Sebastian Tomczak
Electronic Music Unit
Elder Conservatorium of Music
University of Adelaide
North Terrace
Adelaide, 5005
Australia
stomczak@e-access.com.au

DIY Physical to Computer Interactivity: Background, Issues and Solutions

Abstract

This paper will investigate the process of creating an alternative and affordable DIY gestural control system in which the interface can be quite easily built by the hobbyist who is not necessarily the electrical engineer. Also, the system is to be low in cost should the end user require multiple interfaces.

A number of areas will be discussed. The historical and cultural context of such a project with its links to Circuit Bending (under the proposed implementation) will be presented. The current commercial options, their strengths and weaknesses when compared to each other, and how these spur on the reasoning for this interface are also discussed. An investigation into the aims and specific objectives of such a project, in addition to the varied outcomes is offered. Finally, the specific, desirable characteristics of the hardware to be used, along with a candidate option, are presented.

Historical / Cultural Context

This project relies on the basic principles of physical interface design. A simple definition would be a hardware device capable of reading analogue voltages, thus translating events taking place in the physical world into data signals that are eventually processed and mapped to musical parameters in the physical world.

Although not directly categorised under the umbrella of the term, the implementation of this project as outlined in this paper shares a certain philosophy with circuit bending.

Circuit bending, as defined by Reed Ghazala, an early pioneer and creator of 'alien instruments' made by the process is;

> "… an electronic art which implements creative audio short-circuiting. This renegade path of electrons represents a catalytic force capable of exploding new experimental musical forms forward at a velocity previously unknown. Anyone at all can do it; no prior knowledge of electronics is needed. The technique is, without a doubt, the easiest electronic audio design process in existence." (Ghazala)

The above outlines the concept of 'piggy-backing' on existing devices and technologies through simple modifications and additions, thus changing the purpose and thereby the context in which that particular unit is functioning . This idea is held in common with the methodology and proposed implementation for a physical interface outlined in this paper. A technology originally conceived for something else such as a hobbyist - constructed USB gamepad is removed from its original context, inserted into a different framework and utilised in a new fashion, in this case as a musical interface.

Also, the philosophy of simplicity in design and procedure shared in common with circuit bending can be found in this solution

What Are The Current Arguments?

To understand the aims and ideals of this project, a perspective of the current climate in the field of physical interactivity is needed. There are already a number of interface and sensing solutions commercially available with different strengths and weakness. Two examples of current sensor implementation are the Making Things' Teleo modules (Making Things ref. 1) and Infusion Systems' I-Cube-X control system (Infusion Systems ref. 1).

A quick comparison between the two reveals some similarities but also a number of important differences. Since interfacing from the physical world to the computer world (and not the reverse) is the focus of this paper, only input characteristics will be compared. Both systems output a +5V control signal, used in conjunction with physical sensors to measure events in the real world. Dedicated AC power adaptors are required for each system to work also. Neither are bus – powered which is where energy for a device is supplied via the same connection that transmits and receives data from the host computer system. This would conceivably impact on the mobility of an interface. Also, both systems appear to be relatively user-friendly to the uninitiated. That is to say, no formal electrical engineering or similar education is a prerequisite for the successful installation and use of these devices.

The Teleo system by Making Things transmits data over USB to a host computer. The Teleo Introductory Module (US$159 at time of writing, April 2006) that is required in a Teleo system has four analogueue and two digital inputs (Making Things ref. 2). Further input functionality can be accessed by networking different Teleo Modules such as the Analogueue In Module (US$149 at time of writing, April 2006) (which adds an additional twelve inputs to the system). Sixty-three such modules can be linked together, with only one USB cable connecting the first module to the host computer (Making Things ref. 3). The Teleo system outputs data at a depth of two bits (2^2, a range of 0 – 3) to ten bits (2^{10}, a range of 0- 1023), and the lowest sampling interval is one millisecond (Making Things ref. 4).

In terms of host computing, Teleo objects and classes are available for Max/MSP, Macromedia Flash and also supports the C++ programming. Both Max/MSP and Flash are compatible with Mac OS X and Windows XP. C++ further extends the compatibility and widens the possible applications of the system further. However with some modification and programming, it is possible to interface the Teleo systems directly with other music and sound applications such as Super Collider (Wilson).

The I-Cube-X System is compatible with Max/MSP in host mode (wherein a higher bit depth is available to the user), and can communicate with any software or hardware package capable of receiving MIDI messages in the standalone mode.

The I-Cube-X control system comes in a number of packages that offer different bit depths and number of inputs. All require a dedicated MIDI input on a computer MIDI interface for host data transmission, however the I-Cube-X system features a stand alone mode where data with simple processing can be sent to MIDI devices such as synthesiser modules, samplers and so forth.

The most popular package (approx. US$730.43 at time of writing, April 2006) features 32 analogueue inputs, each at a depth of seven (a range of 0 – 127) or twelve bits (a range of 0 – 4095) . The sampling frequency varies, depending on the bit depth and number of inputs used, and starts at four milliseconds when used in host mode (Infusion Systems ref. 2). The high bit depth and the potentially useful stand alone mode are two highlights of the I-Cube-X. There is also a downscaled version of the system, with less inputs and lower bit depth (approx. US$417.64 at time of writing, April 2006) (Infusions Systems ref. 3) and a wireless I-Cube-X package (US$599.00 at time of writing, April 2006) (Infusion Systems ref. 4). The budget range starts with a beginners kit at US$299.00 (at time of writing, April 2006), seemingly comparable to the downscaled version (Infusion Systems ref. 5).

Both the I-Cube-X Control System and the Teleo Modules appear to cater to a similar market, in that they have somewhat comparable specifications, and the lowest associated cost for both systems are of the same magnitude (US$159.00 versus US$299.00 at time of writing, April 2006).

The question arises to which demographics in the market these products might be aimed at. Learning institutions, especially those concentrating on technology and inter media arts education with a musical angle, are prime candidates, as are established artists and musicians with various backgrounds interested in building their own hardware interfaces with minimal electronics knowledge and low level engineering skills.

It is feasible, however, that there are certain demographics to whom these funds are not available to, for a number of socio-economic reasons. For example, financial gain for an individual wishing to explore some aspect of physical interactivity may be too low to warrant an investment into such currently available commercial systems.

Another consideration to be made in this assessment are more global factors of the economy; the Western currencies are performing higher than their developing countries' counterparts. This could also lead to difficulty for interested individuals and institutions in some countries to obtain such physical interactivity.

Even if such funds are available to the individual or group, there is a certain amount of resource use in certain instances. If, for example, such expensive devices are used in installations, where the technology is engaged for perhaps a week or more, then that resource of sampling from the real world cannot be used in other projects . For a significantly less expensive solution, multiple interfaces could be created and used at once in different contexts, resulting in a gain of efficiency.

In light of the current geo-political climate, certain areas of the world may face tough import restrictions. This would make it harder for the people in those areas to get access to the systems mentioned, since they originate from the United States of America (Teleo) and Canada (I-CubeX). A system that uses parts that can be bought in the same country as the creator of the interface would bypass this problem.

Thus, it is obvious that a niche of the physical interface market has remained unfulfilled by the currently available products.

Reasoning For This Project

It has been shown above that there is a gap in the low cost range for a user – constructed physical interface. This project is perhaps a first step in the direction of finding a solution for this problem.

Investigation

The aim of this project is to investigate an economically efficient and practically viable model for physical to computer interactivity, with a number of useful outcomes.

The term *economically efficient* as it relates to this interface should be seen as a comparison of features, simplicity of construction and usability in light of a cost analysis, made up of parts, labour (both money and time) and shipping.

However, it is difficult to measure certain parameters such as bit depth, response time and sampling interval in terms of how they might effect an overall outcome. Surely, the way that the interface is used and integrated into various situations will vary widely from user to user in terms of what such a device should and should not be capable of.

For example, if a glove with bending sensors attached to each finger is to be controlling the pitch parameter for MIDI messages on five different MIDI channels, then the bit depth of MIDI (seven bits, a range of 0 – 127) would suffice for that particular application.

If instead, however, each finger on the glove is to be mapped to various spectral, time or frequency – based parameters, then the seven bit resolution used in the previous example might not be the preferred option but would of course still work- it just might not sound as good or smooth as envisaged.

For a project to be practically viable, it would need to be readily understood by someone without specialist knowledge. This is along the lines and also an extension of the circuit bending philosophy and approach towards technology, where a great emphasis is put on the simplicity of self – modified / self - created processes and methods. The other factor relating to this is how easily-obtainable the parts needed to make such a device are. Are they specialist parts? Or can they be bought, for the most part, at a local distributor of electronic components?

Practicality also relates to the future availability of the parts i.e. if the user wishes to build another interface in ten years time, will the parts still be as readily accessible? These are the factors that will help to determine if the project is or is not a time-consuming and frustrating one for the end user.

The idea of physical to computer interactivity relates to the creation of music where physical actions or events in the real world are used to control certain parameters, in this case, of music.

If the sensors connected to the interface are designed as part of a musical controller to be played by a human performer, or are used in an installation with human – computer interaction, then it is generally possible to categorise the devices into two groups. The first is *haptic* and refers to an interface or sensor that must be physically touched or manipulated in some way (for example a flex sensor or a pressure sensor) in order to record a change in physical state or movement of the player or involved person.

Non – haptic refers to a sensor type that does not involve direct physical contact, for example a motion detector (Baecker et al, 1989). In both cases, the technique is called *direct gestural acquisition,* since the physical events and changes originating from the player are captured (Wanderley).

Outcomes

There are a number of outcomes directly linked to this project. They are wide ranging in their scope of variety.

First and foremost, a free resource will be created containing highly detailed yet concise instructions regarding the process of constructing the interface, complete with diagrams and photos (if it is felt they are required). This resource will be published in the form of an online website. It will also contain suggestions for sensors and how to attach these to the device, as well as suggestions on where to purchase the parts needed for the creation of the interface in at least more than one country. The aim of this is to create an environment in which novice users are able to learn step by step how to build their own interface. A framework is thus presented to the user, in a form that is digestible for them. This framework serves as a reference base, covering the fundamental knowledge required to create such a device.

Another aspect that will be found in this user resource is a forum, in which individuals can communicate ideas, problems and suggestions to one another. An example of an existing forum of a similar nature Aaron Nelson's *DIY Stompboxes* site forum (Nelson), where novice and advanced users ask and respond to questions of a technical and musical nature. The aim of the forum is to involve the community in the project. New ideas and processed will be made public. Novice users can ask direct questions, and will receive many answers. This system complements the knowledge base, in that the general information found in the base is built upon by specific information on a case by case basis.

Finally, the resource will also contain a section regarding considerations when preparing scores for gestural – based music. This category will be, by its nature, more academic than the rest of the resource. The aim of this section is once again education and making the user base aware of issues, challenges and solutions associated with the creation of scores.

A series of Max/MSP patches, objects and standalone applications for both the Windows and Macintosh operating systems will be created, targeted at both novice and advanced users of the visual programming language.

For example, a standalone synthesizer / sampler might be included that can be operated without needing to be configured or programmed by the end user. However, objects made for data interpretation and interpolation will also be included. These obviously need to be used inside of a Max/MSP patch, and as such would be aimed at more advanced users.

A series of graphing and data recording tools, useful for analysis, will also be developed. All of these software components will also be made available to interested parties free of charge via the web resource.

Another outcome of this project will be a completed musical interface. The exact nature of this interface is uncertain at this point in time, however it will most likely be for one performer and will be designed to function in both solo and ensemble concert situations.

One or more works will be written for this instrument, notated using a distinct gestural scoring system. These will be performed during either public or closed concerts at the end of 2006.

Furthermore, a written thesis documenting the processes of creation for the interface, online user resources and scored work, investigating cultural and philosophical aspects of the projects as well as possible wider application and appeal

Characteristics

The hardware portion of the physical interface itself has a number of desirable characteristics. First and foremost it should be cost-effective, in the range of AU$20 – AU$30 for components of the device. This cost estimate does not including a casing or any postage associated with the parts.

In keeping with Circuit Bending philosophy, the interface should be idealistically simple in design. The end user should be able to quite easily assemble the device themselves, without any help.

The interface will be able to transmit data via USB. This is integral, since USB cables are readily available and the USB protocol is ubiquitous in modern computer systems (USB Implementers, Inc ref. 1). Also, through the use of USB, the possibility remains of simultaneously using multiple interfaces on the one host computer .

Additionally, a beneficial feature for the interface would be to be completely powered by the USB bus. If this feature is included, there will be no need for external

AC power adaptors or battery packs. Conveniently, the USB bus carries a +5V signal between pins one and four (Pinouts.ru), so this option is a reality (since many micro controller units as well as some of the sensors, require the +5V power source to function).

Through the use of a bus powered device, benefits such as increased portability, less setup complexity and fewer connections and parts are realised.

Finally, a suitable data protocol for transmission needs to be used. A candidate for this is the HID, or human interface device protocol. It is currently used for such tasks as receiving data from input peripherals like computer mice, keyboards, USB gamepads and similar devices (USB Implementers, Inc. ref. 2). Data from a HID bus can be directly and effortlessly read into programs such as Max/MSP, thus making software development both easier and more efficient time wise.

Candidate Hardware / Software

The perfect candidate, which fulfils all of the aforementioned criteria would be a project based around the Mjoy USB gamepad. These plans and software were written by Mindaugas Milasauskas in 2004 and are available under a GNU General Public License for non-commercial work (Milasauskas, 2004).

The only microcontroller unit used in the Mjoy is an Atmel ATMega8- 16- PI. It is an affordable processor, costing in between AU$11 (at the time of writing, April 2006) (Grantronics, 2006) and AU$17.54 (Dontronics) (at the time of writing, April 2006) depending on supplier. The ATMega8 is used in conjunction with a number of basic components including resistors, capacitors, diodes, zener diodes and a quartz oscillator.

The processor, as used in the way described by Mindagaus, features six analogue to digital inputs, two of which have a ten bit depth. The other four inputs have a bit depth of eight (2^8, a range of 0 – 255). The sampling interval is between five and ten milliseconds.

In addition, Mindagaus uses eleven pins on the Atmel chip to make a twenty-eight toggle matrix, in which a connection between two points, or lack therefore, is shown digitally.

It remains to be seen how useful such a feature would be in a musical interface. Certainly, sensors like a tilt switch (which is either on or off) may be relevant to some users. However, since a *matrix* of toggles is implemented, each toggle is not entirely independent because pins are shared between a row of toggles. This co-dependency creates possible 'blind spots' within the matrix.

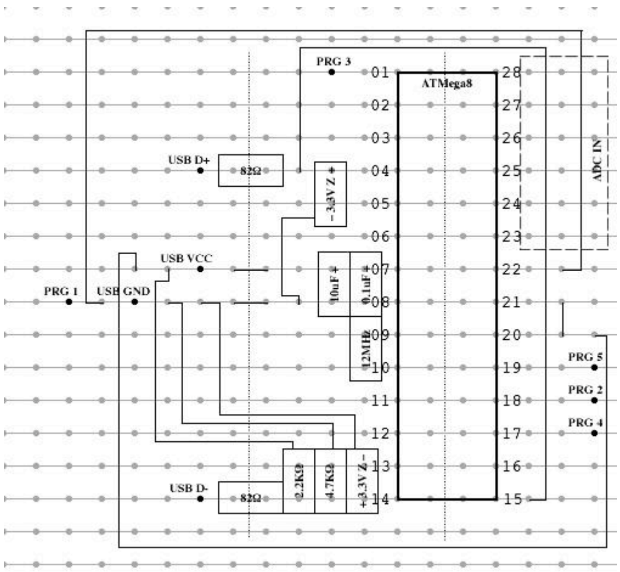

Figure 1. *An example of a strip board layout for an Mjoy interface*

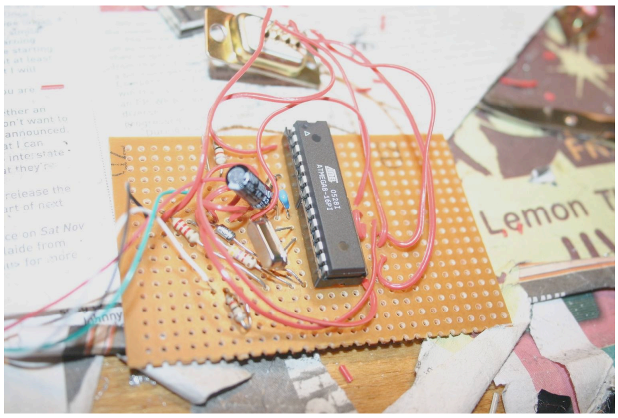

Figure 2. *An example of an Mjoy interface*

Conclusion

This project will achieve a number of outcomes. The interface is an effective hands - on learning tool for the novice. By providing a viable alternative at roughly one tenth the cost of other systems, this venture also presents the user community with a cost – efficient interface from the physical world to the virtual one, thereby filling in a perceived need in this niche.

References

Ghazala, R. *Circuit Bend*. http://www.anti-theory.com/soundart/circuitbend/ (11^{th} April 2006).

Making Things. *TELEO*. http://www.makingthings.com/teleo.htm (11^{th} April 2006).

Infusion Systems. *I-CubeX Online Store: The Ultimate Controller*.

http://infusionsystems.com/catalog/index.php (11th April 2006).
Making Things. *PRODUCTS.* http://www.makingthings.com/products/products.htm#modules (11 April 2006).
Making Things. *TELEO NETWORK.* http://www.makingthings.com/products/documentation/teleo_user_guide/teleo_network.html (11th April 2006).
Making Things. *T.INTRO.AIN.* http://www.makingthings.com/products/documentation/max_externals/t_intro_ain.htm (11th April 2006).
Wilson, S. *[sc-users] Client port is hardcoded? was: OSC and Kroonde.* http://www.create.ucsb.edu/pipermail/sc-users/2005-February/016551.html (18th April 2006).
Infusion Systems. *I-CubeX Online Store.* http://infusionsystems.com/catalog/product_info.php/cPath/21/products_id/28 (11th April 2006).
Infusion Systems. *I-CubeX Online Store.* http://infusionsystems.com/catalog/product_info.php/cPath/21/products_id/91 (11th April 2006).
Infusion Systems. *I-CubeX Online Store.* http://infusionsystems.com/catalog/product_info.php/cPath/21/products_id/98 (11th April 2006).
Infusion Systems. *I-CubeX Online Store.* http://infusionsystems.com/catalog/product_info.php/products_id/110 (11th April 2006).
Baecker, R.M. and Buxton, W.A.S. 1989. "The Audio Channel" *Human-Computer Interaction: A Multidisciplinary Approach.* Morgan-Kauffmann. 393-399.
Wanderley, Marcelo. *Discussion Group on Gesture Research in Music.* http://recherche.ircam.fr/equipes/analysesythese/wanderle/Gestes/Externe/Capture.html (11th April 2006).
Nelson, Aaron. *DIYStompboxes.com- Index.* http://www.diystompboxes.com/smfforum/ (11th April 2006).
USB Implementers Forum, Inc. *USB.org - FAQ: Basic Information.* http://www.usb.org/about/faq/ans1#q8 (11th April 2006).
Pinouts.ru. *USB pinout and signals @ pinouts.ru.* http://pinouts.ru/data/USB_pinout.shtml (11th April 2006).
USB Implementers Forum, Inc. *USB.org – HID Tools.* http://www.usb.org/developers/hidpage/ (11th April 2006).
Milasauskas, Mindaugas. *DIY USB Joystick Controller Project*. 2004. http://www.mindaugas.com/projects/Mjoy/ (11th April 2006).
Grantronics. 2006. *Grantronics.* http://www.grantronics.com.au/components.html (11th April 2006).
Dontronics. *Dontronics - Atmel/AVR Microcontrollers.* http://www.dontronics.com/cat_hard_micro_avr.html (11th April 2006).