off. The two numbers that make up computer language are 0 and 1. When the microprocessor reads some electrical current, then the number is 0 (current on). When the microprocessor doesn't read any electrical current, the number is 1 (current off).

COMPUTER LANGUAGE IS a little different than English, because a computer must know in advance how

many letters are going to be used in each individual word. Computer words also must all contain the same amount of letters. The words of the MIDI language are made up of the different combinations and permutations of the binary number system. If computers used words of only one letter, then the entire language would consist of only two words (0 and 1). If the MIDI language used words that were two letters long, then there would be four possible words (00, 01, 10, 11). If MIDI words were four bits in length, there would be sixteen distinctly different words. Now, perhaps, you are getting the picture. When the bits are put together to form a complete word, they take on a new name called bytes. In other words, a bit is a single digit (0 or 1), while a byte is a combination of bits.

MIDI uses words that are always eight bits in length. If you pull out your trusty pocket calculator, you'll find that there are 256 different permutations possible using eight bits, ranging from 00000000 to 11111111. While 256 different words is certainly a lot better than two, it is hardly a large enough vocabulary to express a musical performance, which is exactly what the MIDI language does. In a nutshell, MIDI identifies and labels each and every aspect of a musical performance, and turns everything into combinations of the binary numbers (0 and 1) that microprocessors can understand.

Back at the beginning of this article, I mentioned that you can simply strike a pad, and trigger the synthesizer to play its sounds. However, you want to be able to tell the microprocessor which note to play, and how loud to play it. You might even want to convey other information about the stroke. What if you want your snare pad to play a sampler, and your kick pad to play a drum machine? We are quickly running out of MIDI vocabulary! It seems that 256 words are just not enough. We need more words!!

The solution is both extremely simple and elegant. We just combine the 256 different words together into a sentence, just like we do in English. In MIDI, a sentence (or a full command) may take from one byte to three or more, before it is complete.

What we are going to do is split the 256 different words into two classifications. Let's say that we have 128 different nouns that label different types of animals, such as dog, cat, fish, gerbil, etc. Let's say we also have 128 different adjectives that describe the noun, such as big, smelly, small, green, etc. If we tried to talk to someone, and only used a single word sentence, then our language would not be very expressive. Now let's say that sentences in our language consist of two words, the first being the noun, and the second the adjective. We could say dog green, cat green, fish green, and gerbil green, and we could say dog smelly, cat smelly, fish smelly, and gerbil smelly. Now our language can be much more expressive. Instead of the original 256 different words, there are now 16,384 combinations of nouns and adjectives.

In MIDI terms, any byte that begins with the number 1 in the first position (1001 0000) is called a status byte. Any number that has a 0 in the first position (0011 1111) is a data byte. Status bytes are related to the noun in the example in that they indicate a particular performance parameter command, such as Note On, Note Off, Pedal etc. Data bytes are like the adjectives and express an amount or a level of the status parameter. If the status byte says Note On, then the data byte indicates which note is turned on. If the status byte says a pedal has been pushed, the data byte indicates which pedal. Some MIDI commands require two data bytes following the status byte. For example, a Note On status byte could be followed by one data byte expressing which note and another data byte which says how loud.

At the most basic level, computers do one, and only one thing; They can read whether an electrical switch is turned on or off . . .

SINCE MIDI IS a language of electronic signals, when computers and drum machines and synthesizers and electronic drumkits speak MIDI to each other, there must be a way to physically carry the electrical signals from one machine to another. Well, the MIDI language moves over a cable with a five-pin DIN plug at each end – much like any other electrical signal. Traditionally, computers and microprocessors use two different styles of sending their signals. One way is called parallel transmission, where each bit moves down a separate wire. Since MIDI uses eight-bit bytes, parallel transmission would require a cable with eight separate wires. The other common type of transmission is called serial transmission, where each bit is sent over the cable one at a time. Serial was adopted for MIDI transmission primarily because it is less expensive.

It's hard to believe, but MIDI moves down the cable at a speed of 31,250 bits per second! That's pretty fast; for every second of time, MIDI is sending and receiving over thirty-one thousand single digits.

Okay, now we've got MIDI messages which make up certain commands, moving over a cable, fast. Is that all there is to it? Well, yes and no. There are two other parts of the MIDI language that are critical to understanding how MIDI works: MIDI channels and MIDI modes.

Let's take a look at MIDI channels first. There are 16 different MIDI channels, but they are not at all like audio channels. In your home stereo system, there is a left and a right channel, each of which requires a separate cable. The signals for these two channels cannot travel over the same wire. If you take a look at the back of a mixing board, you'll see a separate set of wires for each channel. A 32-track board, for example, requires 32 cables in order to keep each audio signal separate.

MIDI channels are a little more like the channels on your TV set. If you've got cable TV, then all the channels that your cable company distributes are sent along the same wire. Your TV set is receiving all of the channels at the same time, but you can decide which channel you want to watch by dialing it in or pushing a button on the front panel of the TV. This is the channel that you are watching, even though a hundred or more channels may share the same wire coming into your home.

Another analogy to MIDI channels is the walkie-talkie. Let's pretend that you and your friend have a walkie-talkie and you're standing at opposite ends of a football stadium. Each walkie-talkie has a selector dial on the front which selects any of 16 different channels. In order to communicate with each other, you must be on the same channel. If you send a transmission over walkie-talkie channel four, and your friend is listening to walkie-talkie channel seven, he isn't going to hear anything. But if he sets his walkie-talkie to listen to channel four, your voice will come through and your communication will be complete. When you connect two MIDI instruments together, they must be set to send and receive on the same channel or the MIDI commands will not be heard. All MIDI instruments allow the musician to select the receiving channel (this is the first thing to check if you can't hear your sounds).

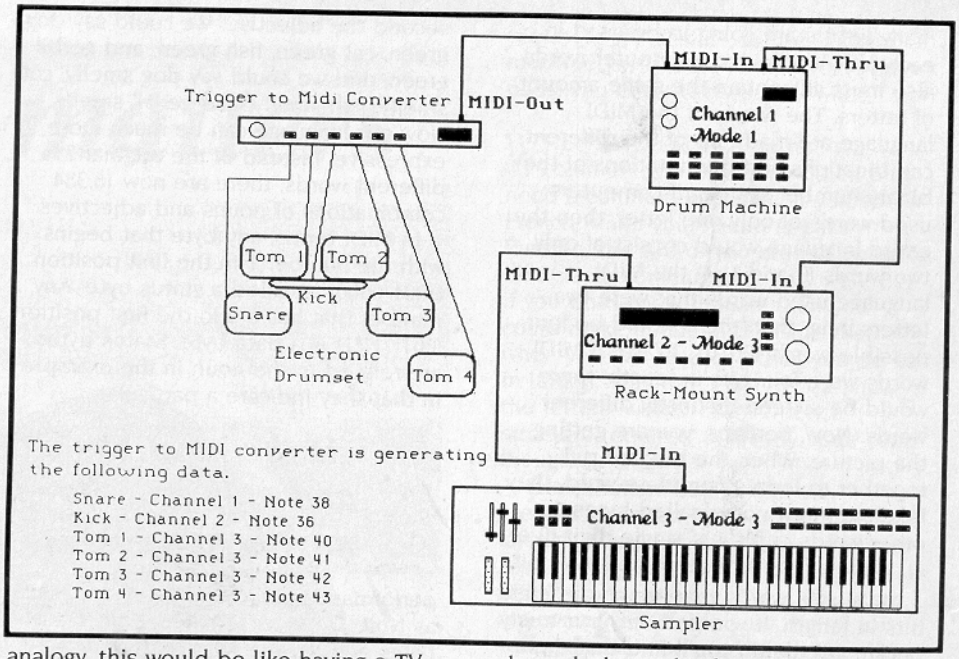
I SORT OF lied to you a little while ago when I said that there were 128 different status bytes that label different parts of your musical performance. If you stayed up all night, you probably couldn't think of 128 different aspects of a performance. The people that put the MIDI language together couldn't come up with 128 either. In fact, there are only seven different status bytes that relate to

FEATURE

musical performance: Note On, Note Off, Poly Key Pressure, Mono Key Pressure, Control Change, Program Change, and Pitch Wheel Change (this relates to pitch bend wheels found on keyboard synthesisers). Why are there only seven? Because each of these status bytes is also tagged with additional bits that determine which of the 16 MIDI channels will carry the signal (among other things).

MIDI modes are something completely different. They can easily be characterized as 'styles of listening'. MIDI modes are only important to the listening device. Since MIDI is a communication, it always takes at least two devices to communicate. The device that is sending the MIDI messages is called the Master and this is the one that is doing the talking. The device that is receiving the MIDI messages is the Slave and is doing the listening. Because the MIDI modes are different ways to listen to the MIDI messages, they don't have anything to do with the Master device.

There are four different ways that a Slave device can receive MIDI messages. These four modes are a result of combining three different types of MIDI mode messages. The first mode message is known as 'Omni'. The Omni message tells the receiving device to listen to all 16 MIDI channels at once. Going back to the TV channels analogy, this would be like having a TV set that could split itself into 16 different screens so that you could watch all the channels at the same time! Pretty neat, huh? The ability to listen to all MIDI channels can either be turned on (Omni on), or off (Omni off). If Omni is turned off, then the receiving device will only listen to a single MIDI channel (the one it is set to) and ignore all the information on any other channel.



The other two MIDI mode messages are 'Poly' and 'Mono'. These two messages are mutually exclusive, that is, selecting Poly will cancel Mono and vice versa. When a listening instrument is in Poly mode, it has the ability to play more than one sound at a time (polyphonic). When an instrument is listening in Mono mode, it can only play one note at a time.

When you combine these three different commands (Omni On, Omni Off, Poly/Mono), you end up with four modes. MIDI modes can be a little confusing because different manufacturers these days are claiming that their instruments have five or six different modes. However, only four modes have been recognized and defined by the MIDI specification, no matter what a brochure might say.

THE FIRST MODE (Mode One) is Omni On/Poly. In this mode, an instrument will be listening to all sixteen MIDI channels at one time and responding to the commands it receives in a polyphonic manner. You might use Mode One when you want a drum machine to receive MIDI commands from many different sources at once (say an electronic kit, Octapad, and a synth).

Mode Two is Omni On/Mono. This mode really isn't used very often and you can probably see why. A device in this mode will listen to all sixteen channels, but only play one note at a time. Why anyone would want to do this is beyond me! I've never met anyone who uses this mode, but it's there if you need it.

Mode Three is Omni Off/Poly. This is perhaps the most common of all the modes. As your MIDI system grows, you will find yourself using Mode Three a lot. In this mode, an instrument will only listen to a single MIDI channel and it will respond in a polyphonic manner.

Mode Four is Poly Off/Mono. Instruments set to listen to MIDI commands in this mode will be listening to only one channel, and playing only one note at a time. While applications for this mode may not be readily apparent at first glance, there are many times when it can be extremely useful. We'll take a look at some special uses of Mono Mode next month.

Okay, now that we've got a couple of the basics taken care of, let's see how an electronic drumset player can make use of the MIDI commands to do something really slick. Looking at the example, you see a set of electronic drums with six pads connected to a trigger-to-MIDI converter. The converter is connected to three different devices: a drum machine, sampler, and a rack-mount synth. In this example, all pads are going to play sounds from the drum machine, the tom-tom pads are also going to play the sampler, and the kick pad is going to play this really hip explosion sound from the synth. Check it out! (Of course, you've got to make sure those sounds are in your synth and sampler.)

Next month, we'll take a look at Ins, Outs, and Thrus; some more MIDI messages in detail (as well as how to use them), and discuss ways that you can control how the trigger pads respond to your touch. Stay tuned (MIDI channel six, mode three)! R