A Low-level Embedded Service Architecture for Rapid DIY Design of Real-time Musical Instruments

Andrew Schmeder, Adrian Freed
Center for New Music and Audio Technologies
Department of Music, UC Berkeley
1750 Arch Street, Berkeley CA 94720, USA
{andy,adrian}@cnmat.berkeley.edu

Abstract

An on-the-fly reconfigurable low-level embedded service architecture is presented as a means to improve scalability, improve conceptual comprehensibility, reduce human error and reduce development time when designing new sensor-based electronic musical instruments with real-time responsiveness. The implementation of the concept in a project called *micro-OSC* is described. Other sensor interfacing products are evaluated in the context of DIY prototyping of musical instruments. The capabilities of the micro-OSC platform are demonstrated through a set of examples including resistive sensing, mixed digital-analog systems, many-channel sensor interfaces and time-based measurement methods.

Keywords: real-time musical interface, DIY design, embedded web services, rapid prototyping, reconfigurable firmware

1. Introduction

The goal for enabling of rapid DIY (do-it-yourself) design is to make tools for independent/non-industrial designers that are easy to use yet also flexible enough to encourage experimentation and powerful enough to produce non-trivial results. For non-experts, these tools provide a simplified and foolproof design process. For experts they enable rapid access to complex designs. This goal is not trivial because several objectives must be simultaneously satisfied:

- Affordability: Rapid prototyping is a multi-iteration design process–low cost is essential to make this possible.
- Process Simplification: Includes elimination of expensive and specialized tools, reduction of user errors, and faster development time.

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.
NIME09, June 3-6, 2009, Pittsburgh, PA
Copyright remains with the author(s).

- Conceptual Abstraction: The target users (DIY designers) work in an interdisciplinary space. There is an opportunity to create a new and powerful technical language to describe design parameters that function at an appropriate level of abstraction.
- Scalability: Scalability in the sense that simple programs/interfaces can be easily repurposed into more complex examples. Repurposing is the main activity of "hacking" and an essential practice for DIY design.
- Signal Quality for Musical Gestures: The system must support high-bandwidth many-channel sensor systems with real-time performance characteristics sufficient for the faithful transduction of musical gesture signals. We expect electronic musical instruments to retain the same sense of visceral interactivity as traditional acoustic instruments, a property that differentiates musical instruments from other types of interfaces [1].

2. Context: State of the art

In this section we examine other approaches to sensor interfacing according to the above criteria.

2.1. Industry Solutions

Industrial design process for microcontrollers is focused on objectives such as affordability in bulk production, reliability, quality assurance and security. Watchdog timers, automated correctness provers, and real-time operating systems are some of the technologies developed for these needs. More recently, conceptual abstraction has come into play as researchers are exploring service oriented architectures that bring the web-services technology stack to the microprocessor [2]. However, the industry has remained indifferent to the specific needs of music-quality gesture transduction and hackability.

2.2. Hobbyist Tools

Reduced cost of small-batch electronics manufacturing combined with increasing microcontroller capability (i.e., Moore's Law) and generally increasing technical literacy has created an explosion in popularity of sensor interfacing platforms in the DIY community.

2.2.1. *Arduino/Wiring*

The Arduino/Wiring platform [3] is a low-cost sensor interface that has made great progress in the area of process simplification by the inclusion of an integrated development environment that features simple controls to save and run the user program on the target platform. The Wiring programming language and libraries make no significant effort to provide conceptual abstraction. Programs are essentially imperative C-style code, and difficult to repurpose. Temporal controls are primitive, and many examples use strategies such as busy-wait delays that introduce temporal errors. No effort is made to abstract over hardware-specific details such as analog-to-digital converter (ADC) bit depth. There is no effort to establish a notion of signal quality in the communications link.

2.2.2. *Arduino/Firmata*

The Firmata project [4] is a reconfigurable service oriented firmware for the Arduino. Firmata uses a protocol based on MIDI citing the need for a highly compact protocol due to a slow serial link speed of 0.115Mbit/s (USB full-speed is capable of 12Mbit/s). We anticipate that this shortcoming is temporary in nature and it is unfortunate that design decisions hinge upon it. Firmata has no temporal semantics or consideration of signal quality needed for musical interfaces.

2.2.3. *The Make Controller*

The Make Controller [5] has a service oriented architecture based on OSC messaging, enabling scalablity of programs. The implementation is based around a real-time operating system, providing good timing behavior independent of program content, although it does not expose temporal information directly to the user. Conceptual abstraction is provided for several sensing/actuating paradigms, however this could be improved, for example by using floating point numbers instead of hardware-dependent integers. Affordability could be improved as the physical platform itself is quite large. No effort made to assure signal quality is sufficient for use in musical instruments.

2.2.4. *Electrotap Teabox*

The Electrotap Teabox [6] is an important example as it recognizes fully the importance of signal quality–by encoding the sensor data into a digital audio stream, minimal latency and jitter is assured. Unfortunately this comes at a great price, compromising affordability. Furthermore, it is primarily a voltage-sensing input interface, excluding interesting interfaces that use more advanced methods of measurement possible with microcontrollers (e.g., as in [7], [8] and further examples in this paper).

3. Design of the micro-OSC Platform

micro-OSC [9] is a low-level service oriented architecture (a low-level, node-level middleware according to [10]) that leverages the Open Sound Control (OSC) [11] protocol to achieve program scalability and temporal semantics. This style of interaction results in a layered development model where expert engineers create the lowest-level code and designers use the OSC service-oriented middleware to configure the device behavior at runtime (**Figure 1**).

micro-OSC addresses the issue of signal quality for musical gestures through use of high sampling rates and the OSC timetag feature to guarantee that information streams are communicated without jitter-induced noise [12].

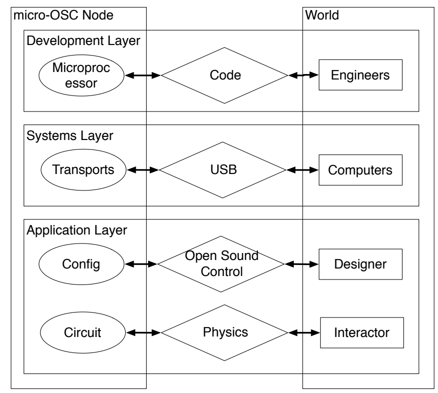

Figure 1. Entity-relation diagram of interaction layers

3.1. A simple program

The following program for the Wiring-Arduino platform reads an analog input pin at a rate of approximately 1Hz and reports it to a host computer.

```
int inputPin = 12;
void setup() {
  pinMode(ledPin, INPUT);
}
void loop() {
  Serial.println("%d",
    analogRead(inputPin));
  delay(1000);
}
```

The equivalent program for micro-OSC is a pair of messages (the reporting is automatic, and there is no error accumulation in the timing)

```
→ /an/12/dir input
→ /an/12/samplerate 1.0 Hz
```

3.2. Conceptual Abstraction

To design new interfaces, it is obviously necessary to expose some operational details. Not all details are equally useful–for example, the bit depth of the ADC is rarely of importance to the designer, however, enabling the configuration of optimal analog-to-digital conversions based on impedance of a circuit, while a more complex topic, is sufficiently useful to merit inclusion. We therefore put forth the following concepts as a set of fundamentals necessary to learn when to designing a new interface with micro-OSC:

- Basic electronics: at the ”network flow” level: sources and sinks, water analogies.

- Temporal semantics: *When* is as important as *what*. Concurrent events, sequential events.

- Methods of measurement: Measurement through ratios between a fixed and a variable quantity (e.g., of voltages, resistances, etc). Measurement of time.

3.2.1. Temporal Semantics enabled by OSC

The use of OSC messaging in micro-OSC includes full support for bundles and bundle timestamps. The device keeps track of time using an on-chip hardware timer that can be synchronized to global time [12]. Timestamps enable micro-OSC to schedule received messages to execute at precise moments in the future, enabling coordination across multiple heterogenous systems. The temporal concurrency of an OSC bundle ([11] for details) enables micro-OSC to semantically describe concurrent hardware events such as the port-write (a state change that applies to multiple pins simultaneously). Furthermore, micro-OSC uses timestamps to annotate the time-of-occurance for internal events such as input sample acquisition and interrupt-on-change triggers. This enables transparent support for high resolution measurements of time, for example as used in the pulse-echo protocols for an ultrasonic range-finder, for measurement of discharge time of a capacitor, and for removal of transport induced jitter-noise.

3.2.2. Measurement with Ratiometric Resistance

In nearly all sensing methodologies, measurements are obtained from ratios between an unknown and a fixed quantity. For example, an analog-to-digital converter measures the ratio between an unknown voltage to some reference by timing the relative speed of charging a capacitor. Measurement of of an unknown variable resistance using the pull-up or pull-down circuit (**Figure 2**) is method used by many position and pressure sensors. In our experience, novice designers ignore the resulting non-linear voltage at the measurement point. micro-OSC presents a simple solution, by enabling a ratiometric resistance sensing mode, the sensor data is automatically linearized. The measured value is reported as a floating point number expressing the resistance of the unknown as a ratio (possibly greater than one) of the reference.

4. Design Examples

This section illustrates the capabilities of the architecture in the context of rapid design. The following projects are each implemented from concept to completion over the span of several hours. Manual labor is the dominating factor in the designs, not software.

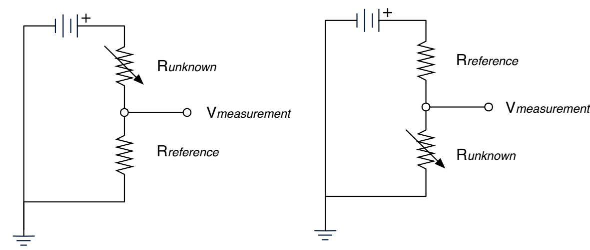

Figure 2. Voltage divider; Right: pull-down, Left: pull-up

4.1. Simple Examples

We have created many new interfaces by using resistive sensing with the pull-up circuit in combination with new malleable piezo-resistive materials [13] (**Figure 3**). Some designs include other components such as linear voltage sensors (e.g., accelerometers, gyros), or LED lighting control for visual feedback.

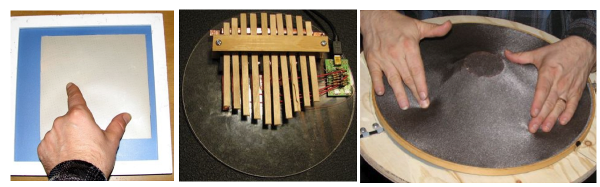

Figure 3. Dual Touch, Kalimba, Tablo

4.2. Scaling to Many Channels

A multiplexing feature of micro-OSC configures microcontroller pins to sequentially step through multiple states. This capability generalizes to multiplexing, shift registers, sequential scanning and switch matrices.

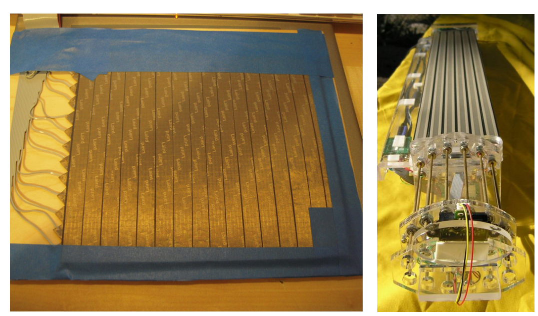

Figure 4. 16x13 scanning multitouch, 12-stringless cello

4.2.1. Scanning multitouch array

A multitouch array is constructed with piezo-resistive fabric (**Figure 4**). The top surface is covered with conductive tape traveling in the vertical direction, the bottom surface with tape in the horizontal. Op-amp virtual grounding prevents cross-talk between channels [14]. A 16-way driveline sequencer is configured with the multiplexing driver where the inactive state is floating and the active state is driving high. Each of 16 steps activates one vertical strip and subsequently reads the horizontal strips to obtain 13 pressure measurements for a total of 208 points.

4.2.2. 12-Stringless Cello

With the objective to create a 12-stringless cello controller [15] (**Figure 4**), we first created a one-channel prototype with a pressure/position sensing strip and bow velocity+pressure sensor. This prototype was then scaled up by a factor of 12 by adding a multiplex chip to the circuit, enabling the multiplex driver and increasing the sample rate.

4.3. Bidirectional LED

A concept music-player interface was constructed where a linear array of LEDs functions as both a volume control and a volume indicator (**Figure 5**). Both pins of an LED are connected to a microcontroller tristate pin, enabling four possible states: output on, output off, reverse-biased, discharging [16]. Cycling through the states rapidly (using the multiplex driver) enables the LED to function as a light sensor and a light output. A charge-rate sensing mode is used to estimate the RC time-constant of the incident-light-dependent LED discharge phase. Simultaneously, the embedded micro-OSC controller handles digital serial communication to drive the LCD display.

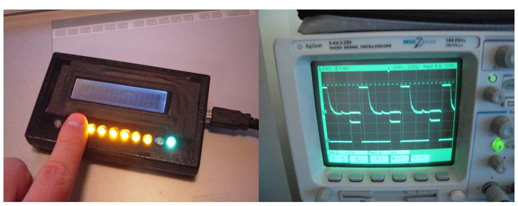

Figure 5. LED controller+indicator concept

5. Conclusion

We have presented a set of goals for the implementation of a sensor interfacing platform with features that accomodate a rapid DIY design process and temporal qualities essential to musical gesture sensing. Through the use of appropriate conceptual abstraction and an OSC messaging service, designs can be expressed in simple statements and easily modified. Multiple diverse examples show that this system is capable of generalizing to many different types of sensors and methods of measurement that go beyond simple voltage sensing and without requiring users to learn microprogramming.

5.1. Future work

The functionality described in this work is only a beginning for a rich area of development. For example, actuation/control methods are another area where conceptual abstraction can be applied in a similar way.

Finally, while the current implementation of micro-OSC is based on a Microchip PIC processor, the choice of hardware is not dogmatic. The future of microprocessor technology is continuously evolving. Increasing computational power, highly parallel architectures and fast networking options will make semantically-rich protocols more common on embedded devices. We anticipate that these developments will make conceptual abstraction and temporal semantics ever more important.

References

[1] D. Wessel and M. Wright, "Problems and Prospects for Intimate Musical Control of Computers," *Computer Music Journal*, vol. 26, pp. 11–22, 2002.

[2] J. Helander, "Deeply embedded XML communication: towards an interoperable and seamless world," in *EMSOFT '05: Proceedings of the 5th ACM international conference on Embedded software*, (New York, NY, USA), pp. 62–67, ACM, 2005.

[3] N. Zambetti and et al, "Arduino." `http://arduino.cc/`, 2008.

[4] Various, "Firmata." `http://firmata.org`.

[5] M. T. Inc., "Make Controller Kit." `http://www.makingthings.com/`, 2007.

[6] Electrotap, "Teabox." `http://shop.electrotap.com/products/teabox`.

[7] A. Freed, R. Avizienis, and M. Wright, "Beyond 0-5V: Expanding Sensor Integration Architectures," in *International Conference on New Interfaces for Musical Expression*, (Paris, France), pp. 97–100, 2006.

[8] A. Freed, "Novel and Forgotten Current-steering Techniques for Resistive Multitouch, Duotouch, and Polytouch Position Sensing with Pressure," in *NIME 2009*, 2009 (submitted).

[9] A. Schmeder and A. Freed, "uOSC: The Open Sound Control Reference Platform for Embedded Devices," in *NIME*, (Genova, Italy), 05/06/2008 2008.

[10] R. Sugihara and R. K. Gupta, "Programming models for sensor networks: A survey," *ACM Trans. Sen. Netw.*, vol. 4, no. 2, pp. 1–29, 2008.

[11] M. Wright, "Open Sound Control 1.0 Specification." `http://opensoundcontrol.org/spec-1_0`, 2002.

[12] A. Schmeder and A. Freed, "Implementation and Applications of Open Sound Control Timestamps," in *ICMC*, (Belfast, Ireland), pp. 655–658, ICMA, 25/08/2008 2008.

[13] A. Freed, "Application of new fiber and malleable materials for agile development of augmented instruments and controllers," in *NIME 2008*, 2008. CNMAT.

[14] M. Shimojo, A. Namiki, M. Ishikawa, R. Makino, and K. Mabuchi, "A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method," *Sensors Journal, IEEE*, vol. 4, pp. 589–596, Oct. 2004.

[15] A. Freed, F.-M. Uitti, A. Schmeder, and J. MacCallum, "Introducing a stringless cello controller for two bow performance." `http://cnmat.berkeley.edu/event/2008/11/11/introducing_stringless_cello_controller_two_bow_performance`, 2008.

[16] P. Dietz, W. Yerazunis, and D. Leigh, "Very Low-Cost Sensing and Communication Using Bi-directional LEDs," in *UbiComp 2003*, (Seattle, Washington), 12/10/2003 2003.