AC 2011-70: AN ONLINE LABORATORY-BASED GRADUATE ENGINEERING TECHNOLOGY COURSE IN PROGRAMMABLE DEVICES AND SYSTEMS

Warren A. Rosen, Drexel University (Eng. Technology)

Dr. Warren A. Rosen received his Ph.D. in physics from Temple University in 1978. Between 1978 and 1985 Dr. Rosen served as assistant professor of physics at Colby and Vassar Colleges where he carried out research in optical physics, solar physics, and medical physics. From 1985 to 1996 he worked at the Naval Air Warfare Center, Aircraft Division in Warminster, PA where he established an optical communications laboratory for development and characterization of optical components, systems, and protocols for high-performance avionics data networks. Dr. Rosen is currently an assistant clinical professor at Drexel University, where he is responsible for developing and teaching courses in microprocessors, microcontrollers, and FPGAs. Dr. Rosen has carried out research sponsored by the National Security Agency, National Science Foundation, the National Oceanic and Atmospheric Administration, DARPA, the Office of Naval Research, and the Missile Defense Agency.

Dr. Rosen is the author or coauthor of over 50 publications and conference proceedings and the holder of five U.S. patents in computer networking and signal processing.

M. Eric Carr, Drexel University

Mr. Eric Carr is currently the Laboratory Technician for Drexel University’s Engineering Technology program. Eric assists faculty members with the development and implementation of various Engineering Technology courses. A graduate of Old Dominion University’s Computer Engineering Technology program, Eric enjoys finding innovative ways to use microcontrollers and other technologies to enhance Drexel’s Engineering Technology course offerings. Eric is currently pursuing a MS in Computer Engineering at Drexel, and is an author of several technical papers in the field of Engineering Technology Education.
An Online Laboratory-based Graduate Engineering Technology Course in Programmable Devices and Systems

Abstract

In this paper we describe the development of an online graduate engineering technology course in programmable devices. The course is intended to provide the graduate engineering technology student with an overview of programmable devices and systems for industrial and embedded applications. Microprocessors, microcontrollers, Field Programmable Gate Arrays (FPGAs), and programmable logic controllers (PLCs) are compared with respect to suitability, performance, and cost (both device and implementation costs) in industrial and embedded environments. Industry standard development and modeling tools are introduced and used to develop representative control programs and to predict performance. A key feature of the course is the extensive use of web-based laboratory exercises using equipment housed in Drexel University’s PLC Laboratory. This equipment includes Dragon12 microcontroller development boards, Digilent Spartan-3E FPGA boards, and Amatrol PLC workstations. For microprocessor experiments students use their own PCs or PCs in the lab.

Introduction

Over the past 30 years there has been an explosion in the number and types of programmable digital devices that can be used in applications ranging from controlling simple processes to controlling a space shuttle or the power grid.

This course is designed to introduce the student to the spectrum of programmable devices that they may choose from when confronted with a complex application. Students learn the key attributes of the most commonly used devices so that they can make an informed selection decision. They learn the internal structure of these devices, their strengths and weaknesses, and costs associated with their deployment and use. They also learn how to use industry standard programming tools to program these components and to make estimates of performance, cost, and power consumption.

The course was developed as part of Drexel’s new Master of Science program in Engineering Technology. This program is aimed at students who are seeking professional development to meet evolving workforce demands, expanding opportunities for professional advancement, or pursuing a managerial position. The emphasis of the program is on the applied aspects of the technological spectrum, such as product improvement, industrial practices, and engineering technology operations. Programmable devices such as microcontrollers, programmable logic controllers, and Field Programmable Gate Arrays are critical components in any modern industrial environment, and an understanding of their characteristics, application space, and associated costs is essential to the advanced practitioner.

There are three principal objectives for the course. The first is to provide the student with a comparative overview of programmable devices, including microprocessors, microcontrollers, programmable logic controllers, and field-programmable gate arrays (FPGAs). The second is to
provide the student with an understanding of the relative suitability of these components in industrial and technical applications with respect to key selection criteria including cost, performance, power requirements, and ease of implementation. The final objective is to familiarize the student with a set of industry-standard tools used to program these devices and to make estimates of their performance and power requirements.

Course Content

The course is designed for a ten-week (plus final exam) quarter system. The topic schedule is shown in Table 1.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to programmable devices, overview of useful programmable technologies, typical applications and selection criteria</td>
</tr>
<tr>
<td>2</td>
<td>Microprocessors – history, CPU organization, how a program is executed, programming, high-performance techniques</td>
</tr>
<tr>
<td>3</td>
<td>Introduction to Programmable Logic Controllers, key components, relay logic, programming</td>
</tr>
<tr>
<td>4</td>
<td>PLC techniques – seal-in logic, latches, interlocks, event sequencing</td>
</tr>
<tr>
<td>5</td>
<td>PLC timers, counters, jumps and subroutines, program documentation</td>
</tr>
<tr>
<td>6</td>
<td>Introduction to microcontrollers – microcontroller organization, programming models, memory maps, selection criteria</td>
</tr>
<tr>
<td>7</td>
<td>Programming – the CodeWarrior IDE, programming languages, assembly instructions and directives, addressing</td>
</tr>
<tr>
<td>8</td>
<td>I/O ports, timing, pulse width modulation</td>
</tr>
<tr>
<td>9</td>
<td>Introduction to Field Programmable Gate Arrays, internal architectures, applications, ASICs and FPGAs, introduction to HDL</td>
</tr>
<tr>
<td>10</td>
<td>Programming, FPGA design flows, design tools, key vendors, course summary</td>
</tr>
</tbody>
</table>

Table 1. Topics by week for the online course.

The first week provides an introduction to what programmable devices are followed by an overview of programmable technologies. The devices of interest are broken down into two categories—“stored program” devices such as microprocessors and microcontrollers, and “data flow” type devices such as Field Programmable Gate Arrays. The operating characteristics, strengths, and weaknesses of each are discussed in terms of typical applications one might encounter in an industrial environment. Selection criteria such as performance, power consumption, ease of use, and cost are also discussed.

In the second week microprocessors are introduced not only as the programmable device the students are most likely to have encountered, but also as the foundation for the next two programmable devices, the microcontroller and the programmable logic controller. First, a high-level description of what a microprocessor is and what it does is presented. Then a brief history
of the development of microprocessors is given. The focus is mainly on Intel processors but a brief description of Motorola/Freescale and AMD machines is given. This is followed by a description of a simple, generic microprocessor’s internal organization including the general functional elements common to all microprocessors (arithmetic and logic unit, special registers, and control circuitry), how they are interconnected, and how they work together with external memory to execute the program written for them by the programmer. Next, the instruction set and memory map are described and the student carries out some basic exercises using his or her own PC and the MS-DOS Debug program. Remote access to computers in Drexel’s Programmable Devices Laboratory is available to students using Macintosh computers or PCs running Windows 7, for which Debug is not available. The discussion of microprocessors concludes with a description of techniques that have been developed over the years to dramatically increase their performance, including multi-layered cache memory, pipelining, and superscalar execution. The advantages and disadvantages of Reduced Instruction Set Computers (RISC) are also described.

The next three weeks are devoted to Programmable Logic Controllers. The first week’s discussion introduces the basic concept of the PLC, describes what PLCs are, their uses and application space, their history and the motivation behind their development, and what differentiates a PLC from a conventional microprocessor or microcontroller. Next, the internal components of a PLC together with some of the external components such as switches and relays needed to make a process control system are described. The relationship between “relay logic” and Boolean algebra is discussed and ladder logic, the graphical method of representing automation processes that predate PLCs, is introduced. The unique way in which a PLC executes a program (by examining all the inputs and then evaluating all the outputs, rather than executing sequentially) and the impact this scheme has on the way PLCs are programmed is described. The students then learn how to program a PLC using ladder logic using the industry standard RSLogix development environment.

The second week of the PLC sequence is devoted to basic PLC techniques. These include the use of the Startup bit, seal-in logic, internal addressing, latching circuits, Master Control Reset instructions, and interlocks and their use for enhanced safety and in event sequencing for time-dependent behavior.

In the third week the discussion of PLCs is completed with a few advanced topics including timers, counters, and program control instructions. The discussion of timer and counter instructions includes complete timing diagrams for all outputs and internal bits for all commonly used timers and counters. Next, instructions used to control the flow of the program are described, including labels, jumps, and subroutines. Suspend and Temporary END instructions used for debugging are also described. Finally, the importance of good documentation is discussed along with techniques provided by RSLogix for self-documentation.

Each week’s lecture is accompanied by a related set of laboratory exercises. These exercises are performed online on PLC workstations in Drexel’s Programmable Devices Laboratory. Each workstation contains an Amatrol 85-P-AB2 Programmable Controller trainer system with Allen-Bradley PLC (now marketed by Rockwell Automation), and electro-pneumatic and motor application panels, interface to a Dell Optiplex 780 computer (Core 2 Duo E8400, 3GHz,
3GB DDR3 1066) hosting an RSLogix 500 PLC software package. The facilities are available at all times. Figure 1 shows the workstation and host computer. The web camera in the foreground allows the student to monitor the behavior of the system visually. The camera’s image can be seen on the computer monitor. The panel at the top of the PLC station is the electro-pneumatic panel. The online laboratory capability is described in detail in the next section.

![Figure 1. Workstation used for online laboratory exercises. The board on the table is used in microcontroller experiments.](image)

Microcontrollers are described in weeks 6 through 8, beginning with a discussion of the similarities and dissimilarities between microcontrollers and microprocessors. A substantial amount of time is devoted to understanding and programming microcontrollers, first, because they’re popular and ubiquitous in low-cost applications and second, because they can give a better understanding of microprocessor organization and instruction sets. Programming is done in assembly because it gives a much better feel for what is actually happening inside the machine and a closer connection with the components of the internal architecture and organization.

In the initial lecture binary and hexadecimal number systems are reviewed. This is followed by the description of a generic microcontroller organization and the function of its internal components. Next, the programming model and memory map are described, followed by a discussion of some of the criteria that should be considered when selecting a microcontroller for a particular application.
In the next lecture a brief history and purpose of the CodeWarrior Integrated Design Environment (IDE) (available as freeware) is presented, together with a summary of commonly used programming languages. Assembly instructions, directives, labels, and comments are then described. An overview of the instruction set is presented. The instructions are broken down into a small number of categories that can be retained by the students. Some examples are instructions that move data, instructions that perform arithmetic, instructions that test or manipulate data, and instructions that control the program flow. A small subset of the instruction set (about 20 instructions) is defined that enable the student to write small illustrative programs that can be run online using microcontroller trainers in the Programmable Devices Laboratory.

The trainers used are the Dragon12-Plus Development Boards from Wytec, which are based on the Freescale HCS12 microcontroller. This board can be seen in front of the PLC trainer in Figure 1. Students can also download CodeWarrior to their PC so that they can test their homework programs in simulation mode.

The final microcontroller lecture covers microcontroller I/O ports and control registers. First, an overview of typical microcontroller I/O ports is presented. These ports include simple parallel digital I/O ports, analog-to-digital converters, and serial communications ports. Next, timing is discussed, beginning with simple delay loops and ending with a discussion of the timer module. This module contains circuitry for timing, counting, and pulse width modulation for generating complex waveforms. This lecture is followed by a set of laboratory exercises that demonstrate the use of parallel ports, timing delays, and the pulse width modulation module, which is used in a motor speed control exercise. For these exercises the web camera seen in Figure 1 is simply aimed at the Dragon12-Plus board. Figure 2 shows a screenshot of the host computer screen taken with a remote iPad using Mocha VNC (Business version).

In the final two lectures of the course, Field Programmable Gate Arrays (FPGAs) are presented. In the first week the motivation behind the development of FPGAs, their internal architectures, and the relationship between FPGAs and Application Specific Integrated Circuits (ASICs) is described. Typical applications are discussed, starting with the original application as simple glue logic and including ASIC prototyping, digital signal processing (DSP), embedded control, realization of physical layer protocols for high-speed communications, high-performance computing, and reconfigurable computing. Hardware Description Language (HDL) is introduced with several representative examples. The final week is devoted to learning how to program an FPGA, FPGA design flows, design tools, and key vendors. For the laboratory exercises the Dragon12-Plus microcontroller trainer is replaced with a BASYS 2 FPGA board from Digilent.

The prerequisite for the course is a standard undergraduate course in digital electronics with a minimum grade of C. This is also one of the requirements for admission to the program.

Due to the far-ranging nature of the subject matter no textbook is used. Instead, selected readings are assigned from websites such as ars technica. Bb Vista is used for course management.
Laboratory Component

The implementation of online laboratory exercises was one of the key challenges in the development of the course. VNC (Virtual Network Computing) is used to connect remotely to PCs in the lab that host the development software and control the hardware. Visual feedback is provided using a webcam attached to the host PC.

Remote access to the computers is provided using UltraVNC Server (freeware; available from http://www.uvnc.com); students can access the computers using the VNC client software of their choice. (We have experienced good results using the UltraVNC client software available from the same site.) Although Windows provides built-in remote access capabilities using Remote Desktop, we have found UltraVNC to be generally easier to use as well as more reliable. In addition, it does not require lockout of the local display and controls. This allows an instructor or TA at the local site to directly assist a student who is logged in remotely, as the instructor or TA could see the student’s actions on the screen and assist if desired by using the local keyboard and mouse. Combining this functionality with the webcam and voice communications using VOIP or standard telephones allows students to experience one-on-one instruction as needed, similar to a traditional brick-and-mortar lab setup where an instructor or TA could make the rounds assisting students.

Network-induced delay (“lag”) is sometimes noticeable and depends strongly on the type of Internet connection. With fast Internet connections such as 4G cellular or even standard DSL, typical lag appears to be on the order of a few hundred milliseconds—enough to be noticed but
not a major hindrance. Students accessing the Internet via inherently high-latency Internet connections such as two-way satellite links will likely experience worse delays. We’ve tested the system with a variety of platforms and networks, including an iPad with WiFi and 3G using Mocha VNC for Business and an Epic 4G smartphone running Android 2.1 over 4G using Android VNC. In practice the remote system works well, even with relatively slow connections. For the slowest connections we found that suitable performance could be achieved by changing the resolution of the web camera from 640x480 to 320x240. Reducing the color depth also improves the response time.

Although the VNC software allows students at remote locations to create and modify programs and download these to the PLC, Dragon12 trainer, or Spartan-3 board, there is no ability to control the physical inputs (e.g., switches, relays, etc.). Since interaction with these components is essential in order to test the functionality of the students' ladder-logic, assembly, or VHDL programs, a way of remotely interacting with the workstation was needed.

The solution we developed uses a "relay box," consisting of a PIC16F887 microcontroller and support circuitry to provide relay-switched inputs to the PLC, etc. The relay box is shown in Figure 3. An onscreen "virtual panel" is provided to allow students to switch inputs on and off at will, just as local students could on the physical switch panel. Figure 4 shows a remote screenshot of the host computer screen showing the virtual panel, taken from a MacBook Pro using the Chicken of the VNC application. The virtual panel runs on the site computer and interacts with the relay box via an RS232 connection. When a student clicks the onscreen switch to turn a input to the PLC, etc. on or off, the corresponding command is sent to the relay box, which implements the command. The intent is to make the remote student's experience mirror that of onsite students as much as possible; where an onsite student would turn a switch on or off, remote students can click on a "virtual switch" to accomplish the same thing. The relay box also monitors up to sixteen outputs from the PLC, microcontroller trainer, or FPGA board and displays these on both the virtual control panel as well as via physical LEDs on the relay box. These LEDs may be seen on the relay box front panel in Figure 3.

The relay box inputs are protected against overvoltage by a 1k current-limiting resistor and Zener diode combination. The outputs are controlled via socketed miniature relays driven directly by the PIC microcontroller, protecting the MCU from high voltages, ESD, and short circuit faults. A schematic of the relay box is shown in Figure 5.
Figure 3. Relay box used to interface the host computer to the physical I/O on the PLC, Dragon12 or Spartan-3 boards.

Figure 4. Remote screenshot of the host computer screen with the virtual panel used to control physical I/O.
Assessment

Student performance is evaluated using a number of in-course and end-of-course assessment tools including written examinations, homework, laboratory reports, and participation in online discussion groups. Student experience in the course is assessed using an online course evaluation at the end of the term.

One notable feature of the written examinations is the inclusion of an online hardware-based question. The student reserves one of the workstations during his or her scheduled examination time and is given a small project to realize using the PLC, microcontroller, or FPGA trainer. The student then submits a file containing the solution with the written part of the exam. The solution can then be tested on the machine as part of the grading process. The instructor or a teaching assistant may also monitor the student’s effort on the workstation in real time if desired.

Summary

This paper described a new online graduate engineering technology course in programmable devices. The course is intended to provide the graduate engineering technology student with an overview of programmable devices and systems used in industrial and embedded applications. A key feature of the course is the extensive use of web-based laboratory exercises in Programmable Logic Controllers, microcontrollers, and Field Programmable Gate Arrays. Industry standard development and modeling tools are introduced and used to develop representative control programs and to predict performance.

References