AN EMBEDDED HARDWARE PLATFORM FOR FUNGIBLE INTERFACES

Avrum Hollinger, Joseph Thibodeau, Marcelo M. Wanderley

IDMIL, CIRMMT
Music Technology Area, Dept. of Music Research
Schulich School of Music, Montreal, Canada

ABSTRACT

A modular and reconfigurable embedded computer system for designing and prototyping electronic musical interfaces is presented. The system is based on an ARM7 microcontroller that configures peripheral devices, collates input signals, and processes/synthesizes audio. Peripheral devices, each equipped with its own processor, are fully reconfigurable enabling processing and conversion of both analog and digital signals. Communication between the ARM and the peripheral devices is accomplished using the I2C protocol, allowing for multi-master-slave operation. Although the system operates without the use of a personal computer, a serial interface was implemented to facilitate prototyping with a computer terminal running a text or graphics-based user interface. Applications for this platform include a standalone electronic musical instrument and an augmented acoustic trumpet.

1. INTRODUCTION

A fungible interface is not merely the portmanteau of a fun and tangible interface, but an interface with the ability to exchange one of its components with another. We present a modular approach to rapid design of embedded human-computer music systems whereby a central module (hub) manages higher-level computation and communicates with each of the peripheral modules (nodes).

The manifold motivations for this project evolved around developing a platform for standalone electronic musical instruments, augmented acoustic instruments, and electronically-controlled audio effects units. These applications, however, are but a small subset of what is possible and could easily be extended (but not limited) to modular robotic and haptic interface systems, multimedia controllers, dance, or tech-art installations.

The advantages of an embedded system stem from its scope of operation, size, and cost. It is purpose-built and as such need not juggle disparate goals, which in addition to increasing the efficiency of the system, protects it from a system-wide crash due to collisions with other processes. An embedded system is also built only as large as is necessary, thus economizing resources and improving the portability and form factor of the end result. In terms of musical instruments, an embedded interface and synthesis system untethers the digital musical instrument from a personal computer. Such an instrument is a self-contained whole with a voice and identity of its own.

2. BACKGROUND

Signal acquisition using microcontrollers as sensor-to-computer interfaces has become the norm in musical interface design [20]. These devices are relatively inexpensive, easy to start working with, and are readily available with a wide range of signal converters and communication interfaces. When a single microcontroller cannot perform all the requisite operations, developers turn toward an approach using a network of microcontrollers [19] [16], DSPs, or a field-programmable gate array (FPGA) [9] [2] [20]. In all of these applications, the synthesis or host component of the bus/network was implemented on a PC.

While FPGAs have the capacity to implement all of the functions of a microcontroller, the hardware description language (e.g. VHDL, Verilog) and system setup are major obstacles for rapidly prototyping a variety of end-user applications. FPGAs are quite capable of out-performing a microcontroller in the realm of synthesis [14] [17], and although fully-encapsulated systems that include data acquisition, mapping, and synthesis are possible, there are few designs that integrate all of these functions [3] [8] and most devices still require a host PC.

3. HARDWARE PLATFORM AND CONNECTIVITY

At the core of the hardware platform is an ARM7-based microcontroller. It is responsible for assigning I2C bus addresses to nodes, initiating data transfers, collating input signals from nodes, performing mapping and audio synthesis, and communicating with a computer terminal if needed.

Each node is implemented on a Programmable System-on-Chip (PSoC) whose functionality—excepting the communication protocol—differs between applications. Possible functions include input or output devices, analog-to-digital or digital-to-analog converters, rudimentary analog or digital signal processing units, or some combination thereof.

Not only PSoCs but any I2C-capable device (such as an Arduino) can be used as a node, so long as it conforms to the communication protocol described in this paper.

Once connected to the hub, a node is automatically configured and can be controlled remotely based on its defined functionality. Using a bus communication infrastructure saves on space, allowing the nodes to be connected to the hub using only four wires (power, ground, I2C data, I2C clock). The drawback to a bus communication system is that the throughput for each node is reduced as more nodes are added. Therefore, analog sensors that require a higher sampling rate than the I2C protocol can provide must be connected directly to the hub's ADC inputs.

3.1. PSoC

The unique aspect of the PSoC (Cypress Semiconductor Corp.) is that it is more than a standard microcontroller. It combines both analog and digital blocks that are reconfigurable depending on the application. As far as electronic music systems are concerned, the PSoC can accelerate development time by reducing external analog components and allowing for digital control of analog signal paths. The PSoC has been used previously in musical applications [11] [6] [5], but for this project the modular and reconfigurable nature of these devices is germane.

3.2. ARM

The current I2C hub design is based on a 72MHz ARM7-core LPC2468. The main features that were utilized included the I2C port, UART, DAC, and ADC. While the ARM's I2C interface was used as the backbone for signal communication, the UART was used in combination with an FTDI chip to communicate with a PC over USB. The ARM's 10-bit ADC was employed for high-speed sensor and audio analog-to-digital conversion (up to 408kS/s) and the 10-bit DAC was used as the primary audio output with 1μs settling time. For higher-quality audio applications, the LPC2468 can interface with off-chip DAC and ADC ICs using SPI or I2S.

3.3. I2C Communication Protocol

I2C bus addressing is administered by the ARM7 hub and initiated by a PSoC node using a handshaking protocol:

1. node requests I2C address from the hub
2. hub sends next available I2C address to the node
3. node responds with its input and output capabilities
4. hub registers the capabilities of the new node which is now ready to be polled

Once a node has gone through the handshaking process, its output data are available to the hub as sources in the mapping scheme.

4. MAPPING AND SYNTHESIS

In addition to pulling data streams from the I2C bus, the ARM processor performs mapping and synthesis. The mapping structure incorporates a cubic scaling function for each source signal. Each synthesis parameter is calculated as a linear combination of scaled inputs. For development purposes, the scaling and mapping parameters can be updated using a computer terminal by querying the current state and sending control messages over a serial interface. The namespace for these parameters is human readable, allowing the user to communicate with the device in a command-line interface or with a graphical user interface. For example, the mapping can be configured during design-time with the McGill Digital Orchestra Tools Mapper using MAX/MSP [12], then stored to the ARM processor's flash memory for live performance use.

Computed synthesizer inputs are updated at a programmable control rate, while audio output is computed and sent to the DAC with a higher-priority interrupt running at a programmable audio rate. Maximum speeds vary depending on the number of input channels and their resolution as well as the complexity of the synthesis algorithm. Two different physical modeling synthesizers were implemented on the ARM: an excitable cell model and a vocal fold model.

4.1. FitzHugh-Nagumo Excitable Cell

The FitzHugh-Nagumo (FHN) model is a simplification of the Hodgkin-Huxley squid giant axon model based on a Van der Pol oscillator [4]. Using a finite difference approximation, the solution is given by:

$$v_{t+1} = v_t + dt(-w_t - v_t(v_t - a)(v_t - 1) + I) \tag{1}$$

$$w_{t+1} = w_t + dt(e(bv_t - gw_t - d)) \tag{2}$$

where v and w refer to the membrane potential and the recovery variable, respectively. I represents the stimulus current, dt the integration time, and a, b, d, e, and g are used as other synthesis parameters. This model provides a simple non-linear oscillator with coupled control over fundamental frequency, amplitude, and timbre. An extension was implemented which incorporated a two-dimensional grid of excitable cells coupled by the Laplacian heat diffusion equation, giving rise to a more complex synthesizer. An explicit topographical analogy to a planar control surface is accomplished by spatially varying the model's parameters and mapping excitatory input at specific locations to stimulus current at corresponding cells of the grid.

4.2. Ishizaka-Flanagan Vocal Fold Model

A simplified Ishizaka-Flanagan (IF) vocal fold model was implemented based on the work by Avanzini [1]. Although Avanzini put much work toward a simplified one-mass model, the two-mass model was utilized to provide an extension to the physically achievable glottal waveforms. By retaining access to each mass and its associated model parameters independently, and by setting those parameters with significantly different values, mode locking does not occur; enabling the synthesis of a huge variety of timbres not available with the simplified model.

5. CURRENT APPLICATIONS

5.1. GGT: Flexible Silicone Instrument

A prototype interface to control physical modeling synthesizers was constructed by embedding fibre optic sensors into a form cast in flexible silicone rubber. The sensors were fabricated to measure deformation of the interface, including bending, pinching, twisting, and stretching gestures. Each sensor consisted of an LED, fibre optic cable, a photo-diode and transimpedance amplifier. The analog signals were multiplexed and further amplified using a programmable-gain amplifier on the PSoC before being digitized, as shown in Fig. 1. Ambient light compensation was performed on the PSoC. The system was alternately tested with both FHN and IF physical models. Mappings were configured using a PC and the command-line interface. The 2-D FHN synthesizer was initially tested with a large 40-by-40 grid, but due to on-chip memory constraints and processor resources this was reduced to a small 5-by-4 grid. Results were underwhelming as the soul of the 2-D synthesis engine relies on the variety of spatially-distributed synthesis parameters as noted in 4.1. The IF synthesizer was implemented with much better results, allowing continuous and nuanced control over pitch, amplitude, and timbre. Due to the complexity of the synthesis algorithm, however, the sample rate was running at an unimpressive (however usable) 10kHz. By using off-chip RAM and a higher-speed ARM processor, these synthesizers will be further explored as part of self-contained electronic instruments using physical modeling synthesis.

5.2. Symbiote: Augmented Trumpet

The Symbiote is a modular trumpet augmentation design platform. Most augmented trumpets in the literature [13] [7] [15] [10] [9] are custom-built projects with a static hardware configuration aimed at satisfying the needs of a specific performer. Whether or not their design is publicly available (as with the Electrumpet [10]), they require significant technical skill to build and maintain. Using the Symbiote however, the necessary level of technical knowledge for development is reduced significantly. Inexperienced designers can use pre-made nodes as swappable building blocks while experienced designers can still custom-make their own nodes if necessary.

Currently a pitch tracking node has been implemented using a zero-crossing pitch estimation algorithm informed by the state of the trumpet valves (similar to [13] and [18]), and initial results are promising. The output of this node will be used to control synthesizers on the hub. Analog sensor interface nodes have also been implemented and will be used to experiment with different interface configurations for the control of delay-line effects. Other nodes currently in development include a keypad controller for live looping and a visual feedback node to display state information. Two example configurations are illustrated in Fig.2.

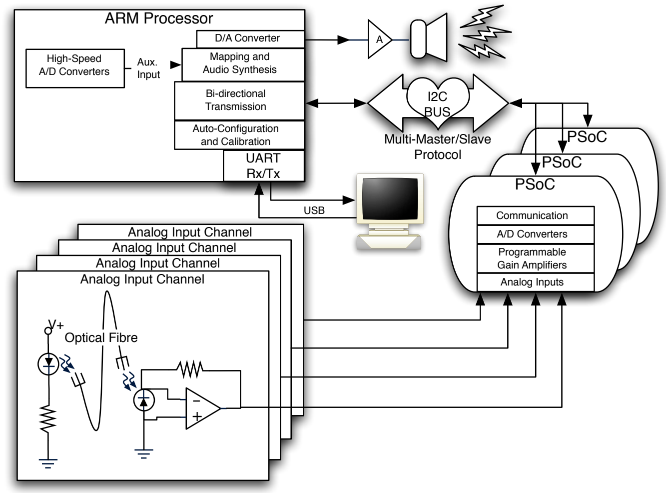

Figure 1. *Example system configuration showing fibre optic sensor-based interface.*

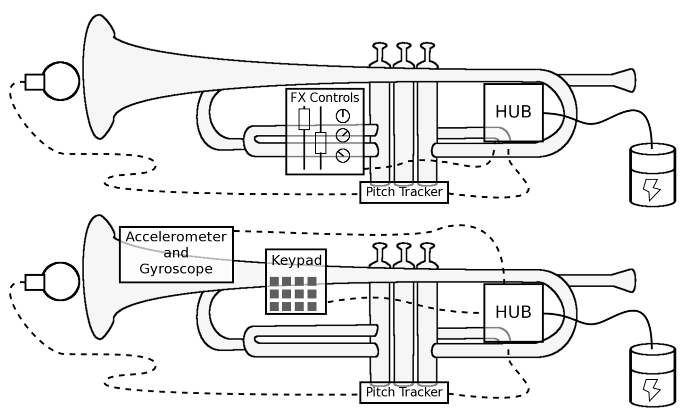

Figure 2. *Example Symbiote configurations.*

6. CONCLUSIONS AND FUTURE WORK

A modular platform for rapid prototyping of computer music systems, from sensor interface to synthesizer, was presented. This system will form a versatile springboard for the development of new digital musical instruments in the

Input Devices and Music Interaction Laboratory (IDMIL)[1]. As subsequent interface nodes and synthesis algorithms are implemented, the ease and speed of development of more sophisticated projects will increase. Using the nodes as replaceable parts will make maintenance of projects using this system quite simple, and changing/upgrading the implementation will not affect the operation of existing elements as long as the new ones adhere to the communication protocol and overall architecture. To address the limitations in computing power as provided by the current ARM7 and PSoC implementations, the system will upgrade to a Cortex M3 processor in the hub and an eventual migration to the PSoC5 (with a Cortex-M3 core) for both nodes and hubs upon their release later this year.

7. REFERENCES

[1] F. Avanzini, "Efficient numerical modeling of vocal fold mechanics," in *Int. Conf. Forum Acusticum*, 2002.

[2] R. Avizienis, A. Freed, T. Suzuki, and D. Wessel, "Scalable connectivity processor for computer music performance systems," in *International Computer Music Conference*, 2000.

[3] P. R. Cook, "Remutualizing the musical instrument: Co-Design of synthesis algorithms and controllers," *Journal of New Music Research*, vol. 33, no. 3, pp. 315–320, 2004.

[4] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical Journal*, vol. 1, no. 6, pp. 445–466, Jul. 1961.

[5] A. Hollinger, "Design of fMRI-compatible electronic musical interfaces," Master's Thesis, McGill University, 2008.

[6] A. Hollinger, C. Steele, V. Penhune, R. Zatorre, and M. Wanderley, "fMRI-compatible electronic controllers," in *Proceedings of the 7th international conference on New interfaces for musical expression*, 2007.

[7] J. Impett, "A meta-trumpet(er)," in *Proceedings of the ICMC*, 1994.

[8] S. Kartadinata, "The gluiph: a nucleus for integrated instruments," in *Proceedings of the 2003 conference on New interfaces for musical expression*, Montreal, Quebec, Canada, 2003.

[9] ——, "The gluion advantages of an FPGA-based sensor interface," in *Proceedings of the New interfaces for musical expression conference*, 2006.

[1] The project web page contains additional information about the project including photos and video links. http://www.idmil.org/projects/fungible_interfaces

[10] H. Leeuw, "The electrumpet, a hybrid electro-acoustic instrument," in *Proceedings of the New Interfaces for Musical Expression (NIME) conference*, 2009.

[11] G. Lemaitre, O. Houix, Y. Visell, K. Franinovic, N. Misdariis, and P. Susini, "Toward the design and evaluation of continuous sound in tangible interfaces: The spinotron," *International Journal of Human-Computer Studies*, vol. 67, no. 11, pp. 976–993, Nov. 2009.

[12] J. Malloch, S. Sinclair, and M. M. Wanderley, "A network-based framework for collaborative development and performance of digital musical instruments," in *Computer Music Modeling and Retrieval - Sense of Sounds*, ser. LNCS, R. Kronland-Martinet, S. Ystad, and K. Jensen, Eds., vol. 4969. Berlin / Heidelberg: Springer-Verlag, 2008, pp. 401–425.

[13] D. Morril and P. Cook, "Hardware, software and compositional tools for a real-time improvised solo trumpet work," in *Proceedings of the ICMC*, 1989.

[14] E. Motuk, R. Woods, and S. Bilbao, "FPGA-based hardware for physical modelling sound synthesis by finite difference schemes," in *Proceedings of the IEEE International Conference on Field-Programmable Technology*, 2005.

[15] B. Neill. Ben neill – mutantrumpet/composer. [Online]. Available: http://www.benneill.com/

[16] B. O'Flynn, G. Torre, M. Fernstrom, T. Winkler, A. Lynch, J. Barton, P. Angove, and S. O'Mathuna, "Celeritas — a wearable sensor system for interactive digital dance theatre," in *4th International Workshop on Wearable and Implantable Body Sensor Networks*, 2007.

[17] T. Saito, T. Maruyama, T. Hoshino, and S. Hirano, "A music synthesizer on FPGA," in *Field-Programmable Logic and Applications*. Springer Berlin / Heidelberg, 2001, pp. 377–387.

[18] B. F. Thomas Craig. (2008) Trumpet midi controller. [Online]. Available: http://instruct1.cit.cornell.edu/Courses/ee476/FinalProjects/s2008/twc22_bef23/twc22_bef23/index.html

[19] N. Villar, "Flexible physical interfaces," PhD Thesis, Lancaster University, Nov. 2007.

[20] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," in *Proceedings of the 2001 conference on New interfaces for musical expression*, Seattle, Washington, 2001, pp. 1–4.