

PDA: Real Time Signal Processing and Sound Generation on Handheld Devices

Günter Geiger
Music Technology Group,
Pompeu Fabra University, Barcelona
email: ggeiger@iuamtg.upf.es

August 27, 2003

Abstract

Not too long ago real time audio synthesis and signal processing used to be restricted to dedicated hardware. In the mid-nineties software synthesizers entered the field of desktop computing.

At the same time software for real-time computer music systems moved from dedicated DSP's to desktop computers with diverging operating systems. PD (Pure Data) is a computer music systems that showed a great deal of flexibility, originally designed to run on SGI Irix and Windows NT machines, it has been ported to Linux in 1997 and runs on all major Desktop operating systems nowadays.

This was only possible because of its openness and availability as free software. This paper describes a new port of PD, this time not for a new operating system but for a new type of computers: the PocketPC. The paper describes the capabilities of these small handheld devices nowadays and then goes on to describe different aspects of the software system derived from PD and called PDA.

1 Introduction

Computing devices get smaller and faster every year. With more or less powerful computers that can be carried around in the pocket being a reality, the question is what does this mean for computer music. One obvious advantage is that computer musicians will be more mobile, it will be possible for them to carry around their instrument. But the concept offers more possibilities than just being able to experiment with computer music wherever

you are.

New ideas of controlling computer music instruments have to be developed in order to take full advantage of the newly gained mobility.

Combining these handheld devices with wireless networking will offer new ways of collaboration and interaction between computer musicians.

All of this is not reality yet, different obstacles have to be overcome, but the most important steps are taken, and running a classical computer music tool like PD on a handheld device surely proves the feasibility.

The problems to overcome can be split into three main topics. First, how to solve the constraints of handheld devices in terms of performance.

Second, how will it be possible to work and develop on these devices, most of the time there is no keyboard, and the user interface basically consists of a touch-screen and a few buttons.

Third, how is it possible after having built a patch, to control the resulting instrument.

2 Handheld Devices

The most powerful processors that are available today for handhelds are Intel StrongARM and XScale processors, with a clock frequency of 200 and up to 400 MHz. Considering that the fastest desktop had 400 MHz not too many years ago and powerful sound applications were possible back then already, the performance problem seems to be easy to solve at a first glance.

This is of course not the case, because small devices have several restrictions that full grown com-



Figure 1: The iPaq running a PD FM synthesizer patch

puter systems do not have. On the other hand in some domains they are not that far behind.

2.1 Sound Quality

One might think that handheld devices would have a far inferior sound quality, they have integrated sound chips, and common experience from laptop computers tells us that integrated sound tends to be of low quality.

Taking a closer look reveals some surprise, though. The specification does not look that bad, the sound chip on the iPaq supports 2 channels 44.1kHz, CD quality sound. The chip used in the iPaq is Philips UDA1341TS, a chip designed for MiniDisc portable players, the data sheet of the chip [6] reveals a over-sampling ratio of 128, a S/N ratio of 100dB (sic).

Because of the design of the handheld this sound quality is conserved in the device, there is no interference of harddisks, no moving parts or influences from the power supply. The sound quality is excellent.

The system where tests have been carried out is a Compaq iPaq. The only audio input on these devices is a builtin microphone, therefore the audio input is of very low quality.



Figure 2: GIGE playing on the iPaq

2.2 Processing Power

The core of the most powerful handheld devices nowadays is based on the the Advanced RISC Machines (ARM) 32 bit embedded microprocessors [3]. This technology has become the most popular chip in use today. Its newest implementation is the PXA255 XScale processor, a pure RISC processor with binary compatibility with the older StrongARM processors, that supports clock speeds up to 400 MHz and several other speed enhancements, like code-compression in order to reduce instruction code size and a Multiply Accumulate (MAC) instruction, borrowed from DSP (Digital Signal Processing) technology.

Besides that the StrongARM processor has every feature that a modern RISC engine needs, hit-under-miss cache management, instruction pipeline and branch prediction. The only thing that makes it different from common Desktop microprocessors is its lack of a floating point unit.

As signal processing software written in higher level languages like C has been concentrated around using IEEE-754 32 bit floating point numbers, and the floating point emulation of the ARM processors is slow, this fact is a major obstacle for porting applications to ARM architectures.

2.3 Memory and Peripherals

It can be said that handheld devices offer a reasonable amount of memory which is about 64 MB [2].

HP's (formerly Compaqs) iPaqs are extensible devices. Using an adaptor for Compaq Flash cards, the system can be extended with different peripherals, ranging from additional memory cards (the base system comes with 48MB of flash memory) to wireless network cards and CF format hard disks. PCMCIA extensions are supported too, making it theoretically possible to use PCMCIA multichannel soundcards.

3 Computer Music on PocketPC

3.1 PD and Linux

PD [5] is a free software implementation of a computer music system based on the MAX paradigm. One of its outstanding features might be its compactness of code. This has led to the decision to use PD as a base for doing software development on handheld devices, rather than writing new software from scratch.

The current implementation runs on Linux only[8], although there is nothing Linux specific in the code. This means that it probably can easily be reused on the Windows PocketPC platform. The choice for Linux was based on the fact that all necessary tools were easily available.

3.2 The Performance

Preliminary tests showed quick results, because of the fact that getting PD to start on the iPaq was merely the task of recompiling it. The performance was bad though, so the system could not be used to do real-time audio processing. In order to improve the performance, all floating point calculation had to be avoided, specially for the data-flow part which is very performance intensive.

Avoiding floating point calculations means basically to replace every signal processing object that exists with an integer version [7] [1]. Luckily PD has a small and orthogonal set of base objects.

Table 1 shows some performance measurements (the typical number of sine oscillators benchmark) of the ported objects. The patches consist of oscillators, audio output and signal adders. The integer only version should have a speedup on standard computers too.

Audio off	7 %
10 oscillators	35 %
20 oscillators	51 %
30 oscillators	68 %
40 oscillators	87 %

Table 1: Performance on a 200MHz StrongARM processor

3.3 Programming with PDA

As mentioned above, the limited possibilities of user input on handheld devices makes it hard to write patches on the iPaq. This is definitely a field where a lot of improvement is possible. Up to now the patching happened on a desktop machine, running the equivalent version of PD, then the patches are transferred to the handheld and played there. It is possible to edit patches on the handheld device itself, but at the time being it is much more cumbersome.

3.4 Controlling the System

The direct way to control the patches on the iPaq is through the touch screen and buttons of the device. The touch-screen delivers data with two degrees of freedom. Handheld devices commonly have character recognition software installed, the software that comes with the iPaq Linux distribution offers a program called xstroke for this purpose. Xstroke allows to define new character mappings in a configuration file, making it the ideal tool for recognizing gestures on the touch-screen in order to improve the control.

All of these control input parameters can be mapped via their PD objects to events in the patch.

Of course it would be advantageous to have external controllers. This is possible through the serial port that is builtin in the iPaq. Accessing the serial port is straight forward by using PD's externals for serial port. There is a wide range of controllers that can be used this way. [4]

4 Future Directions

Future work will mainly focus on improving the integration of the software in the handheld device and

on finding new ways to control the system. Linking together several handheld devices via wireless networks and exchanging synchronization and control data is another field that will be very interesting.

Having a new technology available to generate music is always a challenging experience, only the future will show what kind of applications this will have.

In order to make the system easier accessible it could be recompiled on other PocketPC operating system such as WindowsCE or PocketPC 2002.

5 Conclusions

PDA delivers a proof of concept, it shows one direction into which computer music systems might go in the future, probably similar to the laptop computer music revolution we saw in the last few years. The performance figures show that we are not too early with this work, which hopefully will be one way to cross boundaries and open new paths to computer music practice.

References

- [1] Charles R. Yates, *Fixed-Point Arithmetic*, <http://personal.bellsouth.net/~yatesc/fp.pdf>, 14 April 1999.
- [2] Corp. Hewlett Packard, *Compaq iPaq Pocket PC H3900 Series*, Data Sheet, 2002.
- [3] Intel Corp., *The Intel PXA250 Applications Processor*, <http://www.intel.com/design/pca/>, 2003.
- [4] Joseph Paradiso, *Electronic Music Interfaces*, <http://web.media.mit.edu/~joep/SpectrumWeb/SpectrumX.html>, 1998.
- [5] Miller Smith Puckette, *Pure Data: another integrated computer music environment.*, Proceedings, Second Intercollege Computer Music Concerts (1996), 37–41.
- [6] Philips Semiconductors, *UDA1341TS - Economy audio CODEC for MiniDisc (MD) home stereo and portable applications*, 16 May 2002.
- [7] Robert Gordon, *A Calculated Look at Fixed-Point Arithmetic*, <http://www.embedded.com/98/9804fe2.htm>, 2003.
- [8] The Handhelds Project, *Linux on the Compaq iPaq*, <http://www.handhelds.org/>, 2000.