

# Musical Interaction Design with the CREATE USB Interface

## Teaching HCI with CUIs instead of GUIs

Dan Overholt

CREATE / MAT

Center for Research in Electronic Art Technology / Media Arts and Technology Program  
University of California, Santa Barbara  
dano@create.ucsb.edu

### Abstract

*The CREATE USB Interface, or CUI, is a tool for both research and teaching that was developed to facilitate the design of new musical instruments and other interactive art projects. Teaching Human-Computer Interaction in music and related media arts presents a requirement for extremely responsive control methods, and a very different design theory than conventional HCI. A new course titled Media Interface Technology was developed in order to give students the necessary engineering expertise and aesthetic awareness to create successful input/output devices, and map their functionality onto musical synthesis or other interactive algorithms.*

## 1 Introduction

At the Center for Research in Electronic Art Technology and the Media Arts and Technology graduate program at U.C. Santa Barbara, we explore new metaphors for artistic interactivity that connect the physical world with the virtual realm. We develop new techniques for computing that generate music and visual arts in a myriad of ways; but in order to put forth these techniques, we must create new sensors, and build interfaces that can better grasp their control and generation. The CREATE USB Interface (CUI) allows us to bind physical actions and processes to corresponding digital expressions.

To simplify this process of connecting the real world to the virtual, the CUI provides the basic electronics needed to capture sensor input or control actuator output. The CUI comes with a USB port, power/activity LEDs, a reset button, a programming button, header pins for connecting external sensors, and a prototyping area. At the heart of the board is the PIC18F4550, a versatile microcontroller made by Microchip Technologies, Inc. The PIC18F4550 features thirteen A/D inputs, eighteen general-purpose I/O ports, built-in USB support, and an efficient RISC-like instruction set. The CUI circuit only uses one of the PIC's I/O ports; the remaining ports are available for user applications to be connected to sensors/actuators or additional circuitry built in the prototyping area.

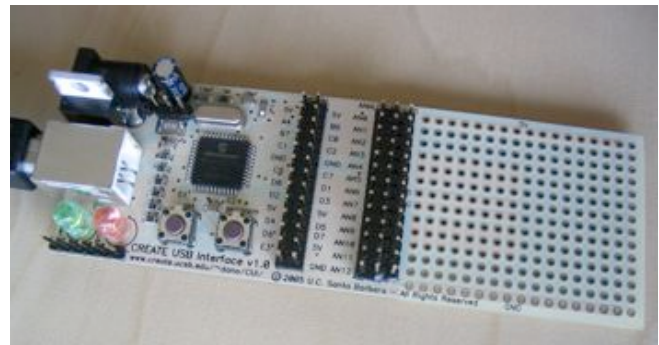


Figure 1. The CREATE USB Interface v1.0.

The CUI provides a flexible, easily extensible platform for designing new musical instruments and other electronic art works. It allows for expedient development of new prototype interfaces, and has several advantages over many commercial solutions – some of these are lower cost, direct USB transfers (no MIDI or serial interface needed), can be powered via the USB cable, and because the circuit and firmware were designed ‘in-house’ from the ground up, there is local expertise in low-level areas (such as firmware details and interrupt handling).

## 2 Media Interface Technology Course

The Media Interface Technology course is designed to explore the use of sensor technologies for computer music, visual art, and responsive performance systems, as well as interactive environments and installations. Students get an in-depth analysis of current technologies, and acquire experience that will allow them to research and develop custom sensor systems in their own work. It introduces the principles and operation of many sensor techniques and discusses their applications in new musical instruments, gestural human-computer interfaces, and the visual and spatial arts. It covers the design of computer interface systems including analog/digital electronics along with human factors and interaction styles. It is an applied theory course with laboratories and student projects that can

potentially continue as independent research in following terms of study. Topics in the course include:

- HCI design theory for music and the media arts
- Microcontroller programming and custom sensor design
- Analog-to-digital conversion and RF wireless systems
- Sensor applications and signal conditioning electronics
- Ergonomics, haptics, and emerging interface technologies
- Pressure, position, optical, inertial, capacitive, magnetic, and ultrasonic sensing techniques
- Communication protocols, signal processing, feature extraction, and mapping schemes
- Output modalities include music/sound, interactive visuals, and motion control/robotics

## 2.1 Course Motivation / Objectives

Technological breakthroughs in computation have impacted both music and engineering, opening up possibilities for expression far beyond those of previous generations. This has especially affected the way we think and produce art, but in many cases has relegated user interaction to simple start/stop or point/click paradigms. The lack of connectivity between the real world and the digital world severely limits the amount of user interaction that is possible—this course therefore focuses on extending computer systems with novel real-time interfaces. It seeks to provide students with conceptual frameworks for thinking and technical skills for the production of new interactive systems.

The goal of this course is the application of sensor technology to areas such as new audiovisual interfaces, music/sound manipulation and spatialization systems, interactive installations/virtual environments, 2D/3D video and graphics interfaces and other innovative devices for interacting with and manipulating computer systems. It is expected that the course will lead to a completed prototype of a new interface as a final project. A student who successfully completes this course will have experienced taking an original interface through a full cycle of design, prototype, evaluation and testing. The system demonstrated at the end of the term should explore the relation between a physical interaction and musical / visual / spatial content, and is to be accompanied with a written report / web site describing the intent and an evaluation of the interface.

## 2.2 Musical Interaction Design Framework

The Media Interface Technology course is supported in the Media Arts and Technology program, CREATE (Music department), and the Art department at U.C. Santa Barbara. As such, the course addresses broader issues than just musical interaction, however many student projects have

focused on musical interaction design. In fact, the course might be somewhat biased toward design frameworks that involve interaction with sound in some way, since the author's background is firmly in musical interfaces. Some of the questions that are commonly raised during discussions about musical interaction design frameworks include:

- How to generate artistic expression from sensor interfaces?
- How to enhance the interactive experience with sensor-mediated content?
- How to control structure from free-form expression?

Issues such as these tend to be universal to many forms of new media art, and are strong indicators of the differences in design theory between musical HCI and conventional HCI. Digital technology today offers compelling new methods for sound synthesis, but typical human-computer interfaces (point/click or on/off type inputs) lack the expressive nuance and subtlety (accuracy and precision in engineering jargon) needed to manipulate them effectively in real-time. With the right framework, students can attempt to address this issue in their own work by designing more responsive control methods and devices with the CUI.

The research aspects of the author's own work with the CUI involve the design of new interfaces attempting to match musical sound in tangible interaction design to human perception through highly responsive input devices. A distinction may be drawn here between purely responsive systems, and those that exhibit some autonomy or other inherent behavior that makes complete control impossible. While autonomous systems may sometimes have interesting output (e.g., algorithmic music), the author's research tends to focus on keeping most of the intelligence in a system with the human rather than the machine, using the CUI as a means to explore the potential for new human capabilities. A non-musical example of such a human endeavor is the development of high-performance aerobatic remote-control airplanes – with these new technological tools, R/C pilots have been able to master their skills to a level beyond that of the most accomplished pilot of a real airplane.

## 2.3 Workshops with the CUI

A series of workshops have been developed that explore designed interactions for new media interfaces using the CUI. Most recently, from May 31<sup>st</sup> - June 3<sup>rd</sup> 2006 there was a 4-day workshop with the CUI at STEIM, the Studio for Electro-Instrumental Music in Amsterdam (see Figure 2). Several other venues are in the planning stages for similar workshops as well, and the CUI will be used at a sensor workshop for artists and performers July 24<sup>th</sup> – 28<sup>th</sup> at U.C. Berkeley's Center for New Music and Audio Technologies (CNMAT) taught by Adrian Freed and Michael Zbyszynski. As artists, we are always faced with the challenge of

investigating and incorporating new technologies into our work while finding our own voice or approach to expression within them. The workshops intend to help in this process.



Figure 2. The *Beyond Hacking* Workshop at STEIM.

While such workshops are by necessity much shorter than the Media Interface Technology course at UCSB, they are intended to provide an accelerated learning model by providing the hands-on experience needed to establish avenues for further development by the workshop participants when realizing their own projects. They are aimed at participants who already have some level of expertise with the real-time software tool of their choice, i.e., Max/MSP/Jitter, SuperCollider, PD, etc., and come to the workshop with an idea for an inspiring way of getting 'hands-on' with their digital audiovisual data.

### 3 CUI as a Teaching Platform

There are many practical problems commonly associated with laboratory courses that involve the building of electronic prototypes. One of the most common is the need for a device programmer in order to update the firmware of an embedded microcontroller. The CUI gets around this problem by using a "bootloader" that allows students to load new firmware into their CUI via the USB connection. This is accomplished simply by resetting the CUI while holding down the program button. The status LED then flashes while the new firmware is uploaded, and upon another reset the CUI runs the new code. In the classroom, this speeds up the debugging process, as there is no need to remove the microcontroller from the circuit and wait for the programmer. The firmware for the CUI is compiled using the free C18 compiler and development environment MPLAB from Microchip, which runs on Windows (or VirtualPC on the Macintosh).

However, it is not always necessary to modify the firmware of the CUI. Although the students in the Media Interface Technology course are expected to learn the

necessary steps to do so, many simple sensor-based (input-only) interfaces can be developed using the default firmware we have developed for the CUI. This is especially useful for people who are using the CUI outside of U.C. Santa Barbara, both because they do not need to purchase their own device programmer to be able to update the firmware, and because the default firmware works with many different sensors and host programs "out of the box".

#### 3.1 HID vs. Serial Data Transfers

The CUI uses a part of the USB specification called HID, for Human Input Device, as do many plug and play peripherals today. It works with OS X, Windows, and Linux hosts, enumerating as a multi-axis, multi-button game controller type device, thereby making it simple to gather data in program such as Pd, Max/MSP/Jitter, SuperCollider, and other programs that support USB game controllers. It sends updates at a rate of 100Hz, and the default firmware is configured to send all 13 analog inputs as well as 12 digital inputs to the host computer. The analog inputs are sampled using the PIC microcontroller's internal A/D converter, which has 10-bit accuracy.

The fact that the PIC microcontroller used on the CUI has built-in USB support, rather than relying on another chip to relay data via USB is one of its main advantages. Many other interfaces (such as Arduino, Wiring, Gainer, etc.) use a microcontroller without hardware USB support, relying on a chip such as the one made by FTDI, inc (Future Technology Devices) to translate the legacy serial port of their microcontroller into USB-compatible data. As such, they are unable to function as a true HID device.

The HID protocol is important for today's modern computers, and may be considered significant especially for musical interfaces, as the latency and jitter can be much lower than devices that use "virtual COM port" style transfers. HID is also what manufacturers use in standard mice/keyboard/joysticks, so the CUI makes use of solid, debugged host drivers that don't exhibit some of the strange behavior the author has seen with virtual COM ports. For example, on Apple OS X, the HID device drivers are part of the lowest level of the OS, using wired kernel memory to store data received from a HID device. Additionally, if a device prototyped with the CUI was desired to begin marketing and selling, most of the engineering is already done since it is already a true USB plug and play device.

There are many benefits to using the HID protocol for USB transfers, but there is currently one caveat – most host platforms tend to focus their HID support entirely around input events, leaving output from the host to the CUI via HID as a future implementation, or using proprietary force feedback protocols. For projects that use the CUI to drive actuators such as lights or motors, there are three different approaches that can be used. The first is to change the firmware of the CUI to a generic (bi-directional) HID device rather than a game controller type device, the second is to

use force-feedback to send data from the host to the CUI, and the third is to step back to virtual COM port style transfers, which can be done entirely in firmware with the CUI.

Unfortunately, there isn't good support yet in programs such as Max/MSP/Jitter, SuperCollider, PD, etc. for the force feedback protocol or generic HID. Some students have written their own host programs in C++ in order to send data to the CUI when it is programmed with generic HID firmware (Newman 2005). Future work may address this issue by creating custom objects/plugins for popular host applications. There is a downloadable force-feedback object (Kosma) for Max/MSP that the CUI has limited support with (by emulating the Logitech force-feedback protocol), but this has some built-in limitations due to the protocol itself. So, for installations and other devices that need output, the simplest approach right now is to use virtual COM port transfers (via the *serial* object in Max or the *comport* object in PD) due to the lack of software support for bi-directional HID in such programs. One example of this approach is a custom firmware written for the CUI that sets the brightness levels of 12 LEDs connected to its digital output pins using PWM (Pulse Width Modulation) in response to serial commands sent from Max/MSP or any program capable of sending serial data.

In addition to the Media Interface Technology course at U.C. Santa Barbara, the CUI is being used pedagogically at several other institutions such as the Kunsthochschule für Medien Köln, Germany, and the City College of New York. Interest in using the CUI from around the world has been increasing, and pre-built/bootloaded boards with the default firmware (ready-to-use for sensor input) can be sent to those who cover the cost of making them/shipping (about \$50 each). Programs at universities can choose between getting pre-built CUIs or bare circuit boards and a bill of materials if they are able to do the surface-mount soldering themselves. Students in the Media Interface Technology course make their own CUIs using a template, solder-paste, and a skillet for the surface-mount parts, learning some "tricks of the trade" along the way.

### 3.2 Project Case Studies

This section examines a few example projects using the CUI from the Media Interface Technology Course at UCSB, which has been offered twice, once in 2005 and once in 2006. Most of the projects have complete documentation online as listed in the reference section of this paper, and therefore will not be covered in as much detail here. These particular projects were only some of those most directly related to musical interaction design – there were many other projects as well as seen on the course web site (Projects from Media Interface Technology Course 2005). Please visit the site to see more projects.



Figure 3. August Black playing *El Lechero*.

*El Lechero* (Black) is an instrument and physical computer interface, mixing the functionality of a joystick with the instrumental feel of a keyboard or accordion. Two blocks, each with 4 pressure sensitive keys, are fastened to a third center block on two joints of rotation. By holding the two outer blocks with the hands, one can play the finger keys while simultaneously unfolding *El Lechero* on its two rotary pivots. The instrument uses magnetic (hall-effect) sensors and an embedded CUI. The design is derived from musical instruments and is geared towards electronic performances for sound and image.



Figure 4. Graham Wakefield's and Wesley Smith's *Tangled Listening-Playing Device*.

The *Tangled Listening-Playing Device* (Wakefield, Smith 2005) is an example of an instrument that uses both input and output from a CUI. Its design stemmed from the creator's desires to merge some of the most crucial audio technology inventions from the previous century into a hybrid object. It is a network of tangled audio technology, based around a turntable deck, electric guitar strings, and optical sound tracks. The speed of the turntable is controlled by the CUI using the force feedback protocol, and a FET transistor to modulate the current to the turntable's drive motor. An infrared optical sensor is connected to the CUI to send data back to the host computer, which in turn generates sound and controls the turntable.

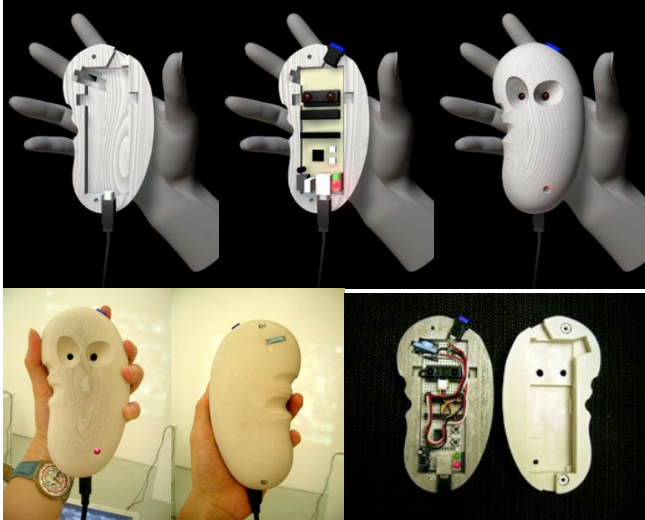


Figure 5. Haru Ji and Lance Putnam's *Airemin* interface, 3-d model shown at top, with embedded CUI at bottom.

The *Airemin* is a USB musical interface that measures its distance from objects using infrared sensing. It is inspired by the desire to have a compact and easy-to-handle Theremin-like device. It can be played with the other hand or scanned against any other object, like another performer. The device has in addition a thumb button on the top and a finger dial on the back. The device effectively gives the user two continuous control axes and a trigger to start/stop events.

The 3-d modeling program Maya was used to construct a virtual model of the device, and the resulting files were sent to 3-d printer (Z Corporation) to make the physical object. The first piece for the interface was titled *Airecell* and allowed the performer to control both sound and graphics composed of cellular-like sound-producing graphic elements drifting over a sea of noise. The cells pulsed in response to user interaction, emitting sounds of harmonic sine waves, and were produced using SuperCollider for the sound and OpenGL for the graphics.



Figure 6. The author's own *BoingBoing* interface.

*BoingBoing* is a new musical performance interface built by the author that provides the musician with four sensor-equipped ping-pong balls. The springs vibrate at frequencies less than 20Hz, thereby lending unusual qualities to the music, such as collisions, bounces, trembles, shudders and shakes. The performer can adjust the spring constants of

each ball by raising or lowering the corresponding rod. A 2-axis accelerometer is contained inside each ball, and there are simple knobs above each of the balls to adjust other parameters of the synthesis. All sensor data is sent to the computer via a CUI. "Quiver", the first composition for *BoingBoing*, was performed at a CREATE concert on 10 November, 2005 along with Lance J. Putnam and Wesley Smith (Overholt 2005).

### 3.2 Current and Future Developments

One of the most recent developments for the CUI is a wireless RF add-on that uses Bluetooth to communicate with the host computer. The Bluetooth transceiver is made by SparkFun Electronics, and the battery is a Lithium-Ion type, very similar to those used in the iPod. These additions add very little size to the CUI, as shown in figure 7. Bluetooth data is received on the host computer via a virtual COM port in Max/MSP/Jitter or PD and achieves an update rate of 100Hz for all of the analog and digital inputs on the CUI. Although the Bluetooth functionality requires different firmware than USB, users may switch freely between USB and Bluetooth using the bootloader.

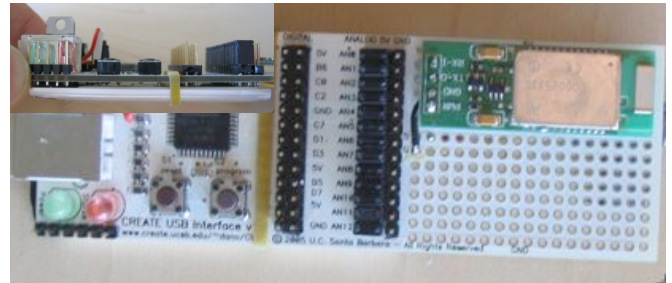


Figure 7. Bluetooth transceiver and Li-Ion rechargeable battery for the wireless CUI.

Another recent addition to the CUI's capabilities is the ability to control an iPod using Apple's remote control protocol over the dock connector. This is a feature that will be used in an upcoming collaborative installation, as well as other future projects. There are many other developments currently underway as well, such as support for I2C communication bus, a protocol that some sensors use to communicate with microcontrollers, and SPI, another popular protocol for devices such as LCD displays. All of these developments (both hardware additions and custom firmware versions) are being made open source in order to enhance the overall community of people working with the CUI.

## 4 Conclusion

Teaching musical interaction design using the CREATE USB Interface brings students with a wide range of backgrounds together, and exposes both technical and theoretical aspects of Human-Computer Interaction for musical expression as well as other artistic areas. The main

goals of the Media Interface Technology course at U.C. Santa Barbara's Media Arts and Technology program are to provide the technical skills necessary to design and build successful interfaces, develop a conceptual framework for musical interaction design as well as visual and spatial arts, and gain experience in an individual or group project that involves the design of a new interface and maps it to musical, visual, or spatial output. Many interesting projects have already been completed, and the CUI is used in the ongoing series of courses taught at U.C. Santa Barbara as well as in a variety of workshop forms tailored to different settings.

## 5 Acknowledgements

Media Arts and Technology students Wesley Smith and Rama Hoetzlein kept things moving through the formative stage of the CUI project. They volunteered many hours to help develop code and teach other students about PICs and basic electronic circuitry. All of the students involved in the Media Interface Technology courses have made this a fun project, and many thanks go to Alex Norman for his role as teaching assistant for the Media Interface Technology course in 2006.

The author gratefully acknowledges the support and inspiration from the following people: Curtis Roads, JoAnn Kuchera-Morin, B.S. Manjunath, Stephen Travis Pope, Matthew Turk, Marcos Novak, and other faculty members of the Media Arts and Technology program, Art, and Music and Engineering departments at UCSB. Some of this research was supported by the National Science Foundation's IGERT Program in Interactive Digital Multimedia (Award #DGE-0221713) at University of California, Santa Barbara.

## References

- Black, A. "El Lechero":  
<http://aug.ment.org/lechero/>
- CREATE USB Interface:  
<http://www.create.ucsb.edu/~dano/CUI/#CUIv1/>
- CUI Ideas and Support Forum:  
<http://www.mat.ucsb.edu/CUI/>
- Future Technology Devices International Ltd.  
<http://www.ftdichip.com/>
- Ji, H., Putnam, L. "The Airemin":  
<http://www.uweb.ucsb.edu/~lputnam/airemin/>
- Kosma, W.: Force Feedback object for Max/MSP/Jitter  
<http://www.wojciechkosma.art.pl/>
- Media Interface Technology Course:  
<http://www.mat.ucsb.edu/594O/>
- Microchip Technologies, Inc.  
<http://www.microchip.com/>
- Newman, E. "DrawBot 1.0":  
<http://www.mat.ucsb.edu/~e.newman/drawBot/>
- Overholt, D. "BoingBoing":  
<http://www.create.ucsb.edu/~dano/boingboing/>
- Projects from Media Interface Technology Course 2005:  
<http://www.mat.ucsb.edu/~dano/sensors/>
- Wakefield, G., Smith, W. "Tangled Listening-Playing Device":  
<http://www.grahamwakefield.net/MAT-S05/594O/index.htm>
- Z Corporation.  
<http://www.zcorp.com/>