

2011

Teaching Control Programming Using Programmable Automation Controllers

Dennis G. Watson

Southern Illinois University Carbondale, dwatson@siu.edu

Seburn L. Pense

Southern Illinois University Carbondale, sebpense@siu.edu

Follow this and additional works at: http://opensiuc.lib.siu.edu/psas_articles

Recommended Citation

Watson, Dennis G. and Pense, Seburn L. "Teaching Control Programming Using Programmable Automation Controllers." *Applied Engineering in Agriculture* 27, No. 2 (Jan 2011): 295-302. doi:10.13031/2013.36478.

This Article is brought to you for free and open access by the Department of Plant, Soil, and Agricultural Systems at OpenSIUC. It has been accepted for inclusion in Articles by an authorized administrator of OpenSIUC. For more information, please contact opensiuc@lib.siu.edu.

TEACHING CONTROL PROGRAMMING USING PROGRAMMABLE AUTOMATION CONTROLLERS

D. G. Watson, S. L. Pense

ABSTRACT. *Introductory control programming was included as a required course for agricultural systems students. A programmable automation controller (PAC) was programmed with a flowchart paradigm to monitor and control applications. An ex post facto research design was used, with a questionnaire to obtain student feedback. The PAC instructional unit and student feedback are described. Ninety-two percent of students agreed the PAC unit of the course helped improve their problem-solving skills.*

Keywords. *Programming, Control, Automation, Problem solving, Curriculum.*

Problem-solving skills are critical for agricultural systems (AS) graduates. Problem solving ranked in the top five in a combined ranking of very important basic knowledge and applied skills in a 2006 survey of over 400 employers across the United States (Casner-Lotto and Barrington, 2006). Two educational objectives for agricultural technology programs are “apply principles of problem solving to be able to solve problems” and “use computer technologies to solve problems” (Harper et al., 2001). Computer control has become increasingly pervasive and complex; AS students would benefit from additional training and experience in problem solving. In 2002, we considered this need when reviewing course requirements for our AS students.

Our initial consideration was to make computer programming a requirement in the curriculum. Many science and engineering programs require a computer programming course and there are good justifications. Programming reveals the mechanical nature of computers and the notion of computer intelligence evaporates when students learn they can control each step of a machine’s behavior (Biermann, 1994). A programming course includes general problem-solving methods (Gries, 1974). A common justification for teaching programming is that it is a “study in clear thinking and problem solving, and it gives students wonderful practice at building representations and working with divide-and-conquer top-down methodologies” (Biermann, 1994). These techniques for breaking a complex problem into increasingly smaller, readily understood components transcend computer programming and are valuable strategies for solving systems problems.

Introductory computer programming experience would provide students with a basic understanding of control structures and the use of input and outputs. The result would be increased ability to interact with system designers and process engineers. Although a computer programming course would offer these important advantages for our students, our faculty had reservations about requiring our students to take an available introduction to programming course. Based on feedback of students who had completed the programming course, a large part of the course focused on constructs and features of the language before applications were developed, and the applications were related to business processing rather than agriculture. A course was needed that adhered to Biermann’s first rule of teaching programming to non-majors, “that the study should focus on interesting applications” (Biermann, 1994). An ideal course would focus on applications from the very beginning and for AS students this would include agricultural or other relatable applications.

Besides writing of typical programming code, we wanted students to have experience with programmable automation devices used in industry. Although programmable logic controllers (PLCs) with a ladder logic programming interface were common in industry, we preferred a device with a more user-friendly programming environment—so students could focus on problem-solving. After comparing various options, we selected a programmable automation controller (PAC) from Opto 22 (Temecula, Calif.). The ARC Advisory Group, an industrial analyst, is credited with coining the term programmable automation controller or PAC to describe devices which combine the best features of a PLC with the best features of a PC (Opto 22, 2007; National Instruments, 2009). According to the ARC Advisory Group, “PACs help end-users to refocus their energies on core competencies, and to relegate automation and control integration functions to their automation providers. PACs support Web-based monitoring and maintenance. PACs address user concerns to better monitor and control the interconnected real-world devices and to become more focused on total system performance than with hardware selection” (Bhattacharjee and Shah, 2004).

Submitted for review in April 2010 as manuscript number EDU 8541; approved for publication by the Education Division of ASABE in January 2011. Presented at the 2009 ASABE Annual Meeting as Paper No. 096560.

The authors are **Dennis G. Watson, ASABE Member**, Associate Professor, and **Seburn L. Pense**, Associate Professor, Department of Plant, Soil and Agricultural Systems, Southern Illinois University Carbondale, Carbondale, Illinois. **Corresponding author:** Dennis G. Watson, Department of Plant, Soil and Agricultural Systems, Southern Illinois University Carbondale, MC 4415, Carbondale, IL 62901; phone: 618-453-6979; e-mail: dwatson@siu.edu.

OBJECTIVE

The objective of this study was to develop a PAC programming unit for a required undergraduate course and assess student perceptions of the intended learning outcomes. The learning outcomes were:

- students would be able to program a PAC for multiple control and monitoring applications,
- students would improve their problem analysis and problem-solving skills.

METHODOLOGY

The methodology for this study was an ex post facto research design. Wiersma (1995) stated ex post facto is appropriate when the variables being studied are not manipulated, but studied in their natural context. Babbie (1986) stated that ex post facto may be employed to determine cause and effect relationships between past events. Wiersma (1995) also noted that ex post facto research designs may explain relationships and effects occurring between the variables.

POPULATION AND SAMPLING

The population of this study included 125 Southern Illinois University students—with about 95% majoring in Agricultural Systems. A census sample of the population included all students enrolled in the required course, introduction to control programming, over seven years. Random sampling of subjects was not feasible given the limitations of testing intact groups. Of the population, 119 provided useable responses.

INSTRUMENTATION

The Instructional Improvement Questionnaire (IIQ) was first developed in 1969 at Southern Illinois University to provide feedback to instructors about their teaching. Validity of the instrument was addressed through a review of existing literature which provided initial guidance for the instrument development process. An item pool was developed from the literature and locally prepared items. Validity was further addressed when the item pool was reviewed and revised by a committee of students, faculty and measurement specialists. Initially, a 72-item form of the IIQ resulted from the committee review and was administered on a voluntary basis for one year. The IIQ was then revised and shortened to a 49-item instrument, based on the following three criteria: “time to complete the form, item variability across courses, and an item’s relationship to factors derived from a factor analysis of the IIQ items” (Pohlmann and Elmore, 1976a).

Reliability of the instrument was initially established in a 1970 pilot test when Chronbach’s alpha coefficients ranged from 0.62 to 0.93 for the five IIQ subscales. A section of forced-choice items was later eliminated in order to limit the time required for students to complete the questionnaire. At present, the IIQ contains four parts: “a student biographic data section, an instructor evaluation section (20 items), a course evaluation section (20 items), and an optional item section where instructors can have students respond to items prepared by individual departments or faculty” (Pohlmann and Elmore, 1976b).

DATA COLLECTION

The IIQ was administered at the end of the introduction to control programming course and completed by students in 10 to 15 min. The anonymous responses were forwarded to the instructional evaluation staff where the data were summarized and reported back to the instructor.

PAC PROGRAMMING INSTRUCTIONAL UNIT

Resources for the PAC programming instructional unit included PAC hardware, a programming environment, and a textbook. These are described in this section along with an overview of the PAC course content.

PAC HARDWARE

The SNAP Ultimate I/O (Opto 22, Temecula, Calif.) was selected as the PAC for this control programming course. (In 2010, the PACs were updated with the current model of SNAP PAC R-series.) A learning center configuration (fig. 1) of the PAC was obtained. The learning center included a PAC, input/output (I/O) modules, mounting rack, control panel, and integrated power supply for about the same price as the PAC alone. The Ultimate I/O PAC was based on the 32-bit ColdFire® 5407 processor with 16 MB of random access memory (RAM). The current PAC R series units have the 32-bit ColdFire® 5475 processor with 32 MB of RAM. Both PACs were equipped with the Internet protocol (IP) for direct connection to a network.

The learning center included five input/output (I/O) modules of: four channel digital input, four channel digital output, two channel analog output (± 10 VDC), two channel analog input for integrated circuit temperature sensors, and two channel analog input (± 10 VDC). The eight channel rack provided space for up to three additional I/O modules.

The front panel of the learning center included monitoring and control indicators and actuators of an alarm indicator (tone generator), three LEDs, two toggle switches, two momentary toggle switches, a variable resistor generating 0-5 VDC output, a 0-10 VDC meter, and an integrated circuit temperature sensor.

The overall dimensions of the learning center were $28 \times 28 \times 18$ cm H ($11 \times 11 \times 7$ in. H) and each was configured with a unique IP address and connected in alternate carrels in a 25 seat computer laboratory. During the course, two students shared a learning center and took turns downloading and testing their programs.

THE PAC PROGRAMMING ENVIRONMENT

Programming of the PAC is based on a flowchart paradigm using the manufacturer’s PAC Project software. The primary components of PAC Project used by students were PAC Control for programming control strategies and PAC Display for developing human-machine interfaces. Each PAC Control program is called a strategy. Each strategy can consist of multiple flowcharts, which can run simultaneously. The PAC Control programming environment (fig. 2) consists of a menu and toolbars; a strategy tree which includes details on the control engine (PAC), flowcharts, variables, and I/O points; and flowchart area. The programming environment has most major features of modern programming environments, including a debugging mode with run, pause and stop controls; step into, over, and out features; auto step



Figure 1. PAC Learning Center with PAC (upper left) mounted on rack with input/output modules. An Ethernet cable inserted in the RJ-45 jack (upper left corner) connects the PAC to a LAN.

feature; breakpoint setting; and watch windows for I/O point or variable values. The auto step feature is commonly used when testing a program, as it highlights the currently active flowchart symbol (fig. 3).

Strategy development consists of five primary steps: configuring I/O points, inserting desired flowchart symbols

onto chart area, applying descriptive name to flowchart symbols, adding instructions to flowchart symbols, and testing. The steps of inserting desired flowchart symbols and naming them serve as a pseudocode phase and allow students to build a representation of their logic before entering the first programming command.

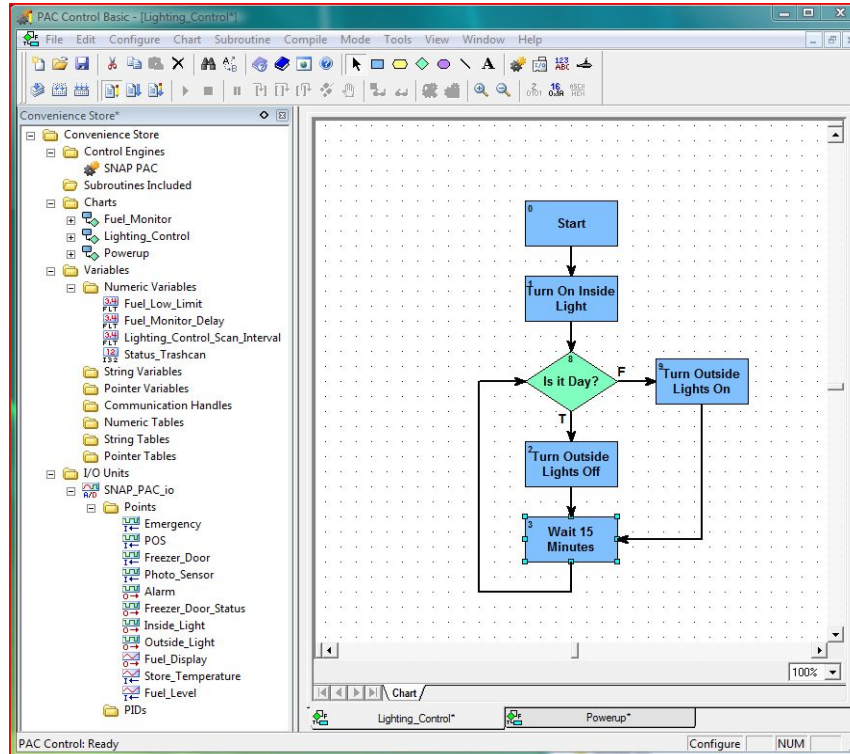


Figure 2. The PAC Control programming environment with strategy tree displayed on the left and one flowchart for the strategy displayed on the right.

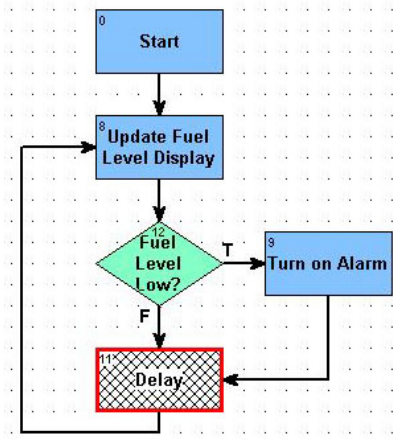


Figure 3. Example of a chart running with the auto step feature highlighting the currently active flowchart symbol (Delay).

Each flowchart symbol can have one or more programming instructions. The programming interface allows students to select instructions from a drop-down menu. As an instruction is selected, dialogs appear for the required parameters (fig. 4). Users also have the option of writing a program using OptoScript (fig. 5) within a special OptoScript flowchart symbol.

The PAC Display program is used to create an operator or human-machine interface for an application. The software comes with a symbol library, allowing students to choose from a wide range of symbols to represent system components. Objects on the operator interface can be programmed to change color, size, or appearance based on status of an I/O point or variable within the PAC Control program.

PAC TEXTBOOK

The *SNAP PAC Learning Center User's Guide* (Opto 22, 2009) was the latest version of the textbook for the unit. The

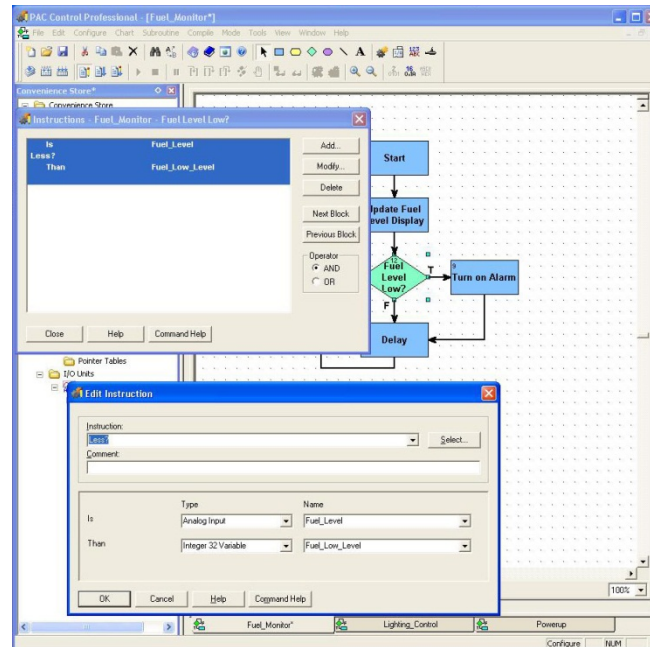


Figure 4. Instructions contained within the decision symbol of “Fuel Level Low?” are displayed.

guide was developed as a training resource for the PAC learning center. The guide uses a single application of a convenience store monitoring and control application. Although not directly related to agriculture, the students could readily relate to the convenience store application with monitoring of fuel tank levels, photo sensors for controlling outside lights, alarms, and other features. The text included step by step instructions for programming the PAC and was divided into nine chapters (see table 1). The text was used extensively throughout the PAC unit of the course. As each of the chapters was assigned for reading, students were required to complete a homework assignment of true/false, multiple-choice, or matching questions related to the chapter.

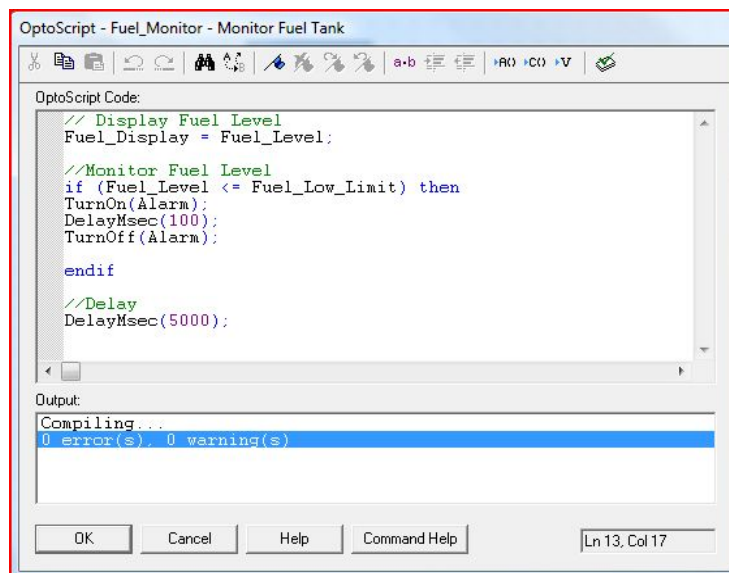


Figure 5. Example of OptoScript programming language available as an alternative to choosing instructions in other flowchart symbols.

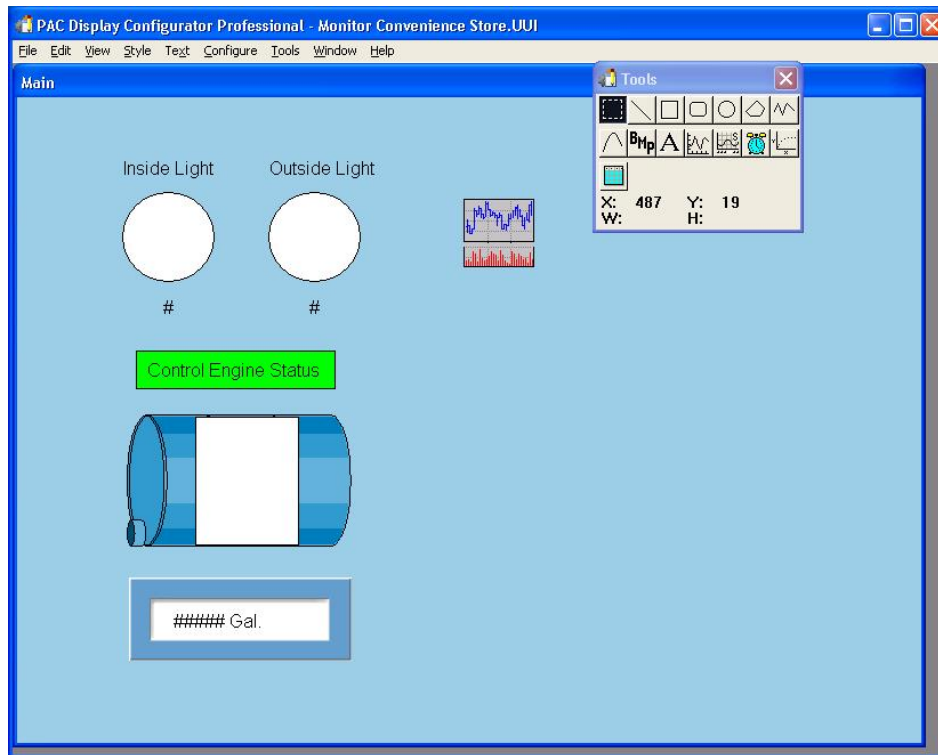


Figure 6. PAC Display configuration software used to develop an operator interface for a convenience store application.

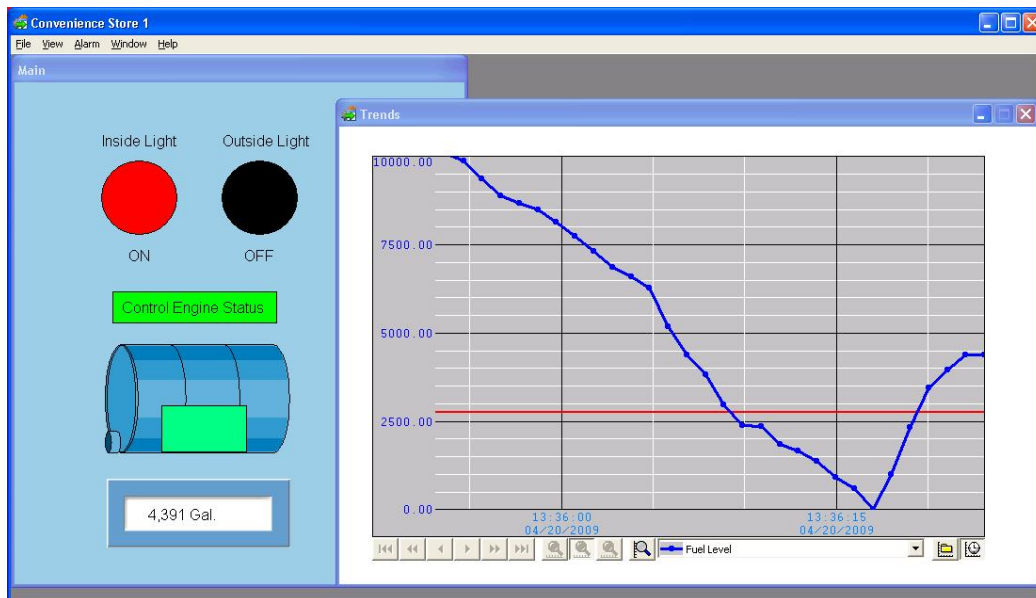


Figure 7. Operator interface for convenience store application in real-time mode with trend chart of fuel tank level.

PAC COURSE CONTENT

The PAC unit consisted of six lectures, six programming assignments (labs), one exam and a final project (table 2). The first five labs were completed individually by students as they followed the steps in the text, to develop a control and monitoring application for a convenience store. The sixth lab and final project were based on other applications and completed in teams of two (or three, if needed). The exam consisted of true/false, multiple-choice, and matching questions derived from the text. The final project served as

a means to evaluate the students' ability to apply PAC concepts, problem-solving skills, and programming skills to an agricultural automation and control problem of their choice.

The class was scheduled to meet for 75-min periods, two days per week. Each class period typically included a lecture slide presentation and time for students to work on programming labs. On most days, the computer lab and instructor were available to the students an additional 35 min at the end of the class. The extra lab time benefited many

Table 1. Chapter topics of PAC unit textbook.

Chapter	Topic
1	Using PAC Manager software to configure a PAC with an IP address
2	Using PAC Control to configure a PAC for I/O modules and I/O points
3	Introducing flowchart programming paradigm with examples
4	Programming for digital control of lights and alarms
5	Programming of analog inputs and outputs with scaling
6	Programming using the OptoScript language
7	Using PAC Display to develop a human-machine interface
8	Using the trending feature of PAC Display
9	Using the alarm features of PAC Display

Table 2. Sample listing of course content as presented each day of class.

Class Day	Course Content
1	Lecture: Introduction to Monitoring and Control Applications in Agriculture Demonstration: Configuring I/O Points Lab 1: PAC - Configure I/O Points Homework of reading and review questions for PAC configuration and programming
2	Lecture: Introduction to PAC Control Homework of reading and review questions for flowcharts Lab 2: PAC Control Charts
3	Homework of reading and review questions for digital control, analog control, and OptoScript Lab 3: PAC programming for digital and analog
4	Homework of reading and review questions for operator interface, trending, and alarming Lecture: Introduction to OptoScript Programming Lab 4: OptoScript Programming
5	Lecture: Introduction to Operator or Human-Machine Interface Lab 5: PAC Display
6	Lecture: Overview of Temperature Control in Swine Building Lab 6: Swine Building Temperature Control
7	Lab 6: Swine Building Temperature Control
8	Lecture: Description of Final Project and Tips for Written Report and Oral Presentation Assignment: Final Project
9	Exam over PAC unit
10	Final Project Lab Day
11	Final Project Proposal Presentations
12	Final Project Lab Day
13	Final Project Lab Day
14	Demonstration of Final Projects

students by helping them keep up with the lab and project assignments. Rather than have the students turn in labs for evaluation, it was more efficient and effective to review and evaluate the labs with students during the lab time.

Since the text included step-by-step instructions for implementing the convenience store application, the instructor had the option of demonstrating the features or letting the students discover them as they completed the step-by-step instructions. Typically, the instructor would demonstrate key steps related to configuring and debugging the PAC and I/O points and allow students to follow the text

instructions with minimal demonstrations. We explained to the students that during this part of the course, the learning method would be more similar to future learning experiences during their career, in that they would be expected to learn how to program new equipment with only a manual or tutorial to guide them. The instructor was still available to answer student questions and provide guidance upon request.

During initial years of the course, we found students would follow steps in the text to complete a programming assignment, but had difficulty applying the experience to program a different application. In recent terms we repeatedly stress the importance of understanding the various steps in a programming assignment, as a subsequent lab and final project will require the steps to be applied to a different application.

During the first four years of the course, the students started on the final project after completing the labs in the text. We found as many as half the students struggled with applying their newly learned skills to the requirements of the final project. The past three years, we introduced a sixth lab to bridge the gap between the textbook labs (with step-by-step instructions) and the final project.

The sixth lab assumes temperature in a confinement building is managed by a heater, ventilation fans, and opening or closing of curtains that cover openings in the side walls of the building. Students are required to develop a program to:

- scale the 0-5 VDC of the potentiometer to simulate a temperature range of 0°F to 120°F
- monitor the temperature and turn on/off heater, open/close curtains, and turn on/off fans as temperature changes
- display a human-machine interface indicating status of the system.

Since adding this lab to the course, we have observed students working more confidently on final projects and completing them with approximately 25% less instructor input.

The final project is based on the following scenario:

The student has been hired by a consulting firm, Control Strategies LLC, which specializes in developing process monitoring and control systems. Their clientele includes farms, elevators, feed processing plants, waste-water treatment plants, water treatment plants, food processing plants, fruit storage facilities, and boiler operators. The company has just negotiated a great deal on PAC Learning Centers. Using these relatively low-cost, off-the-shelf monitoring and control units they envision being able to reach a completely new group of clientele that were previously unable to afford control and monitoring systems. The student's new boss feels that the convenience store example included in the learning center training materials does not relate to the market they want to reach. The student's first job assignment is to work in a small team to come up with a more appropriate, realistic example using as many of the features of the I/O modules as possible, to develop an example for one of the company's existing clients.

The final project consisted of two parts. In part one, students developed a written proposal and make a 10-min presentation with slides. Part two was development of the prototype monitoring and control system and demonstration to the instructor. Initially, part two was scored as 67% of the final project. For the past two terms, the two parts were each worth 50% of the final project. Additional value was applied

to part one to emphasize the applied skills of oral and written communication. In a survey of over 400 employers across the United States, oral and written communication ranked first and fourth, respectively, for desired skills of new job entrants with a four-year degree (Casner-Lotto and Barrington, 2006). During the lecture that introduces the final project, the importance of oral and written communication were stressed as critical career skills. Simple tips were provided for making an effective presentation, such as introducing themselves and their project title, starting the presentation with a thought-provoking question related to their proposal, not reading the report off slides during the presentation, demonstrating an appropriate level of enthusiasm, and rehearsing, rehearsing, rehearsing. A simple outline was provided to guide students in developing the written proposal. The required proposal headings were initial system description, control and monitoring system objectives, and operator interface. Students were required to develop and include high quality graphics in their report and presentation. The rubric had equal weighting for five items in each of oral and written presentations. The oral presentations were evaluated on preparedness, opening, quality and use of visuals, professionalism and equal team member participation. The written proposals were evaluated on content in each of the three required headings, quality and use of graphics, and overall professionalism. The instructor judged the quality of proposal presentations and reports to be much improved during the past two terms, as a result of the increased emphasis on oral and written communication skills.

Student projects during the most recent term included: continuous flow and in-bin grain drying systems, dairy barn automation, shop automation including CO monitoring and ventilation, poultry barn automation, and greenhouse automation. Students implemented their projects using the basic monitoring and control strategy from prior labs—in that the program consisted of concurrently executing control charts with each operating in a continuous loop and control actions were dependent on change in a variable or input. Otherwise, the students developed their own monitoring and control design to meet their project objectives. In most cases, the monitoring and control applications were more complex than the prior lab exercises. The rubric for evaluation of completed projects consisted of functionality (10%), organization and naming (15%), completeness of charts (15%), user-friendliness of charts (10%), and human-machine interface (50%).

STUDENT FEEDBACK RESULTS

Over the past seven years, the instructor ratings by students on the IIQ averaged 4.5 on a 1 to 5 scale, with 5 being excellent (table 3). Since this study employed a census sample of an intact group, descriptive statistics were employed, including mean and standard deviation.

The mean aggregate course rating was 4.56 on the same scale. Course enrollment averaged 17 per semester over the same time period. These ratings were considered above average in our department. We have been pleased with the IIQ results, particularly considering this was a required course.

Table 3. Student mean ratings of instructor in Introductory Control Programming course from 2004-2010.

Term	n	Mean	Standard Deviation
Spring 2004	16	4.40	0.90
Spring 2005	17	4.49	0.62
Spring 2006	9	4.60	0.70
Spring 2007	16	4.46	0.58
Spring 2008	21	4.41	0.76
Spring 2009	18	4.90	0.33
Spring 2010	22	4.85	0.39
Aggregate	119	4.56	0.59

During the last two terms, supplemental questions were added in which students were asked to indicate agreement or disagreement with statements specifically about the PAC unit. All students (n = 39) agreed the PAC unit was a good learning experience and 97% agreed the PAC unit of the course was good, while 92% found the PAC unit of the course very interesting. Ninety-two percent of students agreed the PAC unit helped improve their problem-solving skills. Students were also given the opportunity to answer the question, “In what ways did the PAC unit cause you to analyze problems differently?” Table 4 includes a sampling of student hand-written comments.

DISCUSSION AND RECOMMENDATIONS

After seven years of introducing a new “Introduction to Control Programming” course, we consider it a success based on the positive feedback of the students and their demonstrated ability to synthesize control programming concepts and their own system knowledge into their final projects. The use of a PAC to teach introductory programming helped us achieve objectives of students learning to program, and improving their problem analysis and problem-solving skills.

The PAC unit was essential to the success of this required course. It allowed us to meet Biermann’s first rule of having interesting applications for teaching programming to nonmajors (Biermann, 1994). The PAC applications revealed

Table 4. Sample of student responses to “In what ways did the PAC unit cause you to analyze problems differently?”

Student Comments
It allowed me to apply computer skills to real life farming situations
It made me look at what it takes to make something like a dryer system or manure pit sensors work right and how much you have to look at to get the materials to fit together and work right.
It made me think about a lot of different things at once to make sure that it will all work together.
It taught me to think the same way as the computer works.
Look at different ways of doing things that I didn’t realize could be automated.
Understanding how computer systems work gave me a broader insight on problem-solving.
You have to be able to think in sequence.
Made me try to look for solutions before diving into the problem.
I had to figure out different ways to approach one problem.
It made me look at the whole problem not one area.

the lack of computer intelligence, as students learned they were responsible for controlling every detail of the monitoring and control application (Biermann, 1994). We expect students successfully completing the PAC unit to be less likely to view an automation or decision-making system as a “black box.” Their experience in the PAC unit provided them with an understanding of programming for monitoring and control applications. The PACs allowed our students to focus more on total system performance of monitoring and controlling interconnected devices, rather than on hardware selection (Bhattacharjee and Shah, 2004).

Many students enter the course with some level of technical problem-solving skills, although most often mechanical-based. It is rewarding to see them learn that some of the same approaches that lead to good problem solving in mechanical systems can be applied to control programming. We are confident that most students are finishing the course with a greater appreciation and understanding of problem-solving skills and greater confidence to apply their skills to new areas. Based on the results of the PAC unit described, inclusion of similar content in an AS program is recommended.

ACKNOWLEDGEMENTS

Opto 22’s generous contributions of updated PACs and components, technical support, and training resources are greatly appreciated.

REFERENCES

- Babbie, E. 1986. *The Practice of Social Research*. 4th ed. Belmont, Calif.: Wadsworth.
- Bhattacharjee, A., and H. Shah. 2004. *Programmable Logic Controllers Worldwide Outlook*. Boston, Mass.: ARC Advisory Group.
- Biermann, A. W. 1994. Computer science for the many. *Computer* 27(2): 62-73.
- Casner-Lotto, J., and L. Barrington. 2006. *Are They Really Ready to Work?* New York, N.Y.: The Conference Board.
- Gries, D. 1974. What should we teach in an introductory programming course? *ACM SIGSCE Bulletin* 6(1): 81-89.
- Harper, J., P. Buriak, and B. Hitchings. 2001. University faculty perceptions of the technology education needs in food, environment and natural resources system. ASAE Paper No. 01-8029. St. Joseph, Mich.: ASAE.
- National Instruments. 2009. A Comparison of PACs to PLCs. Available at: zone.ni.com/devzone/cda/tut/p/id/2960. Accessed 2 March 2009.
- Opto 22. 2009. *SNAP PAC Learning Center User’s Guide*. Temecula, Calif.: Opto 22.
- Opto 22. 2007. *Understanding Programmable Automation Controllers (PACs) in Industrial Automation*. Temecula, Calif.: Opto 22.
- Pohlmann, J. T., and P. B. Elmore. 1976a. The instructional improvement questionnaire. *J. Educational Measurement* 13(2): 161-163.
- Pohlmann, J. T., and P. B. Elmore. 1976b. The instructional improvement questionnaire. ERIC Document Reproduction Service No. ED121828. Carbondale, Ill.: Southern Illinois University.
- Wiersma, W. 1995. *Research Methods in Education: An Introduction*. 6th ed. Needham Heights, Mass.: Allyn and Bacon.