Preparing a remote conducted course for microcontrollers based on Arduino

Fotopoulos Vassilis
Hellenic Open University
Digital Systems and Media
Computing Laboratory
vfotop1@eap.gr

Spiliopoulos I. Anastasios
Hellenic Open University
Digital Systems and Media
Computing Laboratory
a.spiliopoulos@eap.gr

Fanariotis Anastasios
Hellenic Open University
Digital Systems and Media
Computing Laboratory
a.fanariotis@eap.gr

Abstract

During the past few years, there is a lot of discussion about remote labs in the fields of informatics and computer engineering. However for courses related to logic design, microprocessors, microcontrollers and computer architecture, usually simulation tools are used only. In this paper, the preparation of a remote microcontroller course will be presented, in which the students may control real hardware via a web interface, directly from their homes while watching the results in real time through web cameras, connected to each of the four laboratory systems. The heart of the lab is the famous Arduino board.

Our target while designing the system was simplicity both hardware-wise and software-wise without degradation of learning volume and quality. Combining these two aspects a Graphical user interface was designed that is both pleasant and functional to use. Ultra fast software has been used server-side keeping the real-time experience true to its definition and finaly a set of lab-exercises was created that is both educational and interesting, maximizing this way the educational gain per time spent.

Key-words: remote lab, remote course, microcontroller, arduino

1. INTRODUCTION

Laboratories in the field of logic design, computer architectures and micros, are included in the curricula of every computer science or computer engineering department around the globe.

However, those labs in their traditional form, require lots of equipment like various ICs, development boards, testing equipment etc. making them impossible to be provided in a remote form. That is the reason that all available relative remote courses just make use of simulation tools that have become widely available and actually one may easily find free of charge tools for each category. For example, one can use logisim (Burch, 2012) for simple logic design circuits or Quartus (Altera Corp., 2012) for more complex designs and computer architecture exercises, not to mention microprocessors and microcontrollers simulation tools. Each micro has its own free tool and there are too many of them.

Of course there are some exceptions. MIT has an excellent record for remote labs with iLabs (Massachusetts Institute of Technology, 2012). In the case of Advanced Digital Lab for example, students are controlling real equipment installed at MIT labs. However, they see waveforms that correspond to running their design on an Altera DE1, development board. So the experience is not so different comparing to a simple simulation tool.

A different approach is presented from the Remote-Digital Signal Processing Laboratory (R-DSP Lab) (Kalantzopoulos, Karageorgopoulos & Zigouris, 2008) of the University of Patras. In this paradigm, users control real DSP equipment, a Tectronix digital oscilloscope and a TTi frequency generator, by means of NI's LabView runtime engine. The student may observe the instruments, mainly the oscilloscope screen by a locally positioned web camera. In the field of microcontrollers, which is of most interest to this work, at University of Technology Sidney (UTS) there is a remote lab called Coldfire Rig (Labshare Australia, 2012), (Kotsuki & Murray, 2011), (Fig. 1). In that setup, 12 Motorola 32-bit microcontrollers are used from the students that may write stand-alone code in C or assembly language or applications that run on the uClinux embedded operating system. However on the aspect of physical output, the web cameras provide images of a 7-segment LED display output board.

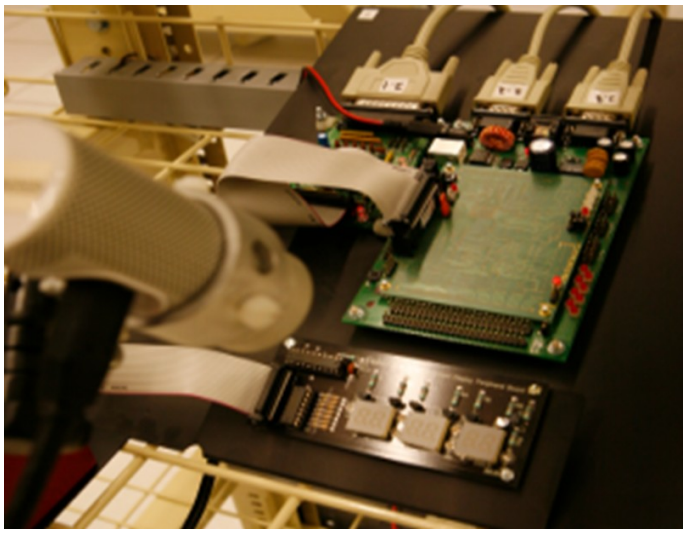
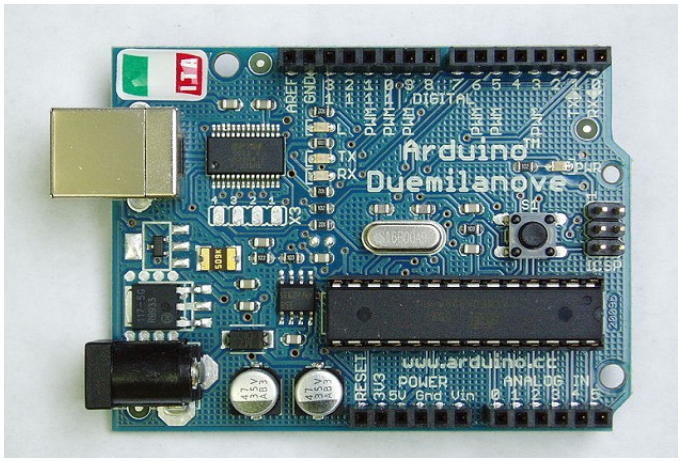


Figure 1: The UTS Coldfire Rig Laboratories

This paper is organized as follows. In section 2, the Arduino board is briefly presented. Section 3 is an introduction to the web interface and the programming environment while in section 4, the selection of lab experiments is discussed and rationalized, along with possible variations. In section 5 setup issues are discussed concerning the required equipment the specifications and the various choices of technologies and server-side software, along with the related costs. Finally section 6 presents the conclusions and directions for future work.

2. THE ARDUINO BOARD

The Arduino board was developed in Ivrea, Italy, around 2005 (Kushner, 2011). It begun as a fork of the Wiring project of Colombian artist and programmer Hernando Barragán. It started as a cheap means of teaching physical computing and control of student-built interactive projects. There is a great variety of different implementations based on the original board; in our case, we selected the low cost but rich featured Arduino Duemilanove (Fig. 2). It is not the latest version but it is well suited for our needs and quite cheaper comparing with the most recent Arduino Uno (Arduino, 2012).



Fugure 2: The Arduino Duemilanove board

The Duemilanove board, is built around an Atmel ATmega328 microcontroller. The board can be powered by batteries or a wall transformer, since the onboard 5V regulator can easily bring down a voltage of up to

12V to the 5V level required for the microcontroller to work. A USB cable can be used for both powering and programming the board from a personal computer. The ATmega328 microcontroller, clocked by a 16 MHz crystal oscillator, is a device powerful enough for the project needs. It has 6 analog inputs, 14 digital I/O pins of which 6 may provide PWM outputs, an SPI dedicated port, I2C support and many more. A flash memory of 32KB is available for user programs, along with 2KB of SRAM and 1KB of EEPROM. Additional boards can be stacked upon the main board, called "shields", providing extra functionalities such as sensor arrays, ethernet connectivity, wireless communications, DC motors driving etc

3. THE ENVIROMENT-WEB INTERFACE

For accessing the remote lab through the web, a XAMPP system is used with the latest versions of Apache and MySQL. A cheap web camera records the output of the system and for streaming the video to the web application's window, the native Linux video encoder "avconv/ ffmpeg" alongside a streaming server are used. Other technologies used include AJAX, Javascript, Java Applets, Flash etc. Of course all programming tools needed for compiling and flashing user code to the microcontroller, are also installed at the system

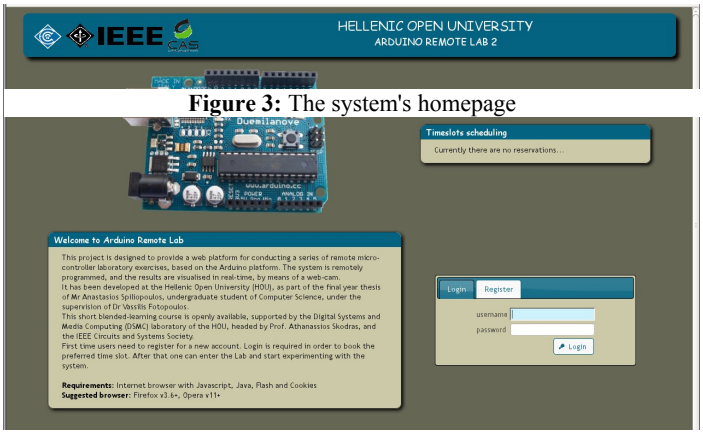


Figure 3: The system's homepage

By accessing the system's homepage (Fig. 3), user may log in or create a new account. At the same page, a table shows the reservations' schedule for the system, since each user can reserve a one hour time slot for experimenting. After the login procedure, if the user has reserved the current time slot he may proceed to the laboratory's main page, or he is given the opportunity to reserve a time slot of his liking

At the main page, there are four dragable windows inside the browser area (Fig. 4). The first one (upper left) is the code editor area, where the user can write his code. Syntax highlighting is provided. There are also three command buttons and a check box. By pressing the first button labeled "compile", the code is compiled and the result of the compilation is shown at the compilation output window right below the button. If there are any errors, by clicking on each error description at the compilation output, cursor moves at the source code area and at the line that contains the error, just like any ordinary programming IDE would do. If the "upload" check box is checked the compile button will also upload , flash and begin execution of the code to the microcontroller. The reset Arduino button functionality is obvious, the systems resets the data and flash memory by

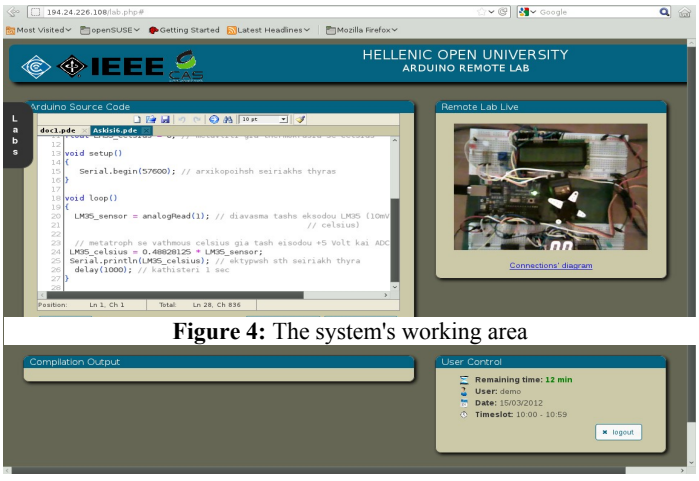


Figure 4: The system's working area

loading an empty program. Last but not least, the serial monitor button opens a terminal window through which the user may interact with the Arduino via its serial port in real time, by typing or viewing data.
It has to be noted that all software needed is installed server side. There is absolutely no other requirement from the user other than having a common web browser with Java and Flash enabled, that nowadays all user-level operating systems have one preinstalled.

4. THE SELECTION OF LAB EXPERIMENTS

Selecting the lab exercises is very important both from the side of the educational requirement and the side of cost and target board capabilities. By examining the structure of traditional on-site microcontrollers training, and taking into account the available pins and capabilities of the Duemilanove board, we concluded to the following nine experiments:

Lab 1 - Blinking LED
Lab 2 - Using an RGB LED
Lab 3 - Printing on LCD
Lab 4 - Controlling a Servo Motor
Lab 5 - Data acquisition from an LDR photoresistor
Lab 6 - Using the LM35 temperature sensor
Lab 7 - Using 7-Segment displays
Lab 8 - Using a TFT-LCD Display
Lab 9 - Using an PCF8754 I2C port expander

These are real classic exercises and the course is quite complete, containing basic I/O, display handling, motors and sensors. The equipment of the first seven experiments can be connected to a sole arduino occupying all its available pins while for more demanding applications such as the TFT-LCD display a dedicated arduino board can be used leaving some space for new, more lightweighted ,processing-wise, experiments like the port expander. Some pins can be released if instead of the classical text lcd or TFT module, another one with I2C capability is selected, but this is an issue that may be dealt in a future version of the system. Sample code is provided for all of the nine experiments by clicking on a link at the tab that pops out at the left side of the window, so that the student may test the setup immediately.

The code is very easy to understand and one of the major advantages of the proposed system, is the fact that one may easily combine code from all the existing experiments to create new ones. A sample list of such combinational experiments follows:

- read the value of the temperature sensor and display its integer part on two seven segment displays
- read a sensor value and print it on the LCD screen
- program some games and use the LCD screen as output. Could be some game like “hangman” or even include action by moving some character using the serial terminal as input
- read the value of a sensor for a time period and analyse it (e.g. mean value, standard deviation etc.). Even more complex analysis can be performed (e.g. FFT) and the results may be shown at the LCD
- turn the servo motor by controlling it with the serial terminal, or by the value of some of the installed sensors
- gradually increase or decrease the intensity of a led, based on the photoresistor's reading
- program a guess game that will get input from the terminal and lights the RGB led green if the answer is correct or red if the answer is wrong

These are only a few of the possible paradigms of additional experiments that can be realized. Some of them may require a good amount of new code, while others (e.g. the first two examples) can be performed by simple copy-paste parts between the available codes of two experiments. However variations can be asked even in the original codes. For example, we intentionally leave the 7-segment displays lit, after executing the corresponding experiment, thus when we ask the student to experiment on this, we have an additional question about how to switch off the displays. There are several other such specially "forgotten details", like stopping the movement of the servo, switch off the LCD backlight or the leds. It worths mentioning that the student may use the system just to compile and verify his code, in case he just wants to practice on coding or the hardware setup does not support the code's requirements.

5. SETUP ISSUES

Five Personal computers have been used for setting up this remote lab. All systems were relatively old, pentium 4 desktops with 512MB of memory and 40GB Hard disks. Storage capacity and processing power are not crucial for the setup but the amount of memory is, since each system hosts an Apache and MySQL server. Having this in mind an Ubuntu (Ubuntu, 2012) based Linux operating system named "Lubuntu" (Lubuntu, 2012) was chosen as the main platform running an LXDE desktop and thus minimizing memory requirement while making the system as energy efficient as possible, two open source software applications were used for encoding and streaming the video captured by the webcam, these are the avconv video encoder that is preinstalled in every ubuntu based Linux and the C++ RTMP Server (C++ RTMP Server, 2012), a powerful and very light RTMP video streamer. All the above complemented by the XAMPP server effectively made a setup totally based on open source software. The PCs have been running trouble-free, 24/7 up to this day and thus it is dimmed a good way to revive old laboratory computers. Some further improvement may be accomplished if the programming environment, web and sql server is installed to a single server PC, leaving media encoding and streaming as the only tasks for the PCs to which the boards are connected.

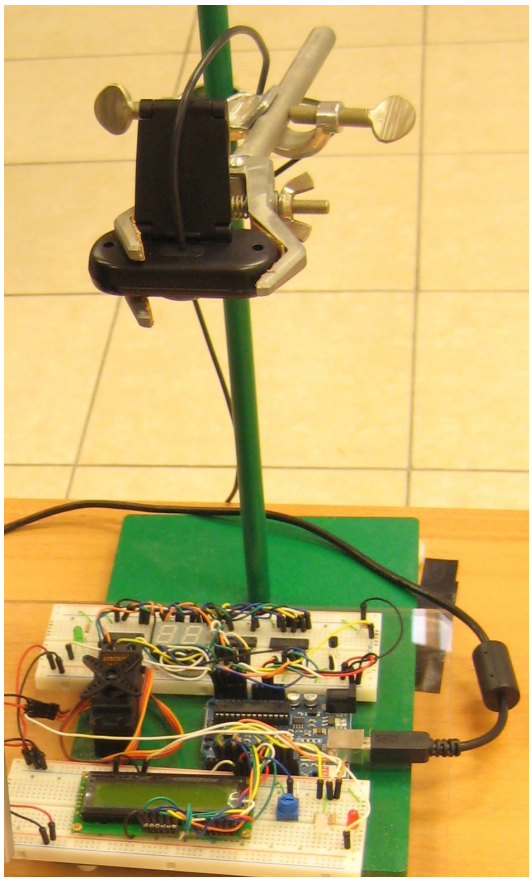


Figure 5: The system fully assembled

As far as the cost is concerned, the electronic equipment used was not costly. Each system consisted of an arduino,two breadboards, a webcamera, and a variety of sensors and other active equipment (Fig. 5) costs less than €100 and actually the cost may go down to €80 depending to the providers of the electronic parts. A total cost of €350-€400 for setting up a remote microcontrollers lab with 4-5 stations, is very affordable even for labs with a limited budget.

6. CONCLUSIONS AND FUTURE WORK

In this paper, the design and implementation of a remote microcontroller laboratory course is presented. The design is novel, low cost and has been successfully tested. The course has been recently offered to a small number of students (~30), in order to

get some early feedback by using a suitable questionnaire. After processing the results we expect to identify further improvements, including hardware expansion and addition of new experiments.

The next step is to offer the lab as a pilot course into the HOU's Computer Science curriculum. It is especially important for an open university such as HOU, to offer laboratory experience which is a missing characteristic of most open universities.

7. ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the IEEE Circuits and Systems Society (CASS) for partially funding the creation of this laboratory.

8. REFERENCES

Burch, C. (2012). Logisim (Version 2.7.1*) [Computer Software]*. Retrieved from http://ozark.hendrix.edu/~burch/logisim/index.html

Altera Corporation. (2012). Quartus II Web Edition [Computer Software]. Retrieved from http://www.altera. com/products/software/quartus-ii/web-edition/qts-we-index.html

Massachusetts Institute of Technology. (2012). *MIT iCampus: iLabs.* Retrieved from http://icampus.mit.edu/iLabs/default.aspx

Kalantzopoulos A. ,Karageorgopoulos D., & Zigouris E. (2008). A LabVIEW based Remote DSP Laboratory. *International Journal of Online Engineering*. 4 (Special Issue, REV2008), pp. 36-44.

Labshare Australia. (2012). *Current and Future Labs.* Retrieved from http://www.labshare.edu.au/project/index.php?option=com_content&view=article&id=123&Itemid=88#Coldfire

Kostulski T., & Murray S. (2011). *Student Feedback from the First National Sharing Trial of Remote Labs in Australia*. Paper presented at the 8th International Conference on Remote Engineering and Virtual Instrumentation of International Association of Online Engineering, Brasov, Romania. Retrieved from http://www.labshare.edu.au/media/img/student_feedback.pdf

Kushner, D. (2011). *The Making of Arduino - IEEE Spectrum.* Retrieved from http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino

Arduino. (2012). *Arduino HomePage.* Retrieved from http://arduino.cc/

Ubuntu. (2012). U*buntu Homepage.* Retrieved from http://www.ubuntu.com

Lubuntu (Version 12.04 LTS) [Computer Software] , Retrieved from http://lubuntu.net

C++ RTMP Server (Version 784) [Computer Software]. Retrieved from http://www.rtmpd.com/