

RandoM1: A PATCH GENERATOR/ LIBRARIAN FOR THE KORG M1

This useful accessory program for the Korg M1 and Atari ST not only saves and loads patches to and from disk, but generates new sounds to delight your ears.

By David Snow

Original Data Format (seven bytes)							
Byte 0	b7	b6	b5	b4	b3	b2	b1 b0
Byte 1	b7	b6	b5	b4	b3	b2	b1 b0
...Byte 6	b7	b6	b5	b4	b3	b2	b1 b0
MIDI Data Format (eight bytes)							
Byte 0 (msb_byte)	0	b7	b7	b7	b7	b7	b7 b7
Byte 1	0	b6	b5	b4	b3	b2	b1 b0
...Byte 7	0	b6	b5	b4	b3	b2	b1 b0

FIG. 1: M1 Data Dump Format.

It was a tough choice: either go on a Hawaiian dream vacation or buy a big black box of metal and plastic (kind of makes you wonder about some people's values, doesn't it?). My wife smashed through the logjam of indecision when she advised me to go ahead and buy the Korg M1; anything was preferable to my constant whining and sulking. Well, I haven't regretted the decision. What an axe! It has 142 waveforms in ROM (4 megabytes worth), including multisamples, drums, and digitally generated sounds, all in sparkling 16-bit fidelity, not to mention two independent, programmable stereo effects units, 8-program multiimbral output with up to 16-voice polyphony, and an 8-track sequencer. It's a nice package. (For more information on the M1, see the review in the November 1988 issue—Ed.)

However, programming a synth as sophisticated as the M1 can be intimidating. A random patch generator can help overcome "programmer's inertia" and provide a few jolts of inspiration for jaded ears, but where do you put all those wonderful new programs? There's no internal disk drive for saving pro-

gram and sequence data, so to make room for your new patches, you'll have to either erase some of the factory patches (most of which are too good to waste), buy a handful of those costly little 256K RAM cards, or save and load data via MIDI system exclusive (sys ex) dumps. Clearly, the latter is the most cost-effective way to go if you already have a computer.

RandoM1 is a GEM-based, integrated patch generator/librarian for the M1 and Atari ST. It stores programs to disk in banks of 50 or 100, depending on which M1 memory allocation option you select. New patches are created by selecting parameters randomly from other patches, a trick Tim Dowty used as the basis for his CZ-101/C-64 patch generator, CZPLUS (August 1987 EM).

HOW IT WORKS

To save a bank of patches to disk, or generate a new patch, it is first necessary to transfer patch data from the M1 to the computer. To do this, RandoM1 sends a MIDI system exclusive (sys ex) All Program Parameter Dump Request to the synth, which responds by sending

a sys ex dump of all its programs. (For information about system exclusive messages, see Tim Dowty's "CZ Patch Librarian" in the February 1987 EM and Jim Johnson's "Fun With System Exclusives" in the March 1987 issue.)

To create a new sound, the software selects 143 parameter bytes randomly from the patches sitting in the ST and stuffs the mongrelized data into a buffer. The new patch then goes to the M1 in a Single Program Parameter Dump, and RandoM1 patiently waits for you to try out the sound on the instrument. If you like what you hear, you can store it in the M1 in the normal fashion (push the synth's EDIT PROG button, go to Page 9, press WRITE, and select OK), or you can keep creating new patches until you find one you like.

Loading the patch bank in the ST and saving it to disk requires binary saves and loads of the bank buffer, while loading the synth with a patch bank requires an All Program Parameter Dump from the ST.

There is a complication, however. According to the MIDI spec, data bytes for any MIDI message (including system exclusive messages) cannot exceed a value of 127. However, many of the M1's parameters exceed 127, so the M1 needs to "compress" its data before sending over MIDI.

The CZ-101 method of compression, accommodated by CZPLUS, splits each byte into two nibbles (4-bit units) and sends them in separate bytes. The M1 uses a more efficient but more complex scheme (Fig. 1). It groups the data into sets of seven bytes each, strips off bit 7 of each byte in the set and places it into an extra byte, then sends this byte before the other seven during the dump. Therefore, for every seven bytes of original data, eight are sent via MIDI (and it's a good thing, too, since a patch dump in the CZ sys ex format would take almost