

Satellite CCRMA: A Musical Interaction and Sound Synthesis Platform

Edgar Berdahl
Center for Computer
Research in Music and
Acoustics (CCRMA)
Stanford University
Stanford, CA, USA
eberdahl@ccrma.stanford.edu

Wendy Ju
Center for Computer
Research in Music and
Acoustics (CCRMA)
Stanford University
Stanford, CA, USA
wendyju@ccrma.stanford.edu

ABSTRACT

This paper describes a new Beagle Board-based platform for teaching and practicing interaction design for musical applications. The migration from desktop and laptop computer-based sound synthesis to a compact and integrated control, computation and sound generation platform has enormous potential to widen the range of computer music instruments and installations that can be designed, and improves the portability, autonomy, extensibility and longevity of designed systems. We describe the technical features of the Satellite CCRMA platform and contrast it with personal computer-based systems used in the past as well as emerging smart phone-based platforms. The advantages and trade-offs of the new platform are considered, and some project work is described.

Keywords

NIME, Microcontrollers, Music Controllers, Pedagogy, Texas Instruments OMAP, Beagle Board, Arduino, PD, Linux, open-source

1. INTRODUCTION

As instructors of CCRMA's course in Physical Interaction Design for Music [17], we have noticed that many of incredibly novel and innovative musical instruments and installations created in our course over the years have very short lives; even projects demonstrated at the NIME conference [18, 15, 19, 8, 3, 1, 4, 14, 6] are often inoperable just a year afterwards. Conversations with researchers and musicians at a wide variety of other institutions indicate that this is phenomenon is not isolated to CCRMA alone, and that, even for the very-motivated, it takes enormous effort to keep NIME instruments in a functional and performable state. This lack of longevity and robustness creates a situation where 1) the quality of music produced by NIME instruments is limited by the fact that the instruments do not last long enough for musicians to develop expertise, to compose great scores, or refine the initial instrument designs, and 2) the quality of shared learning from NIME instruments is limited by the fact that the instruments are seldom transferred or built upon except by the same researchers who



Figure 1: Satellite CCRMA

originated a particular design. While the rapid obsolescence of new musical controllers might be caused by the relatively short attention span of students or by the focus on novelty within the community, we believe that some element of the blame might be placed on the computer platforms that are at the heart of most NIME instruments.

Satellite CCRMA (Figure 1) is a musical interaction design platform designed to support the creation of new instruments for musical expression as well as sound installations. It incorporates a single-board OMAP-based Linux computer, an Arduino-based microcontroller, and a breadboard for electronics prototyping. By creating a platform which includes a small, inexpensive and autonomous Linux computer, we hope to free NIME designs from the constraints and obstacles associated with platforms which use more general-purpose laptops and desktop workstations.

This paper describes the Satellite CCRMA platform design, its underlying rationale, our initial forays into introducing this platform to students in our Physical Interaction Design for Music course, and what we have learned so far. While it will be many years before we can empirically determine whether our platform is indeed longer-lived than the alternatives, our interim assessments of the affordability, adoptability and the extensibility of the platform give us great optimism that that the platform will be of great use to a large number of people within the NIME community and beyond.

2. PLATFORM DESCRIPTION

2.1 Hardware

The Satellite CCRMA platform is centered around the OMAP35x embedded processor line from Texas Instruments, which can run Linux. We currently use the Beagle Board (see the maroon-colored board in Figure 1), which features the OMAP3530 processor. The processor incorporates a superscalar ARM Cortex-A8 core running at 600MHz as well

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NIME'11, 30 May–1 June 2011, Oslo, Norway.
Copyright remains with the author(s).

as an a TMS320C64x+ 16-bit fixed-point DSP running at 430MHz. The Beagle Board connects by way of a USB hub incorporating Ethernet to the Arduino Nano, which is inserted into a solderless breadboard (see Figure 1, rear). The following list shows the parts that we used in our initial Satellite CCRMA design with our students during the Autumn quarter in 2010:

- Beagle Board Rev C4¹
- Arduino Nano²
- Solderless breadboard
- 4GB (or larger) SD card
- GWC Technology HE2440 USB 2.0 4-Port Hub with Ethernet Adapter
- Two GT Max adjustable-length USB cables
- Ethernet cable
- 2.5A 5V switching power adaptor (For example DVE DSA-15P-05 US)

Since this deployment, we have begun moving the platform to the similar Beagle Board XM. Depending on the intensity of the output sound and the loudspeaker size required, we suggest supplementing the kit with compact mobile speakers so that sound production can be localized to the instrument.

2.2 Software

We have prepared a special SD card image for the platform that boots Ubuntu Linux into an environment with pre-installed Linux audio applications including the Jack audio server, Pure Data Extended, Faust, JackTrip and ChuckK. Because Satellite CCRMA can easily be connected to the Internet, many new packages can be easily installed (e.g. `sudo apt-get install lynx`). Other packages may need to be compiled for the ARM architecture; however, cross-compiling may not be necessary because the SD card installation comes with the gcc tools preinstalled. The SD card image is especially valuable because we have tested it—otherwise, compiling a new SD card image from scratch can easily require multiple days of work. For more information on our SD card image and the community we are building for artists, please see the following link: <https://ccrma.stanford.edu/~eberdahl/Satellite>

In our class, we currently have students program their sound synthesis engines using Pure Data Extended. This graphical environment allows rapid prototyping by connecting together objects using patch cords. For example, a student can login to his or her Satellite CCRMA kit remotely over Ethernet, load the audio server using the command `qjackctl &`, start the audio server, and finally start Pure Data Extended by typing `pd &`. This causes an X-Window to be forwarded over the Ethernet connection, including something similar to the Pure Data Extended patch shown in Figure 2.

3. MOTIVATION

A generic architectural diagram for musical interaction design platform is shown in Figure 3. Although the specific microcontrollers, computer operating systems, hardware, software, firmware, communications protocols, etc used by different members of the NIME community differ, this diagram

¹See <http://beagleboard.org/hardware>

²See <http://www.arduino.cc/en/Main/ArduinoBoardNano>

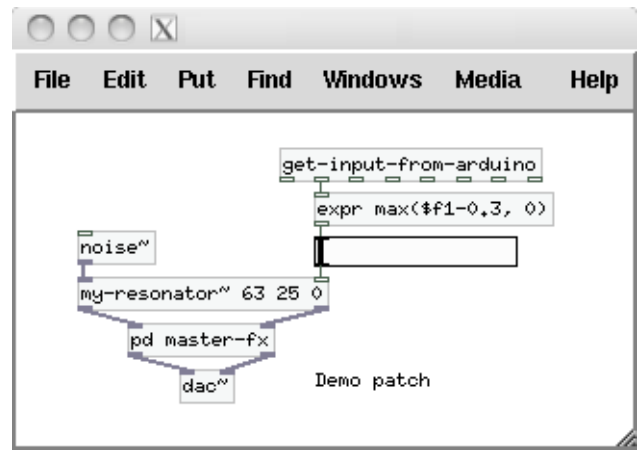


Figure 2: Demonstration patch in Pure Data

reflects the fact that most musical interaction platforms use both a microcontroller (to collect and process sensor data, and also to control actuators) and a microprocessor (to perform more computational challenging tasks such as sound synthesis). What we have observed is that the weak link, as far as longevity and robustness are concerned, is the computer housing the microprocessor. Because computers are expensive, people seldom devote separate computers to their NIME designs; instead they use the same machine that they are writing emails and theses on as the critical engine of their new musical instruments. These instruments thus suffer collateral damage every time we upgrade our operating systems, or install a new version of Java, or switch to new hardware.

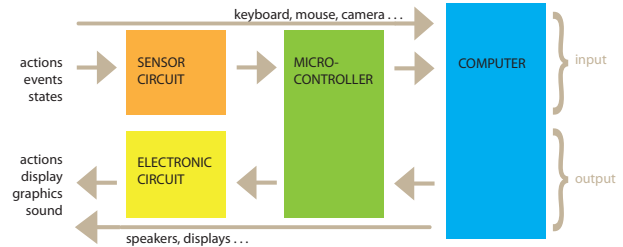


Figure 3: Physical Interaction Design Platform Architecture

At a base level, any system that we would use in our class would need to be affordable (students balk at paying more than \$150—what they pay for textbooks in a normal class), adoptable (able to be picked up and put to novel use within a school term), and extensible (able to support a wide variety of ideas that we the instructors didn't think of when we gave the students the systems). We also had the following criteria in mind when we were developing the system:

3.1 Longevity

Most acoustic musical instruments stay operable for many years with minimal maintenance. Hence, as soon as a musician acquires a musical instrument, he or she can often assume that he or she can practice, learn repertoire, compose, and even make adjustments to the instrument for many years to come. Hence, the *longevity* of an acoustic musical instrument promotes the development of virtuosity.

As we mentioned before, we believe that one of the challenges to longevity in prior music interaction platforms was

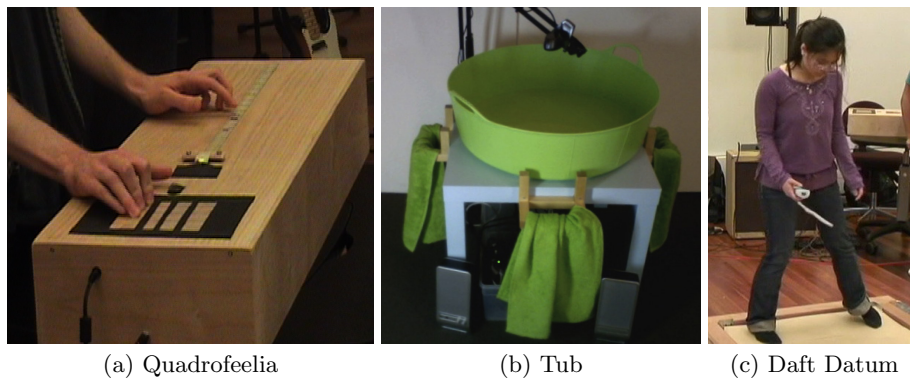


Figure 4: Some student projects made with Satellite CCRMA

that they relied on the use of a non-dedicated computer to stay within a reasonable price point.

3.2 Independence/Autonomy

One of our objectives in our design was to have the Satellite CCRMA platform be stand-alone. To this end, we set up the operating system so that it could boot reliably without a keyboard, mouse or monitor. We strove to minimize the number of wires that needed to be connected to the system. We particularly focused on eliminating the need for the specialized cables that Beagle Board commonly use. Each additional cable or connector or AC adapter that had to be connected to the system at design or performance time was viewed as a liability, for the misplacement of any piece of the system is an obsolescence risk. Configured with battery power and lithium-ion powered speakers, it is possible to sense, synthesize and generate sound from just the Satellite itself.

We hope that the Satellite CCRMA platform will promote the development of long-lived projects by promoting independence of external systems. This is why the platform incorporates both a microcontroller for sensing and a processor for synthesizing sound. Although it is possible to connect Satellite CCRMA to the internet over either an Ethernet connection or wirelessly, we recommend that designers consider permanently disconnecting final projects from the Internet so that they are as independent as is practical.

3.3 Native Floating Point Computation

These projects require synthesis algorithms to synthesize the sound, but these can be computationally demanding. Furthermore, although it is possible to implement most sound synthesis algorithms using integer computations or fixed-point computations, this process can be challenging and time consuming. Indeed, David Zicarelli once quipped that he had heard that a programmer “could either write fixed-point code or stay married.” Such difficulties could distract students from design goals, forcing them to focus on specific engineering problems. Finally, many open-source libraries for synthesizing sound, such as Pure Data Extended, require floating-point computations. For this reason, it was important that the Satellite CCRMA platform was capable of *natively* carrying out *floating-point computations*, as supported by the OMAP3530 processor.

3.4 Compactness

To promote flexibility and integration into other objects, the Satellite CCRMA is *compact*. The Beagle Board itself has a 3”x3.5” footprint, and the Satellite CCRMA board fits into a 5.25”x7”x2” envelope, even with a generic breadboard for

electronics onboard. This is small and light enough that it easily could be attached to a violin. Small systems are easier to carry around, which meant that students were more likely to take their projects home to work on. The upcoming beta release of Satellite CCRMA will be even more compact as it is based on the Beagle Board xM with on-board Ethernet and USB support, so the GWC Technology HE2440 hub with Ethernet as shown beneath the Beagle Board in Figure 1 will not longer be required.

Only one of the projects in our course demanded a form factor too small for the Satellite CCRMA. In that project, the students made three tennis-ball sized balls that held Arduinos, a PIC-based sound synthesizer chip, a speaker and a battery. Looking forward, it would be an interesting challenge to see what the minimum size of the Satellite CCRMA platform could be while still using standard connectors.

3.5 Reconfigurability and Extensibility

The open source and open hardware nature of Satellite CCRMA promotes *reconfigurability* and *extensibility*, which we found to be crucial to supporting the wide range of project ideas that our students had. For us instructors, the fact that Linux drivers for Ubuntu were already available for a large number of external USB devices made it possible, for example, to quickly set up the Ethernet/USB hub and the Arduino for our students. For our students, it meant that they could add new hardware (such as bluetooth dongles, web cameras, and multi-channel audio devices) and new software (such as emacs, alpine, cwiid, jwm) with relative ease.

The incorporation of the Arduino Nano and breadboard was also an important aspect to the Satellite CCRMA platform. To some degree, the Arduino was extraneous, because the Beagle Board’s expansion port includes multiple serial interfaces and general-purpose I/O lines. However, we felt that it was very useful to be able to leverage our previous Arduino-based curriculum to help our students think about how to design interfaces and controls that were appropriate to the use, context, expression and metaphors they had selected. In the future, we will examine whether the extensibility offered by the Arduino over the native Beagle Board I/O is worth the foot print it takes up.

3.6 Community support

We previously mentioned how leveraging open source hardware and software made the Satellite CCRMA platform easy to reconfigure and extend. The open-source platform also makes it easier to develop a system of community support. Beginning users can leverage examples so that they can get the platform up and running, while advanced users can find out how to tweak the system so that they have more con-

trol over the projects that they develop. For example, the source code for any portion of the software can be obtained, modified, recompiled, and then loaded onto an SD card for use. Similarly, the schematics and layouts for all of the core parts are available, enabling users to make custom boards based on the current state of the platform to meet any special needs.

Inspired by the community-based support and development efforts of platforms such as Processing [13] and Arduino [9], we have created a newsgroup aimed especially at artists to help address questions about the platform: <http://groups.google.com/group/satelliteccrma>. To help bootstrap the learning process, we provided our students with working code examples; these are available to other instructors, students or developers on the Satellite CCRMA site.

4. RELATED WORK

4.1 Prior platforms

In the past, our course has been taught with platforms based on the Basic Stamp [17], the AVRmini [20], and the Arduino [9]. Other past and current microcontroller-based platforms used for creating musical controllers include I-CubeX's Digitizer[11], STEIM's SensorLab and junXionboard, Microchip PIC-based Create USB interface [12], and CNMAT's uosc. All of these systems are meant to be interfaced with a computer, which receives data from the microcontroller (via serial, openSoundControl [21], MIDI formats, for example) to synthesize sounds in software. While these systems all do an admirable job of translating physical phenomena from the real world to signals which can be translated to sound, they require the presence of a laptop or desktop computer to perform real-time synthesis of sound.

4.2 Similar platforms

Far fewer platforms are designed to replace the laptop or desktop computer in the physical interaction architecture. One notable platform is the Audiopint [10] platform. The Audiopint, a 17cm x 17cm "mini-itx" VIA Epia EN1500 motherboard running Ubuntu Linux housed in a Pelican case, functions as a low-cost and flexible audio effects processor and synthesizer. It uses Pure data [16] to synthesize sound. Another platform is the Gluiph [7], which features Pure data running on a complex programmable logic device (CPLD).

Satellite CCRMA builds on the ideas embodied by the Audiopint and Gluiph. The choice of the Texas Instruments' OMAP-based Beagle Board provides Satellite CCRMA with a small footprint (the Beagle Board itself is 3" wide x3.125" long x0.625" tall, and the platform footprint is 7"

x 5.5"x2"), relatively low power consumption, and the potential for broader platform support from the wider OMAP and Beagle Board development community. The cost of Satellite CCRMA, critical to the cost-sensitive artists and musician community, is lower than either platform (\$125 at time of submission). Also, the use of a standard Linux operating system makes it easy to extend the system through opensource software and the use of USB-based computer peripherals and drivers available on-line. Finally, the incorporation of the bread Board and Arduino helps to orient the use of the system towards the development of novel musical controllers and sound installations.

4.3 Alternative platforms

One viable alternative to the Satellite CCRMA system is the use of low-cost netbooks in place of laptops. At the time of this paper's submission, low-end netbooks capable of running Pd cost \$300-\$400 USD. Although little has been published about the use of netbooks as musical system platforms, we believe it is only a matter of time before people start to dedicate inexpensive netbooks to their instruments. The catch is that, as small as netbooks are, the minimum size and layout of netbooks is dictated by typical laptop usage. Even though the cost of integrating features such as a screen, a lithium-ion battery and wifi and bluetooth capability to the Satellite CCRMA system would likely cost more than a similarly equipped netbook, the cost and difficulty of hacking and reconfiguring a netbook into a format that can be easily incorporated into an instrument would likely neutralize any potential savings.

Another alternative to using the laptop or desktop computer as a music platform is to use mobile phones or iPods. In earlier mobile music platforms, the phones were just used for their autonomous sensing capabilities, and an external PC was used to generate the sound, but more recently, sound synthesis on-the-phone has been pioneered using the Synthesis ToolKit on the Symbian OS [5], and the MoMu on the iPhone OS [2]. The advent of the Android OS phones and tablets is likely to greatly decrease the cost of compact autonomous systems with the computational ability to generate sound, so that these devices will be cheap enough to dedicate to specific instrument or installation designs. However, it is fairly difficult as yet to interface external sensors with these systems, which greatly limits the range of musical control and interaction capabilities of platforms based on these systems. Most strategies involve using Bluetooth wireless, which introduces some latency due to the Bluetooth protocol.

5. PROJECT HIGHLIGHTS

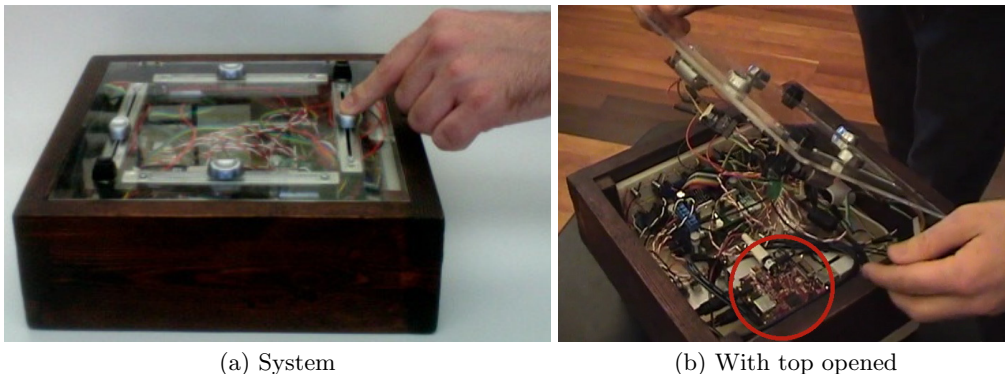


Figure 5: SoundFlinger

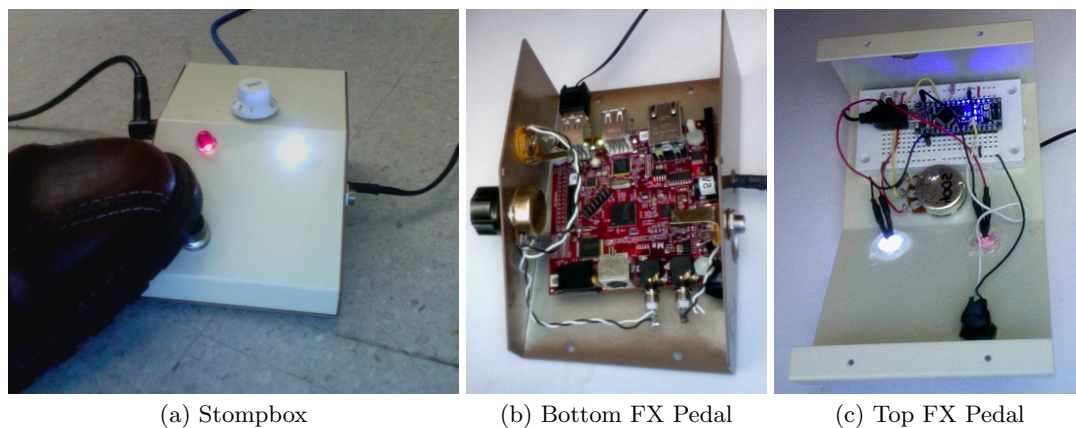


Figure 6: Example Stompbox featuring Beagle Board-xM

To illustrate the range of what is can easily be accomplished with Satellite CCRMA as a platform, we present these examples of what our students created for the class projects. Many of the students were new to the Arduino when they started the class, and all of the students were new to the Beagle Board and embedded Linux.

5.1 Independent Musical Instrument

Quadrofeelia, by Jiffer Harriman, Mike Repper, Linden Melvin, and Locky Casey, (see Figure 4(a)) shows that Satellite CCRMA can be employed to construct an *independent* musical instrument that does not require connections to any external computers. A musician plugs the 5V power supply into a wall outlet, and then the 1/4" audio output jack produces an output signal that can be connected to a guitar amplifier or public address system. As the musician manipulates the sensors embedded in the instrument, output sound is produced in response. Quadrofeelia bears resemblance to a slide guitar, but with more abilities for retuning the chord tunings and inversions due to the buttons manipulated by the right hand.

5.2 Independent Dance Interface

Standard sensing devices typically employed for gaming are incorporated into Daft Datum, a musical instrument played by the feet, by Rosie Cima, Ravi Kondapalli, and Ben-Zhen Sung. A dance pad is hidden beneath the performer's feet and sends data to Pure Data over USB using the `hid` object. To discover the vendor and product IDs for the dance pad, the `dmesg` command can be executed in Linux directly after plugging in the dance pad. No further configuration is necessary. To change the sound synthesis mode, the performer presses buttons on the Wiimote as shown in Figure 4(c). The Wiimote communicates with Pure Data over a remarkably small Wifi USB dongle that can be setup as described on the following Wiki: [https://ccrma.stanford.edu/wiki/Making_a_Wii_remote_talk_to_Pure_Data_\(PD\)](https://ccrma.stanford.edu/wiki/Making_a_Wii_remote_talk_to_Pure_Data_(PD))

5.3 Video-based Interactive Installation

The Tüb, by Marc Evans, Björn Erlach, and Mike Wilson, demonstrates the video capabilities of Satellite CCRMA. The KWC-1301 USB Webcam is suspended above a tub of water (see Figure 4(b), top center) and transmits images to Pure Data running on the Beagle Board stored underneath the tub. Objects in Pure Data's Graphics Environment for Multimedia (GEM) generate sound samples by scanning along circles in the images. If the water is still, then there is almost no sound, but waves in the water create buzzing sounds, whose timbre evolves with the wave motion. Instal-

lation visitors can induce waves in the water using a variety of techniques: poking the water with their hands, pulling the edge of the tub with their hands, poking the water with tubes, blowing water through tubes, etc.

5.4 Multichannel Audio in a Collaborative Installation Piece

The Sound Flinger, by Chris Carlson, Hunter McCurry, and Eli Marschner, demonstrates that Satellite CCRMA is compatible with an external sound interface. The device, as shown in Figure 5, allows users to play back snippets of live recorded sounds through four output audio channels provided by the SIIG IC-710112 USB Soundwave 7.1 Digital audio adapter, which is placed underneath the Beagle Board circled in red in Figure 5(b).

5.5 Audio Effects Stompboxes

Because Satellite CCRMA incorporates audio codecs and native floating-point computation, implementing digital audio effects is relatively straight-forward even for novices. The design history of audio effects incorporates a rich past of audio effects controlled primarily by knobs, switches, and even sometimes pedals. Figures 6 and 6(b) (left) show an example *stompbox*, which incorporates a footswitch and two extra large LEDs. The stompbox demonstrates that Satellite CCRMA can also be employed for teaching audio digital signal processing (DSP) in an embedded systems context. Appropriate languages for DSP include Faust, C++, Pure Data, Chuck, etc.

6. SATELLITE CCRMA WEBSITE

The main website for the project can be found below, which provides links to videos of project presentations, the Satellite CCRMA newsgroup, Wiki, and SD card images: <https://ccrma.stanford.edu/~eberdahl/Satellite/>

7. FUTURE

Since the beginning of this project, we have already had one major hardware change, from the OMAP 3530-based Beagle Board to the OMAP 3730-based Beagle Board-xM. While we believe the OMAP-based Beagle Board family to be a promising platform for the design of new interfaces for musical expression, the core aim of the Satellite CCRMA project is to develop standalone computational capabilities for musical interfaces in an open-source setting. As such, we are tracking related platforms such as the Hawk Board, Panda Board, Crane Board, or GumStix, in case any of these should become more viable for our project goals. In

any case, it is our intent to maintain a similar set up of core platform software across hardware changes, and also to make migration from one platform to the next graceful for users.

We have found that the existing platform is stable and easy enough for NIME students to use, reconfigure and extend within the confines of a single school term. In the future, we hope to see if the Satellite platform helps promising projects grow beyond their short-term novelty "stunt" phase to become more mature musical instruments. In particular, we hope to see whether inventors develop more expertise with their designed installations and instruments, perhaps by making more refinements, developing greater expertise in playing their instrument, or by making scores for the systems' unique capabilities. We also intend to perform more detailed analysis to see if the Satellite CCRMA system helps students to more fully understand full-scale system design.

8. CONCLUSIONS

In conclusion, we found that the introduction of the Satellite CCRMA platform has provided our students with a more full-featured, robust and flexible system for lab exercises and musical interaction design projects. This platform extends the portability, scalability and support community of our previous microcontroller-based system to the microprocessor as well. We have found the platform is useful for instrument-based and installation-based systems, and we are actively promoting the platform to see if it is useful in a wide variety of other applications as well.

9. ACKNOWLEDGMENTS

In addition to Chris Chafe, Fernando Lopez-Lezcano, Bill Verplank, Max Mathews, Perry Cook, Julius Smith III, Michael Gurevich, Carr Wilkerson, and the open-source community, we would like to thank our students who helped us test the initial release of Satellite CCRMA: Chris Carlson, Locky Casey, Roseann Cima, Björn Erlach, Marc Evans, Francesco Georg, Jiffer Harriman, Ravi Kondapalli, Eli Marschner, Hunter McCurry, Linden Melvin, Michael Repper, Mike Rontondo, Spencer Salazar, Ben-Zhen Sung, and Mike Wilson. This research was made possible by a generous equipment donation from Texas Instruments.

10. REFERENCES

- [1] A. Bowen. Soundstone: a 3-d wireless music controller. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 268–269. National University of Singapore, 2005.
- [2] N. J. Bryan, J. Herrera, J. Oh, and G. Wang. MoMu: a mobile music toolkit. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME), Sydney, Australia*, 2010.
- [3] J. Carlile and B. Hartmann. OROBORO: a collaborative controller with interpersonal haptic feedback. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 250–251. National University of Singapore, 2005.
- [4] L. Dahl, N. Whetsell, and J. V. Stoecker. The WaveSaw: a flexible instrument for direct timbral manipulation. In *Proceedings of the 7th international conference on New interfaces for musical expression*, pages 270–272. ACM, 2007.
- [5] G. Essl and M. Rohs. Mobile stk for symbian os. In *Proc. International Computer Music Conference*, pages 278–281. Citeseer, 2006.
- [6] M. Gao and C. Hanson. LUMI: live performance paradigms utilizing software integrated touch screen and pressure sensitive button matrix. In *Proceedings of the 2009 Conference on New Interfaces for Musical Expression*, 2007.
- [7] S. Kartadinata. The gluiph: a nucleus for integrated instruments. In *Proceedings of the 2003 conference on New Interfaces for Musical Expression*, page 180. Citeseer, 2003.
- [8] R. Lugo and D. Jack. Beat boxing: expressive control for electronic music performance and musical applications. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 246–247. National University of Singapore, 2005.
- [9] D. Mellis, M. Banzi, D. Cuartielles, and T. Igoe. Arduino: An open electronic prototyping platform. In *Proc. CHI, 2007*, 2007.
- [10] D. Merrill, B. Vigoda, and D. Bouchard. Audiopint: A robust Open-Source hardware platform for musical invention. *Pd Convention*, 2007.
- [11] A. Mulder. The I-Cube system: moving towards sensor technology for artists. In *Proc. of the Sixth Symposium on Electronic Arts (ISEA 95)*, 1995.
- [12] D. Overholt. Musical interaction design with the create usb interface. In *Proc. ICMC2006, International Computer Music Conference, New Orleans*, 2006.
- [13] C. Reas and B. Fry. Processing: a learning environment for creating interactive web graphics. In *ACM SIGGRAPH 2003 Web Graphics*, page 1. ACM, 2003.
- [14] D. Schlessinger and J. O. Smith. The kalichord: A physically modeled Electro-Acoustic plucked string instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 98–101, 2009.
- [15] H. Shiraiwa, R. Segnini, and V. Woo. Sound kitchen: designing a chemically controlled musical performance. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 83–86. National University of Singapore, 2003.
- [16] H. C. Steiner. Building your own instrument with pd. In *Proceedings of the 1st International Pd Conference, Graz, Austria*. Citeseer, 2005.
- [17] B. Verplank, C. Sapp, and M. Mathews. A course on controllers. In *Proceedings of the 2001 conference on New interfaces for musical expression*, pages 1–4. National University of Singapore, 2001.
- [18] C. Wilkerson, C. Ng, and S. Serafin. The mutha rubboard controller. In *Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–4. National University of Singapore, 2002.
- [19] S. Wilson, M. Gurevich, B. Verplank, and P. Stang. Microcontrollers in music HCI instruction: reflections on our switch to the atmel AVR platform. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 24–29. Citeseer, 2003.
- [20] S. Wilson, M. Gurevich, B. Verplank, and P. Stang. Microcontrollers in music HCI instruction: reflections on our switch to the atmel AVR platform. In *Proceedings of the 2003 conference on New interfaces for musical expression*, pages 24–29. Citeseer, 2003.
- [21] M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the 1997 International Computer Music Conference*, pages 101–104, 1997.