Sonic Banana: A Novel Bend-Sensor-Based MIDI Controller

Eric Singer
LEMUR
New York, NY
eric@ericsinger.com

ABSTRACT
This paper describes the Sonic Banana, a bend-sensor based alternative MIDI controller.

Keywords
Interactive, controller, bend, sensors, performance, MIDI.

1. DESCRIPTION
The Sonic Banana (Figure 1) is a MIDI controller and alternative musical instrument in the form of a 2 foot long flexible rubber tube. Four bend sensors, each 5" long, are mounted in a row along the inside of the tube on a flexible metal bar running the length of the tube. A single pushbutton switch is mounted at the top end of the tube.

The pushbutton and sensors connect to a BX24 microcontroller which converts data from these inputs into MIDI messages. The four bend sensors send MIDI continuous control messages 1-4, respectively. Bending the tube in different locations changes the control value corresponding to the sensor in that location.

The pushbutton switch sends a MIDI note-on with pitch 60 and velocity 127 when depressed and 0 when released.

Data from the Sonic Banana can be sent into Cycling 74 Max and processed to create a variety of algorithmic improvisational instruments. Examples of performance algorithms are described below.

The Sonic Banana may also be used to generate MIDI data for directly controlling another MIDI instrument. This can be done by mapping the pushbutton and controllers as presented, inside a synthesizer, or by reprogramming the firmware on the BX24 to generate different controller values more appropriate to the particular application. Alternatively, of course, control data may be remapped in Max before being sent to a synthesizer.

2. BACKGROUND
Bend sensing has been incorporated into a number of alternative MIDI controllers, including body suits such as the MidiDancer [1] and MIBURI [2], gloves such as the Lady Glove [3] and Mattel PowerGlove-based instruments, and the CyberShoe [4]. The approach with the Sonic Banana differs from these instruments in that bend sensing is not used to sense joint positions or muscular flex, but rather to provide a direct gestural user interface.

The Sonic Banana is an attempt to create an expressive instrument with an unusual performance interface using a "less is more" approach – to create a rich musical output from a limited set of control streams. After the mapping and generation algorithm has been set, the interface encourages experimentation to explore what sort of bending and twisting gestures produce interesting musical effects.

3. PERFORMANCE PATCHES
Examples of improvisational performance algorithms developed in Max include an Arpeggiator and a Harmonica Simulator.

In the Arpeggiator patch, the pushbutton is used to start and stop arpeggiation. The bend sensors control various parameters of the algorithm such as speed, note duration, velocity, chord type, root pitch and pitch bend, with their ranges being scaled appropriately to the parameters.

The example Max patch in Figure 2 shows one particular mapping of the sensors. Sensor 1 selects one of six chord interval sets stored in the coll object (maj, dom7, min, min-maj7, min7, dim). Sensor 2 controls both note velocity and duration of the makenote object. Sensor 3 is mapped to pitch bend. Sensor 4 controls the speed of the metro object.

In the Harmonica Simulator, data from all four sensors is first summed to obtain a single overall bend value. This value is then sampled and differenced to derive a bend velocity value. The bend velocity value is scaled and used as force in a simplified mass-momentum algorithm to simulate pushing a mass along a surface, with adjustable coefficients of friction and air resistance. On each sample, the algorithm uses the following formula

$$v_{t+1} = v_t + F_t / m - \mu * m - d * v_t$$

where F is the force (scaled bend velocity), μ is the coefficient of friction and d is the drag coefficent due to air resistance.

The velocity of the mass maps to MIDI volume, simulating the air flow into a harmonica. A chord is played and held when absolute velocity goes above a small threshold, with positive velocity selecting one chord and negative selecting another. The pushbutton changes the pair of chords selected by velocity.

Values for mass, coefficients and scaling were experimentally determined. A reasonable set of values which gave a good "feel" to the simulation were as follows: sample period of 50 mS, bend velocity scaled by 0.5, m = 0.4, μ = 10, d = 0.001.

Example video of these algorithms can be found at http://ericsinger.com/SonicBanana.

4. FUTURE DEVELOPMENT
In addition to ongoing performance algorithm development, two improvements to the instrument firmware are planned. One will enable scaling and mapping of the sensors and pushbutton via MIDI input; another will put several useful performance algorithms on-board so the instrument may directly drive a synthesizer or sound module. Algorithms may then be selected from the MIDI input using patch changes.

5. REFERENCES

[1] Coniglio, M., MidiDancer, http://www.troikaranch.org/mididancer.html.

[2] IEEE Spectrum, 1997, The Yamaha Miburi System, http://www.spectrum.ieee.org/select/1297/miburi.html.

[3] Sonami, L., Lady's Glove, http://www.sonami.net/lady_glove2.htm.

[4] Paradiso, J., Hu, E. and Hsiao, K.-Y. The CyberShoe: A Wireless Multisensor Interface for a Dancer's Feet. Proceedings of International Dance and Technology 99,, Tempe, AZ; FullHouse Publishing, Columbus, OH (2000), pp. 57-60.

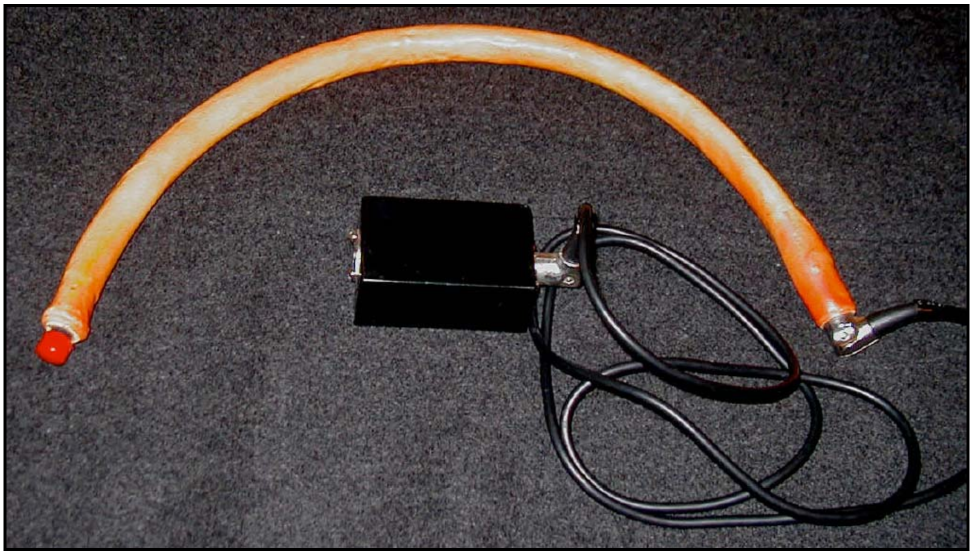

Figure 1: Sonic Banana

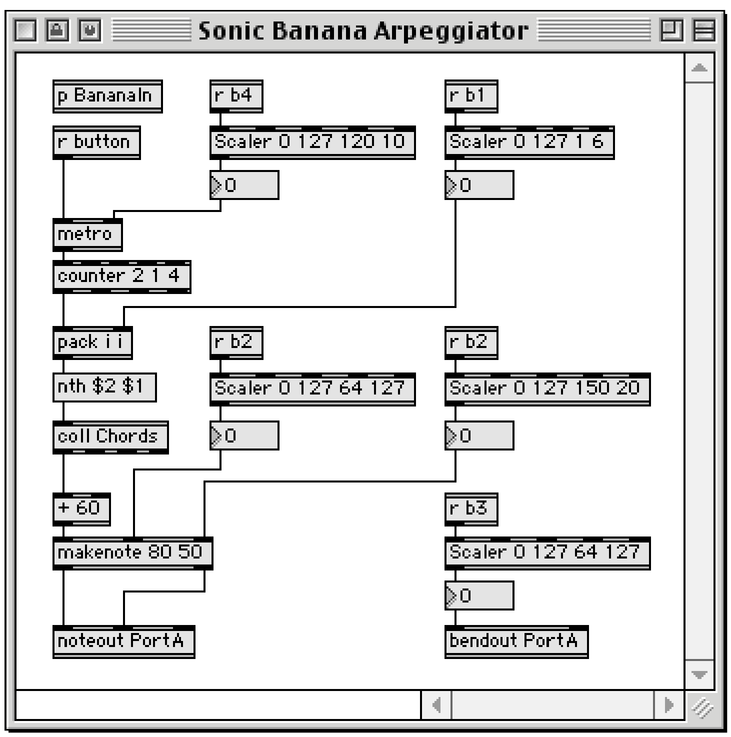

Figure 2: Arpeggiator Max Patch