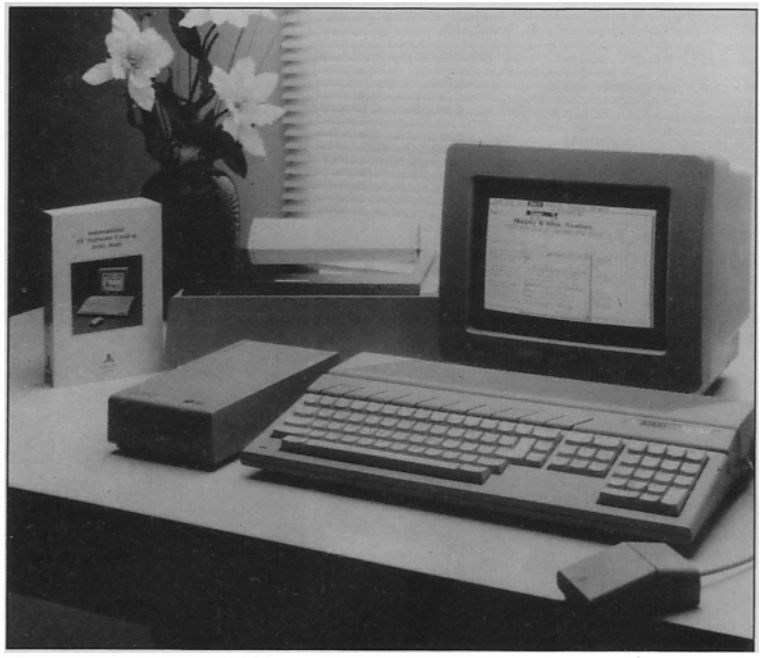
THE MIDI MUSIC BOX: More Cheap Thrills for the Atari ST

In our continuing quest to deliver something for nothing, we present a new algorithmic composition program.

By David Snow



The MIDI Music Box algorithmic composition software for the Atari ST is essentially a bare-bones sequencer that generates its own tracks and is sure to provide hours of mind-numbing diversion. More important, the program illustrates some interesting programming techniques that you can incorporate into your own programs.

The Music Box requires no input parameters. Pressing the "G" key will generate four tracks, each 675 notes long, that draw their rhythm and pitch material from predetermined source sets. Pressing "P" will play the four monophonic tracks on MIDI channels 1 to 4, which is dandy if you have a CZ-101 or other multitimbral synth. During playback the ST's sound chip provides a metronome click on every beat. You can change playback tempo from the default value of 120 beats per minute by pressing "T" and entering a new value.

The program requires two files: a "shell" written in LDW BASIC (**Listing 1**) that initializes the program and manages the user interface, and a collection of assembly language routines called MIDIMUSE.BIN (**Listing 2**) that do the dirty work. The GENST.PRG public domain editor/assembler was used to assemble the latter file. (*Note: LDW BASIC is a compiled language that is somewhat different than ST BASIC, the language supplied by Atari. See the sidebar for details on converting the program to ST BASIC. You can assemble MIDIMUSE.S with just about any 68000 assembler, though you may have to make a few slight changes, especially in the comments and labels.—Ed.*)

The Music Box records its track data in four arrays (track_a, track_b, track_c, track_d), each array being 2,700 words (5,400 bytes) long. Each MIDI event requires four bytes, and each note in the track requires two MIDI events (a note on and a note off), giving a total of 675 notes per track (5,400÷[4+4]=675). The data of each MIDI event is structured as follows: timestamp, two bytes; pitch, one byte; velocity, one byte.

The timestamp is a word-long value that indicates the time of the event, relative to the beginning of the sequence. The pitch and velocity bytes are normal MIDI data. The only difference between note on and note off events in Music Box is the velocity value: 64 for a note on, 0 for a note off. If a note on is not followed by a note off, the note will hang.

Every track is terminated with an end-of-track marker to which the arbitrary value of 253 is assigned. This number is placed in the third byte of the last MIDI event of each track, just after the last timestamp.

That's it. This rigid four-byte structure doesn't allow for niceties like pitch bend, but that was a tradeoff to keep things simple.

GETTING WITH THE PROGRAM

The eight source rhythm patterns from which the program constructs each track are encoded in data statements under the LOAD RHYTHM PATTERNS heading in listing 1. Each pattern consists of ten numbers, organized as five pairs of two numbers; the first in each pair represents the position of each note in the pattern, the second represents the duration of the note. Each beat is subdivided into 24 pulses. Therefore, in a pattern of four quarter notes, the first would occur on pulse 0, the second on pulse 24, the third on pulse 48, and the last on pulse 72. Since each quarter note is 24 pulses long, the pattern would be represented as: 0, 24, 24, 24, 48, 24, 72, 24. To show where the next note will fall, the pattern must end with the next pulse position, followed by a duration value of 0, in this case, 96, 0.

Using this formula, it's easy to read the patterns encoded in this set of data state-