Designing a Microcontroller Training Platform for Active Distance Learning Engineering and Technology Students

Steve C. Hsiung
Old Dominion University, shsiung@odu.edu

James E. Eiland

John R. Hackworth

John Ritz
Old Dominion University, jritz@odu.edu

Original Publication Citation

Hsiung, S., Eiland, J., Hackworth, J., & Ritz, J. (2009). Designing a microcontroller training platform for active distance learning engineering and technology students. *Journal of Engineering Technology*, *26*(2), 34-44.

Designing a Microcontroller Training Platform for Active Distance Learning Engineering and Technology Students

Steve C. Hsiung, James E. Eiland, John R. Hackworth, and John M. Ritz

Abstract

This is an active distance-learning project that addresses the hands-on microprocessor/ microcontroller-related courses. A research team designed a low-cost training system with supporting instructional materials to assist the teaching of these concepts. Individual laboratory activities are being developed to reinforce student learning and skill development in programming concepts. This basic system format eventually will support an array of engineering and technology courses. This project involves two community colleges, Blue Ridge Community College (BRCC) and Olympic College (OC), and a four-year university, Old Dominion University (ODU), in a collaborative research team to design and develop a specific PIC microcontroller training system with custom-designed software and curriculum materials to support related engineering technology courses. The functions of the hardware and software cover different areas of engineering technology courses and majors to maximize the use of the microcontroller training system.

1. Introduction

Microcontrollers have become ubiquitous helpers in our daily lives. They are compact, single-purpose computers running embedded application software that are widely utilized in modern electrical devices and systems to control operations, such as temperature settings of ovens, remote control of television sets, or extended features of cell phones. Now automobile mechanics must work with microcontrollers to control fuel mixtures and ignition timing. Because microcontrollers are so important to our high-tech world, demand is high for workers trained to design, maintain, and embed them into products. But many people who want the training cannot take time away from work or families to enroll in engineering and/or technology programs on university campuses.

Digital electronics and microprocessors/microcontrollers are a major component of the high-tech world and important subjects in Electrical Engineering Technology (EET) and related technology curricula. To educate students in these fields and accommodate the growing needs in distance learning, the methods of delivering these educational materials must be changed. Studies show the obstacles in delivering hands-on education in distance learning environments[1], but all issues can be resolved with modified instructional strategies. Currently, most of the solutions to laboratory-related courses in distance learning are to use computer simulations and sometimes Internet virtual labs, which have fundamental difficulties in solving this issue[2]. For example, the circuit design, testing, implementation, debugging, and performance checking can not be covered by pure use of software simulations and virtual laboratories[1,3]. In addition, the cost of doing all the learning exercises and experimentation is another issue for instructors and students.

The design and implementation of a microcontroller training system for hands-on distance-learning projects can provide opportunities to students in rural and urban areas to learn current technology concepts and become prepared to qualify for high-tech jobs. Following is a description of a microcontroller training system including hardware and software designs, and their implementation using active distance-learning instruction.

2. Active and Passive Distance Learning

In recent years distance learning has become an increasingly common way to reach place-bound students. There are two types of systems for the distance delivery of teaching materials. *Passive distance learning*, the most common method, makes teaching materials or simulation software available online via a computer server using Blackboard™/WebCT™ or special licensed software or shareware. In a similar manner the delivery of teaching material could use a CD/DVD/Flash Drive/iPod, which makes material available to the students all the time, and it does not need a constant Internet connection. Students can gain access to the necessary material anywhere and anytime. Mentoring, consulting, and questions and answers rely heavily on e-mail and discussion boards, and there is very limited real-time interaction between students and instructors.

In the other delivery method, called *active distance learning*, real-time interaction between students and instructors occurs much the same as in a regular classroom, but the students are not in face-to-face contact with instructors. It is still accessible to students anywhere anytime, and if students miss the real-time session, there is an archive for later viewing or studying. This implementation can involve a televised connection via satellite, video streaming via Internet, or a live Web camera lecture/discussion via Internet using Acrobat Connect, Skype, MS Messenger, Yugma, Autodesk, etc. It involves all the activities included in a conventional classroom except the face-to-face student-instructor interaction. The developed microcontroller training platform described herein is specifically designed for hands-on laboratory teaching materials using an *active distance learning* environment for engineering and technology students in a very cost-effective way.

3. The Needs of a Hands-on Microcontroller Training Platform

According to Gokhale's[3] 2004 findings, the effective integration of computer simulations into lecture-lab activities enhances the understanding and performance of students. From the findings of Michael[1], the use of computer simulations to enhance product creativity was not supported. To simply apply computer simulations in distance-learning classes will not be effective to support students' understanding. There must be an association with hands-on experiments or laboratory activities to achieve the maximum learning results, a key in understanding engineering concepts.

There are courses that can easily fit into the distance-learning format, but there are also curricula that have fundamental difficulties in offering the course material online. The most common problems are the courses that require hands-on laboratory experimentations/exercises and their associated high costs, such as those offered in the Engineering Technology (ET) areas. The implementations of virtual labs in which students can remotely log onto and control the laboratory equipment to do the needed exercises via the Internet will solve some of the learning difficulties but they have limitations. Especially, when considering tests, experiments on real circuits and software designs, troubleshooting, and debugging in the microcontroller control-related material, it is a major obstacle to students in understanding the concepts in remote locations. Without letting students actually build the circuits and test their designed software on real hardware set-ups, it is very difficult to visualize the course instructional materials. Without a common training system platform, it has been increasingly difficult for teachers to guide and assist students in troubleshooting their circuits/systems and give them proper suggestions or answers to their problems in a remote environment. On the average, this significantly increases the time required to assist students performing laboratory work online when compared to students taking the course on campus. The distance learning students spend more time in understanding the course materials, since they can only obtain help from teachers via Internet postings, chats, or e-mail discussions.

The cost of learning microprocessor/microcontroller applications is another major issue for implementing ET distance-learning programs because students usually need to purchase parts and equipment themselves to meet the course requirements. This significantly increases the cost of the hands-on courses; oftentimes these financial burdens force potential students to have second thoughts in selecting the major. Converting the course to computer simulation will reduce these costs, but the authors consider hands-on experiences to be vital to the success of microprocessor-related laboratory courses. The project idea is to make the training system with associated instructional modules available to students to buy in place of a textbook and parts list for purchasing through a bookstore or vendor arrangement.

There is currently a wide variety of PIC programmers and in-circuit emulators (ICE) available on the market. The Easy PIC development tool made by Midro-Elektronika[4] (an East European company) uses the PIC16877, which has inputs, outputs, keys, an LCD, and LED display for $149.00. The PIC TUTOR made by AMS[5] (a British company) has a keypad, LEDs, and an LCD display for $340.00. The MR1-MC-05 PIC Emulator made by MCPros[6] (a U.S. company) is a full ICE (when the POD[6] and adaptor modules are added) and costs from $1,200.00 to $2,400.00. The BASIC Stamp made by Parallax, Inc.[7] provides software and hardware in various module configurations for microcontroller applications in a BASIC programming environment. The cost of the individual module is attractive and within a range from $30.00 to $100.00. Implementing all the needed laboratory exercises and projects requires different modules from the BASIC Stamp and the total price becomes unaffordable to most distance-learning courses. Microchip[8], the manufacturer of PIC microcontrollers, has development systems ranging from $199.00 to $2,560.00. These systems are all designed for professional applications, debugging, and simulation. Most of their applications are single purpose and of limited educational value for classroom and laboratory exercises which require multiple functions for various learning experiences. They were not developed on the basis of academic needs and are not suitable especially for a distance-learning environment.

The research of the available training systems proves they are either too expensive for the distance learning student to own or the design focuses are not suitable for distance-learning applications. This microcontroller training platform described here is designed to address these active hands-on distance learning problems and associated cost issues. Use of this training platform, with a mix of Internet-based real-time audio, video, e-mail, chat, conference meeting, and individual consultation software can make the distanced students' learning experience equivalent to that of their on-campus counterparts.

4. Project Objectives, Curriculum, and Hardware Design

The research team transformed its ideas for this learning system into project objectives.These included:

1. **Training System Development:** Design and develop the hardware and software for a training system board that uses PIC medium family members, such as PIC16F84A[9], PIC16F88[10], and PIC16F877A[11], for two-year and four-year institutions in the areas of digital, microprocessor/ microcontroller, automation control, and senior project courses to directly resolve the problems of cost and learning from a distance.
2. **High- and Low-Level Programming Languages:** This system will serve as a common platform for high- and low-level software programming design, hardware circuit trouble-shooting, evaluation, and final project control.
3. **Hardware Modules and Components:** The training system will be designed with many basic modules such as power supply, input/output switches, keypad, interrupt inputs, LED outputs, LCD display, serial interface, parallel interface, PC communication interface, high-power motor driver, sensing, etc. It also will have the flexibility to accept advanced module connections for future expansions.

To fulfill the design requirements and align them to course curriculum needs in two-year and four-year institutions, an extensive collaboration is needed. Research and meetings between the institutions' faculty have been ongoing since May of 2007 in order to reach a common consensus of what are the essential elements in this training system. Researchers believed that courses taught at two-year colleges should be transferable to a four-year university. Table 1 lists course topic comparisons that were summarized from the related course syllabi of the three institutions involved in this project.

Table 1. Microcontroller/microprocessor-related course topics.

Topic	BRCC	OC	ODU	Project
Introductory Concepts/ Number Systems	X	X	X	X
Safety	X	X	X	X
Logic Gates Arithmetic	X	X	X	X
Logic Circuits	X	X	X	X
Memory Devices	X			
Microprocessor Architecture	X	X	X	X
Programming	X	X	X	X
Counters/Time Delays	X	X	X	X
Subroutines	X		X	X
Interfacing	X		X	X
PIC Microcontrollers	X	X	X	X
Pro Basic	X			
Digital Indicators	X			
LCD Modules	X		X	X
Data Memory	X			
Analog to Digital	X	X	X	X
Data Transmission	X	X	X	X
Sensors	X		X	X
Signal Conditioning	X		X	X
Instrumentation	X			
Computer Languages		X	X	
Internal/External		X	X	
Assembly Languages	X	X	X	X
Flowcharting	X	X	X	X
Input/Output	X	X	X	X
Peripherals		X	X	
Hardware Installation		X		
Series Communication	X	X	X	X
Troubleshooting	X	X	X	
Input/Output Ports	X	X	X	X
LED Display	X	X	X	X
Multiplexed Keypad		X	X	X
Circuit Board Fabrication		X		
PIC Addressing Modes			X	X
Interrupts Handling			X	X
Serial Interfacing			X	X
Peripheral Interfacing			X	X
Design Projects			X	
High-Precision Math Routines			X	
Number Conversion Routines			X	
PIC/PC Integration			X	

Due to the differences in program design and institution mission, an extensive information exchange was used to help the team members to reach consensuses on curriculum modules. After many discussions and exchanges of experiences between the design team members, a common list of instructional topics was developed. The design team members elected the following units to develop the course curriculum modules that can be integrated into various courses:

SESSION # **CONTENT**

0. **Microcontroller Technology:**
A Brief History of Microprocessor Develoment, Differences between Microcontrollers and Microprocessors, Microcontroller Applications, Microprocessor Architectures, Memory Types, Microcontroller Packaging/Appearance, PIC16F84A, PIC16F88, and PIC16F877A Memories
1. **Gates, Number Systems & PIC Environment:** Different Number Systems, Number System Conversions, Logic Gates, Logic Arithmetic (Add & Subtract), Header File and Source Codes, The Environment and Software Operations, and Header File and Source Codes
2. **PIC Instructions in Assembly Language Programming:**
Assemble Language Format, PIC Instruction Sets and Registers, 16F84A, 16F88, and 16F877A Internal Blocks and DRAM Distributions, C, Z, and DC Flags in STATUS Register, Setting and Clearing Bits, Logic and Math Operations, and Addressing Modes
3. **I/O Interface:**
PIC Embedded System Designs, Use of Internal Oscillator and External Resonator, Ports Configuration, I/O Port Interface, DIP Switches Inputs, LED Controls, and 7-Segment Interface
4. **Assembly Language Software Designs:**
Programming Controls, Flowcharts, Counters, Loops, Time Delays, Subroutines, DRAM Memory Banks, and PRAM Memory Pages
5. **The Uses of WDT:**
CONFIG Register Configuration, Watchdog (WDT) Configuration, Controls, and Applications
6. **The Uses of IRQs:**
Source of Interrupts, Flags and Enable Setup, Interrupts Handler, IRQ Configuration, Polling vs. IRQ, IRQ Service Routines, Prioritize IRQ Services, and Multitask Applications
7. **Parallel Data Communication:**
Parallel Interface, Data Transmission Protocol, Long and Short Table Lookup Implementations, and LCD Module Interface
8. **3*4 or 4*4 Matrix Keypad:**
Software Debounce Designs, Key Decoding Designs, Matrix Keypad Interface Designs, Interface Software Design, and Testing and Verification
9. **Stepper Motors:**
Unipolar and Bipolar Stepper Motors, Stepper Motors Interface, H-Bridge, Driver, Speed, and Direction Designs/Controls
10. **DC Motors:**
H Bridges Controls, DC Motors Interface, Driver, Speed, and Direction Designs/Controls, and PWM Controls
11. **The Uses of ADC Block in the PIC:**
Analog-to-Digital Conversion Configuration, Software Design, and Controls
12. **The Uses of Sensors & Signal Condition Circuits:**
Sensors Designs and Interface, Signal Condition Circuit Designs, 16 Bit Math Routines Designs, and ADC and DAC Applications

5. PIC Training System Hardware and Software

After finalizing the session topics, work began on the hardware design. The initial goal was to design a hardware circuit that will both enable PIC microcontroller programming and provide limited debugging functions. Information required to develop the PIC microcontroller programming of the PIC flash and EEPROM memories was obtained from the Microchip Website [9,10,11]. Designs of the PC parallel/printer port hardware and software used to program the PIC microcontroller flash and EEPROM memory are widely available on the Internet, and these were modified for use in this project by the design team members[14]. However, due to the limited availability of PC parallel ports on newer computers (particularly laptop computers), it was determined that a USB programming port would also be needed. Nevertheless, available software in the public domain using a USB port is limited, mainly because hardware designs vary and the accompanying software differs for each design. This posed a challenge to the project team.

To provide limited debugging functions on a PIC processor, an understanding of the "Background Debugger Control" and the "On-Chip Debugger" specifications is essential, but there is lack of sufficient documentation of these materials[16]. After consultation with Microchip Inc., it was found that full documentation of the debugger routines is usually not available to the general public and is only shared with Microchip's affiliated third-party tool development companies. Following extensive research, trial and error, and additional consultation with Microchip design engineers, it was their suggestion that the best approach would be to use the available Microchip public domain software.

It was decided that Microchip's "PICKT2" hardware and software architecture would be followed for the design of this development system[17]. In implementing this scheme, the system would be designed around the

"PICKT2" USB communication criteria, thereby using a dedicated PIC18F2550.

To be able to better communicate with the project team members and to create clear and effective documentation, hardware blocks were used to initiate different design ideas. The hardware block design is also aimed toward better links in fulfilling the needs of the curriculum sessions listed earlier. The core circuit design, shown in Block #1 of Figure 1, was tested and verified with the "PICKIT2" software. After three revisions of the hardware functional blocks that were mutually agreeable to the participating colleges, a final circuit design was completed. Figure 1 shows a block diagram representing the training system hardware.

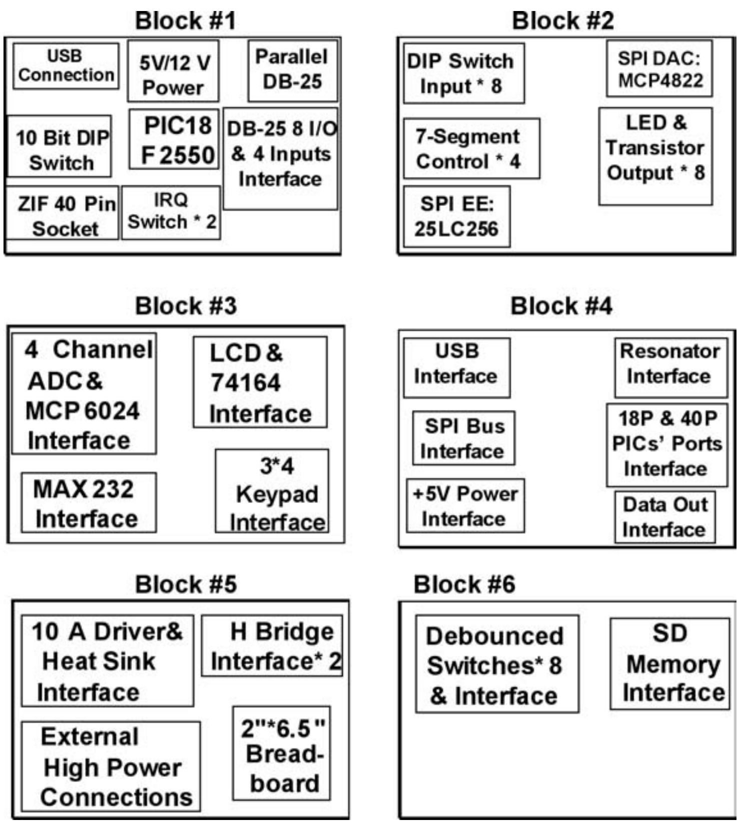

Figure 1. PIC microcontroller training system PCB function blocks.

6. The Circuit Designs

The circuit design of Block #1 meets the "PICKIT2" software requirement, i.e., the core of the training system communication to a PC via the USB port. There is a DB25 parallel port that serves as an alternate communication port, and it provides additional high-level language programming for different curriculum exercise needs. Having two communication ports created contention issues, should both the USB and parallel ports be connected simultaneously. To resolve this potential problem an analog/digital bilateral switch, a CD4066, is used to sense the presence of a parallel port connection and automatically disable the USB. Each of the two communication ports operates with its own unique software. The USB port uses "PICKIT2" software and the DB25 parallel port uses "ICPROG" software18. Additionally, the DB25 parallel port can also be used for high-level language programming controls in C or C++ running Microsoft Visual Studio15. Figure 2 presents the detailed Block #1 circuit design.

Not all PIC families share the same footprint. To accommodate this difference, a PIC package selection switch, a ten-position DIP switch, is provided to select different PIC microcontroller package family members. This allows the system to be able to accommodate an 18-pin, 28-pin, or 40-pin package PIC. The power supply circuit in Figure 2 provides +5V for normal operation and +12V for programming purposes.

In this training system design, it is not necessary to move the PIC from the programming socket (40-Pin ZIF) to a different IC socket for custom-designed code evaluation/testing. A PROG_OPER_SW that allows the PIC to be programmed and tested in the same ZIF socket is provided. Due to the limited available footprint package[19], the same DIP switch and symbol are used in the circuit but the parts are actually two-position switches packed in a DIP format that serves as program/operation functions. Two SPDT switches are intended for IRQ (Interrupt Request) exercises.

The hardware circuit for Block #2 is shown in Figure 3. Four seven-segment LED displays with current limit resistors and driver transistors are used for experiments involving numeric displays. There are eight general-purpose digital inputs provided by an eight-pole DIP switch and corresponding 10K pull-up resistors. The eight general-purpose LED indicators with transistor drivers are for digital output functions, a 32K byte EEPROM[20] is for external data storage, and a dual-channel DAC[21] is ready for data conversion uses. The EEPROM and DAC are configured with an SPI bus interface and are intended for serial communication laboratory experiments.

The circuit design for hardware Block #3 has an RS232 conversion IC[22] which can be used for any TTL to RS232 signal conversion and four general-purpose OPAmps for signal conditioning design. A jumper selectable OPAmp power supply voltage of +5V or +12V is provided for different OPAmp operational requirements. Additionally, a separate 20MHz resonator is added for those PIC family members such as the PIC 16F84A[9] and PIC 16F877A[11] that do not have an internal clock signal oscillator. These are all presented in Figure 4.

Block #4 (not presented in schematic format here) contains various breakable single-line, 10-pin female sockets to be used as interface jumper wire connections

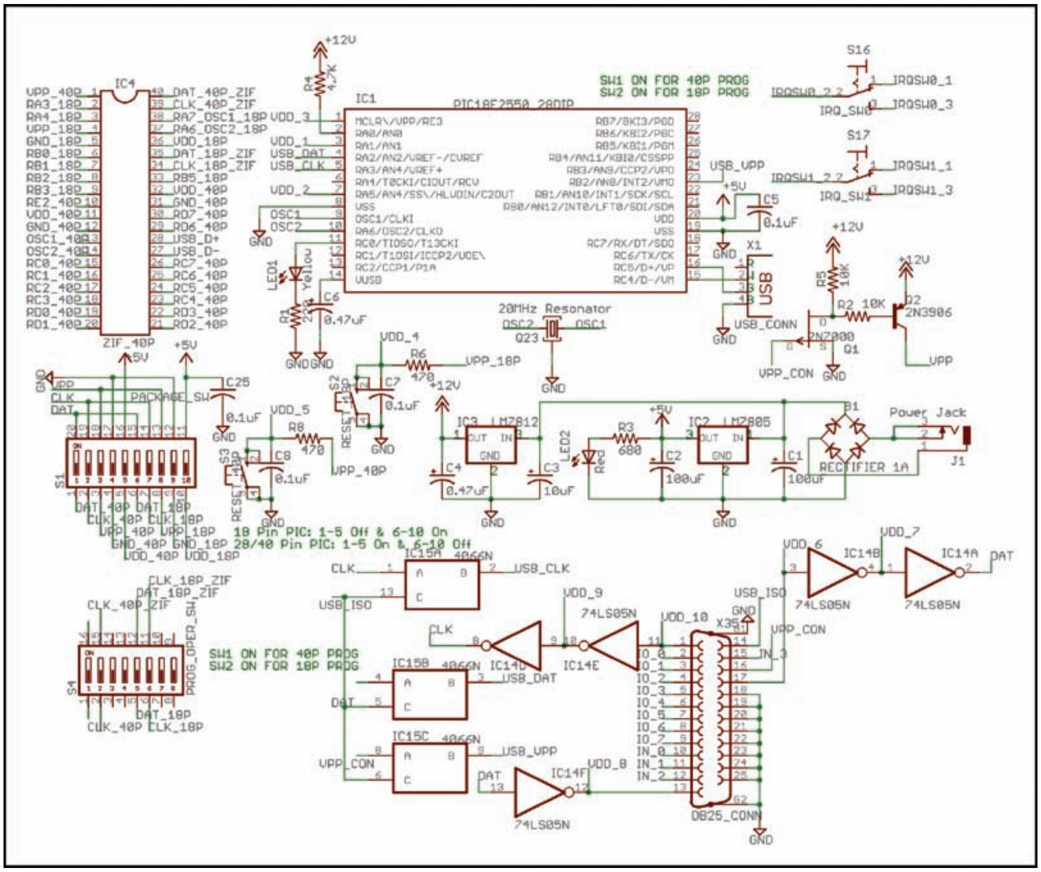

Figure 2. The circuit design for hardware Block #1.

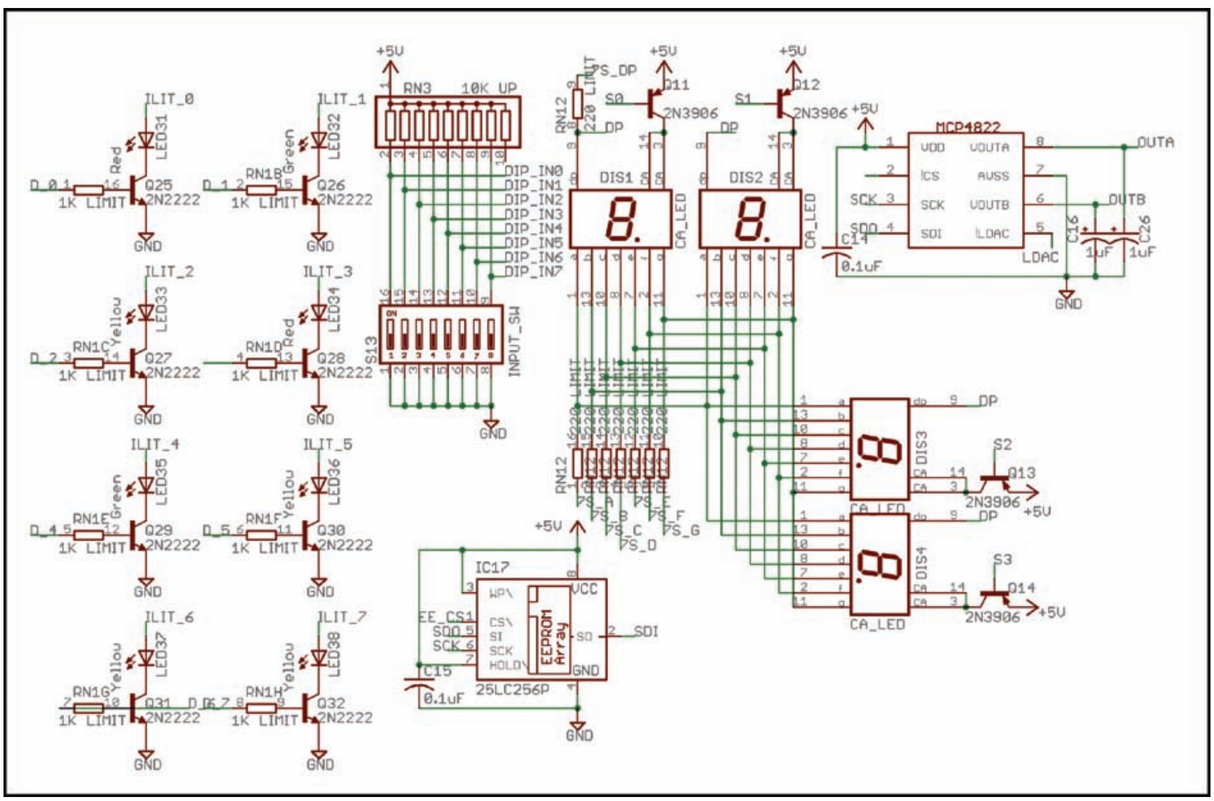

Figure 3. The circuit design for hardware Block #2.

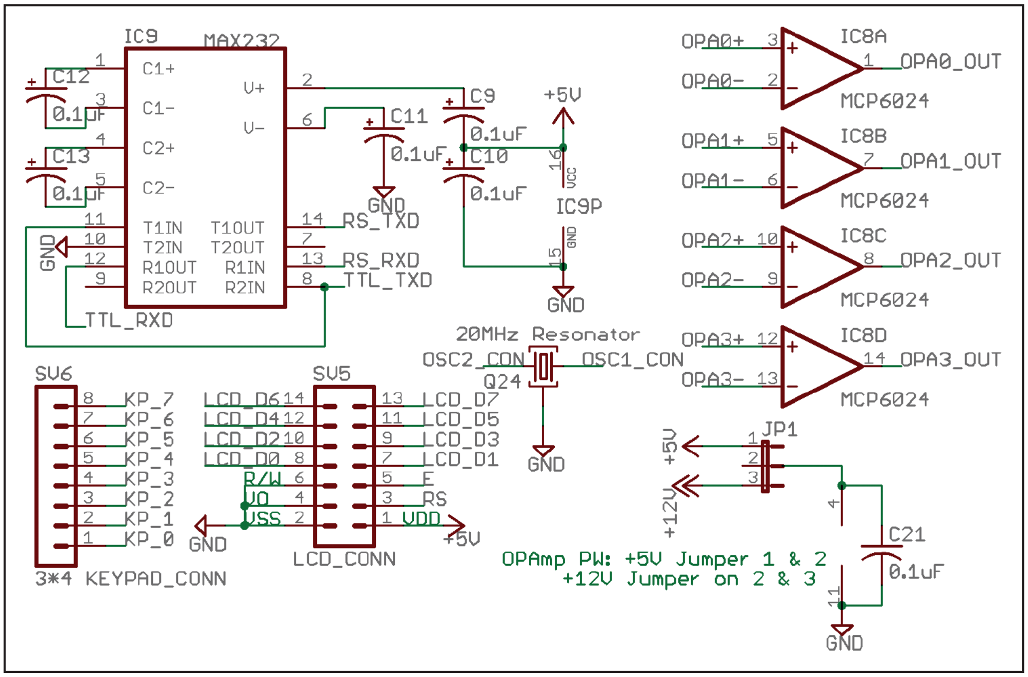

Figure 4. The circuit design for hardware Block #3.

for different needed signals between peripheral circuits, components, and the target PIC microcontroller. These include an LCD module, 3*4 matrix keypad, 8 bit IOs, SPI, RS232, 20MHz resonator, and associated pull-up resistors. They are placed around a 2.2"*6.5" breadboard for easy access of desired interface connections.

The power control section of the hardware, Block #5, is shown in Figure 5. To eliminate any possible noise contamination while running a stepper or DC motor control experiment, an eight-channel optical isolator is included in the design. Eight IRF530 power FETs are the drivers for those high-power needs. Separate terminal block-type connectors are used for small-gauge wire connections for different high-power exercises. The high-power On-Off controls are made available through associated IO_X input signals in this section's designs. When implementing an H-bridge experiment, current direction is indicated by the LEDs, thus providing students with both insight into current directions and indicators for troubleshooting.

The circuits for Block #6 are presented in Figure 6. Block #6 consists of eight debounced pushbutton switches, which provide general-purpose I/O signals that can be used for microcontroller exercises or any standard digital course experiment. This adds flexibility to the training system so that it can be used in traditional digital logic experiments if desired.

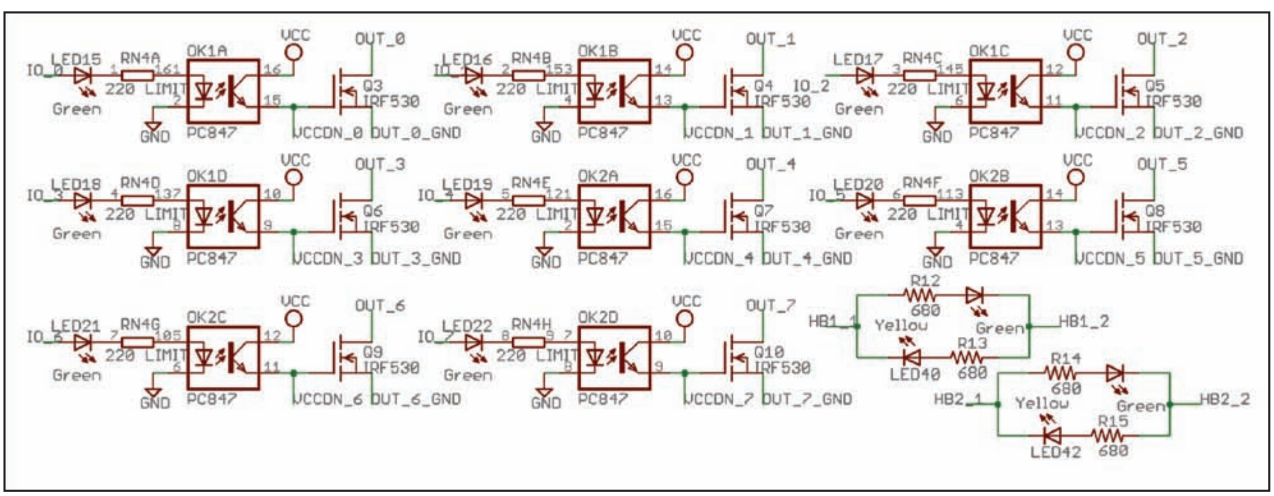

Figure 5. The circuit design for hardware Blockw #5.

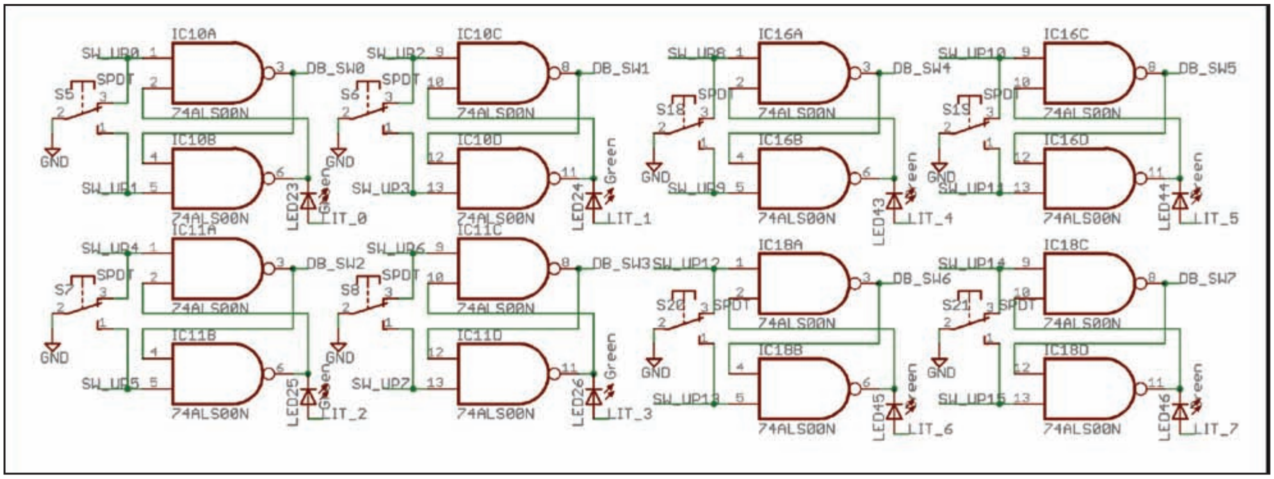

Figure 6. The circuit design for hardware Block #6.

7. Printed Circuit Board (PCB) Implementation

Based on the bill of materials from the designed circuits[19], there are a total of 205 electronic components/parts needed on the PCB. The goal is to make a PCB that has to be less than a standard page size of 8.5”*11” format for easy transportation. When implementing the designed circuits into a desirable fit in a PCB, there are several factors that need to be considered:

1. All the parts should be used in a through-hole format; any surface-mount component will make the assembly and troubleshooting very difficult.
2. Even if footprints and prices for the surface-mount parts are lower, they do not justify the difficulty in replacing and updating the training system in the future.
3. Not all the available parts’ footprints for the PCB layout software can be perfectly matched with the parts from available vendors, so making customized footprints is necessary.
4. Different adjustments on the parts’ footprints are critical design process.
5. To better meet budget constraints, an adjustment on parts’ footprints with available parts should work coherently during the PCB layout designs.
6. A four-layer PCB is preferable because of ease of layout, but the PCB manufacturing cost has forced the design to be a double-sided board.
7. The size of the training PCB can only shrink to its absolute minimum of 8”*10” to host a total of 205 electronic components. This makes the routing a very challenging task. The auto route function performed by the software will not be able to do the job. Several trials and manual routes are the solution to meet the goal.
8. High-power and low-power sections of the circuits should be separated.
9. The routing traces of the high-power signal should be wider in order to carry higher current.
10. High-frequency components, such as the USB, resonators, and SPI bus lines, should be placed as close as possible to their communication partners.
11. All the interface connectors should be placed around the 2.2”*6.5” breadboard for easy access in building interfaces.
12. All the low-power, USB, DB25, and ribbon cable connections are placed at one side of the PCB and all the high-power connectors for motor controls are placed on the other side. These arrangements are designed for user friendliness and easy access in performing laboratory experiments.

After applying these PCB design considerations, a final assembled PCB for this PIC training system is presented in Figure 7.

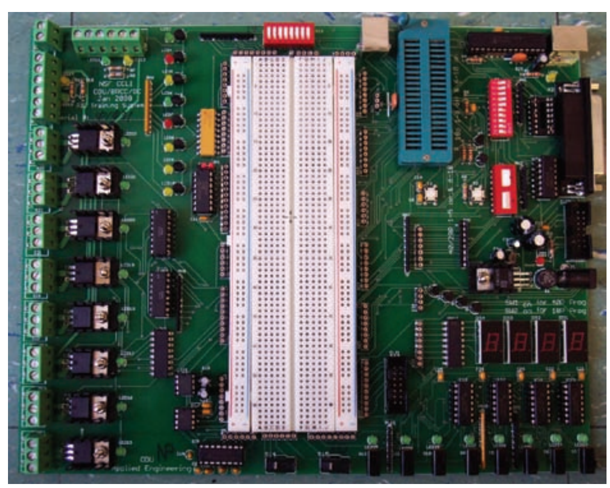

Figure 7. The Assembled PIC Training System.

8. Training System in Distance Learning

Currently, this newly developed PIC microcontroller training system is under field testing and evaluation at Blue Ridge Community College (BRCC), VA, Olympic College (OC), WA, and Old Dominion University (ODU), VA. The intent is to use the teaching/learning system in active distance learning and on-campus courses. After beta testing and refinements, the system will be available to other interested institutions. The objectives are to solve the cost, hands-on troubleshooting, and teaching/learning efficiency issues while doing laboratory activities with active distance-learning classes. To meet the objectives of this system, all students in a course need to use the same platform (this training system). The student issues with hardware interface and software source code design can be resolved quickly via group discussions, online conference lectures, one-on-one videoconferences, and e-mail. The unique feature of this training system is that it matches the curriculum objectives of the three institutions involved in this project. This approach differs from existing products available on the market and can be adopted universally. The other advantage is that the cost of this system is below similar commercial competition because it will not be sold for profit. The purpose of this system is solely to provide an improvement in the quality of active hands-on, distance-learning applications.

The following are steps that are currently used or intended to be used at BRCC, OC, and ODU for on-campus and active distance-learning applications:

1. All students are required to purchase this training system at a price of $125. The instructor and students use the same platform. Lecture materials and lab descriptions are made based on the trainer labels and part numbers thus eliminating any undesirable differences in parts/specs/manufacturers. Under this environment, it makes the distance learning as well as on-campus, hands-on laboratory teaching of troubleshooting efficient and issues much easier to resolve.
2. All course material, course curriculum modules, trainer system manuals, and software (MPLAB IDE), PICKIT2, ICPROG, & Sample Source Codes) are made available online for free downloading. A total cost of $125 for this training system is even lower than a single textbook in similar classes. The curriculum modules were designed to fit different course objectives in either two- or four-year institutions. These cover sufficient content for various levels of courses. It is up to the instructor to choose the proper modules to fit their course needs.
3. Provide Internet links to all the support material and reference books that can be used for the class activities.
4. Weekly class lectures and lab explanations are available via real-time, televised broadcasting or real-time conference lectures via Internet (Acrobat Connect or any other available software).
5. Any experimental issue with hardware interface and software source code designs can be resolved quickly via group discussions on Blackboard, online conference lectures and one-on-one videoconferences via Acrobat Connect, and/or e-mail via the institution's mail server.
6. Weekly real-time audio and video group meetings with the instructor are used to ensure the integrity of the lab requirement is fully understood by all students. The recorded archive of the session is always available for those who miss the weekly group meeting or simply want to review discussed topics.
7. Laboratory performance checks are implemented via real-time, one-on-one videoconferences.
8. Set-up of various topics related to class lectures and lab activities with the help of Blackboard™ or other course management software/tool is done to resolve issues through discussion/consultation between students and instructor. Dividing the class into groups, assigning and rotating group leaders, and grading individuals on group discussion activities will motivate discussion. This will save instructor time by not being required to duplicate answers to similar questions. Students can assist each other in troubleshooting design issues. This aligns with ABET accreditation criteria for group work.
9. There are lab activities where students are required to integrate the content learned throughout the course. This requires students to solve problems and apply the knowledge they have accumulated to design a required final lab project. This strategy develops skills that students will need as they transition to real-world applications in the workplace.
10. The ABET accreditation criteria require that any form of course delivery should have the same course content, evaluation, and assessment methods. All institutions involved with this project are TAC/ABET accredited. All course materials have clearly stated objectives. The content of the modules and module evaluations are based on these objectives. The system has been designed to meet accreditation guidelines. This training system can be used as a campus-based or a distance-learning course since the only difference is the physical distance between the instructor and students.
11. The trainer can be used in all digital-related courses, such as fundamental digital electron-

ics, beginning/medium/advanced level of microprocessor/microcontroller courses, senior capstone project design, etc. The associated curriculum sessions cover microprocessor/microcontroller courses in assembly language programming.

9. Conclusions and Suggestions

In addition to the goals of this project, several additional results were achieved. First, this has been a rewarding educational experience for the research team members. Team members have realized the vast amount of work required to develop new training hardware, software, and accompanying instructional support materials. It was a challenging learning experience for everyone on the design team.

Also there are current demands for this type of training system. This was determined through conversations with faculty during conference meetings. The team has learned that there is a common concern of the obstacle in implementing hands-on distance learning—a lack of a good teaching platform that is effective and affordable.

Since the beginning of this project in April 2007, several groups of students at BRCC and ODU were involved in organizing electronic components, assembling PCBs, evaluating and testing assembled PCBs, testing circuits, and completing surveys for evaluating the effectiveness of this training system. Responses were positive during their learning experiences while assisting with this project.

The other issue for this project was the budget. There was always a lack of funds to get the project completed. Spending more time and paying attention to details enabled the research team to meet the optimal goals without jeopardizing the quality of the project. The final cost for this training system board was set to be approximately $100, but the current bill of material calls[19] for a total expense of $124.19, not including assembly cost. This would have increased the budget by over 25%. However, by increasing the search for components and negotiating volume purchases, it is expected that the bill of material cost can be lowered by 15%-20%. Using the students' help (with pay) in PCB fabrication can lower the cost of assembling the system, and at the same time teach students manufacturing processes, quality control, and troubleshooting skills. It also provides students with practical training experience for their future employment. These approaches bring the $100/training system board closer to a reality.

The prime goal was to make affordable technology-related course materials, activities, hardware, and software available to students who do not have access to on-campus college and university laboratory equipment in microcontroller-related training that is required for many high-tech careers. This project produced microcontroller prototype hardware and software and instructional materials needed to support the active distance delivery of several microprocessor-related courses. Without allowing students to actually build circuits and test their designed software programs on real hardware set-ups using a common platform, it is difficult for them to understand the course content through distance-learning programs.

As this project evolves, individual laboratory activities are also being developed to reinforce student learning and skill development in programming concepts as well as provide a platform for individual student research and development after course completion. The expected outcomes will be better trained/educated students who will qualify for positions in the technical knowledge-based workforce.

References

1. Michael, K. (2001). "The effect of a computer simulation activity versus a hands-on activity on product creativity in technology education". (2001). *Journal of Technology Education*. 13(1), 31-43.
2. Bernard, R.M., Abrami, P.C., Lou, U., Borokhovski, E., Wade, A.W., L. et al. (2004). How does distance education compare with classroom instruction? A meta-analysis of the empirical literature. *Review of Educational Research*. 74(3), 379-439.
3. Gokhale, A. (2007). Effectiveness of computer simulation for enhancing higher order thinking. *Journal of Industrial Teacher Education.* 33(4), 36-46.
4. MikroElektronika (2006). Development Tools. Retrieved April 23, 2006, from http://www.mikroelektronika.co.yu/english/product/tools/etools.htm
5. AMS (2006). Burning Software Corporation, PIC Training & Development System. Retrieved March 20, 2006, from http://www.burnningsoftware.co.uk/pricelst.htm
6. MCPro (2006). Microcontroller Pros Corporation, Microchip PIC Emulator PICE-MC. Retrieved April 25, 2006, from http://www.microcontrollershop.com/ product_info.php?cPath=21_65&products_id=208
7. BASIC (2006). BASIC Stamp, Parallax Inc. Retrieved February 25, 2006, from http://www.parallax.com/index.asp
8. Microchip (2006). Technology Inc., Development Tools. Retrieved April 30, 2006, from http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=81
9. PIC16F84A Data Sheet. (2004). Microchip Technology Inc. http://www.microchip.com/.
10. PIC16F88 Data Sheet. (2004). Microchip Technology Inc. http://www.microchip.com/.
11. PIC16F877A Data Sheet. (2004). Microchip Technology Inc. http://www.microchip.com/.

12. PIC16F87/88 Flash Memory Programming Specification. (2002). Microchip Technology Inc., DS39607B.
13. PIC16F87X EEPROM Memory Programming Specification. (2003). Microchip Technology Inc., DS39025F.
14. PIC16F8X EEPROM Memory Programming Specification. (2003). Microchip Technology Inc., DS30262E.7.
15. Hsiung, S. & Ward, D. (2006). Design Software and Hardware for PC I/O Control. *Journal of Engineering Technology*. Spring, 16-24.
16. On-Chip Debugger Specification. (2001). Microchip Technology Inc., DS51242A.
17. PICKIT2. (2007). Available: http://www.microchip.com/pickit2, December, 2007.
18. ICPROG. (2007). Available: http://www.ic-prog.com/index1.htm, December, 2007.
19. Eagle. (2004). *Easy Applicable Graphical Layout Editor, 2nd Edition*, Cadsoft Inc.
20. 25AA256/45LC256 Data Sheet. (2007). Microchip Technology Inc. DS21822F. http://www.microchip.com/.
21. MCP4821/MCP4822 Data Sheet. (2005). Microchip Technology Inc. DS21953A. http://www.microchip.com/.
22. MAX232-MAX249 Data Sheet. (2006). Maxim Integrated Products. http:///www.maxim-ic.com/.

Steve C. Hsiung

Steve Hsiung is an associate professor of electrical engineering technology at Old Dominion University. Prior to his current position, Dr. Hsiung worked for Maxim Integrated Products, Inc., Seagate Technology, Inc., and Lam Research Corp., all in Silicon Valley, CA. Dr. Hsiung also taught at Utah State University and California University of Pennsylvania. He earned his B.S. degree from National Kauhsiung Normal University in 1980, M.S. degrees from University of North Dakota in 1986 and Kansas State University in 1988, and Ph.D. degree from Iowa State University in 1992.

James E. Eiland

James Eiland has been an assistant professor of electronics at Blue Ridge Community College in Weyers Cave, VA for more than 25 years. In addition, he is the owner/engineer of JEELAND Research & Development, which supports various defense contractors, local industry, and inventors. He is a retired Naval Intelligence Officer and earned his B.S. in Electrical Engineering from the University of Texas, El Paso prior to his military service in 1972.

John R. Hackworth

John Hackworth is an associate professor and director of the Electrical Engineering Technology program at Old Dominion University. He holds a B.S. degree in Electrical Engineering Technology and an M.S. degree in Electrical Engineering, both from Old Dominion University. Prior to joining the Old Dominion University faculty, John had approximately 20 years of industrial experience in test engineering and plant automation with General Electric Company. He is the co-author of two textbooks, which are currently in use by several electrical engineering technology programs at universities within the U.S.

John M. Ritz

John Ritz is professor and graduate program director of Occupational and Technical Studies in the STEM Education and Professional Studies Department at Old Dominion University, Norfolk, VA.