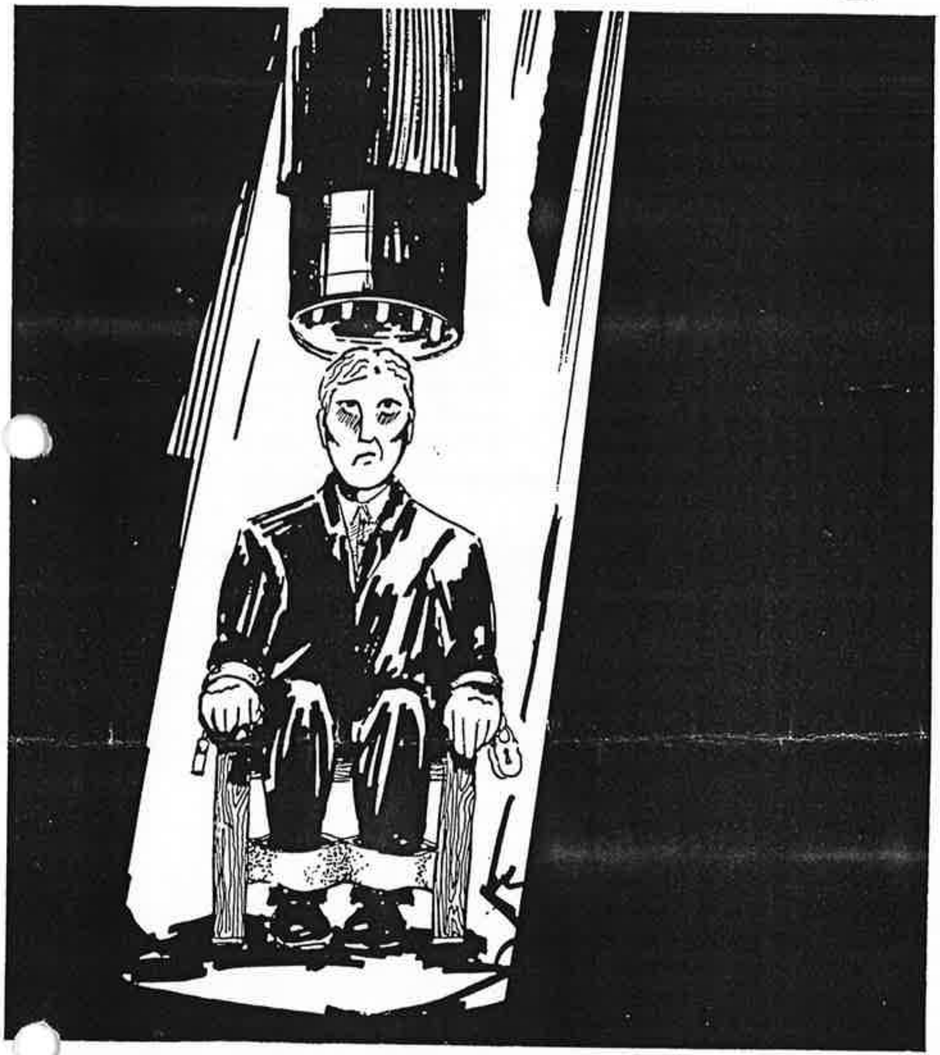
MIDI for the Unwilling



If you have an innate fear of technology, and words like bit and byte leave you cold, then here's the guide to using MIDI that you've been waiting for.

Meeting MIDI for the first time can be intimidating, and the subject is more often than not made more confusing by books and articles that purport to explain how it works. I've decided to take a totally different approach - I've absolutely no intention of telling you how it works and I've no real wish to explain it! Nobody has explained how gravity works yet we stick to the floor just the same! What you really need to know is how to use it which is a totally different ball game. You don't need to know how a digital telephone exchange works to make a long distance phone call and you don't need a degree in electronic engineering to flip through the channels on your TV set so why on earth should you be expected to know how MIDI works before you can make use of it?

Having decided what we don't really need to know, it's time to decide what we do need to know. We may not know how MIDI works but we need to know a little about what it does. Even this isn't as simple as it seems because MIDI can do several different things, so I'll start with the most useful and leave the more esoteric stuff 'till later.

THE IDEA BEHIND MIDI

You've probably heard that MIDI (Musical Instrument Digital Interface) is an electronic system that allows information to be passed from one MIDI compatible piece of gear to another - other explanations say it's a system that allows pieces of equipment to communicate with each other. To the uninitiated this is totally meaningless, even though it is more or less true. Let's take this business of allowing instruments to communicate with each other - this might suggest that they do this spontaneously, that they are in some way intelligent. They are not! And what is this mysterious information anyway: do your synths and sequencers stay up at night swapping share price information or discussing the rising interest rate? I don't think so.

Before going on, let's just look at a very simple MIDI setup and see what's going on. Imagine two MIDI keyboard synths connected as in Figure 1. The MIDI Out of the first keyboard (which we'll call the master) is connected to the MIDI In of the second (which we'll call the slave) by means of a MIDI lead, a fairly ordinary piece of cable with a five-pin DIN plug at each end.

You may have heard of things called MIDI channels, and such mystery as surrounds these will be dispelled shortly. But in order to preserve a sense of wonderment and anticipation, I'll simply say that for the example we are about to perform, the MIDI transmit channel of our master keyboard must be set to the same as the MIDI receive channel on our slave. This isn't 100% true because we haven't discussed Omni mode yet, but as with any form of learning process, we have to start off with a simplistic model we can come to grips with and then apologise for the lies later. This article will contain many such lies - but they won't do you any lasting harm.

THE FIRST STEP

If we now take our right index finger and use it to prod middle C on the master keyboard, the slave machine should also play middle C. Round of applause and break for tea - but don't take your finger off the key yet!

Now, if this was all MIDI could do, the designers needn't have bothered, but fortunately, MIDI can do all kinds of musically useful things as we shall see shortly. However, this simple little example can teach us most of what we want to know about MIDI. Let's look firstly at this so-called 'information' that travels along the MIDI cable from one machine to another.

As mentioned earlier, we don't really need to know anything about the computer language used to carry these messages, that's the computer's job and it will perform it equally well whether we understand how it does it or not. But we do need to create a mental picture of that information so that we can exploit it. The first important fact is that MIDI messages don't carry sounds, they carry instructions. An example? If you open a recipe book, you'll find that it tells you how to make a curry - but it doesn't actually contain one!

When you press a key on your master keyboard, the electronic gubbins inside that box knows which key you've pressed and how hard you pressed it (assuming the keyboard is velocity sensitive). Normally this information is passed directly to the internal sound-generating hardware and the appropriate note plays with whatever sound you have selected. With MIDI, the same information is also turned into a neat little parcel of MIDI code and transmitted down your MIDI cable at a very high speed. When the slave synth (or it could be a sound module with no keyboard at all), receives this message, its MIDI circuitry reads the contents and carries out the order unquestioningly.

So far then, the message relates to the note to be played and how hard it should be played. This note then continues to sound indefinitely until told to stop. When you release the key on the master keyboard, a second message is sent which the slave interprets as an order to turn the note off. If your master synth transmits additional information such as afterpressure, then this is also transmitted over MIDI along with the other note information, but it can only be interpreted by the slave keyboard if it is designed to do so. It's rather like television: all aerials receive colour transmissions, but if the TV at the end of the aerial lead is a black and white set, then you'll only get a black and white picture. Logical so far?

MIDI messages travel down the MIDI cable one after the other - MIDI can only turn notes on and off one at a time, so you might reasonably ask if this causes problems when you try to play a chord. Fortunately, MIDI is so fast that all the notes of, say, an eight-note chord, can be turned on in just a few thousandths of a second, which is fast enough to fool the human ear into thinking they all came on together. That's not to say that MIDI delays can't happen though, if you try to turn a great number of notes on at the same time, you'll eventually end up with a delay that you can hear. Fortunately, not all notes fall on the same beat and not many songs have two dozen instruments playing two-handed chords at the same time!

THE SEQUENCER

Because MIDI is a computer-based message system, the messages can be read by, and stored in a computer. All a MIDI sequencer is, is a computer designed to store MIDI messages along with their timing information so that they can be played back at a later time. Remember that the computer is only storing MIDI messages, not sounds, so these messages must be plugged into a MIDI synth when you come to replay your sequence. To do this, the MIDI Out of the sequencer is connected to the MIDI In of the synth you are using to replay the sounds. If you increase the sequencer's tempo, then the messages come out quicker, but the pitch doesn't go up as it would if you speeded a tape up: a MIDI message that says 'play middle C' will always say 'play middle C' regardless of how fast it follows the piece of information that went before it.

From the above, we can immediately see two distinct advantages of using a sequencer: we can alter the tempo of a performance without changing its pitch and we can, retrospectively, choose any sound that our synth is capable of producing. We may have recorded the tune with a piano sound but it can be just as happily replayed using a brass or synthetic sound. Furthermore, if the sequencer software allows you, (and virtually all do to some extent) you can access and edit your MIDI performance to correct mistakes or insert new notes.

But there's much more to a sequencer even than this - there's the possibility of multitrack MIDI recording. And that's where the incredibly simple subject of MIDI channels rears its head.

MIDI CHANNELS

A typical MIDI system will be more complicated than just one master keyboard and one slave synth, especially if you wish to use a sequencer. Even so, the wiring system is very simple and several MIDI instruments can be daisy-chained together with just one MIDI cable joining one to the other. You've probably noted the MIDI In, Out and Thru sockets on the back of various bits of MIDI gear and the exact way in which these are used will be revealed in due course. At the moment, suffice it to say that in a daisy-chained system, the same set of MIDI information arrives at the MIDI In socket of each and every MIDI device, so why don't they all play the same thing? The answer is: MIDI channels.

There are 16 different MIDI channels numbered, logically enough, from 1 to 16. When a MIDI note message is sent, it has a MIDI channel number allocated to it - the transmit channel number of the master device. Think of this like an address label on an envelope and you won't go far wrong. If you send a MIDI message, such as play middle C with a velocity of so and so on MIDI channel 2, then all the synths in the system receive it, but only the one (or ones) set to receive MIDI channel 2 will take any notice of it. It's like a rather polite society where nobody reads any mail not marked for their attention.

It's possible to make use of more than one MIDI channel, even if you don't have a MIDI sequencer. Some keyboards allow you to set MIDI keyboard splits where the keyboard can be divided into two or more zones, each on a different MIDI transmit channel. A simple example would be a split with bass on the left hand and chords on the right. You might then set the left hand side of the keyboard to channel 1 and the right hand side to channel 2. You could then connect two MIDI synths or samplers to your master keyboard, a bass sound on channel 1 and a chordal sound such as strings on channel 2. Figure 2 illustrates this set-up. Now the left hand side of your keyboard will play only bass sounds and the right will play only strings or whatever sound you chose. With a sequencer, things get much more interesting - but no more complicated.

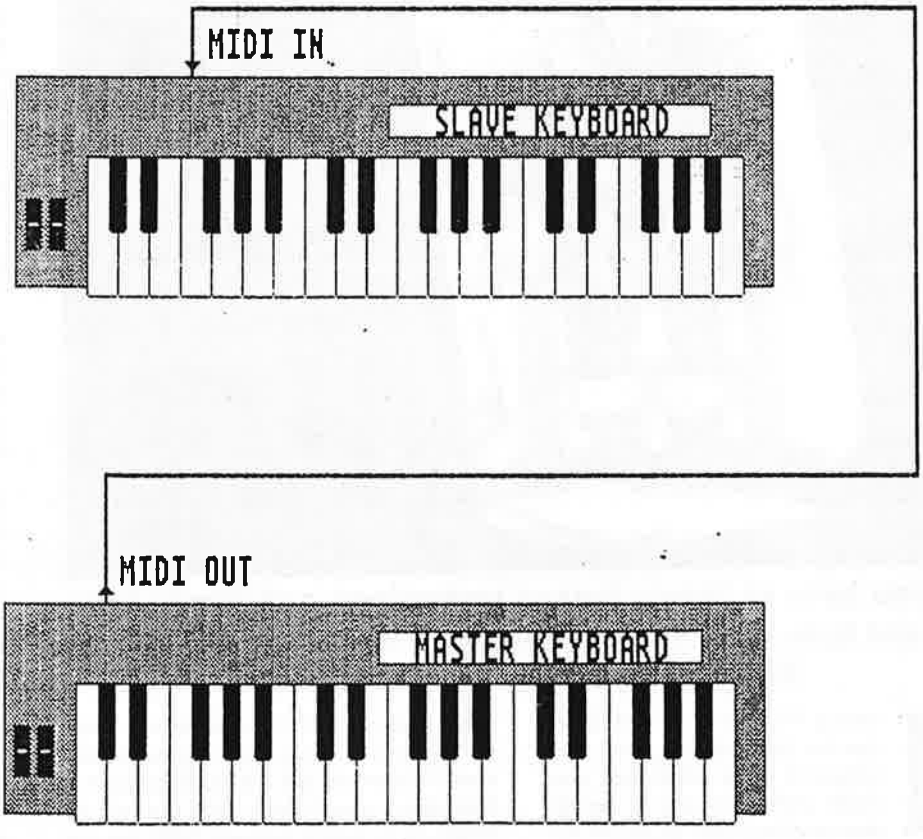


Figure 1. Two keyboards linked via MIDI

As we have 16 MIDI channels, we have the power to send information to 16 different synths, all connected together via MIDI, so a multitrack MIDI sequencer could control all these at once, giving us the ability to build up compositions just as we would using multitrack tape. There are pros and cons though: the pros are the editing facilities, the ability to change tempo, the flexibility of editing and the freedom of choice to change sounds at ►

any time; the cons are that, because you can only record MIDI performance information, not real sounds (unless via your MIDI sampler) you need to have enough MIDI sound modules to play back all the parts you've written into the sequencer.

Both the above limitations have fairly painless solutions. The lack of ability to record real sounds can be got around by running a sequencer side by side with a multitrack tape recorder. Some form of MIDI-to-tape synchroniser will be needed to make the sequencer stay in sync with the music on tape, and you do lose one tape track needed to record a sync code, but you end up with an extremely powerful music production system. Synchronisers have already been covered at some length in previous articles and so I'll not pursue the subject further here.

Before leaving the subject of MIDI channels, you may have heard of something called the Omni mode. This is found in the MIDI receive parameters setup of your synths and should be selected if you want your synth to play all MIDI messages, regardless of what MIDI channel they arrive on. To be honest, Omni isn't much use in sequencer setups as the synth set to Omni will try to play all the parts including drums if there are any! It's of more use to the inept salesman who can plug just about any MIDI source into a piece of gear set to Omni and at least some sound will come out. For this reason, some synths, including Akai and a few budget Yamaha units default to Omni when switched on and Omni needs to be turned off manually before you start work.

In fact there are four MIDI modes which are obtained from the permutations of Omni On or Off and Poly On or Off. Normally, synths default to Poly mode, Omni Off, which is how they are normally used, but Mono mode is useful if you are setting up a MIDI expander to work with a guitar synth. In this case, you can set each guitar string to send on a different channel so that you have, in effect, each string controlling its own monophonic synth. Again, Omni would be left switched off. This has the advantage of allowing independent pitchbend on each string - you've probably noticed that when you use the pitchbend on your synth, all the notes bend together.

MULTITIMBRALITY

The multiple MIDI sound source problem can be got around too - you don't necessarily have to go out and buy 16 different synths. We now have multitimbral sound modules, and for all the fancy descriptions, what this adds up to is a box that can receive several MIDI channels at the same time and pretend to be several synths at the same time. Most models offer up to eight different MIDI-controlled sound sources but you are limited as to the maximum polyphony of such units (often eight or 16 notes) and you may only have a stereo output carrying a mix of all the sounds on the cheaper models. Because of these limitations, you may find that all a single unit can handle is three or four parts at one time. More than this, especially if the music is very busy, and notes start to get cut off early as they make way for new ones. Even so, there are some very cheap multitimbral modules on the market and they really do increase the flexibility of a MIDI system.

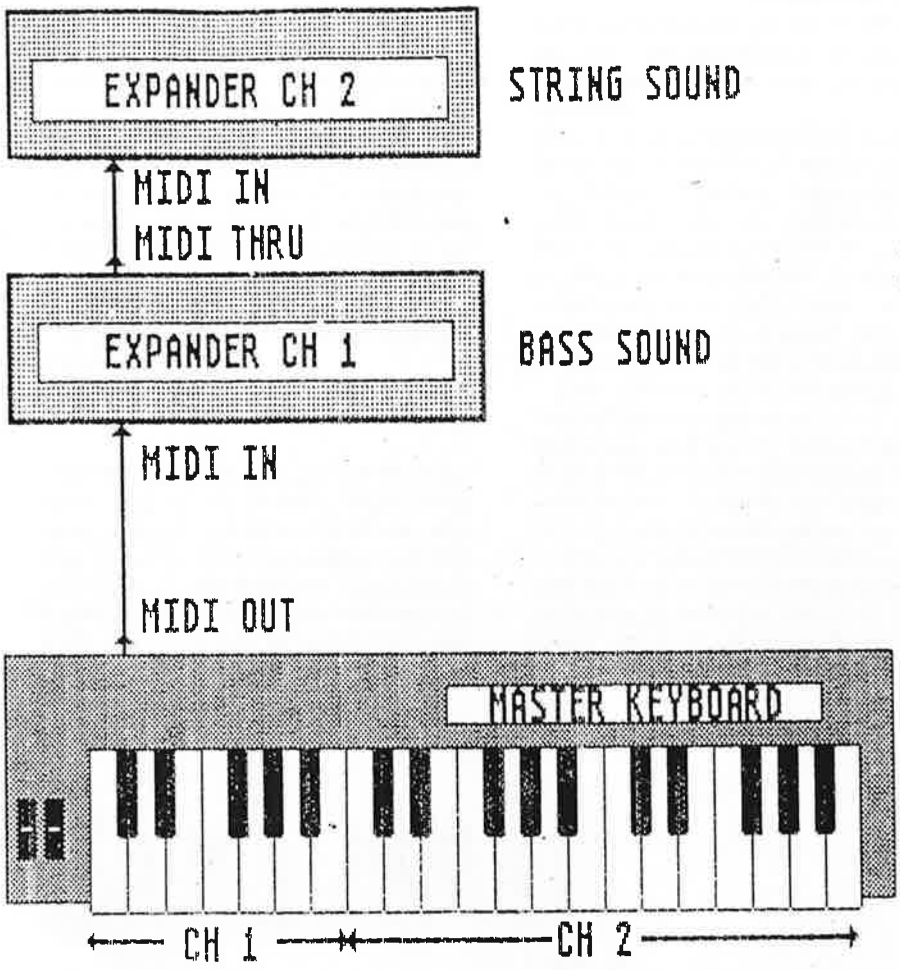


Figure 2. Keyboard split with expanders: bass sound on channel 1 and a chordal sound such as strings on channel 2

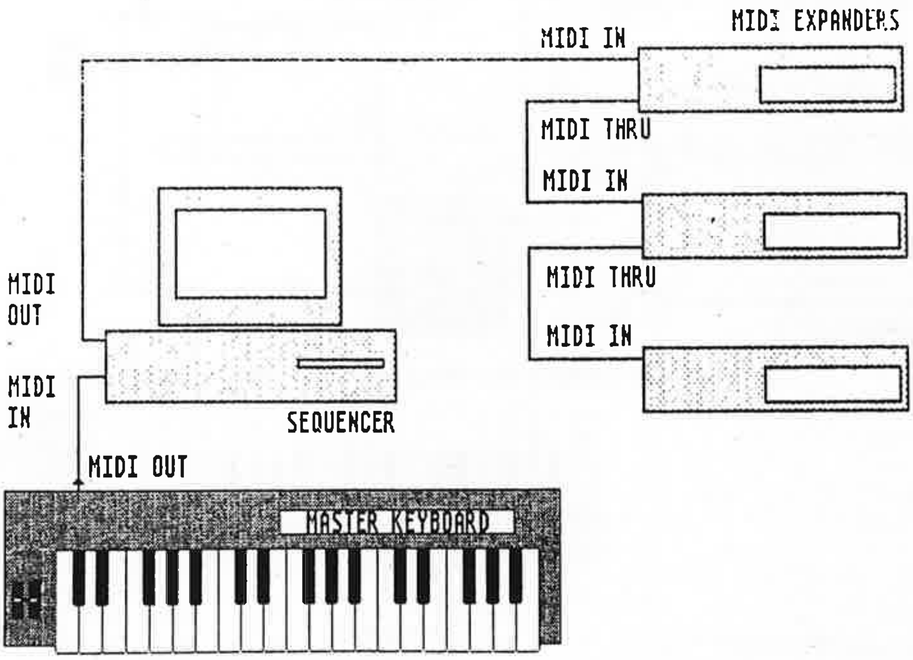


Figure 3. Basic sequencer setup

WHAT ELSE?

There's more to MIDI than sending note information around a system - just about any button or knob you can press or turn on your synth could be given a MIDI code which would then whizz around the system doing your bidding. Virtually all synths will transmit patch changes over MIDI, allowing you to ▶

change the sounds on your expander modules by remote control or from a sequencer, and of course, there are the performance controls such as sustain pedal on/off, pitchbend or modulation depth which are also handled by the system.

There are a couple of minor pitfalls in this area as not all synths work in the same way. A pitchbend of a fifth on one synth might be read as pitchbend of a third on the slave module and unless the receiving synth has the ability to select a different scaling for the bend range (as many now do), then you're stuck with it. Patch changes and controller information are channel specific, just like notes, so they will only affect the expander set to the appropriate MIDI channel.

CONNECTIONS

And now a few words about the MIDI sockets on the back of your synth and how they're used. The three sockets on a complete system are: MIDI In, MIDI Out and MIDI Thru. On some budget systems, Thru may be missing or may be combined with Out. MIDI In is pretty obviously the way to get MIDI messages into a unit so all slave devices in a system must have their MIDI In connected or they won't work. MIDI Thru is a system for passing on the MIDI In signal to another piece of equipment and a simple piece of circuitry is used to beef up the signal so it doesn't get exhausted when several pieces of gear are daisy-chained together. What comes out of MIDI Thru then is essentially the same as what goes into MIDI In.

In simple MIDI systems, there will only be one MIDI Out used and that will be on the master keyboard. MIDI Out is used to send the master information that will be passed onto all the other synths. There are, as always, exceptions. If you have a sequencer, then when you are recording, the master keyboard is in charge - but when you come to play back the sequence, the sequencer calls the shots. To accomplish this, the MIDI Out socket of a sequencer is usually designed to behave as a Thru when you're recording and as an Out when you're playing back. A basic sequencer-based setup is shown in Figure 3.

Problems arise if too many synths are daisy-chained together because the MIDI messages get corrupted in just the same way as a phrase passed along a line of people at a party. In the case of MIDI, this can mean stuck or missing notes. Four or five units is about as many as you can get away with safely. The solution is to use a MIDI Thru box which splits the MIDI signal into several 'first-hand' copies, each of which is routed directly to the expander modules. Figure 4 shows how a MIDI Thru box is incorporated into a system.

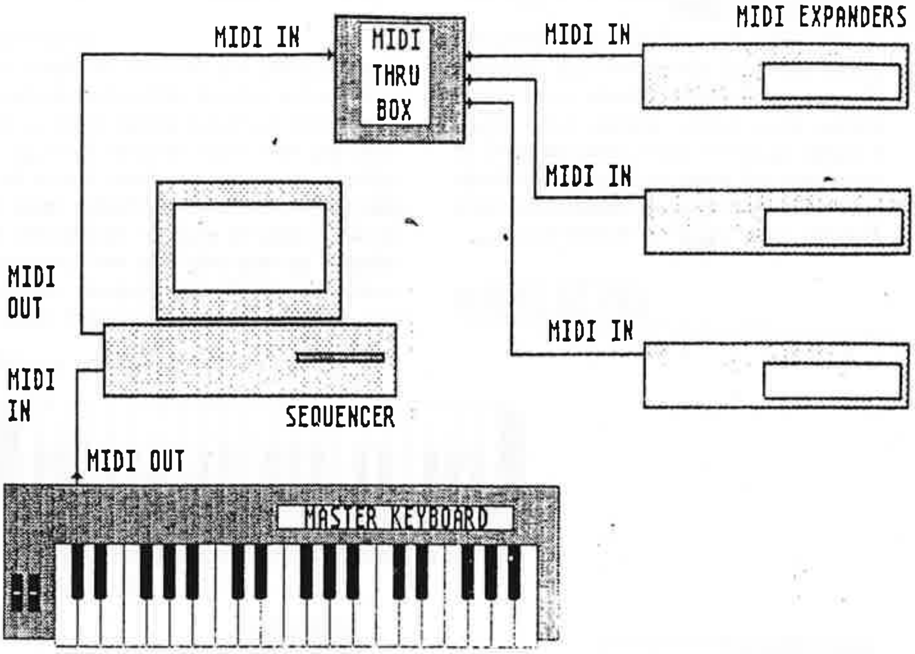


Figure 4. Shows how a MIDI Thru box is incorporated

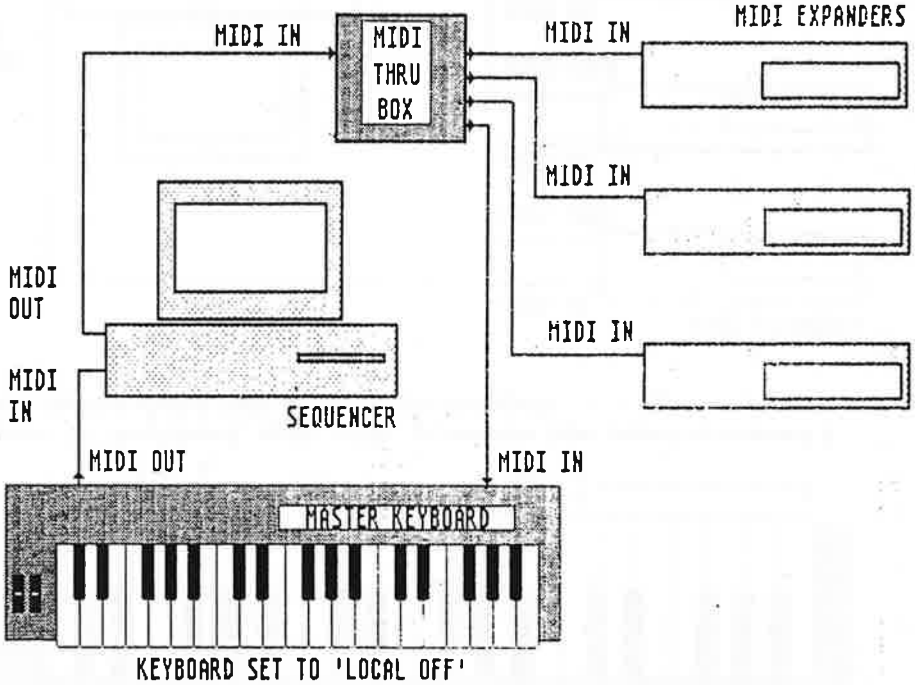


Figure 5. Sequencer system with Local Off

So, what happens if you want to have two master keyboards plugged into a sequencer at the same time so that two of you can play and record together. Unfortunately, we can't just join two MIDI cables together as the resulting mess of data would be meaningless and you might even damage your instruments. We have to use a special MIDI merge unit which properly integrates the two data streams. Some sequencers have these built in, but in most cases, you'll need to buy a separate MIDI merge unit.

LOCAL OFF

A keyboard synth can be thought of as a master keyboard and a separate MIDI sound module sharing one box where the keyboard is internally connected to the sound section. This can cause problems when you want to set up a sequencer system, because if you feed the output of the sequncer back to the MIDI In of the keyboard, then you have a MIDI feedback loop. What happens is this: You play a note on the keyboard/synth, it sends a MIDI message via the MIDI Out socket and this goes into the sequencer's In socket. The same signal will appear at the MIDI Out/Thru of the sequencer and is then fed back to the keyboard/synth after a delay of a tiny fraction of a second. The keyboard/synth responds to this note information and plays the note again sending yet another MIDI message to the sequencer and round it goes, usually resulting in a horrible noise.

The easiest way to avoid this undesirable state of affairs is to separate the keyboard from its sound section by using the Local Off switch - if your keyboard has one. This may be a physical switch or may be an option in the MIDI setup menu. Now the keyboard doesn't talk directly to the synth section, it has to go via the sequencer so there's no loop and no more problem. Figure 5 illustrates this point.

If you don't have a Local Off, then you can set most sequencers to block the MIDI Thru information on a channel of ▶

your choice. This breaks the loop again, but it means that you have to turn down the sound on your synth when you're playing new parts destined for other synth modules if you don't want to hear it.

DRUM MACHINES

MIDI devices that have a built-in tempo generator of some kind, such as drum machines or multitrack sequencers, don't just send out note, performance controller and patch change information - they also send out what is known as a MIDI Clock. This is the MIDI equivalent of a metronome and it allows drum machines to be run in perfect sync with sequencers or other drum machines. As well as the tempo information, MIDI also tells the slave device when to start and when to stop.

You don't have to use a drum machine's internal tempo generator and programming system though, you can connect it and use it just like any other synth expander module. If you play your master keyboard now, you'll find that different drum sounds appear on different keys while some keys play nothing at all. This is because there are nearly always more keys than drum sounds. You can write your drum patterns directly into a sequencer by tapping in the rhythms on your keyboard, usually layering a couple of drums at a time, and this way, you're just using your drum machine as a dumb box of voices that can be activated via MIDI. Of course, you'll need to go into the MIDI edit page of your drum machine and turn off the MIDI Start facility, otherwise, whatever internal patterns it has will start to play as soon as you run your sequencer!

AND THE REST

So far, I've only looked at the very basic aspects of MIDI and even so, I've had to gloss over quite a few of the facts to protect the innocent. There's quite a bit more lurking underneath but if you don't need to use it yet, don't let it worry you.

I'll finish off by mentioning a few other attributes of MIDI which you may or may not feel you need to explore more deeply, so that when you come to read a more conventional MIDI article, it will at least make sense.

You may have heard of System Exclusive or SysEx MIDI information. Normally this doesn't much concern the average MIDI user and is more often used by the people who write editing software. It is a way of accessing the editing parameters within a synth, sampler or drum machine via MIDI, but as no two synths have the same controls, the codes are exclusive to specific manufacturers and models, hence the name. SysEx can also be used to output the whole memory of a machine to a MIDI storage device or cassette tape and is often used for dumping drum patterns to tape or for saving libraries of synth sounds. As far as the user is concerned, you just follow the procedure in the manual and don't worry about how it works.

Then there are MIDI Song Position Pointers or SPPs for short. Again you don't need to worry about them but it's nice to know what they do for you. Essentially, they are part of the MIDI code generated and read by modern sequencers and drum machines so that they can be started up from any point within a song, not just at the beginning. This is great if you have a sequencer sync'd to a tape machine, because a sync unit that handles SPPs will allow you to run the tape from any position in the song and the sequencer will be right there with you. There's also MIDI Time Code which is MIDI's way of working with SMPTE time code and we'll be hearing a lot more about that in the future.

That's as far as I want to take the subject for now. This guide is grossly over-simplified, full of white lies and hopelessly incomplete, but if it helps someone who was previously scared stiff of the subject to make use of MIDI and get results, then I make no apologies. And, having read so far, you should now be able to face the more serious texts on MIDI with much less trepidation.

Text: **Paul White**