THE PEARL DIGITAL ELECTRONICS LAB: FULL ACCESS TO THE WORKBENCH VIA THE WEB

J. M. Martins Ferreira[1], Ricardo J. Costa[1], Gustavo R. Alves[2], Martyn Cooper[3]

[1] FEUP / DEEC
Rua Roberto Frias
4200-465 Porto - PORTUGAL

[2] ISEP / DEE
Rua Dr. Ant. Bern. Almeida
4200-072 Porto - PORTUGAL

[3] OU / KMI
Walton Hall
Milton Keynes MK7 6AA - UK

Abstract

Web-based course management and delivery software is becoming common in many areas of education, but the facilities provided by such systems do not support practical laboratory work. In many technical areas this limitation constitutes a serious restriction on the usefulness of any web-based educational frameworks, since "pen-and-paper" courseware is but a small part of the overall pedagogic materials that must be provided to the students. The system described in this paper addresses this need in the area of digital electronics and is being developed in the scope of an IST project called PEARL.

1. Introduction

Experimental work is a vital component of science and engineering teaching at all levels. The increasing use of multimedia packages or "virtual science" has much to offer in terms of teaching scientific facts and principles, but does not generally focus on the process of scientific enquiry or engineering practice. The PEARL project aims at developing a system to enable real-world experiments remotely in an e-learning context. The students will be able to interact with the remote experiment, change parameters and in some cases modify and design experiments. They will also be able to discuss their actions and what they anticipate will happen, and observe the results and analyse them, using Internet-based collaborative tools (e.g. CUSeeMe) embedded in the PEARL system. The process is real and so has an authenticity and unpredictability that simulations or descriptions cannot replicate. PEARL uses a thin-client approach and considers four areas of demonstration: bio-chemistry (remote access to an electron microscope), fundamental physics, PCB visual inspection, and digital electronics.

This paper deals with the digital electronics area and starts by presenting the overall system set up, followed by a description of the three types of experiments supported: 80C51 (microcontroller) programming, introductory logic design, and design for test using the IEEE 1149.1 and 1149.4 standards. The results of the trials conducted thus far are then presented, and a final section discusses the main conclusions derived from the work done to this date and the main guidelines for future developments.

2. Overall system set up

The implementation of the PEARL system is based on three main components (excluding the user computer and the network, i.e. the Internet): the Web server, the Lab server, and the remote laboratory infrastructure. The Web server hosts the course management and delivery software (in the case of PEARL, WebCT was used), processes all the actions / requests from the user side, and establishes the link with the Lab server (a PXI system from National Instruments with a 500 MHz Pentium III embedded controller), which is responsible for controlling the remote lab infrastructure. In the case of the digital electronics area, this infrastructure is illustrated in figure 1 and consists of a PXI-compatible Boundary-Scan controller from Goepel Electronics, a function generator, a multimeter, a two-channel 100 MHz oscilloscope (all PXI-based and supplied by National Instruments), and a hardware board that is specific of each type of experiment.

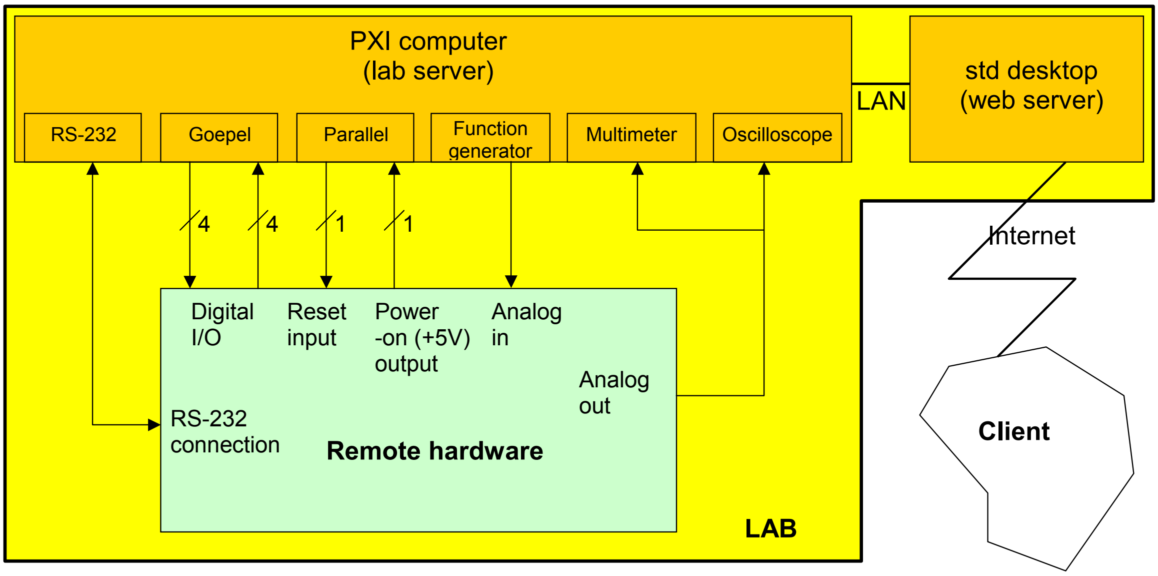


Figure 1: Overall experiments set up.

The interfaces to the previous instruments were built using LabView and are made available via the web in the form of Java-based versions, which are generated with a commercial tool from Nacimiento (AppletView). The main HTML page that provides access to the remote experiments comprises a text chat window, a videoconference window, and one window controlling the interface to the lab equipment.

3. 80C51 experiments

The remote hardware available for microcontroller experiments consists of an 80C51 board providing four digital inputs, four digital outputs and one analog output. The sequence of steps to be carried out by the students may be summarised as follows:

1. Write the code (in assembly, C, etc.) and use a compiler to produce an executable file in Intel hex format - **off-line**
2. Verify / debug the code using a simulator - **off-line**
3. Connect to the remote lab server
4. Upload the code, reset the remote hardware and transfer the hex file to be executed – **on-line**

Transferring the hex file to the 80C51 board immediately starts execution. Steps 1. and 2. may be carried out using public domain tools, while access to the remote lab server requires nothing else than a standard web browser.

At the present the first two steps are carried out using KEIL’s demonstration software (the only restriction is related to the size of the executable code, but 4 KB is more than enough for introductory experiments). Figure 2 illustrates the interface of the 80C51 programming experiments (the program uploaded in this case generates an ascending sawtooth waveform at the analog output).

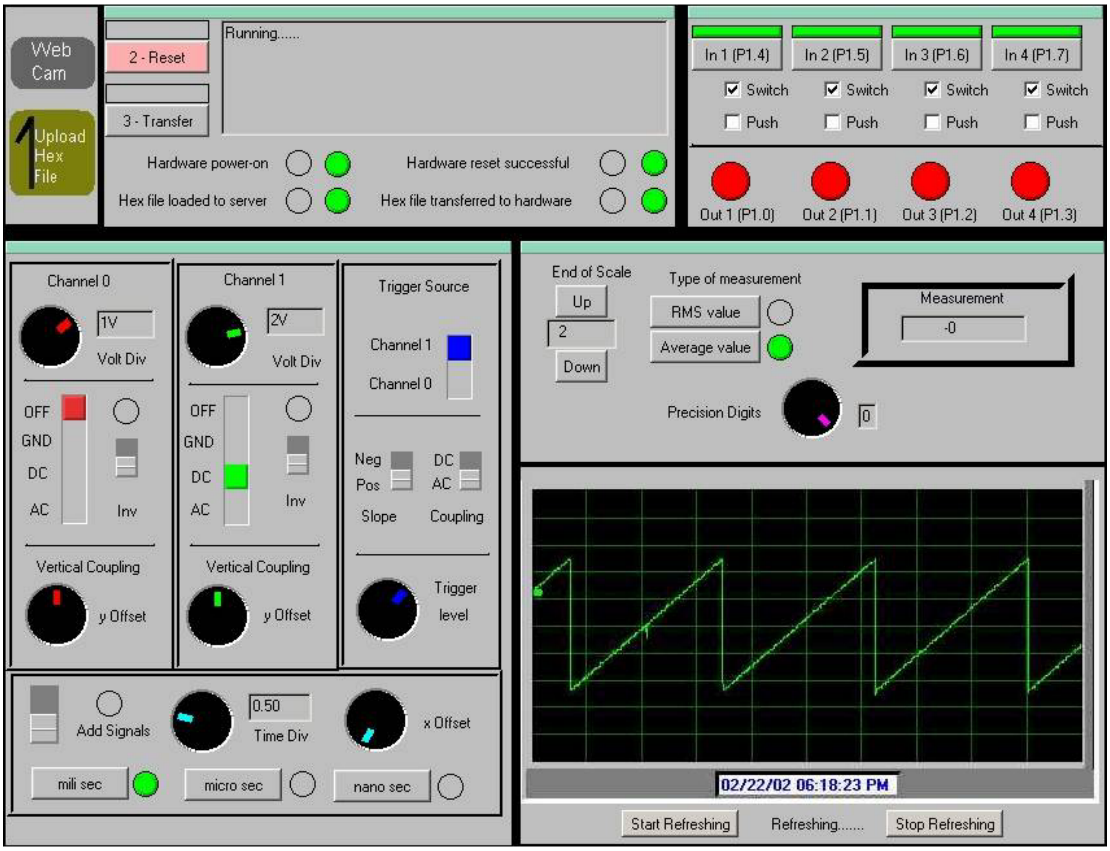


Figure 2: User interface for the 80C51 programming experiments.

Notice that the four digital inputs are directly controllable through the four keys represented in the upper right of figure 2, while the state of the four digital outputs is displayed in the four red leds located underneath the keys. The students also have available a webcam image showing the real hardware, which may be visualized in an extra window using the button present on the upper left.

4. Hardwired logic experiments

The hardwired logic experiments were designed to support an Introductory Logic Design class, where the students deal with such structures as counters, shift registers, finite state machines, etc. The remote hardware comprises a medium-complexity FPGA, programmed by the user through a sequence of steps that may be summarised as follows:

1. Design the circuit (schematic diagram, hardware description language coding, etc.) and produce the programming file for the FPGA – **off-line**
2. Verify / debug the circuit specification using a simulator - **off-line**
3. Connect to the remote lab server
4. Upload the code, reset the remote hardware and transfer the programming file to configure the FPGA – **on-line**

Notice that this is exactly the same sequence of steps as for the 80C51 programming experiments, and in fact uses the same interface as illustrated in figure 2 (the FPGA board does not possess an analog output, but the oscilloscope is used to visualise digital output waveforms). The remote hardware used in this experiment is the Xilinx FPGA board illustrated in figure 3.

Once again, the software required on the client side consists only of public domain tools. The packages currently used comprise the WebPack design entry and compilation software from Xilinx (step

1) and the ModelSim simulator (step 2), both available free of charge.

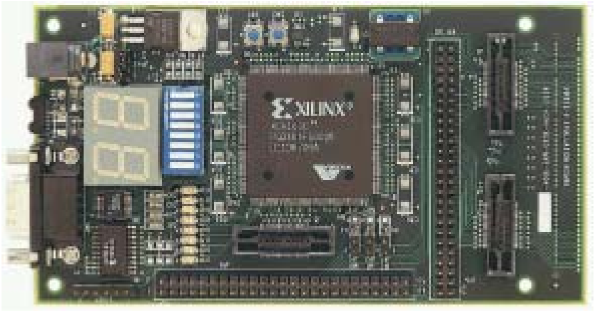
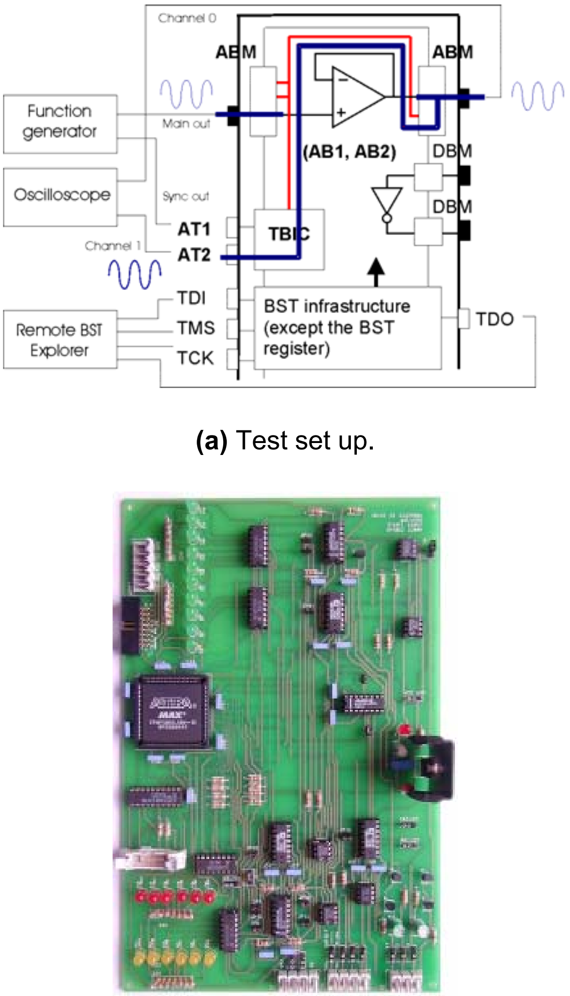


Figure 3: The FPGA board used to support hardwired logic experiments.

5. IEEE 1149.x experiments

The two experiments of this type are based on demo boards that contain one or more components compatible with the IEEE standards 1149.1 (standard test access port and boundary scan architecture) and 1149.4 (standard for a mixed signal test bus). In relation to the 1149.4 experiment, the demo board illustrated in figure 4 is accessed using two interfaces: one to control the digital part (i.e. the four test access port pins connected to the boundary scan controller) and the other one to control / observe the analogue part (both the functional and test circuitry). The two ATAP (analogue test access port) pins are connected to the PXI instruments as follows: AT1 is connected to an output of the function generator (controllability) and AT2 is connected to one of the oscilloscope channels (observability). The two analogue functional pins are connected to the other output of the function generator (the input pin) and to the second channel of the oscilloscope (the output pin).

The sequence of steps to be carried out by the students includes setting up the oscilloscope and the function generator, and exercising the four digital TAP pins (TDI, TDO, TMS and TCK), using a public domain tool that is made available to work with these two IEEE test standards.



(a) Test set up.

(b) The IEEE1149.4 board.

Figure 4: The IEEE 1149.4 experiment.

6. Trials

The first trials took place at FEUP on November 2001 and provided initial feedback on the strengths and weaknesses of the PEARL system. At that period no special measures were put in place concerning the quality of the network connections from FEUP to the outside world. Also, and since the Remote Lab interface was not incorporated in the collaboration tools interface, each pair of students had to open a series of windows to perform the proposed experiments. These two aspects were the main drawbacks that were identified in the first trial, where only one pair of students (in four) managed to conclude the experiment within the proposed time frame (30 minutes). Based on the previous experience, the FEUP team introduced the following corrective measures:

1. Ask the FEUP IT services to provide / guarantee sufficient bandwidth for the trial. In face of this request, the IT services reconfigured the gateway that serves the Lab, and established a special 90 kbps channel for all inbound traffic to FEUP. The outbound traffic from FEUP was not critical.
2. The interface to the FEUP lab was integrated into the collaboration tools page. Some re-arrangements on the page layout had to be done, in order to have an all-in-one interface, as illustrated in figure 5. However, if the collaboration tools are not required, they are excluded from this interface.
3. All the computers used on the trials were monitored, in terms of network traffic, using an application called Etherpeek.
4. Two video cameras monitored the students' computers, in order to obtain detailed information regarding all their actions, so as to better identify possible causes of failure.

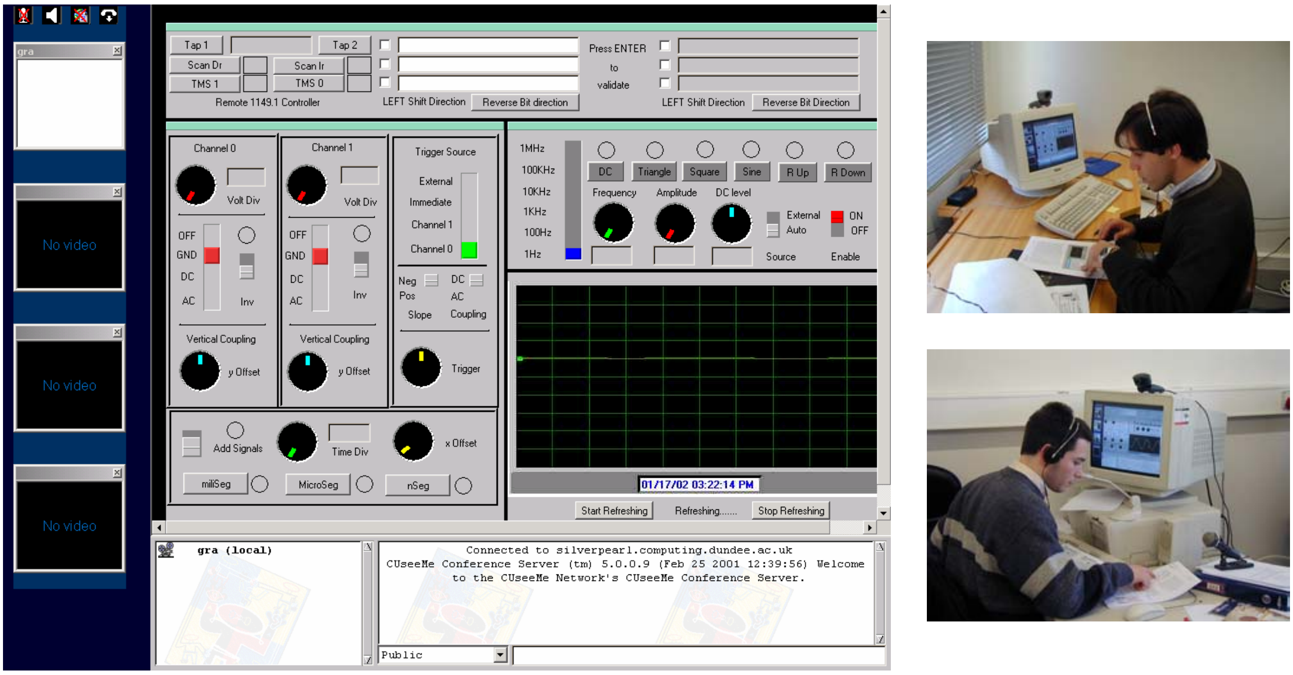
A subsequent trial was then carried out, using one single pair of students to perform the IEEE 1149.4 test infrastructure experiment. The tutor was available on-line in a different place, with the two students – working as a team - also located in different places. They had 30 minutes to complete two experiments, the first of which was fully described in a detailed lab script (e.g. what vectors to shift and what buttons to push), while the second experiment involved some further work from their part (e.g. to determine what vectors to shift).

The first experiment was successfully done in about 15 minutes, while the second one took approximately another 25 minutes, with some hesitations and errors by the team. All the sequence was videotaped, and the network traffic was monitored. The two students made full use of the audio communication facilities and used the chat frame to communicate with the tutor. Text-based chatting had two benefits: it had a low impact on the bandwidth available for audio communication, and it allowed them to see / read / discuss each message (one or more times). The tutor listened to some parts of the dialogue between the two students so as to better understand what they were doing, in spite of the negative impact on the quality of the audio link between them (which could be noticed, when the tutor switched on / off is own speaker). Finally, a series of photos were taken during the trial, as illustrated in figure 5.

7. Conclusion

The work done so far proved the interest of providing remote experiments to students in technical areas, as a vehicle to convey an awareness of practical issues that cannot be obtained through simulation alone. Moreover, it has proven that a thin-client approach is feasible, presenting minimum requirements on the students accessing such systems from home. The availability of large-scale remote hardware experiments for classes with hundreds of students is however dependent on technical and non-technical aspects, such as economic factors, network traffic implications, firewall protection policies, etc.

The work planned for the immediate future, until the completion of PEARL, addresses the following aspects: development of pedagogical complements, including troubleshooting and "what-if" instructions in case of partial system unavailability, reliability improvements by the introduction of redundancy (e.g. replicating the web server), facilities for scheduling of resources / on-line booking, and session log strategies for student assessment.



01/17/02 03:22:14 PM

```
Connected to silverpearl.computing.dundee.ac.uk
CUseeMe Conference Server (tm) 5.0.0.9 (Feb 25 2001 12:39:56) Welcome
to the CUseeMe Network's CUseeMe Conference Server.
```

Figure 5: Interface used for the experiment on the IEEE 1149.4 Std., integrated with the web-based collaboration tools (CUSeeMe), and photos from the trial.

References

[1] Martyn Cooper, Eileen Scanlon, Stuart Freake, “Remote Controlled Teaching Experiments, in Science and Engineering Subjects, Accessible over the World-Wide-Web: the PEARL project", *Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications* (EdMedia), Montreal, Canada, 2000.

[2] Martyn Cooper, Liliana Patricia Santacruz Valencia, “User interface approaches for accessibility in complex WWW applications - Examples from the PEARL project", *Proceedings of the Association for the Advancement of Assistive Technology in Europe Seminar* (AAATE), Hatfield, UK 2000.

[3] André V. Fidalgo, Ricardo J. Costa, José M. Ferreira and Gustavo R. Alves, "Experimenting the 1149.1 and 1149.4 test infrastructures in a Web-accessible remote Lab (without Plug-ins!)", *Proceedings of the XVI Conference on Design of Circuits and Integrated Systems* (DCIS’01), Porto, Portugal, 2001.