MIDI *matters*

Sequencing in Sync

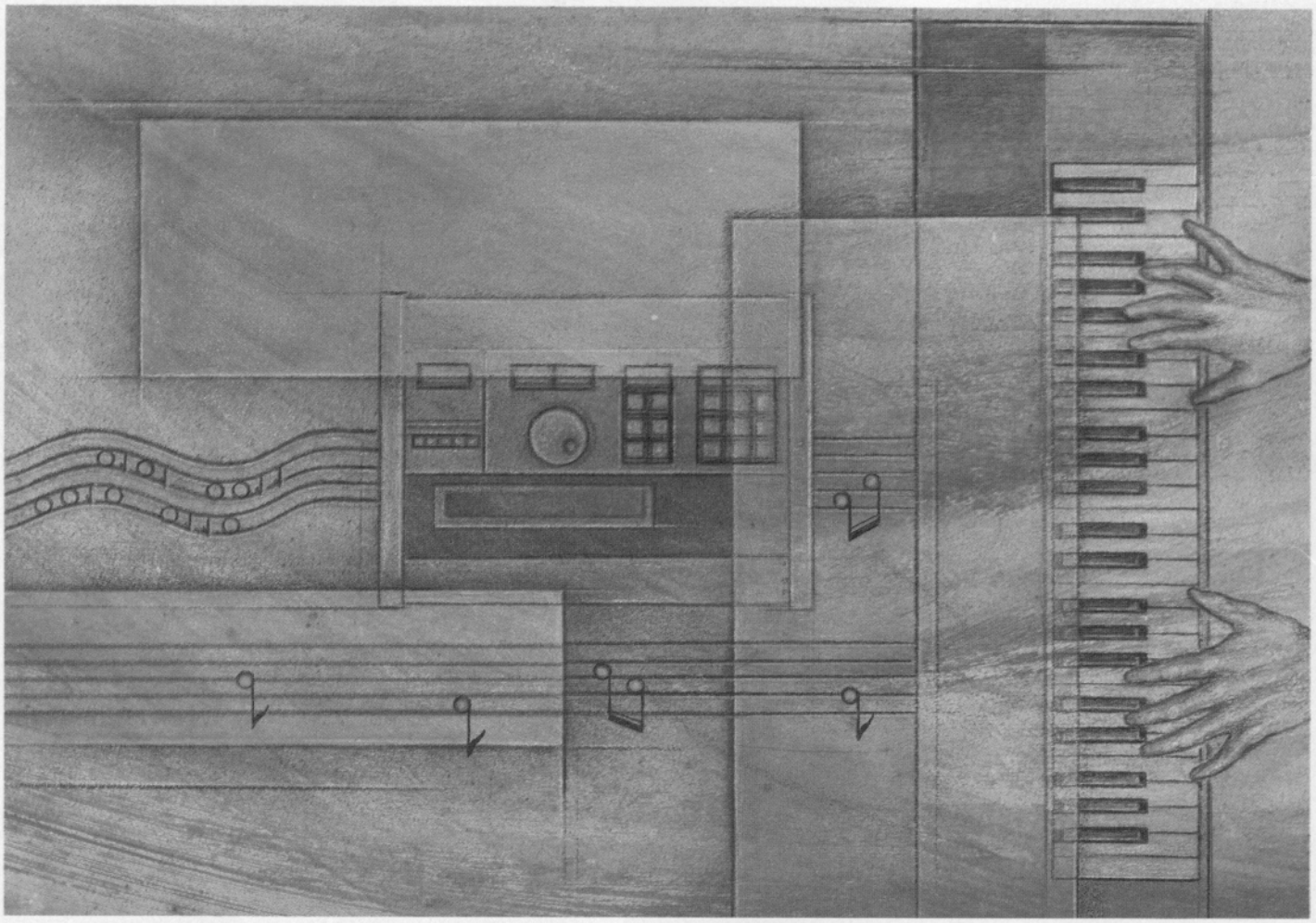


Illustration *Rick Lohnes*

Synchronizing your sequencer and all your external MIDI devices is easy with time code . . .

Text by Norman Weinberg.

I'M SURE YOU'VE heard about sequencers. But you may not know just what they are or how they work. Sequencers are really nothing more than fancy digital versions of tape recorders. They are called sequencers because they record the sequence, or order, of a MIDI data stream. MIDI information moves too fast to be recorded on normal audio tape – any signal that clips by at 31,250 bits per second must use a dedicated computer (or computer chip) in order to record the data. Sequencers can be found as an added feature in many keyboard synths and rack mounted units, and as dedicated software programs for computers. In fact, if you own a drum machine, you already own a sequencer. Whenever you set a drum machine up to record a pattern, you are sequencing the MIDI instructions to play certain notes at certain times.

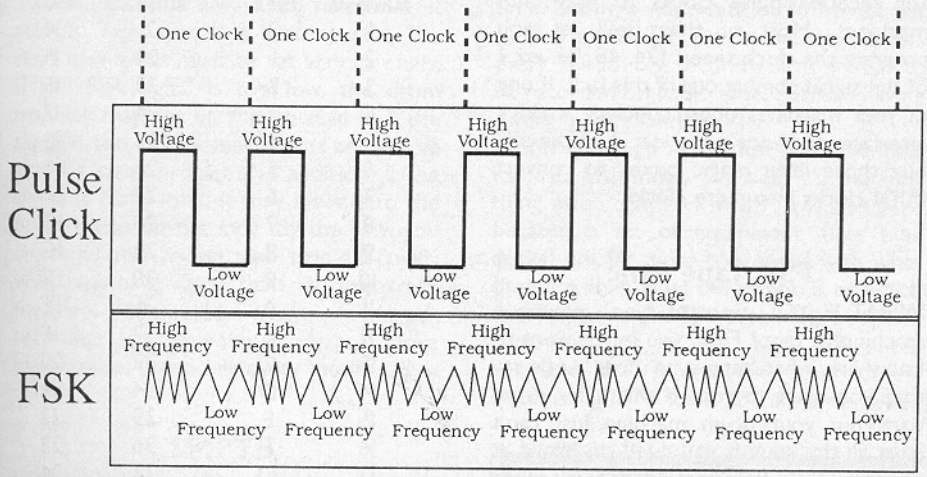
Before we get any further into sequencers, you must realize one thing: sequencers **do not** record any sound at all! Sequencers do not record music, they only record the MIDI data stream, which consists of instructions for other devices. If you don't hook up a sequencer to an instrument, you're not going to hear anything. In order to make music, sequencers need to send the instructions to another device. That's why you've never seen a sequencer with a headphone jack.

Tape vs Sequencing

OK, SO WHAT'S the big deal with sequencers? Everything! Once the sequencer has recorded the MIDI data stream in its digital state, you can do many marvelous things, all of which fall into a set of functions called editing.

With audio tape you can only edit in a very limited fashion. For example, you can ▶

Example #1



► take a razor blade and get rid of some of the tape (called splicing). You can take one piece of tape and graft it to another to create a longer tape. You can punch in and punch out, and you can overdub. Yes, you can also play the tape backwards, crush it between your hands, or scratch off part of the magnetic coating with a fork – but these techniques are special effects, not real editing.

Once the MIDI data stream is inside the sequencer, you can do many different types of editing. You can still splice material out of the data stream, slide material over, or even repeat (loop) certain sections over and over. Not only do sequencers allow these basic edits, but you now have the ability to make them at a resolution of 1/1000th of a second. Try doing that with a razor blade! You can also overdub to your heart's content. The sequencer that I'm using will handle over 300 tracks (so the manual says, but I've never needed over forty). However, this is just the tip of the iceberg; sequencers can do several tricks that just can't be done with conventional audio tape.

Beyond Audio Tape

SEQUENCERS CAN PERFORM some neat tricks – like playing your music faster or slower without changing the pitch. When you change tempo, the MIDI messages are sent from the sequencer faster or slower, the messages themselves are unchanged. If you want the song (or any certain section) to play at a different pitch, you can transpose anything to any key. This is really a very simple task for computers. If you ask a sequencer to bring everything up a half-step, the computer chip just adds 'one' to every MIDI note number. Middle C (MIDI note number 60) would become C sharp (note number 61).

Besides playing around with the pitch, sequencers let you edit any note's (or groups of note's) velocity level, duration, position in time, or just about anything else. Since sequencers record the MIDI data stream in its entirety, they will even record and allow you to edit pitch bend information, controller data, patch change commands, aftertouch, mode messages, and system exclusive data. Think about it.

Some sequencers (most often only the computer software type) will let you edit with a high degree of sophistication. Taking just velocity, for example, you can make all the notes the same velocity, add or subtract a certain number to all velocities, add or subtract a certain percentage to all velocities, change smoothly from one velocity value to another, change smoothly from one percentage to another, and so on. In short, a sequencer can be your MIDI workstation and master controller and be placed in charge of all the other devices and instruments in your system.

Clocking In

WHENEVER YOU CONNECT two or more sequencers together (drum machines, computers, or synths with internal sequencers) you've got to have some way to synchronize them to the same clock. Every electronic system can have one and only one master clock. When you sync several MIDI devices together, they perform in perfect ensemble, in essence creating a single instrument which has the power of many.

As promised, we're also going to get into the new MIDI Time Code this month. MTC (as they say in the trades) is pretty heady stuff, so it might be easier to understand if we cover the other timing standards that are used with electronic music first.

Last month, we discussed the MIDI timing clock message which can be used to sync-lock two MIDI devices (such as two drum machines). But what are you going to do if you've just got to use some pre-MIDI gear in your system? Just throw it away? Of course not (if you do plan to chuck it in the trash, call me first). ►

Back before MIDI, drum machines used something called 'clocks' to stay together. Each unit would have a pair of jacks called clock out, and clock in. When you connected a cable from the clock out of Drum Machine A to the clock in of Drum Machine B, Machine B would 'slave' to Machine A. In other words, the slave machine would use the clock signals from Machine A instead of its own internal clock. This always worked out great, except for a few problems. Some companies built drum machines that used a rate of 24 clocks per quarter note while other companies used 48 clocks per quarter.

Whenever you try to connect two different clock rates, you're just asking for trouble. If you had a 48 clock master connected to a 24 clock slave, the slave machine would play everything twice as fast. Since there were about four major clock rates running around, a drummer who owned gear by several different manufactures, was in **big** trouble. In order to solve this problem, some companies invented things called Clock Interfaces. These little boxes could read any type of clock speed, and convert it into any other clock speed. Want to connect a 24 clock unit to a 96 clock unit? No problem, just buy one of these little gizmos and you're home free.

The end result is that all but the cheapest (and some of them too!) drum machines offer clock in/out jacks to send (or receive) pulse clocks to non-MIDI machines. Most of them will let you program the clock speed (24, 48, 96, etc.) of the signal coming out of this jack. If one of your machines doesn't include a clock interface, all is not quite lost. You can still buy those little magic boxes to convert MIDI clocks into pulse clocks.

FSK Tape Sync

WHAT IF YOU want to sync your drum machine to tape? First, you might want to know just what 'tape sync' does. Let's say that you want to record this killer drum part, but your drum machine just can't have all the sounds you need on board at the same time (this beat needs forty-seven different drums). Here's what you do. Record as many sounds as you can on one of the tape's tracks, then load the remaining sounds into your drum machine, and overdub the extra instruments. How do you synchronize the overdub with the track already on the tape? With tape sync.

Tape sync uses a particular kind of clock which is well suited to recording onto tape. Instead of alternating high and low voltages (like pulse clocks), tape sync makes use of alternating high and low frequencies (see example number 1). Tape machines do a much better job at recording pitches than raw voltages. These alternating pitches are called FSK (Frequency Shift Key), and many drum machines (particularly Roland units) have FSK jacks on the back panel. You connect a cable from the tape out jack to an input on the tape deck in order to 'write' the sync signal to the tape. When you want the drum machine to use this signal as the master clock, you connect a cable from the tape's output to the tape in ►

Example #2

Base 10	Hex	Base 10	Hex
1	0	21	14
2	1	22	15
3	2	23	16
4	3	24	17
5	4	25	18
6	5	26	19
7	6	27	1A
8	7	28	1B
9	8	29	1C
10	9	30	1D
11	A	31	1E
12	B	32	1F
13	C	33	20
14	D	34	21
15	E	35	22
16	F	36	23
17	10	37	24
18	11	38	25
19	12	39	26
20	13	40	27

► jack of the drum machine. What could be easier?

Well, it's not always as simple as it sounds. Tape sync is a little touchy. The recording levels must be set very carefully. If the FSK signal is too low, the drum machine may not be able to read it. If the signal is too loud, it may distort and you've got the same problem. In addition, if the signal is really loud, it may bleed into the other tracks on the tape. By the way, you won't be able to use tape sync effectively with anything less than a four-track machine. But take heart, four-track recorders are becoming more and more inexpensive, practically every month.

SMPTE

SO FAR WE'VE taken a brief look at three different types of synchronization systems: MIDI clocks, pulse clocks, and FSK. Now comes the big one, the one that ate Wyoming, the heavy-weight champ, the ultimate in sync – SMPTE. SMPTE stands for the Society of Motion Picture and Television Engineers, and while this is really the name of an organization, the time code that they approved back in 1969 has taken their name.

SMPTE is based on the universal idea that time is measured in hours, minutes, and seconds. In addition, because this time code was developed for cinema, seconds are divided into frames (film and video frames). Just like MIDI, SMPTE is a digital language that is used to interface two or more machines together. One of the differences between these two electronic languages is the speed of communication. You might remember that MIDI moves at 31,250 bits per second. Even if you had a machine which converted the on and off voltages of MIDI into high and low frequencies (to let you record the MIDI data stream on a tape deck), you would need a tape deck with a frequency response of over 31kHz. Since the best machines on the market can't record much above 22kHz, you're out of luck. So MIDI cannot be recorded to regular audio tape, but SMPTE can.

There are four different SMPTE rates that are used around the world. One runs at 24 frames per second (actual film – cinema), while one runs at 25 frames per second (for both black and white and color TV for most of the world). Then there's 30 frames per second (American black and white TV), and finally what is called 30 drop frame (for American color TV). Thirty drop frame is a weird animal. It really runs at 29.97 frames per second. To work this out, it counts off 30 frames per second and 'drops' or leaves out 108 frames per hour.

Here's how SMPTE works. A SMPTE 'word' contains the time plus some additional information needed by video. All in all, this requires 80 bits to relate. One SMPTE word is sent for every 'frame' of time. Taking a video rate of 30 frames per second and 80 bits per frame, a bit occurs every 1/2400th of a second – slow enough to be recorded on audio tape, but also having a very high resolution.

SMPTE's high resolution makes it ideal for use with video (in fact, it's the only thing video can work with). It is also used because a lot of equipment (like tape decks) are far more concerned with what time it is than what the tempo is or which chorus you're on. The preceding sentence illustrates one of the main differences between SMPTE and the other timing systems: SMPTE locks to real-time, while the other systems lock to the speed of a quarter note.

Please don't let all of these details about the code scare you. SMPTE is very easy to work with. Thankfully, machines that write the code to the tape, and read the code back from the tape, do all the hard stuff internally. As a drummer using SMPTE as your synchronization signal, you just connect a cable from the SMPTE out jack to the tape deck and record whatever comes out. When you want the machine to slave to the tape, send the tape's signal to the SMPTE in jack. It all works like magic!

There are a small number of drum machines on the market that contain their own SMPTE generators and readers. The ►

► E-mu SPI200, the Akai/Roger Linn MPC60, the Sequential Studio 440, and the Simmons SDX, will all read and write their own SMPTE codes. If you don't have one of these machines, but still want to use SMPTE as your timing reference, you must add a dedicated reader/generator. They aren't too expensive (around $1000 – less if an accessory to a computer), and just like any other digital device, the prices of dedicated SMPTE units are falling.

MIDI Time Code (MTC)

BACK AT THE beginning of this article, I said that MIDI Time Code was new. The format and codes that make up MTC were only approved late last year. MTC is a means of translating and locking SMPTE to MIDI (did that sentence have enough acronyms?). In order to do this, you need a device that can read SMPTE, make the conversion into MIDI, and send the proper messages to other MIDI devices in the system. Because MTC is so new, the only drum machine that I know of that will do this is the Akai MPC60 (there may be others by the time you read this). Currently, a dedicated SMPTE to MTC converter is probably the easiest and least expensive choice. So, what makes MTC so great anyway? Why do you need it? You don't mind if I ask the questions for you, do you?

Let's look at some of the messages that MTC can convey:

1. Full Message. Whenever a tape transport is fast-forwarded or rewound to a new location, a full message is sent to all MIDI devices connected. This is a long series of bytes (ten to be exact), and therefore, is not sent when the tape is up and running at its normal speed. This message indicates the exact hour (along with the SMPTE frame rate), minute, second, and frame of the tape's position and is very useful for getting all devices up to the proper spot in time.

2. Quarter Frame Message. This is the message that is used whenever the tape is running. You might remember that we mentioned this message as part of the System Common classification. By using system common instead of system exclusive, these messages take less time to send because they don't require all the accompanying ID numbers that are used with system exclusive commands. As the name suggests, this message moves down the MIDI cable four times per frame. If your SMPTE rate is 30 frames per second, then there are 120 of these messages during every second.

If the MIDI Time Code only provided a reliable time lock, it would still be great, but MTC enables you do something that no other time code can. With MTC, you can lock certain events to places in time, not to counts within a measure, but to real-time! Let's have a little fun, and you'll see how locking events to time can be a better way to go under some circumstances. How about this? You're laying down a sound track for a short commercial. Just when the boy on the screen hits a baseball through an old lady's kitchen window, you want to fire your 'glass break' sound from your sampler (originally triggered, of course, from your MIDI drumset). After seeing the commercial, you figure out that the glass break happens on the seventh frame of the fourth second. You pull out your pocket calculator, the time to tempo conversion handbook, and two aspirin. Four cups of coffee later, you have decided that the glass break sound should fire the second thirty-second note after the third beat of measure seven. You write the music, sequence the sampler to fire at the proper point in the measure, and all is well. And then the producer calls you up and says: "The music is just a little too slow, and by the way, we're going to have the camera follow the ball into the window in slow motion. Won't that be great?" You see the problem now? If you only program the sequencer to play the music faster, then the glass break is going to happen too early.

The solution? Lock the glass break sound to a specific frame. Whenever that frame comes up, the sound will fire regardless of the tempo of the music. When the producer calls up with the changes, just say, "No problem." Edit the tempo of the music, and change the frame for the glass breaking sound. This is much easier than messing around with a calculator!

The MTC messages that let you lock events to frames (and therefore to real-time) are called SetUp Messages. In addition to locking 'single-fire' events (like our glass break sound effect), SetUp Messages will let you do many other groovy things, such as:

1. Punch In/Punch Out. For tape decks that respond to MTC, you can indicate a punch in or out time for overdubbing. This can be about nine zillion times more accurate than making punches in or out by hand.

2. Event Start/Event Stop. Yes, it is possible to have an entire musical sequence start and stop at certain frame numbers. If the same sequence has several different starting points, it would be just like hitting the continue button on a drum machine.

3. Cue Point. Unlike an event, a cue point just happens one time, and exists as a single point within the time flow. The glass break sound effect used above is an example of a cue point.

4. Cue Point with additional information. The additional information can be 'human' type info such as the name of the cue ('glass break' for example), or it can be another MIDI message. In other words, cue points can also be used to send a program change message or a song select command.

If you think that all of these different messages would give you plenty of flexibility, then think about this: MTC can keep track of over 16,000 distinct events in each setup type over each MIDI channel. Yes, that means 16,000 different punch in points, 16,000 different cue points, and 16,000 different events.

To be honest with you, there is a certain amount of doubt about the future of MTC. First of all, not many instruments fully support MTC. Some manufacturers are waiting to see what everybody else does with it before they commit to including MTC in their own units. As the price of SMPTE hardware and software continues to fall, it may just be easier and less expensive for a company to include SMPTE capability instead of MTC. Only time will tell, but I hope that many other manufacturers jump on the MTC bandwagon. MTC can do several things that are just not possible with SMPTE alone. It can be a very powerful tool, and it can make your life easier under the proper conditions.

Next month: controlling a complete MIDI studio from your drumset and drum machine. How would you like to use your electronic percussion instruments to write a piece of music, record it, and print out the sheet music all in a single afternoon? Stay tuned. ®

Example #3

FULL MESSAGE – 0000 0000

Hex	Binary	Explanation
F0	IIII 0000	System Exclusive Status Byte
7F	0III IIII	ID Number for Real-Time
??	0??? ????	Device Channel Number (0-128)
01	0000 0001	Sub ID For Long Form Time Code
00	0000 0000	Sub ID For Full Message
??	00?? ????	Hours (0-23) and SMPTE Rate Type (0-3)
??	00?? ????	Minutes (0-59)
??	00?? ????	Seconds (0-59)
??	000? ????	Frames (0-30)
F7	IIII 0III	End Of Exclusive Status Byte