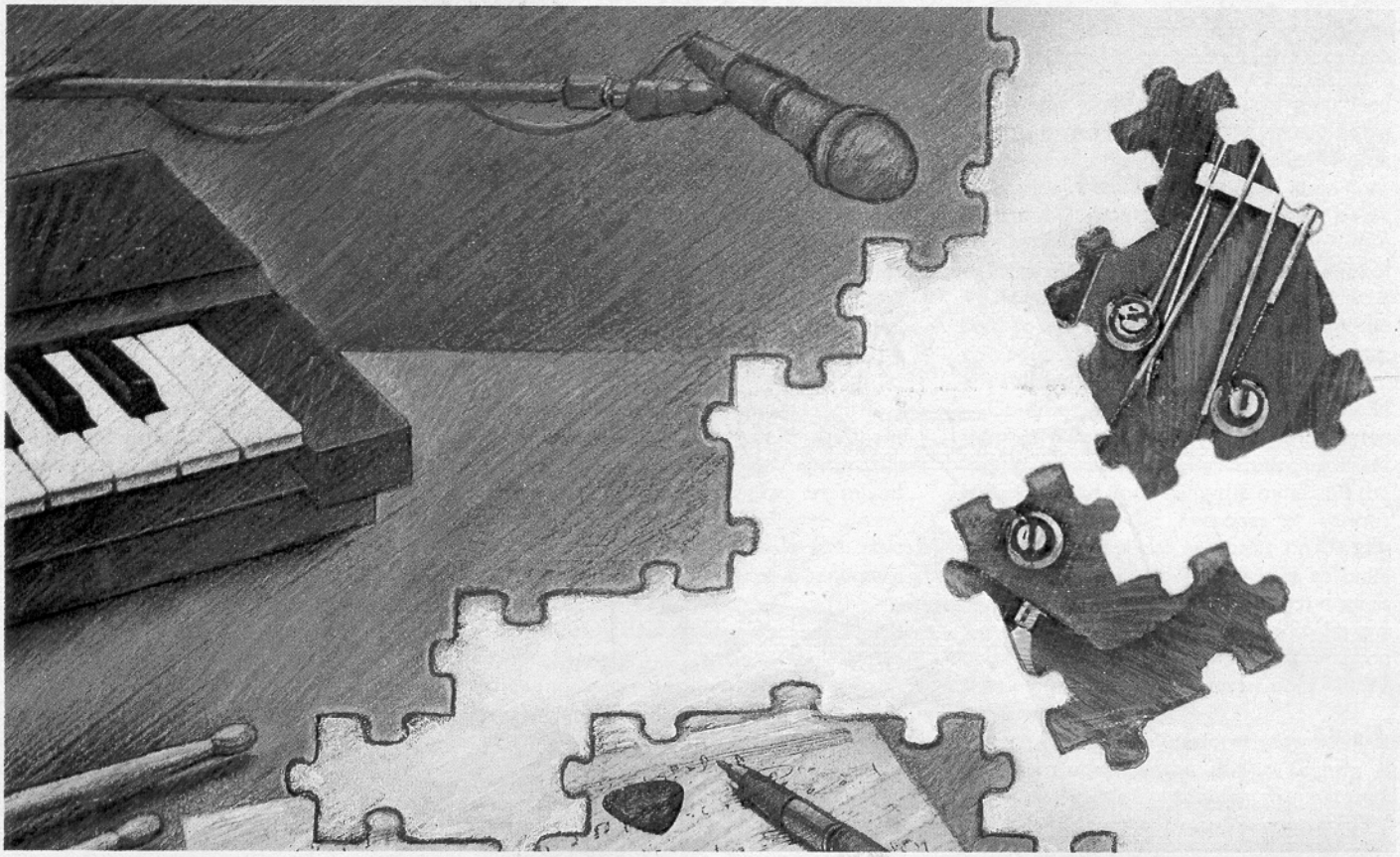
MIDI *matters*

Putting It All Together



Now that you know all about getting MIDI to do your bidding, it's time to put it all together and make some music . . . right from your own MIDI-controlled drum studio at home.

Text by Norman Weinberg.

THIS LAST INSTALLMENT of *MIDI Matters* is designed, as the title suggests, to help you put it all together. Previously, this column has covered all of the different MIDI commands such as channel voice, channel mode, system common, system real time, and system exclusive messages. In addition, we've taken a look at timing and synchronization systems (SMPTE, FSK, Pulse and MIDI Time Code) and sequencers. Now it's time to take all this information, mix it together, say a few well chosen incantations, and create a musical composition using electronic percussion instruments, MIDI messages – and the power of the computer. For those of you who may be new to MIDI, this column will give you a taste of the power that is available with electronic percussion instruments. And even if you've been involved with MIDI for some time, you may find a few new ways to use your gear.

First, let me describe the system that will be used for this composition. In the electronic drums department, there is a drum machine with an internal SMPTE reader/generator (E-mu Systems SP12), an electronic drumset (Roland DDR30), a multi-pad (Roland Octapad), and a mallet controller (Simmons Silicon Mallet). Additional sound generators include a multi-timbral expansion module (Yamaha FB-01), another synth (Casio CZ-101), and a sampler with its own built-in sequencer (E-mu Systems E-Max). The main 'brain' of the system is a computer (Apple Macintosh Plus) with a ton of software, including sequencer, librarian, and editing programs.

If the computer is the brain of the system, then the MIDI patch bay (JL Cooper MSB+) is the heart. The patch bay will be in charge of directing all the MIDI signals from one instrument to another. The audio portion of the system includes a sixteen channel mixer (Roland M160), two outboard effects processors (Alesis Midifex and the DigiTech DSP128), and a four track cassette deck (Akai MG614).

Before working on the piece, there are a few decisions that need to be made: the style and musical structure, as well as the sounds that will be needed. The tune is going to be used as the opening theme for a local news show, so it needs to be 45 seconds long. I thought a Latin/funk tune might be fun to do, so here is the structure I settled on: a four-bar introduction followed by sixteen bars of 'body'.

Step 1. Along with the sounds in the drum machine's internal ROM memory, I need

some additional sounds in RAM in order to get that Latin flavor. Starting up the drum machine's editing program on the Macintosh, I dump over my samples for a berimbau, two agogo bells, two guiro scrapes, claves, and a surdo. Next, I load a disk into the sampler which contains a super funky slap bass sound and a very nice electric guitar. This gives me enough material to work with for a while, so I load up the computer's sequencing program. Now I'm ready to go.

Step 2. The first portion of the recording plan involves creating a click track. I don't really like the metronome included on the drum machine (it's much too sterile), and the metronome from the computer sequencer isn't quite loud enough when other instruments in the system are playing. So, I create my own. First, a loop must be created with the MIDI patch bay so that the drum machine's MIDI signals are fed into the computer, and the computer's MIDI-Out leads back into the drum machine. Putting the sequencer into its step-record mode, I select a value of quarter notes, and record four cowbell strokes from the drum machine. After selecting step-record, I instruct the computer to loop my newly created track indefinitely. This track is assigned to MIDI channel one, and the drum machine is also programmed to use channel one as its basic channel. Now, I hit the play button on the sequencer and the drum machine's cowbell sound is supplying the metronome. A calculator inside the sequencer informs me that twenty measures of music will fit into 45 seconds, using a tempo of 106.7 for the quarter.

Step 3. Okay, now I'm ready to work on the drum part for the introduction of the tune. Using the patch bay, I route the MIDI signals from the multi-pad to the drum machine and also route them to the computer (make sure to turn off any 'MIDI echo' at the sequencer). This way, I can hear the drum machine's voices, and record whatever I play into the sequencer. Since the multi-pad is going to be used to fire the sounds from the drum machine, it's important that the proper MIDI note numbers are programmed into the multi-pad's surfaces.

Step 4. Once the pads are sending the proper numbers, I build a track on the sequencer that is four bars long (the length of the introduction) and tell the computer to loop this track. Punching the record button on the sequencer, I program the hi-hat parts on the first pass, and when the pattern loops, add the surdo. As this four bar pattern continues to loop, the agogo bells, guiro, claves, and berimbau are added. Since these rhythms are being programmed into the computer through the multi-pad, all the subtle dynamic contrasts of the performance are captured the first time.

Step 5. While the introduction sounds pretty good, it's not quite perfect. I decide that the attack points for that track need to be quantized, but I don't want the track to sound too mechanical. I tell the computer to quantize only those notes that are outside of a particular range. In other words, if the specified range is fifteen ticks (a tick is a subdivision of a quarter note and this sequencer runs as 480 ticks per quarter), only those attacks that fall outside of this number will be quantized. If a stroke is within these boundaries, it will be left alone. This works out pretty well, as the notes that were played extremely early or late are fixed, yet most of the data will keep its original 'human' feel.

Step 6. Now that the extra percussion sounds are looping on the drum machine, it's time to record the drumset parts. After telling the computer to record the MIDI data from the electronic kit to a new track, the next step is to use the MIDI patch bay to send the commands from the computer to the drum machine. This way, when I push the play or record button on the software sequencer, the drum machine will play the introduction, but the computer will be listening to the electronic kit. I direct the data from tracks one and two (the metronome and the Latin sounds with cymbals) to go out over MIDI channel one, and the data intended for the kit to MIDI channel two. I then set the drum machine to listen to channel one in Mode Three ▶

(Omni Off/Poly) and the electronic kit's brain to listen to channel two, also in the third mode.

Step 7. After the signal routing is complete (electronic set to computer, computer to drum machine), I hit the record button on the sequencer, and play all of the drum parts for the introduction.

Step 8. Next, I need to record some lines that will be played using the slap bass sound loaded into the sampler. Since I'm not a very good keyboard player (if I have to play little black and white keys with my fingers), I route the MIDI signal from the mallet controller to the sampler. Now, I can play more comfortably, and still fire the sounds that are on the sampler.

Step 9. Here's where it gets a little tricky. I want to be able to record the MIDI data stream from the mallet controller on the computer's sequencing program. But I want the controller to play only the bass sound that's in the sampler, instead of its own internal sound generator. By turning the mallet controller to 'local off' (this is a channel mode message), the controller will still send MIDI data, but the keys won't fire the internal sounds from its own brain. So far so good. But how am I going to send the MIDI signal to both the sampler and the computer? Simple! I use the patch bay as a giant MIDI-Thru box, and send the mallet controller's MIDI data stream to the sampler and the computer at the same time. The computer's MIDI data goes to both the drum machine and the electronic set so that it will play the tracks already recorded.

Next, I set the electronic drum track to loop for the entire introduction, add a new track to the sequencer (number four), name it 'bass line', and hit the record button. At this point, the drum machine and the kit begin playing the introduction over and over, and the computer will record whatever is played on the mallet controller (which is triggering the bass sound from the sampler).

Step 10. While the introduction is looping, I can improvise on the mallet controller until something hip catches my attention. All during the improvisation, the computer is recording every note played and every dynamic nuance. After a few minutes, a couple of good ideas come out of my hands.

Step 11. In order to hear everything played back correctly from the sequencer, I've got to change the settings for the patch bay. Again, I'm going to use it as a giant Thru-box. The signals from the computer will be sent to every other device in the system. Going back to the computer's sequencer, I direct the data from track four to MIDI channel three and tell the sampler to listen to that channel in Mode Three.

Step 12. I hit the play button on the sequencer, and it all works like magic. Listening to the bass lines exactly as they were recorded, it doesn't take long to find the ideas I want, copy them, and paste them into a new track. In essence, I've used the original bass track as a type of scratch pad for recording my improvisation in real-time. Once the final bass track for the introduction is complete (also assigned to MIDI channel three), the old bass track is erased. Since I'm going to use this same four measure bass line for the body of the tune, I set this track to loop five times.

Step 13. Now that the introduction is complete, it's time to come up with some ideas for the body. After saving all the information of the sequence off to floppy disk, I call up a new program for the computer. This is an 'intelligent' musical composition program which reads a certain amount of MIDI data, and then generates its own additional musical material from the original input.

I create a new track on the program, play seven well chosen notes on the mallet controller (using a vibe sound), and let the computer do its thing. These seven notes are repeated over and over, but with some slight variations. According to instructions that I enter into the program, some of the notes will be slightly louder or softer than others, some will be slightly shorter than others (in duration only, not faster rhythmically), and some will be transposed by an octave. The end result is a slick, continuously running pattern of sixteenth notes.

On the second track of this program, I play thirteen different pitches. Again, a few of the parameters are adjusted to create random alterations, and I have the computer play both patterns at the same time. I decide to send the vibe pattern out to MIDI channel four, and the second pattern out to MIDI channel five. They sound pretty good together, and these patterns will serve as a background to the melody during the body of the tune.

Step 14. Getting the data from the computer composition program to a sequencing program is not as hard as it may seem since there is an internal sequencer on board the sampler. I set the sampler to listen to the MIDI data stream of the computer, tell it to sync to an external clock, and hit the record button on its front panel. When the computer begins playing the two patterns, the sampler's sequencer records both MIDI channels at the same time.

Step 15. Once the MIDI data has been sent to the sequencer inside the sampler, it's a simple task to dump it back off to the computer's software sequencer. This involves simply having the computer slave to the clock from the sampler, telling it to record MIDI channel four into one new track, and MIDI channel five into a second new track, then hitting the record button on the computer and the play button on the sampler's sequencer.

Okay, let's do a quick review of what MIDI data is on each track (take a look at example number 1). Track one is the cowbell metronome sending on MIDI channel one and playing the drum machine. Track two is also assigned to MIDI channel one, but playing the Latin percussion and cymbal sounds from the drum machine. Track three is the electronic drumset data on MIDI channel two, and track four is the funky slap bass data on channel three for the sampler. Tracks five and six are the two new computer generated patterns created with the other program. They are using MIDI channels four and five, and the information is being sent to the mallet controller's brain. The data on channel four is being played using a vibe sound and the data on channel five is triggering a marimba sound (the mallet controller's brain is capable of reading two different MIDI channels at the same time). Now it's time to finish the body of the tune.

Step 16. Going back to track two (Latin instruments and cymbals), I turn off the loop and instruct the computer to punch-in on the downbeat of measure five. Again, I use the multi-pad to trigger the sounds from the drum machine, and at the same time, record that data into the track.

Step 17. Next, I follow the same process for the third track which contains the data for the electronic drums. Turn off the loop, punch-in after the introduction, and record the new material.

Step 18. Now that the background material, the bass line, and the percussion parts are complete, it's time to build a melody over the top. Since the constant sixteenth notes comprise a good deal of activity (representing our society in total chaos), a slowly moving, long phrase seems to be the best idea (representing level-headed news broadcasters). I set all the tracks to loop from bars five through sixteen, and adjust the patch bay. This particular routing sends the MIDI signals from the mallet controller to the multi-timbral sound generator (using a fuzz guitar sound with a lot of sustain) and the computer. I send the computer's data out to the drum machine, electronic set, sampler, and mallet controller. Remember that local control is still turned off on the mallet instrument, so it won't play sounds from its own keyboard. It will still respond to MIDI messages however, and it will send MIDI data generated from the keys to the MIDI-Out port. I push the record button on the sequencer, and improvise until I get a line that fits well with the other parts.

Step 19. Once the data that I want to keep is pasted into a new track, I erase the scratch track (the same process that was followed when creating the original bass line).

Step 20. In order to make the soaring melody a little more interesting, I make a copy of the entire track and paste it into two additional tracks. One of the new tracks is transposed up an octave while the other is transposed up a fifth. Now, there are three tracks that contain the exact same musical material, but at three different pitch levels.

Step 21. The original melody track is assigned to MIDI channels six and seven (it's possible to send the data from one track over multiple channels), while the two newly created tracks go to channels seven and eight.

Step 22. Since the sampler is also multi-timbral, it can listen to channel three (playing the slap bass sound) and also play the guitar sounds that are coming in on channel six. The multi-timbral sound generator is capable of listening to eight different channels at once, so I go to its front panel to assign some sounds. I program the fuzz guitar sound and something called 'hip synth 4' to listen to MIDI channel seven. In addition, the sounds of 'glass' and 'glocken 2' are assigned to channel eight.

Step 23. When the computer plays these three new tracks, the data is heard by seven different voices. Three are playing the melody at original pitch, with two each playing the transposed versions (take another look at example number 1). By adjusting their balances with the audio mixer, an interesting layered sound is ►

achieved. But, it's not quite what I'm looking for.

Step 24. It seems that the sampled guitar needs a little bend as it is sustaining. Simple. I set up a new track called 'bend', and have the computer listen to the MIDI output of the sampler. I hit the record button, and wait for the sustained notes. During the held notes, I roll the pitch wheel slightly, letting it fall back down before the next note. This gives a much more lifelike quality to the guitar sound.

Step 25. Back at the computer, I assign this track (containing only pitch wheel changes) to MIDI channel six. This way, the sampled guitar bends, but the sounds produced by the expansion unit listening to channels seven and eight, ignore these messages.

Step 26. Things sound pretty good at this point, and it's time to lay it down on tape. The very first action is to 'stripe' the tape with SMPTE code. Since the drum machine is capable of generating its own SMPTE signal, I connect the sync output from the drum machine to the fourth track of the tape deck. After adjusting the volume to a good level, I hit the record button on the tape, and tell the drum machine to start the sync signal.

Step 27. Just to make sure that everything is working okay, after striping the tape I send the audio output of the deck's fourth track to the sync input of the drum machine. By telling the drum machine to use the SMPTE signal as its timing reference instead of its internal clock, it locks to the tape. The next step is to send the MIDI output of the drum machine to the MIDI-In of the computer, and tell the computer to use an external clock. I hit the play button on the sequencer and the drum machine. Nothing happens. Great... Everything is waiting for the signal from the tape deck.

Step 28. I rewind the tape, hit the play button, and when the SMPTE signal begins, so does the music. Let's take a look at the sync process once more. The tape is serving as the master clock for the drum machine. The drum machine reads the SMPTE stripe and begins playing an empty pattern at quarter equals 106.7. While it plays, it sends MIDI timing clocks to the computer which uses them as its timing reference. Believe it or not, the whole thing works like a charm!

Step 29. While all the parts are playing, I adjust the volume levels of each instrument with the mixer, and decide to send some of the outputs to the audio effects processors. But, I just can't achieve the impression that I'm looking for. I need more audio effects. Instead of running out and buying two more echo and reverb units, I decide to multi-track the tape. Going back to the computer, I mute the tracks that have the flowing melody line, and those with the background sixteenths (leaving only the bass, drums, and percussion tracks). I send the drums' audio signal to a processor with a gated reverb patch, and the bass sound to the other processor using a chorus effect. I decide to leave the additional percussion sounds dry (no audio processing). It sounds good.

Step 30. After adjusting the balances of each instrument, the amount of processing, and the input volume on the tape deck, everything is ready to roll. I hit the record button on the tape deck, and all these sounds are recorded into the first tape track.

Step 31. Returning to the computer, I mute the tracks that just played (bass, drums, and additional percussion), and turn on one of the tracks containing data for the background sixteenths (vibraphone), the track containing the original melody line from the sampler, and the octave transposition. I send the signals from the vibraphone and the multi-timbral expander to a processor using a short reverb and a lowpass filter, while the sampled guitar's signal goes to the other processor with a much heavier and longer reverb.

Step 32 Again, the levels of all the voices are adjusted, the input level of the deck is set, and we're ready to roll for the second pass. After rewinding, I hit the record button, and lay these sounds on the second track of the tape. Since the SMPTE signal is controlling everything, both recorded tracks are in perfect sync. (If your tape deck has separate record and playback heads, make sure you're in 'sync' mode – off the record head.)

Step 33. Now for the last pass. Back at the computer, the tracks for the vibes and the melody transposed an octave are muted. I leave the sampled guitar track, and turn on the marimba and the last transposed melody tracks. The guitar keeps the same processor as before, but the marimba sound is flanged by the second processor.

Step 34. Set all the levels, rewind the tape, and the remaining sounds are recorded into the third track of the tape deck.

Step 35. After all the parts are on the four-track, its time to mix down to a two-track cassette that can be used by the TV studio. During the mixdown, the first track is assigned to both the left and right stereo channels. The second track goes to the left side, and track three is sent to the right side of the stereo field (see example number 2).

The project is now complete, and it was done entirely on a MIDI controlled drum studio at home. If your current MIDI experience is at the elementary level, you may find this long 'recipe' difficult to follow. But believe me, it's not all that tough to actually do. The entire project could be completed in less than two hours. ®