The MIDI Pick

Trigger Serial Data, Samples, and MIDI from a Guitar Pick

Roy Vanegas
Interactive Telecommunications Program
New York University
721 Broadway, 4th Floor
New York, NY 10003-6807
r.o.y@nyu.edu

ABSTRACT

The guitar pick has traditionally been used to strike or rake the strings of a guitar or bass, and in rarer instances, a shamisen, lute, or other stringed instrument. The pressure exerted on it, however, has until now been ignored. The MIDI Pick, an enhanced guitar pick, embraces this dimension, acting as a trigger for serial data, audio samples, MIDI messages [1], Max/MSP patches, and on/off messages. This added scope expands greatly the stringed instrument player's musical dynamic in the studio or on stage.

Keywords

guitar, MIDI, pick, plectrum, wireless, bluetooth, ZigBee, Arduino, NIME, ITP

1. INTRODUCTION

Developed for NIME, a course offered at New York University's ITP graduate program, The MIDI Pick is designed to allow the guitar pick to act wirelessly as either an analog or digital trigger, freeing the musician from the constraints of a foot controller. The device is a tripartite audio controller, comprised of a conventionally-shaped plectrum, a microcontroller, and a wireless module. The unit interprets finger pressure as analog or digital values and transmits that information as serial data to a computer wirelessly. Max/MSP handles the wireless information, via the `serial` object, redirecting the data to other Max objects[2]. Any stringed instrument player could benefit from the added dimension that The MIDI Pick offers.

2. SIMILAR PROJECT

A review of the literature yielded only one project of a similar nature: The Chicipick. Based in Australia, Steve Chick Research has developed a guitar pick that senses contact with the strings of a guitar and sends that information to proprietary VST plug-ins. No wireless or hard-wired information could be found, and documentation for The Chicipick is scant and pithy [Chick, Steven].

3. DIFFERENT IMPLEMENTATIONS

A discussion of the two phases of this project, and of the forthcoming implementation, follows.

3.1 Original Prototype

Materials and Electronics

The MIDI Pick's pick is made of wood, rubber, double-sided tape, and a force-sensing resistor, or FSR (see Figure 1 at the end of this document). The resistor connects via a pair of braided wires to a mini Arduino microcontroller. A bluetooth module/radio, powered by the Arduino, is responsible for transmitting data wirelessly, while a 9-volt battery provides power via a 5-volt regulator to the microcontroller. The Arduino microcontroller, the regulator, the bluetooth radio, and the battery are all mounted on a wriststrap.

How It Works

Using The MIDI Pick is quite simple: squeeze. The harder one exerts pressure on the pick, the higher a numerical value is transmitted[3]. Conversely, less pressure yields smaller values.

Testing

Tests were originally carried out on an Apple PowerMac G5 and on a MacBook Pro, each machine running Mac OS X, version 10.4.8 and Max/MSP, version 4.6.2. The PowerMac proved subpar: latency at times exceeded 5 seconds. Thus, testing was restricted to the MacBook Pro, which provided a latency-free environment.

This project was tested solely within the Mac OS X environment. However, the future implementation of The MIDI Pick is projected to be studied under Ubuntu and Fedora distributions of Linux and under 2000, XP, and Vista versions of Windows.

[1] MIDI functionality forthcoming in the next version of this project, due in late May 2007.

[2] Any Max object that responds to numerical data may be triggered by The MIDI Pick, making the device a truly universal Max/MSP controller.

[3] The Arduino sends numerical serial data in the form of byte characters, ranging from 0 to 127.

NIME07, New York, NY USA

In terms of power, the 9-volt battery lasted about 75 minutes before The MIDI Pick showed signs of transmission lagging.

3.2 Present Version

Developed during a two-week residency at STEIM, The Netherlands, the current implementation of The MIDI Pick (see Figure 2 on page 3) now includes LEDs connected to resistors for finger pressure notification, a PCB to mount the electronics, a plexiglas to mount two switches (of which only one has been realized), and a power source consisting of two 2032 button cells (which collectively replaced the original 9-volt power source).

In addition, a generalized Max patch was designed (see Figure 3 at end of this document) as a base patch for other Max objects.

The testing environment remained the same: a MacBook Pro running Mac OS X, version 10.4.8 and Max/MSP, version 4.6.1. The MIDI Pick behaved the same as its previous variant. The new power source, however, was not tested beyond the five-minute mark. Consequently, battery life is unpredictable in live performance situations.

3.3 Future Version

Slated to be ready by May 2007, the next version of The MIDI Pick will be smaller, lighter, smarter, and brighter than its predecessors. In addition, its pick will be cast with an FSR embedded in its cavity, and ZigBee units will provide wireless functionality via the microcontroller and a receiver.

Structure and Size

The unit will be built in a three-tier structure: power source on tier one; microcontroller on tier two; LEDs and switches on tier three. The first tier will have loops through which to run a wristband, much like a belt loop on a pair of pants.

The future version will also be small, with a projected footprint of 30 mm (1.18") × 12 mm (0.47") × 60 mm (2.36"), length, width, and height, respectively.

Weight

At 2.0oz, the unit will weigh much less than its two precursors. In addition, it is likely that the unit will weigh even less, as plastic standoffs may replace the weighty metal standoffs.

Switches

The future implementation of The MIDI Pick will provide a smarter interface with a series of three switches, all mounted on plexiglas or clear plastic. The first switch will simply turn the entire unit on or off.

The second will provide the option to be either in digital mode or switching mode, such as a light switch, or in analog mode. If the unit is in digital mode, squeezing the pick would send a 1, and the device would remain in that state until the user squeezes again, in which case the device would send a 0. In analog mode, the unit would simply react to pressure, as described under "Original Prototype" above.

The third and final switch will be used to choose between a 9600 kbps baud rate, or the MIDI standard, 31250 MIDI baud rate.

Illumination

Eight LEDs will allow the user to see The MIDI Pick's behavior. Five LEDs will indicate finger pressure: one active LED will indicate minimal pressure; all five active LEDs will indicate maximum pressure.

The sixth LED will indicate power.

The seventh LED will notify the user of data transmission.

The eighth LED will notify the user whether they are in analog or digital mode.

Pick

The last improvement to this project will involve a custom-made pick, cast of durable plastic, with a 0.558 mm FSR implanted inside its chamber.

Wireless

The bluetooth radio performed well in all versions where fast and reliable wireless communication in a near-latency-free application was critical. Nevertheless, for the future implementation of The MIDI Pick, the bluetooth protocol will have to be abandoned for the ZigBee architecture. The latter is cheaper (at $19 per unit compared to $60 per bluetooth radio), more versatile, and conserves more power.

The ZigBee modules will also allow me to develop a receiver that will run an Arduino, which will convert serial data to MIDI data. The receiver will be fitted with MIDI connectors, thus making The MIDI Pick a generalized MIDI controller, in addition to a universal Max/MSP controller.

Please refer to Table 1 at the back of this document for a comparison of specifications and capabilities for each implementation of The MIDI Pick.

4. PERFORMANCE

On 15 December 2006, I used the prototype version of The MIDI Pick in a live performance that showcased newly minted instruments developed for NYU's NIME graduate course. I employed the unit as an analog controller that manipulated a white noise-generating Max object. The MIDI Pick/Max track complemented an original composition that employed pre-recorded audio tracks, a live violinist, and a live electronic percussionist.

The MIDI Pick performed perfectly.

The MIDI Pick displayed no problems with latency, while located within two feet of the MacBook Pro that was acting as the bluetooth receiver for The MIDI Pick's streaming data.

It is projected that future versions will perform the same at much longer distances.

5. CONCLUSION

In its inaugural form, The MIDI Pick has proven to be stable and performance-worthy — good signs as the project is developed further. In either the studio or on stage, the unit frees its user from being bound to his or her foot controllers, as it allows he or she to trigger effects from the pick.

The MIDI Pick has become lighter with each implementation, and is projected to be at most 2.0oz in future implementations. Additionally, the unit will be smaller, at about the size of a pair of quarters, and more intelligent, by way of switches and lights.

Lastly, the future version will encompass all of the previously mentioned capabilities in addition to MIDI functionality. In essence, The MIDI Pick will become so versatile a trigger that it will be used for everything from stage lighting to hand-held wah-wah effect triggering.

Updates and developments:

`http://roy.vanegas.org/itp/nime/the_midi_pick/`

6. ACKNOWLEDGMENTS

Thanks to Jamie Allen and Gideon D'Arcangelo for guidance and suggestions throughout the development of this project for the fall 2006's NIME course. Thanks also to Robert van Heumen, Nico Bes, Jorgen Brinkman, and René Wassenburg at the Studio for Electro-Instrumental Music, STEIM, The Netherlands, for providing help and resources during the development of the current version of The MIDI Pick.

7. REFERENCES

Chick, Steven. *NAMM Oddities 2005 - Techno Geek Toys.* 30 Jan. 2007
<http://www.otheroom.com/namm05/Techno.htm>

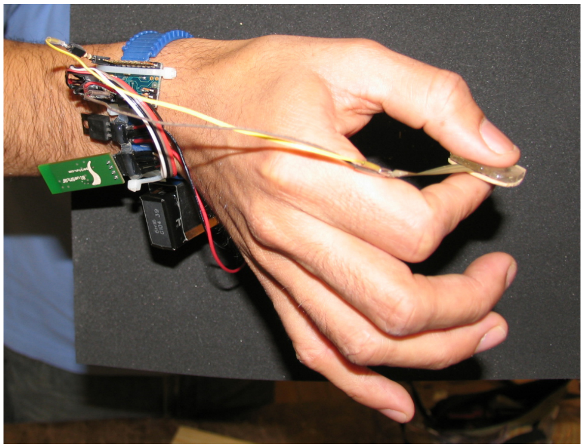


Figure 1: Original prototype

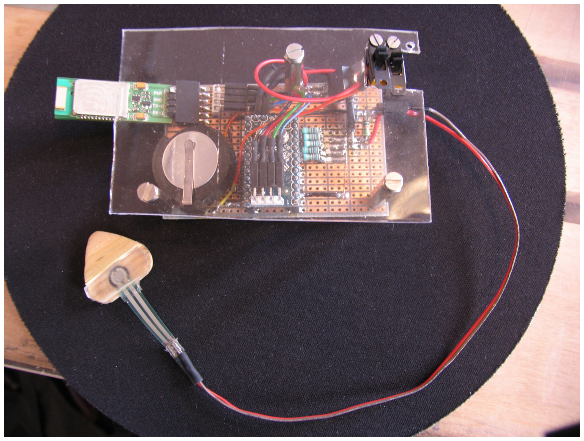
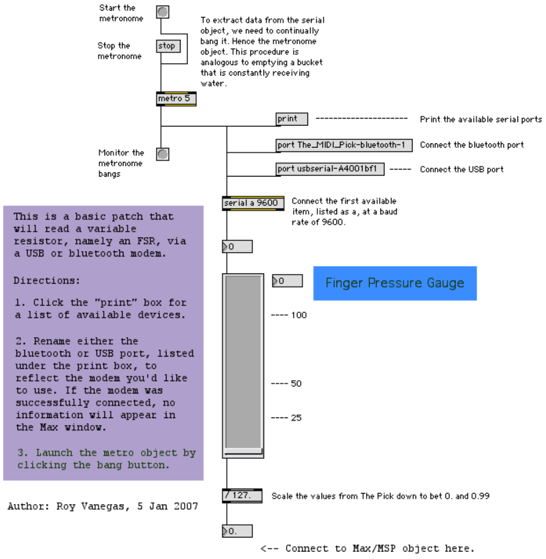


Figure 2: Current implementation (wristband not shown)



Start the metronome

Stop the metronome

To extract data from the serial object, we need to continually bang it. Hence the metronome object. This procedure is analogous to emptying a bucket that is constantly receiving water.

Print the available serial ports

Connect the bluetooth port

Connect the USB port

Monitor the metronome bangs

Connect the first available item, listed as a, at a baud rate of 9600.

This is a basic patch that will read a variable resistor, namely an FSR, via a USB or bluetooth modem.

Directions:

1. Click the "print" box for a list of available devices.

2. Rename either the bluetooth or USB port, listed under the print box, to reflect the modem you'd like to use. If the modem was successfully connected, no information will appear in the Max window.

3. Launch the metro object by clicking the bang button.

Scale the values from The Pick down to bet 0. and 0.99

Author: Roy Vanegas, 5 Jan 2007

<-- Connect to Max/MSP object here.

Figure 3: Basic Max patch that drives Max/MSP objects

Table 1: Specifications and Capabilities

	Prototype	*Current*	*Future*
Power	9-volt (6LR61)	6-volt (2 × 2032)	6-volt (2 × 2032)
Data TX	Serial	Serial	Serial and MIDI
Weight	$2.8oz/80g$	$2.4oz/70g$	$2oz/56.69g$ (projected)
Illumination	None	5 LEDs	8 LEDs
TX Rate	9600 kbps	9600 kbps	9600 or 31250 kbps
Mount	Wristband	PCB mounted on wristband	Three PCBs mounted on wristband
Wireless Protocol	Bluetooth, 802.15.1 (Class 1)	Bluetooth, 802.15.1 (Class 1)	ZigBee, 802.15.4
Switches	None	2	3