

Use of a dual-core DSP in a low-cost, touch-screen based musical instrument

Paolo Bottoni, Riccardo Caporali, Daniele Capuano, Stefano Faralli, Anna Labella, Mario Pierro

Università di Roma La Sapienza Dipartimento di Informatica
Via Salaria, 113, 00198 – Rome, Italy

(bottoni, faralli, labella, pierro)@di.uniroma1.it - <http://hci.uniroma1.it/multimedialab>

ABSTRACT

This paper reports our experiments on using a dual-core DSP processor in the construction of a user-programmable musical instrument and controller called the *TouchBox*.

Keywords

dual-core, DSP, touch-screen, synthesizer, controller

1. INTRODUCTION AND BACKGROUND

Inspired by state-of-the-art controllers such as the *Lemur* [6], the *TouchBox* project was our proposal for the *Atmel DSP Programming Contest 2005* [1], where it won the first prize. Our aim was to merge a configurable sound synthesis engine with a configurable interface, in a compact and portable device using commercial low-cost components. There are many examples of programmable hardware-software synthesizers (e.g. Native Instruments *Reaktor* [9] and Clavia *Nord Modular* [4]). Touch-screen interfaces have received increased popularity in the last years, being produced commercially (as in the *Lemur* [6] and the Korg *Oasys* workstation [8]) and being subject of academic research (as in the Large Multitouch Display [5] or ReactTable [7]). Nevertheless, to the best of our knowledge, a compact, programmable touch-screen-based synthesizer has not yet appeared on the market.

2. DUAL-CORE DSP FEATURES

The Atmel DIOPSIS740 is a high performance dual-core processing platform aimed at audio, communication and beam-forming applications [2]. It includes two distinct CPU cores in a single chip: an *ARM7TDMI* RISC microcontroller and a *mAgic* VLIW DSP. The ARM7 core is the master, able to control mAgic execution; it also provides a great number of on-chip peripherals such as USART ports and AD/DA converters communication channels. The most prominent mAgic DSP core feature is its specially crafted dual register file: registers are coupled so that complex or vectorial

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.

NIME07, New York, USA

Copyright 2007 Copyright remains with the author(s).

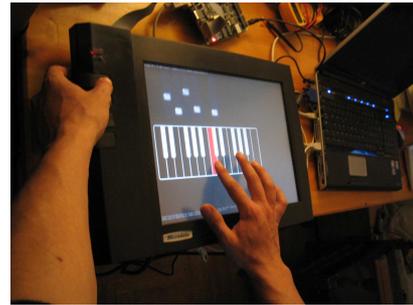


Figure 1: Playing the TouchBox prototype.

operations can be performed in a single cycle (single cycle complex multiply or multiply and add). In our application we have exploited the peculiar DIOPSIS740 processor's architecture so that the user interface and peripherals are handled by the ARM core, while all the audio-rate calculations are performed by the mAgic DSP core.

3. TOUCHBOX FEATURES

The TouchBox has been built using standard low-cost components available on the market. Lacking the multi-touch capabilities featured in more sophisticated controllers, the TouchBox employs a 15-inch single-touch screen, a group of 4 user-programmable hardware buttons, 4 stereo audio in/out connectors, and MIDI in/out ports (Fig. 1). The hardware buttons and two LEDs are housed in a custom aluminium case attached to the screen. The instrument is designed so that there is no predefined orientation: it can be used in a horizontal, vertical or left-handed fashion. The hardware keys can be used as shift-buttons, either in conjunction with pressure location on the screen, or independently (Fig. 2). This is achieved because the corresponding modules can be connected both to audio generators and to interface widgets, being thus exploited to show/hide the widgets and modify their behavior, as well as to trigger audio events. As an example, it is possible to switch between several sets of on-screen control modules by connecting the hardware buttons to a dedicated module "visibility switch" input. The hardware buttons abstraction modules can output different values according to the state of the hardware switch: every module can act as a "toggle" or as a "hold" switch. By combining hardware keys with software on-screen controls, it is possible to obtain several "pages" of controls as well as superimposed layers, in which

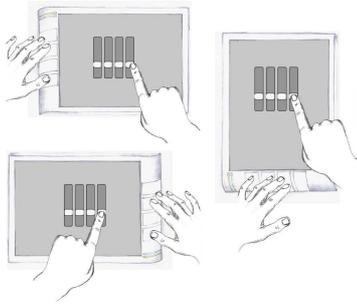


Figure 2: Different UI orientations.

a single touch screen press can activate multiple on-screen controls. The currently implemented interface modules are standard widgets such as sliders, buttons and a piano keyboard, while the available audio modules implement subtractive synthesis-oriented elements such as oscillators, envelope generators and filters. The current version of the OS operates with a latency of 32 samples.

4. OS AND EDITOR

The embedded operating system (*TouchBoxOS*) handles the user interface and peripherals using the ARM7 core, while the mAgic core performs audio rate processing. The two processes are synchronized using software semaphores, and share a common memory area where shared buffers are stored. Data transferred between the two processes is up/downsampled to audio/control rate. The TouchBox is programmed by connecting predefined software modules in a visual, Java-based editor (Fig. 3). The available modules implement either interface components, audio generators/processors or abstractions of physical hardware such as audio and MIDI in/out channels and the hardware buttons. Interface components correspond to widgets in the touchscreen interface, which can be placed in the editor by means of a dedicated interface preview window. The connections between modules transport 40-bit floating point audio data: the conversions between control and audio sample rates are performed transparently to the user. The editor generates two distinct programs for the two cores, which are compiled and uploaded into the TouchBox memory. The first program is actually a custom language script for the TouchBoxOS specifying which user interface modules are used, their on-screen positions and connections; the DSP core program is instead fully generated by the editor in mAgic assembler. Once compiled and uploaded, the module connection scheme is not modifiable: this allows the compiler to optimize the DSP code specifically for each TouchBox program during the generation phase. It is easy to add new, custom DSP modules to the TouchBox editor, by extending a common Java base class and specifying their mAgic ASM source. Interface modules are written in C and are integrated into each TouchBoxOS build, so that their graphics elements can be preloaded in the video RAM at boot time.

5. THE PROTOTYPE

We have built a working prototype of the TouchBox, using an Atmel JTST test board [3] to provide a set of basic functions (48Khz, 20-bit AD/DA converters, serial port communications, memory). We have developed custom boards for

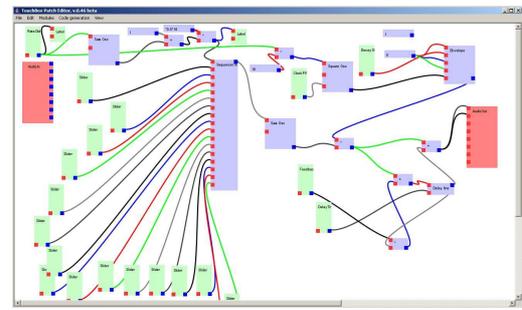


Figure 3: The Java-based editor.

the hardware buttons and MIDI connectors. Since a video card which meets our requirements cannot be connected directly to the JTST, we have also developed a video card emulator which runs on a personal computer and accepts basic video drawing commands from the serial port. The emulator also transmits control data from the touch screen to the JTST. The total off-the-shelf cost of the hardware used to build the prototype is under 1000 Euros (the graphics card emulator can run also on low-end PCs since it merely acts as a medium between the serial port and the PCs hardware). Also, by playing the prototype we have found that the 15-inch screen can actually be reduced to a 12-inch screen without sacrificing playability, which can help reducing the hardware costs further. One of the demonstration scenarios shows the TouchBox implementing simultaneously an audio effect processor, two touchscreen-driven monophonic synthesizers, plus a MIDI/touchscreen-driven polyphonic one. The touchscreen-displayable parts of the various sections interfaces can be dynamically switched and/or superimposed on the screen using the hardware keys.

6. FUTURE WORK AND CONCLUSIONS

The use of a dual-core DSP/microcontroller combination simplifies the development process of a project such as the TouchBox: the first working prototype was developed by a team of 3 people in 4 months. We plan to build a new, self-contained version of the prototype featuring a dedicated graphics card, thus eliminating the emulated video subsystem. This should allow us to experiment with more innovative control widgets requiring higher framerates and faster feedback times.

7. ACKNOWLEDGMENTS

The TouchBox prototype was developed for the *Atmel DSP Design Contest 2005*. The contest has been promoted by Atmel Roma (<http://www.atmelroma.it>).

8. REFERENCES

- [1] Atmel. <http://www.atmelroma.com>.
- [2] Atmel. *DIOPSIS740 Reference Manual*, 2005.
- [3] Atmel. *JTST AT572D740-DK1 User Manual*, 2005.
- [4] Clavia. <http://www.clavia.com>.
- [5] P. Davidson and J. Han. Synthesis and control on large scale multi-touch sensing displays. In *NIME*, pages 216–219, 2006.
- [6] JazzMutant. <http://www.jazzmutant.com>.
- [7] S. Jordà and M. Alonso. Mary had a little scoretable* or the reactable* goes melodic. In *NIME*, pages 208–211, 2006.
- [8] Korg. <http://www.korg.com>.
- [9] NativeInstruments. <http://www.nativeinstruments.com>.