

AC 2009-71: EMBEDDED SYSTEMS DESIGN: RESPONDING TO THE CHALLENGE

Steven Barrett, University of Wyoming

Steven F. Barrett, Ph.D., P.E. received the BS Electronic Engineering Technology from the University of Nebraska at Omaha in 1979, the M.E.E.E. from the University of Idaho at Moscow in 1986, and the Ph.D. from The University of Texas at Austin in 1993. He was formally an active duty faculty member at the United States Air Force Academy, Colorado and is now an Associate Professor of Electrical and Computer Engineering, University of Wyoming. He is a member of IEEE (senior) and Tau Beta Pi (chief faculty advisor). His research interests include digital and analog image processing, computer-assisted laser surgery, and embedded controller systems. He is a registered Professional Engineer in Wyoming and Colorado. He co-wrote with Dr. Daniel Pack five textbooks on microcontrollers and embedded systems. In 2004, Barrett was named "Wyoming Professor of the Year" by the Carnegie Foundation for the Advancement of Teaching and in 2008 was the recipient of the National Society of Professional Engineers (NSPE) Professional Engineers in Higher Education, Engineering Education Excellence Award.

Jeffrey Anderson, University of Wyoming

Jeffrey R. Anderson received his PhD in electrical and computer engineering at the University of Wyoming with a research emphasis in image processing in 2004. He received his BS and ME from the University of Utah in 1989 and 1992, respectively. He has worked on a closed loop controller for mechanical ventilation of patients with adult respiratory distress syndrome. Additionally, he has worked on a servo lung simulator with related control method that was awarded a U.S. patent. Current research work is in segmentation of objects in serial sectioned images. He is an assistant academic professional lecturer at the University of Wyoming in the electrical and computer engineering department.

Jerry Hamann, University of Wyoming

Jerry C. Hamann received the B.S. in Electrical Engineering with a Bioengineering Option from the University of Wyoming in 1984. He then worked for the Loveland Instrument Division of Hewlett-Packard before returning to the University of Wyoming to complete the M.S. in Electrical Engineering in 1988. Sharing time as a lecturer and National Science Foundation Graduate Fellow, he completed the Ph.D. in Electrical Engineering at the University of Wisconsin in 1993. As a faculty member at the University of Wyoming since 1993, Jerry has pursued research interests in applied robotics and control, signal processing, and higher education teaching and learning. He is the former director of the University of Wyoming's Hewlett Foundation Engineering Schools of the West Initiative, which is focused upon enhancing the recruitment, retention and quality of undergraduate engineering students. He now serves as Head, Department of Computer Science.

Robert Kubichek, University of Wyoming

Robert F. Kubichek received his Ph.D. from the University of Wyoming in 1985. He has worked in research positions at the BDM Corporation and the Institute for Telecommunication Sciences (U.S. Dept. of Commerce), and was an adjunct professor at the University of Colorado from 1989-1991. He joined the University of Wyoming in 1991, where he is now an associate professor. Current research interests include remote sensing and speech analysis and processing applications.

Suresh Muknahallipatna, University of Wyoming

Suresh Muknahallipatna received his B. E. degree in Electrical Engineering and Master's of Engineering from the University of Bangalore, India, in 1988 and 1991, respectively. He

completed his Ph.D. degree at the University of Wyoming in 1995, with an emphasis on Neural Networks. He is a currently Associate Professor in the Dept. of ECE at the University of Wyoming. His current areas of expertise are performance analysis, modeling and simulations of storage area networks, mobile ad-hoc networks, and nano-satellite network.

John Pierre, University of Wyoming

John W. Pierre received the B.S. degree (1986) in EE with a minor in economics from Montana State University. He also received the M.S. degree (1989) in EE with a minor in statistics and the Ph.D. degree (1991) in EE from the University of Minnesota. He worked as an electrical design engineer at Tektronix before attending the University of Minnesota. Since 1992, he has been on the faculty in the Electrical and Computer Engineering Department at the University of Wyoming where he is currently a professor. He received UW's College of Engineering Graduate Teaching and Research Award in 2005. His research interests include signal processing education, statistical signal processing, system identification, and signal processing applications to power systems. He is an active member of IEEE and ASEE Societies.

David Whitman, University of Wyoming

David L. Whitman received the B.S. degree (1975) in EE from the University of Wyoming. He also received the Ph.D. degree (1978) in Mineral Engineering from the University of Wyoming. He worked in the synthetic fuels arena prior to becoming a faculty member in Petroleum Engineering at the University of Wyoming in 1981. From 1989 to 2005, he was the Associate Dean of Academics and since 2005 has been a professor of Electrical and Computer Engineering. He received UW's College of Engineering Outstanding Undergraduate Teaching Award in 1990 and 2004 and the ASEE Rocky Mountain Section Outstanding Teaching Award in 2001. He is currently the President-Elect of the National Council of Examiners for Engineers and Surveyors (NCEES) and an active member of IEEE and ASEE Societies.

Cameron Wright, University of Wyoming

Cameron H. G. Wright, Ph.D, P.E., is an Associate Professor with the Department of Electrical and Computer Engineering at the University of Wyoming, Laramie, WY. His research interests include signal and image processing, real-time embedded computer systems, biomedical instrumentation, and engineering education. He is a member of IEEE, ASEE, SPIE, BMES, NSPE, Tau Beta Pi, and Eta Kappa Nu. E-mail: c.h.g.wright@ieee.org Web page: <http://wwweng.uwyo.edu/electrical/faculty/wright.html>

Embedded Systems Design: Responding to the Challenge

Abstract

A recent *IEEE-USA Today's Engineer Online* article indicated that U.S. institutions of higher learning need to provide embedded systems design programs. The article cited that “only a few colleges and universities have good embedded systems programs in place.” The article further mentioned that “many engineers in embedded systems development are getting close to retirement age.” In this paper we will outline our program in embedded systems development including coursework in microcontrollers, digital signal processors, programmable logic controllers, and embedded systems design. We provide education on embedded systems concepts starting with middle school enhancement programs through graduate level coursework. A common thread through these programs is heavy emphasis on design. The curriculum allows a student to become an expert in embedded systems or allows them to selectively choose certain portions to enhance and augment their chosen area of expertise. Much of the design exercises and laboratories have been developed by student engineers. As a case study, we will discuss recent efforts to expand our course and laboratory coverage of programmable logic controllers (PLCs). We made these modifications in response to our constituents in the field who requested engineers be better prepared to design industrial systems controlled by PLCs.

Background

In the February 2008 *IEEE-USA Today's Engineer Online*, Mike Anderson, Chief Scientist of the PTR Group, Incorporated wrote a definitive article entitled, “Help Wanted: Embedded Engineers - Why the United States is losing its edge in embedded systems...[1]” An abbreviated version of the paper was later published in the *IEEE-USA Today's Engineer Digest* in March 2008 [2]. In the article, Anderson describes the prevalent use of embedded systems in a wide variety of industries and notes that on average the citizen of an industrialized nation comes into contact with over 48 embedded systems everyday [1].

To be considered an embedded system Anderson notes it must operate in a constrained resource environment, be economically viable, interact with users and the environment, contain both hardware and software firmware, and operate on low power devices [1]. Anderson further notes that common design characteristics of an embedded system are resource limitations, the application of low power processors, battery operation, limited memory and storage, and the use of many features at low cost [1]. In addition to these design characteristics, Anderson indicates that an embedded design engineer must also be comfortable with the concepts of thermal loading, the MIPS/watts ratio, an intimate knowledge of the target hardware to the register level, and the software interaction with the registers. Furthermore, the engineer must be aware of the overhead involved with the specific choice of software compiler employed to program the target system [1].

Anderson laments that the U.S. is facing an embedded system developer shortage crisis. He notes that it is not the number of engineering graduates that are of concern; however, the

shortage of engineers with a detailed understanding of embedded systems and related concepts. We too share Anderson's concerns. Through various feedback instruments our alumni have requested additional coursework in embedded systems design, and most recently coursework in programmable logic controller (PLC) systems design. A PLC is a modularized, self-contained industrial grade embedded system. It is pre-configured with industrial grade, buffered input and outputs for seamless interface to hardware common in the process control industry. Due to the modular design concept, a PLC based system can be easily expanded with multiple inputs and outputs and specialized subsystems.

We have developed an extensive embedded systems program at our university. The overall goal of our program is to establish an interest in embedded systems concepts as early as middle school. We follow up with programs for high school students. Once students formally enroll in the electrical or computer engineering major, they have a wide variety of embedded systems coursework to choose from. At a minimum, we require each of our majors to complete an Introduction to Electrical and Computer Engineering course containing an embedded systems module, a basic course in digital design, and a first course in microprocessor programming and interfacing. Students may then elect to complete a wide range of coursework to become an embedded system expert or they may selectively choose coursework from the program to augment their desired course of study.

In this paper, we discuss the component courses of the program. Each course provides a different aspect of embedded systems design. We place a heavy emphasis on practical, real world design and application in each course. All of the courses have heavy design content and several have an accompanying laboratory.

It is extremely costly in both time and resources to develop such a program. Throughout development we have extensively employed the engineering skills of our own undergraduate and graduate students to great success. We review the pro and cons of such a venture and use the development of a new course in Industrial Control as a case study to illustrate this process.

Methods

A. Introductory overview. In this section we describe in detail the components of our embedded systems curriculum. We have divided the curriculum into four different components as illustrated in Figure 1: Pre-College Enrichment Programs, required undergraduate coursework, elective undergraduate coursework, and elective graduate coursework. We will discuss each component in turn.

B. Pre-College Enrichment Programs. Three Pre-College Enrichment Programs based on embedded systems concepts are regularly taught at the university. Each of the programs is briefly described below. These pre-college programs are described in detail in [7]. A brief synopsis of each program is provided here for completeness.

B.1. “Computer Science, Engineering and Math (CSEM) Middle School Girls Camp. The CSEM Middle School Girls Camp offers a variety of learning experiences for girls who have completed 6th, 7th or 8th grade and have an interest in broadening their knowledge about

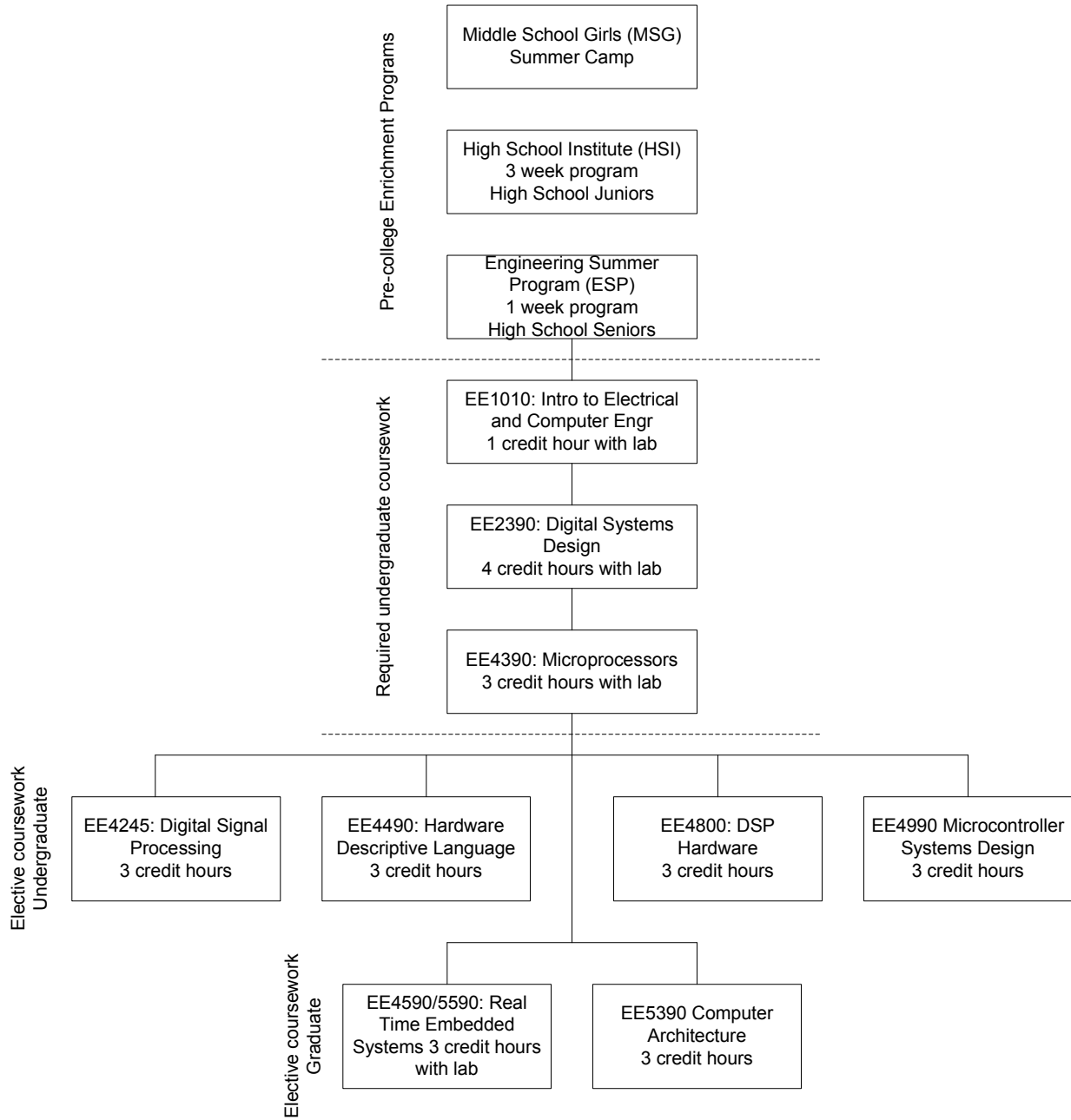


Figure 1. Embedded systems educational opportunities.

computer science, engineering and mathematics. The camp, directed by Professor Jerry Hamann of the Department of Electrical and Computer Engineering, is part of the Engineering Schools of

the West Grants Initiative provided by the William and Flora Hewlett Foundation. The camp focuses on educational, hands-on study to increase and maintain interest in science and mathematics. This one-week camp offers experimentation and exploration in the area of robotics, graphical programming, 3-D design prototyping, and applied mathematics. In addition, there are several social activities such as picnics, swimming, climbing and field trips to nearby engineering and natural resource sites. All activities are provided by faculty from the College of Engineering and College of Arts and Sciences with assistance from graduate and undergraduate students in the disciplines [Adapted from 4, reported in 7].”

B.2. “Summer High School Institute (HSI). The mission of HSI is to provide a place where some of Wyoming’s most intellectually talented high school sophomores can gather before their junior and senior years, living and studying in an environment with no pressure for grades, and sharing ideas and friendship with other gifted students. The primary purpose of the program is to annually draw 100 talented high school sophomore students to the university for an intensive examination of unanswered questions and unresolved challenges. Among the areas that are probed include: world hunger, plants and people, knights and cowboys, drama, ethics and society, communicating with computers, understanding cultural development, pharmacy, fundamentals of computer design and programming, and the links between life and the arts. The goal is not to require students to learn another body of knowledge and pass yet another test. It is, rather to challenge imaginations, focus diverse disciplines on specific issues or problems, and integrate various individual talents into a larger perspective. In the process it is hoped that the selected high school students achieve their academic and personal potential by cultivating their leadership capabilities; to expanding their horizons, developing their adaptability, creativity, and critical thinking abilities, and to heighten their sensitivity to future possibilities for themselves, Wyoming society; and to stimulate and reward excellence in Wyoming schools [Adapted from 5, reported in 7].”

B.3. “Engineering Summer Program (ESP). The College of Engineering and the Wyoming Engineering Society, in conjunction with the J. Kenneth & Pat Kennedy Endowment and the University of Wyoming College of Engineering Hewlett Foundation Engineering Schools of the West Initiative offer high school juniors an opportunity to participate in a summer program of hands-on experiences in various engineering fields. For example, students may design and build a digital circuit, study solutions to an environmental issue, test the aerodynamics of a tennis racket or model rocket, fabricate advanced composite materials, or design timber trusses. Laboratory sessions provide basic instruction and give students the opportunity to put new found knowledge to the test. ESP participants work one-on-one with faculty members and advanced students. This one week program is designed to expand student horizons, develop creative thinking and problem solving skills, and challenge imagination [Adapted from 6, reported in 7].”

In all three of the pre-college enrichment programs (MSG, HSI, and ESP) students are provided a series of hands on motivational exercises to learn the fundamentals of programming using Lego Mindstorm robots and also the fundamentals of digital circuit design. The computer control or robotics class that is offered as part of the High School Summer Institute (HSI) and the Engineering Summer Program (ESP) is based on the Lego MindStorm Education kits. The course focuses on programming the NXT (microprocessor) to solve various problems. The problems include sensing and following a black line, staying within a box and locating gray

mines, and transmitting data to other Lego Mindstorm NXT devices. This course is designed to build on itself were students will quickly build a set of commands that allow for solutions to difficult tasks. The programming platform used is called NXC (Not Exactly C). However, no prior programming experience is required. Fundamental programming constructs are taught and issues associated with microprocessors, such as data acquisition, digital numbers and logic are discussed.

C. Required Undergraduate Coursework. Every electrical and computer engineering major is required to take a minimum of three course containing embedded systems concepts. In this section we provide a brief review of the course content, objectives, and related design activities.

C.1. Introduction to Electrical and Computer Engineering

C.1.1. Course description: EE 1010. Introduction to Electrical and Computer Engineering.

1. Introduction to Electrical and Computer Engineering through a laboratory experience. Students perform both hardware and computer laboratory exercises in a wide range of areas of electrical and computer engineering [3].

C.1.2. Textbook (Optional): *Introduction to Electrical and Computer Engineering* by C.B. Fleddermann and M.D. Bradshaw, Prentice Hall, 2003

C.1.3. Course objectives and design activities: The Introduction to Electrical and Computer Engineering course is motivational in nature and exposes the students to a wide range of areas of electrical and computer engineering. This single credit hour course meets only one day a week for two hours, with a format that combines both lecture and experimentation. The course is divided into a series of one to three week modules, each covering a selected topic in electrical and computer engineering. Details about that course can be found in [8]. Two of the topics covered relate to imbedded systems. There is a two week module on real-time DSP using TI's TMS320C6717 chip. This module is discussed in detail in [9]. There is a three week module on Microprocessors using an HC11 to control a traffic light. That module is summarized here.

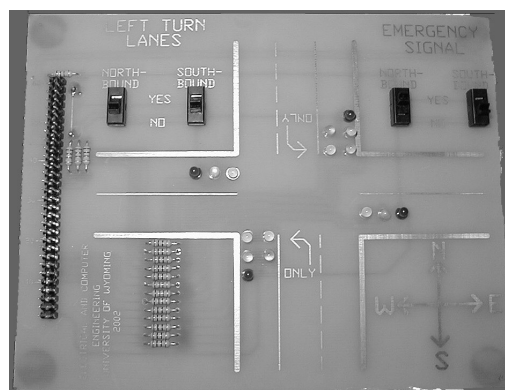


Figure 2. HC11 Controlled Traffic Light Board

The microprocessor module investigates an introductory-level understanding of microprocessor based systems. This module emphasizes that through a software program, one can control hardware. A Motorola MC68HC11 evaluation board is attached to a custom built traffic light board. This board, shown in Figure 2, represents traffic lights at an intersection, with green, yellow, and red LEDs for North/South and East/West traffic. The North/South direction also has LEDs for left-hand turn lanes. The boards have switches which represent a car in the turn lane and other switches which can be used for additional features. The Motorola evaluation boards are attached to the computer that has the ImageCraft ICC11 editing and compiling program for the HC11 microprocessor.

During the first week, an introduction is given to microprocessors, including a description of the ALU, registers, clock, RAM, ROM, and I/O. The initial experiment has the students use the HC11's Buffalo Bug software to manipulate port outputs. The traffic light LEDs are connected to the output ports of the microprocessors. Thus, the students directly control the lights by modifying the contents of the memory locations associated with the output ports. This helps the students see the connection between software and what happens in hardware. Then the lecture explains how programs can be written for the microprocessor and stored in on-chip RAM. The idea of a simple flow chart is introduced to alternate green and red lights between N/S and E/W. The second experiment has the students implement the flowchart in C code, compile the code, and download it to the microprocessor. The second and third weeks expand on the first. The students first implement a timed traffic light controller similar to that of a typical intersection. By the third week there is no lecture element, and the experiment exercise is very open ended. A feature list is provided to the students including adding turn signals, flashing traffic lights after midnight, timed turn lanes, and an emergency vehicle green light. Based on the end of semester student surveys, this module is the one most enjoyed by students but is also the most challenging.

C.2. Digital Systems Design

C.2.1. Course description: EE 2390. Digital Systems Design. 4. Binary logic, digital logic gates, reduction of Boolean expressions, combinational logic design. MSI and LSI combinational logic ICs, flip-flops, synchronous and asynchronous sequential systems design, MSI and LSI sequential system ICs, and algorithmic state machines. Prerequisite: MATH 2205: Calculus II [3].

C.2.2. Textbook: M. M. Mano and M. D. Ciletti, *Digital Design*. Prentice Hall, 4th ed., 2007.

C.2.3. Course objectives:

- Work effectively with a variety of number systems and numeric representations, including signed and unsigned binary, hexadecimal, 2's complement.
- Apply fundamental analysis skills to correctly describe the behavior of a given digital logic circuit.
- Translate system requirements into a practical digital design, making use of modern EDA tools such as schematic capture, Verilog HDL, and logic synthesis programs.

- Demonstrate hands-on test bench skills and the ability to communicate appropriately via a lab notebook while functioning as part of an engineering lab team.

C.2.4 Design activities: Students in EE 2390 encounter a variety of open-ended problems in which they must design some circuit or system to achieve the solution. The most obvious example is the final project in the laboratory portion of the course, in which they are given only the system requirements. They design a solution, build and test it using programmable logic (typically a CPLD) and some supporting peripheral devices, and must demonstrate it working to their TA. Other design activities take place in the lecture portion of the course, with a particular example being the final exam. The EE 2390 final exam provides the students with the detailed requirements for some embedded system such as the control units for railroad crossing gates, highway intersection traffic lights, or a building elevator. The students must come up with a design solution as part of the exam, and must also demonstrate the ability to make judgments about implementation tradeoffs and show proper emphasis on public safety.

C.3. Microprocessors

C.3.1. Course description: EE 4390. Microprocessors. 3. Design of microcomputers, controllers and instruments which use microprocessors. Semiconductor memory design, CPU architecture, bus structure and timing, input-output interfaces and devices, assembly language programming, assemblers, compilers, editors and simulators. Emphasizes design techniques. Laboratory. Prerequisite: EE 2390: Digital Systems Design [3].

C.3.2. Textbook: D.J. Pack and S.F. Barrett, *Microcontroller Theory and Applications HC12 and HCS12*, 2ed, Pearson Prentice Hall, 2008.

C.3.3. Course objectives:

- Design assembly language and C programs which utilize the built-in functional unit of a specified microcontroller.
- Demonstrate a working knowledge of the hardware components of a microcontroller.
- Implement and debug an interface between a specified microcontroller and other hardware.
- Demonstrate the ability to design, construct, test, and debug circuits in a laboratory environment.
- Apply appropriate design techniques to design circuits to given parameters.
- Evaluate, analyze, debug, and modify a given program to improve its execution of a specified task.
- Practice correct safety procedures in the laboratory environment.
- Practice team member skills in a laboratory environment.
- Demonstrate the capability design, implement, and document programs based on top down design, bottom up implementation techniques.
- Demonstrate the capability to properly document code.

C.3.4. Design activities: In the microprocessors course students begin with a series of fundamental laboratory exercises to learn the operation of specific microcontroller subsystems. As the course advances students are required to tie several of the subsystems together into more

complex systems. In response to our Department Advisory Board recommendation, we have added a team-based system design exercise as the final laboratory for the course. A team of 10-12 students are required to design, prototype, test, and document a multi-function weather station. The students are required to elect a team leader. The leader is responsible for working with other team members to partition the large project into sub-team size pieces. Each sub-team is responsible for one of the weather station's subsystems (temperature, wind direction, user interface, etc.) This project has proven to be very popular with the students and helps to partially fulfill ABET requirements for teaming exercises.

D. Elective Undergraduate Coursework. We provide a wide variety of undergraduate elective coursework. A student interested in pursuing a career in embedded systems could elect to complete all of the coursework. Students may also selectively choose from the available course offerings to enhance their chosen field of study. In this section we provide a brief review of the course content, objectives, and related design activities.

D.1. Digital Signal Processing

D.1.1. Course description: EE 4245. Digital Signal Processing. 3. Sampling and oversampling A/D's; FIR and IIR digital filter design, effects of quantization, practical realizations; applications of the discrete and fast Fourier Transform (DFT and FFT); correlation, periodograms, window effects, multi-rate techniques, multi-dimensional signal processing, and other topics in digital signal processing. Prerequisite: EE 3220: Signals and Systems [3].

D.1.2. Textbooks: B. Porat, *A Course in Digital Signal Processing*, Wiley, 1997.

D.1.3. Course Objectives:

- To review basic signals and signal processing. In depth coverage of sampling, quantization, correlation, and linear systems.
- To understand and implement basic spectral analysis using the DFT and FFT. Cover related applications including zero padding, fast convolution, and windowing effects.
- Review Z transform, and obtain a thorough understanding of classical FIR and IIR filter design. Be able to implement digital filters in software.
- Optional topics are changed from year to year. Most recently the DCT and its use in compression was studied. Additional topics included multi-path systems, inverse filters, and comb filters.

D.1.4. Design Activities:

Primarily, design content in this course is software based. Students are required to design FIR and IIR filters to meet given specifications in terms of required cut-off frequencies, pass-band ripple, and stop-band attenuation, while using minimum filter order. Students are introduced to the effects of finite-word length, which may result in instability, and learn how to design and implement more robust filter structures. Optional topics typically involve significant design

content. For example, recent students designed a DCT-based speech coder that minimizes output bandwidth while preserving as much voice quality as possible.

D.2. Hardware Descriptive Language (HDL) Digital Design

D.2.1. Course description: EE 4490. Hardware Descriptive Language (HDL) Digital Design.

3. Hardware Description Language design of digital systems. Industrial CAD tools are used to produce a functional description of hardware that is both simulated and then synthesized into hardware. Methods to describe both combinational logic and synchronous devices are given. Devices such as CPLDs and FPGAs are targeted in this design process. Emphasizes design techniques. Prerequisite: EE 2390: Digital Systems Design [3].

D.2.2. Textbooks:

Starter's Guide to Verilog 2001, Michael D. Ciletti (MC), Pearson Education, Inc, 2004, ISBN: 0-13-141556-5

Verilog HDL: A Guide to Digital Design and Synthesis, Samir Palnitkar (SP), Sun Microsystems Press – A Prentice Hall Title, 2003, second edition, ISBN: 0-13-044911-3

D.2.3. Course objectives:

- Demonstrate knowledge about the target hardware for an HDL
- Demonstrate an understanding of the fundamentals for an HDL
- Demonstrate an understanding and application of State Machine diagrams
- Implement digital design concepts from EE 2390 with an HDL
- Demonstrate an ability to take an algorithm from design specification to HDL implementation
- Demonstrate an ability to create testbenches to validate correct operation of HDL implementation
- Demonstrate an ability to correct and improve upon faulty HDL code
- Demonstrate the ability to use HDL tools correctly
- Be expected to submit assignments that clearly indicate an understanding of the lecture concepts, and provide HDL code that is easy to follow
- Do their own work, and submit design results that are consistent.

D.2.4. Design activities: In this course students are required to complete ten homework design exercises. The exercises are completed using the Xilinx webpack software. Students are required to demonstrate the correct operation of the design exercise with self-written testbenches. The proper testing and documentation of solid digital design techniques are emphasized throughout the course. As a final project small student teams are required to research a complex digital design of their choosing, establish project requirements, prototype and test a design solution, and provide a written and final report of their findings. Students find this exercise to be highly motivational (fun) because it allows them to explore an area of interest to them. Students have developed new video games, encryption and decryption modules, robot controllers, etc.

D.3. DSP Hardware

D.3.1. Course description: EE 4800. DSP Hardware. 3. The thrust of this course is to reinforce the enduring fundamentals of digital signal processing through hands-on implementation of practical DSP applications. This is accomplished through studying techniques of real-time DSP programming using TI TMS320C6713 DSK hardware development boards.

D.3.2. Textbook:

T.B. Welch, C.H.G. Wright, and M.G. Morrow, *Real-time digital signal processing: from Matlab to C with the TMS320C6x DSK*, CRC Press, 2006.

D.3.3. Course objectives:

- Study basics of DSP Hardware and its limitations. Learn how to code and debug programs using TI's Code Composer Studio environment. Understand sample-based processing verses interrupt-based processing.
- Review theory of FIR and IIR filters, and learn how to implement them efficiently in a real-time environment.
- Investigate hardware issues such as interrupts, direct memory access, and accessing external devices such as the on-board DIP switch bank and LED lights. Understand and utilize memory maps, boot procedures, and the multi-channel buffered serial port.
- Learn how to mitigate stability problems in finite word-length machines. Implement IIR filters as second-order stages in a real-time environment.
- Implement filters using frames with and without the enhanced DMA. Learn how to avoid edge effects.
- Explore and implement different methods for direct digital signal synthesis.
- Review the fast Fourier transform and implement it in real time. Use the FFT for fast convolution and filtering applications using the overlap-and-add method.

D.3.4. Design activities:

The course is not specifically design oriented. However, students are confronted with many problems where critical design decisions are required. As an example, initial projects require them to design FIR and IIR filters using Matlab, and figure out how to code the filters without breaking the real-time schedule. They determine a filter order that is high enough to meet specifications, but small enough that all computations can be completed within the available time slot. Additionally, they are constantly confronted with practical trade-offs throughout the course. For example, a high-order FIR filter gives better performance, but may not fit in the fast internal memory space. The coefficients can be stored in slow external memory, but now the real-time schedule is not met. Toward the end of the semester, students propose and implement a final project. These typically involve significant design content, involving both software and

occasionally hardware. For example, in a recent project students devised a digital “scratch machine.” This sound effect is typically achieved by manipulating an old vinyl record to abruptly speed it up or slow it down. The students designed a small motor-generator interfaced to the DSP, which they could manually spin like a record.

D.4. Microcontroller Systems Design

D.4.1. Course description: EE 4990. Microcontroller Systems Design. 3. Architecture and instruction set of Intel family of microprocessors; Intel System Development Kit and its monitor program; Microsoft Macro Assembler (MASM) and Visual C/C++ Express; modular programming; High level language compilers of object code; Interface design issues of peripheral devices to Personal Computer. Prerequisite: EE 4390: Microprocessors [3].

D.4.2. Textbooks:

Intel Microprocessors, Architecture, Programming, and Interfacing, Barry B. Brey, 8th Edition, Pearson Prentice Hall, 2009.

Assembly Language for Intel-Based Computers, Kip R. Irvine, 5th Edition, Pearson Prentice Hall, 2007.

D.4.3. Course objectives:

- Demonstrate an understanding of the architecture of Advanced Intel family of microprocessors.
- Demonstrate an understanding of real and protected mode of operation of Advanced Intel Microprocessors.
- Demonstrate the functions of each instruction with the Intel family of microprocessors.
- Demonstrate assembly programming using MASM and the use of Visual C/C++ Express with inline assembler and separate assembly language applications.
- Demonstrate basic memory and I/O interface of Pentium microprocessors.
- Demonstrate advanced I/O techniques that include interrupts and direct memory access.
- Demonstrate operation and programming for the Pentium 4 family of arithmetic coprocessors, as well as MMX instructions.
- Demonstrate interfacing of small systems to the personal computer through the use of parallel port, serial ports, ISA and the PCI bus interfaces.
- Introduction to Core2 microprocessors.

D.4.4. Course activities:

- Assembly language programs to demonstrate the working of Real-Address Mode of Intel family of microprocessors.
- Procedural programming in Assembly language using MASM.

- Advanced procedure development dealing with stack frames, recursion, and multimodule programs.
- Developing graphical windows application in assembly language.
- Interfacing a small LED system to a PC through the serial port.
- Program to demonstrate hardware interrupt handling.
- Programs to control PC Sound.

E. Elective Graduate Coursework. We also provide several graduate elective courses within the area of embedded systems. In this section we provide a brief review of the course content, objectives, and related design activities.

E.1. Computer Architecture

E.1.1 Course description: EE 5390. Computer Architecture. 3. Examines the various methodologies used in the design of high-performance computer systems. Topics include CISC and RISC architecture and instruction sets, pipelining, instruction-level parallelism, memory hierarchy (including cache) design and computer networks. Prerequisite: EE 4390: Microprocessors [3].

E.1.2. Textbook:

Logic and Computer Design Fundamentals, 4th edition, Mano and Keim, Pearson- Prentice Hall, 2008.

E.1.3. Course objectives: The objective of this course is for students to learn the fundamental concepts and components of computer architecture and design. Students study register transfer language, memory systems, instruction set architecture, RISC and CISC architectures, advanced memory concepts, and input/output processing with the ultimate objective of designing a functional processor in Verilog HDL.

E.1.4. Design activities: Students complete a series of design exercises using Verilog HDL. Students are required to design different subsystems of a given computer architecture. They are also required to work on a small design team on a final project of their choosing.

E.2. Real Time Embedded Systems

E.2.1. Course description: EE 4590/5590. Real Time Embedded Systems. 3. Emphasizes a systems approach to real time embedded systems. Students are expected to apply methodical system design practices to designing and implementing a microprocessor-based real time embedded system. Students employ a robot-based educational platform to learn the intricacies of real time embedded systems, distributed processing, and fuzzy logic. Students learn processor

input/output interfacing techniques. Students use state-of-the-art design and troubleshooting tools. Prerequisites: EE 4390: Microprocessors [3].

E.2.2. Textbooks:

Embedded Systems Design and Applications with the 68HC12 and HCS12,
S. Barrett and D. Pack, Pearson Education, Inc, 2005.

and

Microcontroller Theory and Applications HC12 & S12
D. Pack and S. Barrett, Prentice Education, Inc, 2008.
(students have this text already from EE4390 Microprocessors)

E.2.3. Course objectives:

Demonstrate knowledge about the target hardware for the microcontroller based system

- Demonstrate an understanding of advanced microcontroller concepts
- Demonstrate an understanding and application of hardware interface techniques
- Implement complex embedded systems employing multiple microcontroller subsystems
- Demonstrate an ability to take a design from concept to implementation
- Demonstrate an ability to apply proper testing techniques to validate correct operation of an embedded system
- Demonstrate an ability to correct and improve upon faulty microcontroller code
- Demonstrate the ability to use microcontroller programming and troubleshooting tools correctly
- Be expected to submit assignments that clearly indicate an understanding of the lecture concepts, and provide well-documented code that is easy to follow
- Do their own work, and submit design results that are consistent.

E.2.4. Design activities: Students complete a series of open-ended laboratory design exercises. Early in the semester students use the Freescale HC12 microcontroller to measure the gravitational constant using a drop tube instrument. They then must measure and control the speed of a DC motor equipped with an optical tachometer. The remaining laboratories are directed at equipping a robot platform to autonomously navigate through a maze by avoiding maze walls and “land mines.” The final laboratory is a maze competition to see which student robot can process through the maze in the least amount of time.

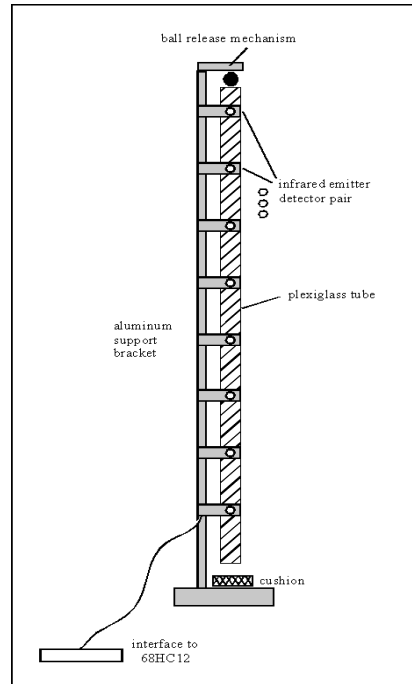


Figure 3. Drop tube instrument to measure the gravitational constant [reprinted from 11].

F. Coursework Development by Student Engineers

Engineering departments are often faced with the need to update laboratory exercises and equipment without adequate funds to do so. This is especially prevalent in an Embedded Systems based curriculum where processor technology and programming tools are rapidly changing. We have made a conscious effort to base new course developments on emerging technology and plan for a 5-6 year classroom lifetime [Adapted from 16].

Another challenge faced by departments is satisfying Accreditation Board for Engineering and Technology (ABET) Engineering Accreditation Commission (EAC) criteria for a major capstone design experience within the curriculum. ABET Criterion 4. Professional Component guidelines state, “Students must be prepared for engineering practice through the curriculum culminating in a major design experience based on the knowledge and skills acquired in earlier course work and incorporating engineering standards and multiple realistic constraints.” These constraints are further defined in Criterion 3. Program Outcomes and Assessment which states, “Engineering programs must demonstrate that their students attain: (c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability[10, adapted from 16].

We have solved these two challenges by updating our embedded system related coursework and laboratories on a limited budget using student engineers. We have successfully used student engineers in the past to develop custom laboratory equipment and teaching aids including a [adapted from 16]:

- Verilog HDL controlled robot [12],

- Labview based digital signal processing and bioinstrumentation laboratory program [13],
- Robot to teach complex real time embedded systems concepts [14],
- Visual simulator to teach real-time operating systems [15], and
- A teaching platform based on the Motorola/Freescale HC12 and HCS12 microcontrollers [16, 17].

We have reported on these success stories in prior ASEE Conference presentations and *Computers in Education Journal* articles [11-17]. As an example, a student designed and fabricated HC12-based microcontroller trainer was developed in house in 2002. Fifteen of the trainers were fabricated by a student led fabrication team during the Summer of 2002. It is worth mentioning that students also developed the laboratory exercises to support the course. We have used the trainers daily since that time for the laboratory portion of several embedded system courses. They have required minimal maintenance.

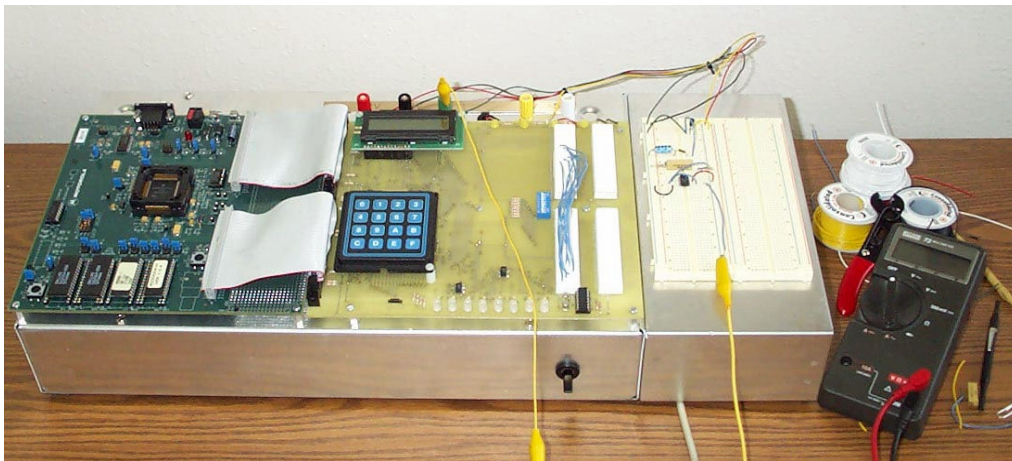


Figure 4. Student developed and fabricated laboratory trainer [reprinted from 16].

From the department's point of view, state-of-art, custom laboratory equipment was obtained at a lower cost than commercially available trainers. Furthermore, students were exposed to a real world design problem and all of the inherent related issues such as: working on a design team, interacting with highly skilled technicians, budget constraints, timelines, manufacturability issues, reliability issues, and customer satisfaction [10, adapted from 16].

G. EE5880 Industrial Control Case Study. To better illustrate the development of a new course employing student engineers, we use EE5880 Industrial Control as a case study. In Spring 2007 in response to alumni surveys, the department decided to develop a course in Industrial Controls. Early on it was decided to evenly divide course topics into advanced micro control concepts and programmable logic controllers (PLCs). Both microcontrollers and PLCs are used throughout industry to control machinery and processes.

To develop a course syllabus, the course director investigated a number of textbooks on these two topics and selected two of them: one based on the Atmel microcontroller and the other PLCs.

- *Fundamentals of Programmable Logic Controllers, Sensors, and Communications*, J. Stenerson, 3rd edition, Pearson/Prentice Hall, 2004.
- *Atmel AVR Primer: Programming and Interfacing*, S.F. Barrett and D. J. Pack, Morgan & Claypool Publishers, 2008.

A course syllabus and outline was then developed based on knowledgeable faculty in the area, a graduate student who had extensive prior experience in industrial control field work, and textbook content. The following course description and objectives were developed followed by a lesson-by-lesson outline.

Course description: Industrial Control. 3. This course emphasizes the control of industrial devices and processes using state-of-the-art programmable logic controllers (PLCs) and microcontrollers. We will investigate control algorithm design in detail and also discuss sensors, transducers, and interfacing. Students will also use state-of-the-art design and troubleshooting tools. Students will apply control theory to a series of hands on laboratory exercises.

Course Objectives: Students shall:

- Demonstrate knowledge about the target hardware for PLCs and microcontrollers.
- Demonstrate an understanding of the fundamentals of PLCs and microcontrollers.
- Demonstrate an understanding and application of UML Activity Diagrams.
- Implement top down design/bottom up implementation concepts.
- Demonstrate an ability to take a design from specification to implementation.
- Demonstrate an ability to troubleshoot and apply appropriate test procedures to validate correct operation.
- Demonstrate an ability to correct and improve upon faulty code and hardware.
- Demonstrate the ability to use compiler and support tools correctly
- Be expected to submit assignments that clearly indicate an understanding of the lecture concepts, and provide code that is properly documented and easy to follow.
- Do their own work, and submit design results that are consistent.

With a draft syllabus in place, the course director then enlisted the help of two different students to develop the laboratory exercises for the course. The students then worked over a period of months to develop the laboratory exercises for the course. The first four labs were developed to microcontroller based control while the remaining labs were developed for PLC based control. The following laboratory exercises were developed:

- Lab 1: Introduction to the Atmel ATmega16 - Rain Gauge Lab
- Lab 2: Temperature Sensor with LCD Lab
- Lab 3: Pulse Width Modulation – Automated Fan Cooling System
- Lab 4: Motor Speed Control with Optical Tachometer
- Lab 5: PLC Introduction Laboratory
- Lab 6: PLC Temperature Control Laboratory
- Lab 7: PLC Motor Control Laboratory
- Final Project: Small team based laboratory design project

Course and laboratory development funds were quite scarce. The course director was able to fund the laboratory portion of the course with a combination of funds from his home department, the department of Chemical and Petroleum Engineering, and a student led fund for engineering excellence. A modest laboratory program was implemented for approximately \$3000.

The course was first offered in Spring 2008 with 26 enrollees. Class success is provided in the Results and Discussion section below.

Midway through the first offering of the course, an alumnus who is a practicing expert in industrial control asked us to study the feasibility of an entire course on programmable logic based control. We are currently working closely with the alumnus to develop such a course with a supporting laboratory.

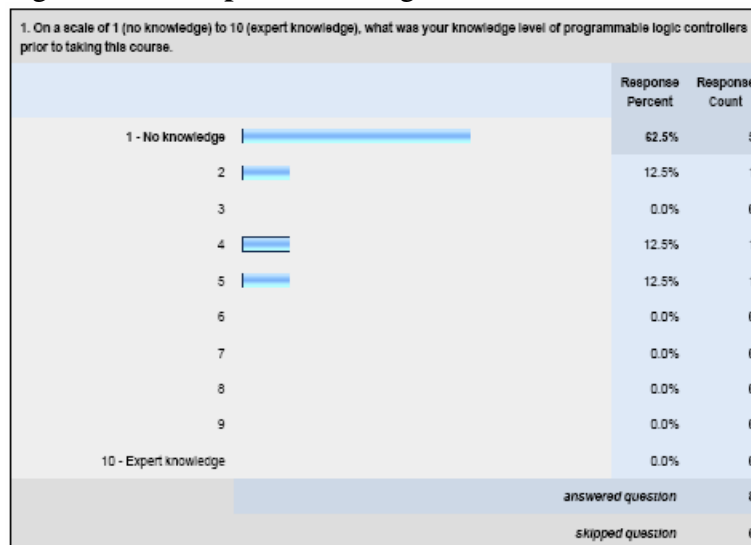
Results and Discussion

To measure the success of the Industrial Control course, we recently polled students who had completed the first offering of the course in Spring 2008. Students were asked a series of questions related to course structure and content. Eight of 26 students completing the course responded to the survey. At the time of the survey many had already graduated and were not available for comment.

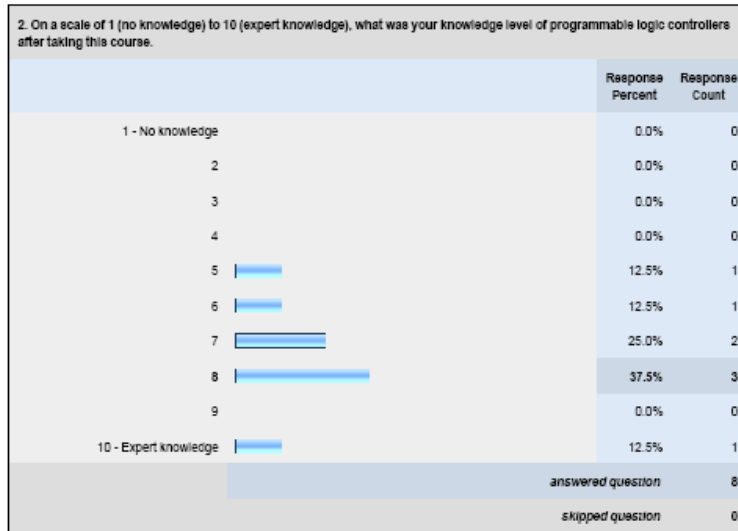
Survey Instrument and Results

In Spring 2008 you completed a new course EE5880 Industrial Control. In an effort to further improve course delivery and content we would like your candid opinion on this short survey. Your name will not be tied to your specific comments. Thank you for your help. Your opinion is vital to us and the continued success of this course.

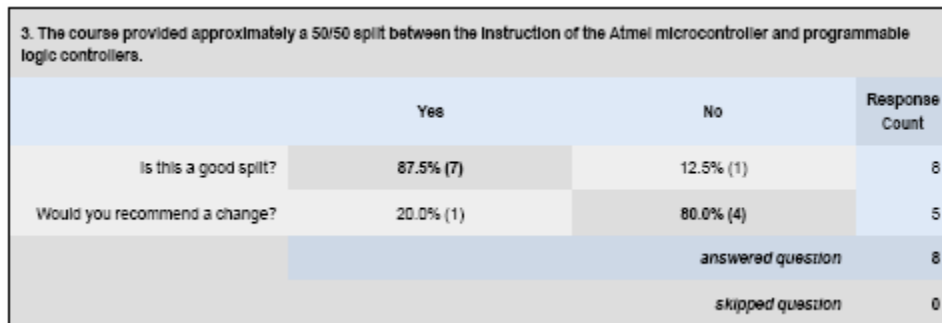
1. On a scale of 1 (no knowledge) to 10 (expert knowledge), what was your knowledge level of programmable logic controllers **prior** to taking this course.



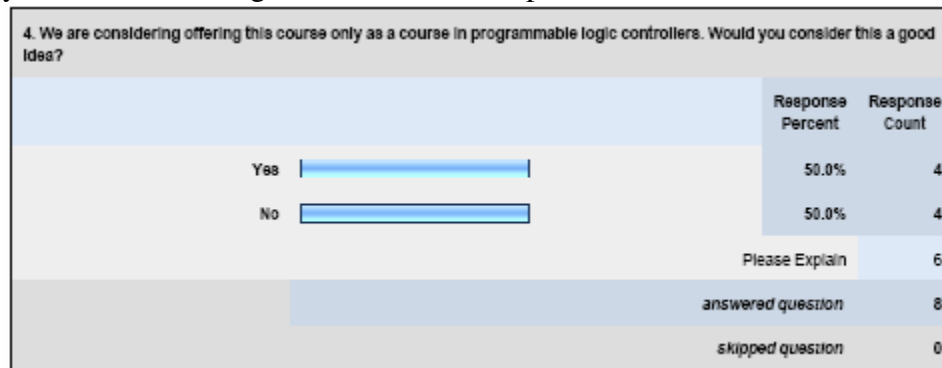
2. On a scale of 1 (no knowledge) to 10 (expert knowledge), what was your knowledge level of programmable logic controllers **after** taking this course.



3. The course provided approximately a 50/50 split between the instruction of the Atmel microcontroller and programmable logic controllers. Is this a good split? Would you recommend a change?



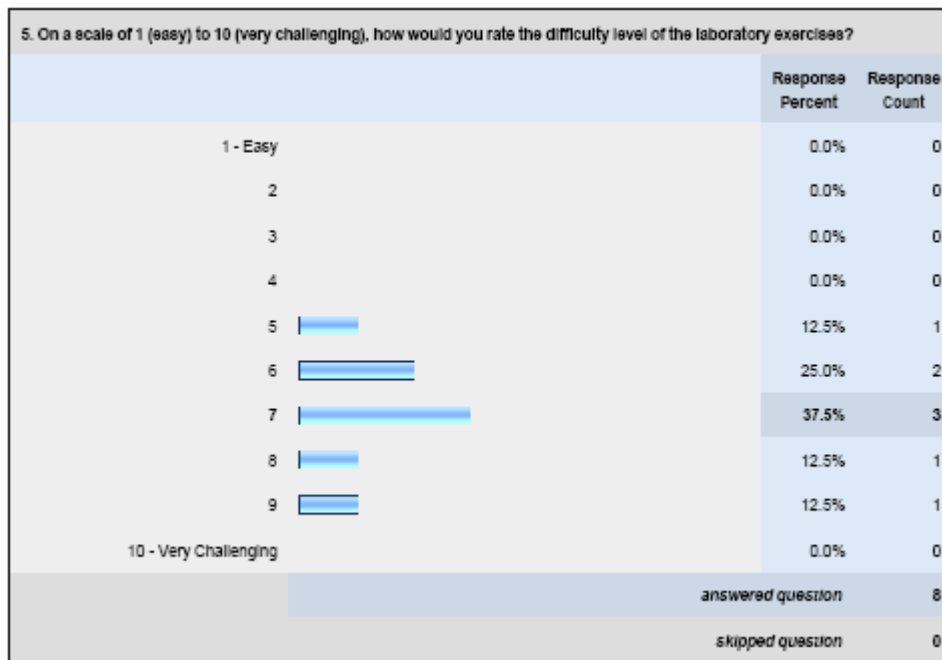
4. We are considering offering this course only as a course in programmable logic controllers. Would you consider this a good idea? Please explain.



Comments from students:

- PLC's are great and very important to those going into any industrial area. However, it's nice to have experience with more than just the HC12 from EE4390 but maybe that experience can come from another course.
- Only if you offer a separate Atmel course also
- The course should still include microcontrollers but more of the semester should be spent on PLCs
- Its good to get hands on experience with PLCs since lots of companies still use them
- PLCs are a great tool in industry; however, their design is intended for techs. An overview, in my opinion, was good enough.
- The Atmel microcontroller introduction give hands on for student and easy understanding, which I think is also very useful for courses such as EE4830 (Senior Design II) and personal knowledge. The transition between PLCs was at a good pace. May be just a bit more emphasis on PLCs but not the whole course on PLCs.

5. On a scale of 1 (easy) to 10 (very challenging), how would you rate the difficulty level of the laboratory exercises?



6. What improvements are necessary for the course?

- If you do go to a PLC only course I'd suggest using the donated PLCs (Allen Bradley) and the purchased one's (EATON) for some variety
- A little more focus on the PLCs would be nice
- This course was perfect. It was fun and challenging. Again it was perfect, please don't change anything. Dr. ____ did an excellent job on this course.
- More emphasis on PLCs and more challenging HW. Great course otherwise.

We were quite pleased with the students' candid and overall positive response toward the course and its contents. We are currently working on the second offering of the course and will take their valuable opinion into consideration as we decide on specific course content.

Conclusions

In his article "Help Wanted: Embedded Engineers," Mike Anderson has issued a challenge to U.S. universities to improve their coursework in embedded systems design. We believe that we have a solid program in embedded systems. It has taken some time to develop the program to reach its current state. The development of new courses and laboratory exercises would not have been possible without the expertise of our student engineers. The student engineers are provided Independent Study credit for their work on these projects. More importantly, they gain practical real world experience working on challenging design projects.

The area of embedded systems rapidly changes due to new evolving technologies. It is a tremendous challenge to keep courses up-to-date. As we look forward to course innovations in the next several semesters, we are working on the following efforts:

- A new set a laboratory exercises for the EE2390 Digital Systems Design course including a large team exercise,
- Several new laboratory exercises for the EE4390 Microprocessors course one of which will be a large team exercise,
- A revised second offering of our Industrial Controls course,
- The concept of Zigbee wireless, networked sensors introduced into the EE4590/5590 Real Time Embedded Systems course, and
- A new course that combines microcontroller technology and hardware descriptive language concepts in the same course.

Acknowledgments

We acknowledge the tremendous support and dedication of our student engineers who have made this curriculum possible. Their names are provided as bold print co-authors in the papers listed in the Bibliography.

We also extend our appreciation to Bruce and Carla Pivik of Infinity Power and Control, Rock Springs, Wyoming for their gracious support of our Industrial Control curriculum and laboratory.

Bibliography (Note: student co-authors are highlighted in bold print)

1. *IEEE-USA Today's Engineer Online*, Feb 2008,
http://www.todaysengineer.org/archive/date.asp?long_dat=Feb+08
2. *IEEE-USA Today's Engineer Digest*, March 2008
3. University of Wyoming, General Bulletin 2008-2009, Office of the Registrar, Laramie, WY. 82071
4. J. Hamann, N. Peck, L. Hutchison, S. Steadman, and D. Coon, "The University of Wyoming Center for Excellence in Engineering Education: Recruiting, Retaining and Educating the Best in the West," Executive Summary for the William and Flora Hewlett Foundation, December, 2002.
5. High School Institute, <http://uwadmnweb.uwyo.edu/Provost/hsi/>
6. Engineering Science Program, <http://wwweng.uwyo.edu/highschool/esp/>
7. S.F. Barrett, J. Hamann, D. Coon, P. Crips, J. Pierre, "Show them NAND Gates and They Will Come," *Computers in Education Journal*, Vol. XVII, No. 2, 26-36, Apr-Jun, 2007.
8. J.W. Pierre, F.K. Tuffner, J.R. Anderson, D.L. Whitman, A.H.M.S. Ula, R.F. Kubichek, C.H.G. Wright, S.F. Barrett, J.J. Cupal, and J.C. Hamann, "A One-credit Hour Hands-On Introductory Course in Electrical and Computer Engineering using a Variety of Topic Modules," *IEEE Transactions on Education*, May 2009.
9. J.W. Pierre, F.K. Tuffner, R.F. Kubichek, and C.H.G. Wright, "Real-Time DSP in a Freshman DSP Course," *Proceedings of the IEEE 12th DSP & 5th SPE Workshop*, Marco Island, Florida, January 2009.
10. Criteria for Accrediting Engineering Programs, Effective for Evaluations During the 2007-2008 Accreditation Cycle, ABET Engineering Accreditation Commission, Baltimore, Maryland, March 18, 2007.
11. S. F. Barrett, **A. Wells**, **C. Hernandez**, **T. Dibble**, **Y. Shi**, **T. Schei**, **J. Werbelow**, J. Cupal, L. Sircin, G. Janack, "Undergraduate Engineers for Curriculum and Laboratory Equipment Development," *American Society for Engineering Educators 2003 Annual Conference*, Nashville, TN, June 2003.
12. **Griffith**, S. F. Barrett, D. Pack, "Verilog HDL Controlled Robot For Teaching Complex Systems Design," *Computers in Education Journal*, accepted, in press.
13. C.H.G. Wright, **D. Mares**, S.F. Barrett, T. Welch, "Digital Signal Processing and Bioinstrumentation Using Labview, the New ELVIS Benchtop Platform, and BIOPAC," *Computers in Education Journal*, Vol. XVII, No. 2, 104-112, Apr-Jun, 2007.
14. S. F. Barrett, D.J. Pack, **P. Beavis**, **M. Sardar**, **A. Griffith**, L. Sircin, G. Janack, "Using Robots to Teach Complex Real Time Embedded Systems Concepts," *Computers in Education Journal*, Oct-Dec, 2006.
15. S. F. Barrett, D.J. Pack, **C. Straley**, L. Sircin, G. Janack, "Real-Time Operating Systems: A Visual Simulator," *Computers in Education Journal*, April – June 2005.
16. S.F. Barrett, **A. Wells**, **C. Hernandez**, **T. Dibble**, **Y. Shi**, **T. Schei**, **J. Werbelow**, J. Cupal, L. Sircin, G. Janack, "Undergraduate Engineers for Curriculum and Laboratory Equipment Development," *Computers in Education Journal*, Vol. XIII, No. 4, 2003, 46-58.
17. S. Barrett, **C. Hager**, **M. Yurkoski**, **R. Lewis**, **M. Jespersen**, **Z. Rubel**, "Undergraduate Engineers for Curriculum and Laboratory Equipment Development: A Freescale S12 Microcontroller Trainer," *Computers in Education Journal*, June 2008, October – December 2008, 22-32.