

Building Low-Cost Music Controllers

Alexander Refsum Jensenius^{1,2}, Rodolphe Koehly², and Marcelo M. Wanderley²

¹ University of Oslo, Department of Musicology, P.O. 1017 Blindern,
N-0315 Oslo, Norway

`a.r.jensenius@imv.uio.no`

² McGill University, Schulich School of Music, 555 Sherbrooke Street West,
H3A 1E3 Montreal, Quebec, Canada
`{rkoehly, mwanderley}@music.mcgill.ca`

Abstract. This paper presents our work on building low-cost music controllers intended for educational and creative use. The main idea was to build an electronic music controller, including sensors and a sensor interface, on a "10 euro" budget. We have experimented with turning commercially available USB game controllers into generic sensor interfaces, and making sensors from cheap conductive materials such as latex, ink, porous materials, and video tape. Our prototype controller, the *CheapStick*, is comparable to interfaces built with commercially available sensors and interfaces, but at a fraction of the price.

1 Introduction

A number of different sensor interfaces and sensor technologies have emerged for musical and artistic purposes in recent years, and have made it possible for "everyone" to build their own custom-made music controllers. However, most of these systems are far too expensive to allow for example all students in an undergraduate music technology class to build their own controllers, or build many self-contained instruments for a performance. This has led to our interest in exploring various ways of creating sensor interfaces and sensors that would allow for making a complete music controller on a "10 euro" budget. Another guiding idea has been that "everyone", also people with a limited technical background, could manage to build their own controller.

The paper starts by discussing our experience with turning game controllers into generic sensor interfaces, then presents how we can make contact sensors from conductive materials, and finally shows a prototype of a music controller and how it can be used for musical applications.

2 Game controllers as sensor interfaces

Commercially available sensor interfaces intended for musical and artistic purposes generally use either MIDI or Open Sound Control (OSC) to communicate

Table 1. Comparison of a number of popular sensor interfaces for musical and artistic purposes (prices and specifications are taken from the manufacturer’s websites, and have been generalized for the sake of comparison)

Product	Manufacturer	Price	Inputs	Speed	Resolution	Protocol
Pocket El.	Doepfer	€80	16	-	7 bit	MIDI
MIDItron	Eroktronix	€125	8	-	7/10 bit	MIDI
Teleo	Making Things	€130	4	-	10 bit	USB
miniDig	Infusionsystems	€330	8	100 Hz	7/14 bit	MIDI
Teabox	Electrotap	€350	8	4000 Hz	12 bit	SPDIF
GluiOn	Glui	€445	16	1000 Hz	16 bit	OSC
Eobody	IRCAM	€480	16	-	10 bit	MIDI
Wi-miniDig	Infusionsystems	€500	8	100 Hz	7/14 bit	MIDI
Digitizer	Infusionsystems	€580	32	24-244 Hz	7/14 bit	MIDI
Wise Box	IRCAM	€950	16	200 Hz	16 bit	OSC
Toaster	La Kitchen	€1200	16	200 Hz	16 bit	OSC
Kroonde	La Kitchen	€1200	16	200 Hz	10 bit	OSC

with a computer (Table 1). Popular MIDI devices (e.g. iCubeX, MIDItron, Eobody) are generally cheaper, and have the advantage of being able to connect directly to any MIDI-compatible equipment. In practice, however, most people tend to connect the interfaces to a computer. Thus, a separate MIDI interface is also required, boosting both the total prize and potential problems of such a setup.

The shortcomings of MIDI in terms of resolution³ and speed, make OSC based sensor interfaces (e.g. Ethersense, Toaster, GluiOn) more attractive. Such devices typically allow for higher resolution and sampling rates, and long-distance communication over standard high-speed ethernet connections [1]. There are even devices that use digital audio for communication (e.g. Teabox [2]), but this requires the computer to be equipped with a digital audio input.

The problem is that none of these devices come close to our "10 euro" budget. Most of them are also too bulky for our needs, since we are interested in integrating as much as possible on the music controller itself. A solution, of course, might be to use microcontrollers such as the Atmel, Pic or Basic Stamp, as suggested by [3, 4], but the background in programming and electronics needed to succeed with this is not something we expect from the regular music student taking a music technology class.

Using consumer electronic devices such as game controllers then seems like a better option. They are easily available for a low prize at any electronics store, and work out of the box using the generic Human Interface Device (HID) driver available with most operating systems [5], and supported in programs such as Max/MSP, PD and Matlab. This means that no extra interfaces are needed and

³ Usually 7 bit (0-127), although some products allow for sending dual messages giving a 14 bit range.

no software has to be installed to make them work. Since they also draw all their power from the USB-port, no external power supply is needed either.

Of the many different types of game controllers available, we find gamepads the most interesting. They are generally cheaper and smaller than other types of controllers (such as joysticks, flightsticks or wheels) and often have the largest number of inputs. Taking a typical gamepad apart reveals a motherboard with 4 analog and 12 digital inputs. These inputs comply to the standard 0-5 volt sensor input range, and thus most sensors can be connected to the input points on the main board directly. In most cases the motherboards are clearly marked, so it is easy to see where the different connectors are, but in some cases it might be necessary to check with a multimeter which connectors carry signal, +5 volt and ground. For simplicity and for making a generic sensor interface, we usually de-solder the small joysticks, and solder on cables with 3-pin connectors. This makes it is easy to test different types of sensors with the interface.

The result is a generic sensor interface with 4 analog and 12 digital inputs, 8-bit resolution and 100Hz sampling rate. Using a "rumblepad" will even give a couple of analog outputs and small motors, and a wireless gamepad can be turned into a wireless controller. For large projects with a need for lots of inputs, it is possible to connect several controllers through USB-hubs.

3 Sensors

For our musical applications, we are mostly interested in contact sensors that allow for the same type of sensitivity and tactile feel as acoustic instruments. A problem with many commercially available products, such as bend, pressure and position sensors, is that they ship only in standard sizes and defined materials [6]. This constrains the possibilities of the controller to the shape and size of the available sensors, and draws the focus away from the musical applications.

Another problem with commercial sensors is the lack of tactile feel and response, either because they are too thin and small, or they are too big and have too low resolution to be interesting for musical purposes. These problems can be overcome by adding extra material around the sensors, but this also reduces the sensor's response since the padding effectively filters the gestural energy.

This is not to forget that the price of only one commercial sensor is often many times our "10 euro" budget, and in most cases it is necessary to have a number of different types of sensors on a controller. But rather than looking at adding lots of sensors to increase the response of a controller, a solution might be to use the sensing material itself as a transducer. For example, for a "shoe controller" the sole of the shoe could be the sensor itself, rather than adding various sensors to a regular sole such as in [7]. Ideally, conductive materials could be bought for example in rolls and then be adapted to the desired thickness and sensitivity. This would also make it possible to adjust the sensitivity of the sensor to the envisioned usage, for example high resolution for a hand controller and lower resolution for a foot controller. Thus, any form and dimension could be realized with the same technology.

Such conductive materials already exist, for example artificial piezoelectric polymers (PolyVinylidene Fluorid, PVDF) tested in several musical interfaces such as the Magic Carpet [8], or the electret, a porous conductive polymer film used for pick-ups in some acoustic instruments. They are, however, quite expensive and since they are mass-produced they are only available in certain standard sizes from the manufacturers. Other surface conductive materials, for example some types of video tape, can be used directly for making sensors [9,10]. Another low cost alternative is to make new materials mixed of conductive parts and some non-conductive or low-conductive and highly visco-elastic material, usually some polymer, such as the plubber, a pressure sensitive silicon/carbon composite used in the z-tile [11].

We have succeeded in making various types of position, pressure and bend sensors from components generally available in hobby or artwork stores, including conductive ink, adhesive, rubber, tape, elastics and porous materials. Since such materials can be bought in many different sizes, it makes it easy to customize the sensing surface to the interface we are interested in building. As can be expected, the conductive qualities of these materials varies quite a lot, so further research is needed to test different types of materials and improve the consistency and reliability of the response. Also, since we are working with materials where the range and conductivity is unknown, more work will also need to be put into improving the electronic circuits for optimizing the signals to match the 0-5 volt range used in the sensor interfaces.

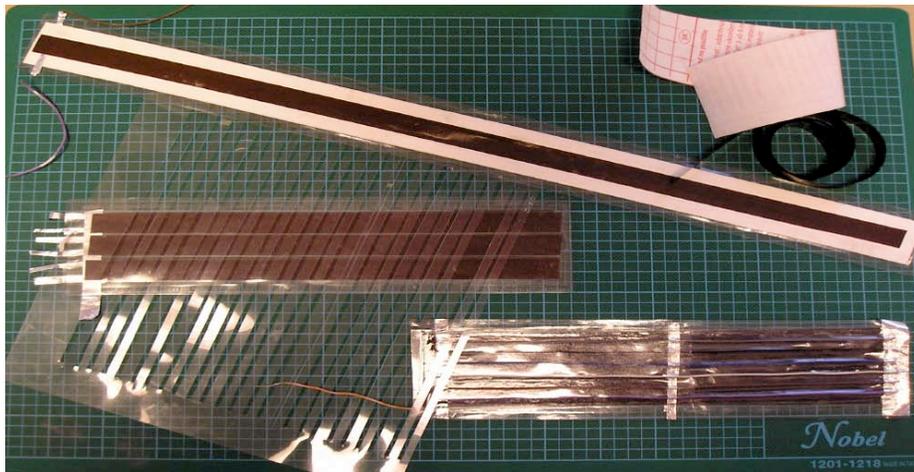


Fig. 1. Various sensors made from conductive materials

4 A prototype musical controller

To test how our interfaces and sensors work in a musical context, we built the *CheapStick*, a simple prototype controller with three pressure sensors made from paper and a position sensor made from video tape (Figure 2, 3). With the sensor interface mounted on the board, this makes a self-contained music controller very close to our "10 euro" budget, and with a USB-plug that can easily be connected to a computer and used with various music applications.



Fig. 2. The *CheapStick* was built with pressure sensors made from paper and a position sensor made by video tape

HID-compliant devices can be accessed directly with the HI object in Max/MSP or the HID object in PD. To help in making an easy-to-use setup for musical applications, we have made the *MultiControl*⁴ program (Figure 4). This program is somewhat similar to the MIDI-based Junxion [12], but adds a number of useful features. The program can access any HID-compliant controller connected to the computer and will automatically detect the active channels from the device. The incoming values are automatically scaled to a usable range of floating point numbers between 0 and 1, and these values can also be smoothed which might be useful if the sensors send a noisy signal. The program can output either Open Sound Control (OSC) messages internally or on the network, or output MIDI so that it can be used with any MIDI-compatible equipment. This program makes it easy to experiment with different mappings, and test out the musical possibilities of the controller.

5 Conclusion and future work

We have presented our current work on turning consumer game controllers into generic sensor interfaces, and making sensors from various conductive materials. This "10 euro" controller performs quite similar to systems that cost many times this price. Although initially motivated by cost and easiness rather than high quality, speed and resolution, we are impressed by the tactile feel and response of some of our low-cost sensors.

⁴ Available from <http://musicalgestures.uio.no>

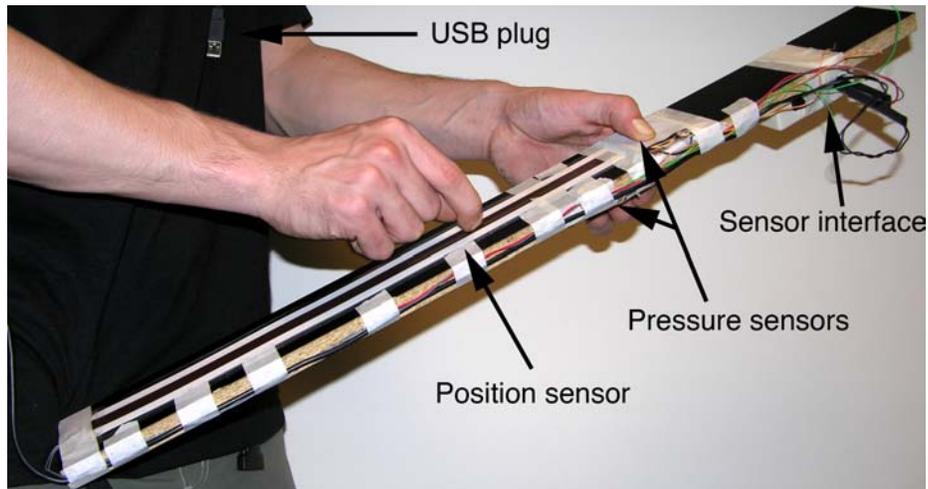


Fig. 3. The *CheapStick* can be played using the right hand to control position, and the left to control the pressure sensors

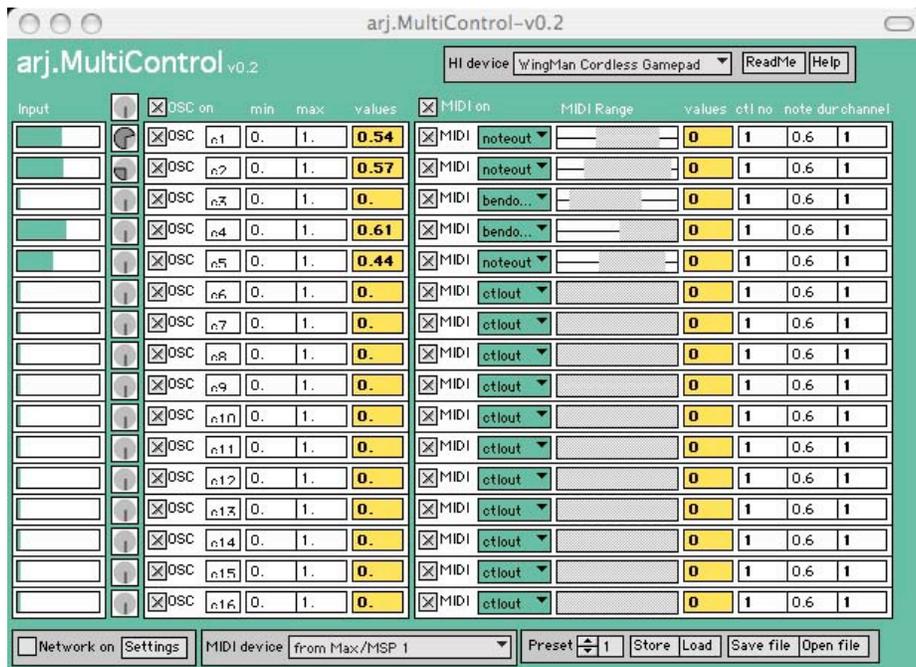


Fig. 4. Screenshot from the MultiControl software which allows for smoothing and scaling of controller data, and output to OSC or MIDI

References

1. Emmanuel Fléty, Nicolas Leroy, Jean-Christophe Ravarini, and Frédéric Bevilacqua. Versatile sensor acquisition system utilizing network technology. In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04)*, Hamamatsu, Japan, pages 157–161, 2004.
2. Jesse Allison and Timothy Place. Sensorbox: Practical audio interface fo gestural performance. In *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, Montreal, Canada, 2003.
3. Dan Overholt. The create usb interface - where art meets electronics. <http://www.create.ucsb.edu/dano/CUI/>, 13 November 2005.
4. Dan O. Sullivan and Tom Igoe. *Physical Computing - Sensing and Controlling the Physical World with Computers*. Thomson Course Technology, Boston, MA, 2004.
5. <http://www.usb.org/developers/hidpage>, 13 November 2005.
6. Marcelo M. Wanderley and Rodolphe Koehly. Methods for the in-house development of sensors for musical applications. Manuscript in preparation.
7. Joseph Paradiso and Eric Hu. Expressive footwear for computer-augmented dance performance. In *Proceedings of the First International Symposium on Wearable Computers, Cambridge, MA*, pages 165–166. IEEE Computer Society Press, 1997.
8. Joseph Paradiso, Craig Abler, Kai-yuh Hsiao, and Matthew Reynolds. The magic carpet: Physical sensing for immersive environments. In *Proc. of the CHI '97 Conference on Human Factors in Computing Systems, Extended Abstracts*, pages 277–278, NY, 1997. ACM Press.
9. T. Escobedo. The synthstick. <http://www.geocities.com/tpe123/folkurban/synthstick/synthstick.html>, 7. May 2005.
10. Andrej Stordeur. Ribbon controller. <http://www.angelfire.com/music2/theanalogcottage/ribcont.htm>, 8. March 2005.
11. Lisa McElligott, Michelle Dillon, Krispin Leydon, Bruce Richardson, Mikael Fernstrom, and Joe Paradiso. *UbiComp 2002, LNCS 2498*, chapter 'ForSe FIElds' - Force Sensors for Interactive Environments, pages 168–175. Springer-Verlag Berlin Heidelberg, 2002.
12. Steim. Junxion (computer program). <http://www.steim.org/steim/junxion.html>, 8. March 2005.