A MIDI Foot Controller - The PodoBoard

Eric Johnstone
Electronic Music Studio
Faculty of Music
McGill University
555 Sherbrooke Street West
Montreal, Quebec
Canada H3A 1K3
Telephone: (514) 388-4552

Abstract

A new MIDI controller for the feet is described. The device consists of a 91 cm X 102 cm X 2 cm thick flat board covered with 2.54 cm square aluminum tiles. Special shoes with metal plates on the toes and heels are pulsed sequentially by a small dedicated computer which then extracts the coordinates of the positions at which the toes and heels have touched. Each toe and heel contains a piezoelectric film microphone from which velocity signals are generated.

The coordinates are sent as MIDI signals to an Amiga computer and into the program SoundScare from Mierastics. SoundScare provides a MIDI environment which can incorporate new modules. A module was written which takes the MIDI coordinates of the toe and heel contact points on the PodoBoard and maps them onto four drawings (one for each toe and heel) representing areas which produce defined MIDI signals. These signals are then processed by an iconic programming language (somewhat like the IRCAM Max program) to drive other modules of the SoundScare environment and external synthesizers.

General

The PodoBoard is being built for Alain LaMontagne, a Quebecois performer. He is a storyteller, harmonica player, and practitioner of the art of clackaste, a sort of step dancing performed while seated on a chair. The PodoBoard replaces the maple plywood board that M. LaMontagne normally uses. This gives him the ability to play synthesised or sampled sounds instead of the usual acoustic sounds of step dancing.

Hardware and Firmware

The PodoBoard consists of an array of 36 X 40 one inch square aluminum tiles. Each tile is connected to
two isolating diodes which connect to horizontal and vertical grid wires. The heels and soles of the shoes are covered with separate metal contacts. A small dedicated computer, a New Micros 'Max' board with an F68HC11 CPU, scars the heels and toes, pulsing them sequentially. The F68HC11 was programmed by Chris Boward in Forth and assembler.

When one of the heels or toes touches tiles on the board, a pulse appears on the horizontal and vertical grids connected to the touched tiles. This 80 bit 'image' of the foot contact is pulled into the F68HC11 from shift registers at 1 megabit per second through the F68HC11's 'SPI' (serial peripheral interface). The F68HC11 examines the image and sends two MIDI note-on messages to an Amiga computer. The first message contains the identity of the touch contact (each left, right heel and toe is assigned its own channel) and the coordinates of the touch point: the second contains the velocity of the contact.

The transducer for the velocity signal is sheets of Kynar piezo film which has been installed between the metal sole and heel contacts and the leather of the shoe. The film generates a charge proportional to the energy of the impact of the foot with the PodoBoard. Analog circuits condition this charge signal which present it as a d.c. voltage to the F68HC11. An on-chip eight-bit A/D converter digitizes the velocity.

If the foot is moved while maintaining contact, pitchwheel messages are sent with the coordinates of the new position. When contact ceases, a note-off signal is sent to the Amiga.

Amiga software

At the Amiga, the MIDI messages are received by SoundScape, an old program from Mimetics written by Todor Fay. Mimetics sells a developer's package that allows the Soundscape program to be augmented with other programs. Thus, a sequencer, MIDI clock, MIDIscope, sampler and many other utilities and processors become an immediate part of the environment. This is accomplished very easily by virtue of the multitasking capability of the Amiga.

The signals from the PodoBoard are relatively useless because they are from a two dimensional source and do not follow MIDI conventions. The PodoBoard, as a keyboard, has 36 X 40 = 1440 tiles, which, with interpolation, results in 1440 X 4 = 5760 identifiable points. The Podo software converts the output of the PodoBoard to the usual one-dimensional MIDI messages.

The Podo software, written by the author, appears as an icon on the SoundScape patch bay. The SoundScape
MIDI-source icon (where the PodoBoard MIDI signal arrives) is patched to the Podo software in icon. The Podo software out icon is patched to the SoundScape MIDI-send icon. The electrical MIDI output of the Amiga then goes off to synthesizers and samplers.

The Podo software is an iconic data flow system programming language (the writing of which was begun in June 1989). The MIDI signal is processed by what appear to the user as boxes with labels in them (function icons) that are connected to other boxes with labels by 'rubber bands' to form a data dependence graph. For instance, the in icon ('in' for short) is where the MIDI messages come in. A transpose icon ('transpose') can be fed the MIDI signal by clicking with a mouse from 'in' to 'transpose.' 'Transpose' can then be patched to an 'out' which will send its input signal out of the Podo software and thence to whatever the Podo-out icon is connected to in the SoundScape patch bay. It is possible to connect any number of function icons to or from any other function icons. Thus, in the above patch, connecting the 'in' to the 'out' in parallel with the 'transpose' will result in a two note chord. It is also possible to have feedback paths (which are very nasty unless a delay or counter or some kind of limiting is included.)

When double clicked, a function icon brings up an edit window. For example, double-clicking the transpose icon brings up a window in which the amount of transposition may be set.

A window in a screen can be saved to disk as a function, which can then be recalled as an icon and replicated as seeded. Inputs and outputs to the function are via 'connect' icons. A function recalled from disk is presented as a name with in and out connects below to the left and right, respectively. Double clicking on a connect will show the edit window of the function with the double-clicked connect circled in a different color from the rest of the diagram.

Movement through a large program is implemented via this double-click and circle technique.

The board icon when double-clicked brings up an edit screen. A screen in Amigese is the basic unit of display; it sets the display mode for the windows it contains. In this case the edit screen is a low-res (320 X 200 pixels) 16 color paint program with four windows which display the mapping of the PodoBoard for each heel and toe. The performer can create areas for each heel and toe that, when touched on the actual board, will generate a MIDI message with a defined channel number, note value, and velocity mapping. Pitchwheel events can be filtered out if desired. There
are a variety of menu operations for copying, deleting, etc. When the performer touches the PodoBoard with the
shoes, sprites appear in the windows showing him where
he has touched the board. It is relatively easy to
watch the screen to know what one's feet are doing —
like using a mouse.

Each defined area on the board may be patched
individually if desired. When an area is defined, one
can choose to send its MIDI signal from the main board
icon (there is one for each heel and toe) or from a
private icon (i.e., a box containing the name of the
area), or from both. Thus, each area can have its own
processing. For example, one area might generate a drum
roll, another could initiate a sequence, and another
could play a chord.

Performance specs

A few timing figures might be of interest to
readers. The Amiga used is a venerable Model 1000 with
no hardware enhancements. A MIDI message sent to the
SoundScape program and then through the Podo software
without any processing has an inherent delay of 1.7
msec; i.e., patch a signal from 'in' to 'out' in the
Podo software and measure the delay from MIDI in to MIDI
out of the Amiga with an oscilloscope. (Since the MIDI
message occupies the first 960 usec of this 1700 usec,
the actual time taken by the Amiga, including task
switching from SoundScape to the Podo software and back
to SoundScape is actually 740 usec.) The transpose
function in the Podo software, which checks to see that
the signal is note-on or note-off and has not been
transposed out of bounds, takes 145 usec (1/10 of 1.45
msec for 10 transpose functions connected in series).
The Kynar film takes 1.6 msec to reach its peak value
(by which time the coordinates of the contact have
already been sent to the Amiga). From the moment of
contact to the time the MIDI signal passes through the
board function and out of the Amiga takes 5.7 msec,
although if measured from the peak of the microphone
signal, the delay is 4.1 msec. The software was written
in Lattice C version 5.04 and has not been optimized.
It is currently about 13000 lines, not including the
structures for windows, menus, etc.

The work has been funded by federal and provincial
grants and supported by the Electronic Music Studio of
McGill University.