AC 2009-108: INTEGRATED SIMULATION AND ASSESSMENT SOFTWARE FOR PROGRAMMABLE LOGIC CONTROLLER LABORATORY INSTRUCTION

Carl Spezia, Southern Illinois University, Carbondale

Carl J. Spezia is an Assistant Professor in the Electrical Engineering Technology Program located in the Department of Technology at Southern Illinois University Carbondale (SIUC). He joined the program in1998 as a Visiting Assistant Professor. He worked as a power systems engineer for electric utilities for eight years prior to seeking a career in higher education. He is a licensed professional engineer in Illinois. His industrial assignments included power system modeling, power systems protection, and substation design. He received his M.S. and Ph.D. from SIUC in 1991 and 2002 respectively. He teaches courses in electric power and machinery, industrial automation, and electric circuits. His research interests include power systems economics, power markets, and electric energy management.

Integrated Simulation and Assessment Software for Programmable Logic Controller Laboratory Instruction

Abstract

Providing students with engaging laboratory experiences in sequential process control is challenging. Simple training devices that use lights and switches to teach basic principles fail to capture the complex interactions of industrial processes. Scale model process simulators are bulky and expensive to purchase and maintain in an educational environment. Laboratory exercises that represent process control responses accurately and engage students visually are the most relevant. Realistic industrial processes demonstrate typical automation systems to expand student programmable logic controller knowledge. This paper presents process simulation and assessment software for gathering student performance data. Students develop programs that reproduce control actions demonstrated by the software. An associated assessment module tests student understanding. Results from a student survey measuring the software effectiveness indicate the value of this laboratory experience as a learning tool.

I. Introduction

The programmable logic controller (PLC) is a fundamental part of modern industrial automation systems such assembly lines, robots, and machine tools. These devices implement sequential control schemes using a variety of programming methods. Ladder logic uses symbolic instructions similar to schematic symbols to program control applications. Developing students' sequential control design abilities and honing PLC programming skills requires a wide variety of exercises with increasing complexity using a number of subsystems. Industrial sequential control systems involve large, expensive, mechanical systems that include hydraulic, thermal, pneumatic, fluid, and electrical subsystems. These systems are costly and difficult to maintain in an educational setting.

Educational laboratory equipment vendors sell PLC trainer systems that are less costly and have simple interfaces made of lights and switches. These trainer systems can demonstrate basic programming principles but fail to show the interactions of complex industrial automation systems. Commercial software to simulate electromechanical systems and link to PLCs lacks debugging functions that help students learn program design.[1]

Student performance assessment is a critical part of any educational experience. Instructors can evaluate student performance in PLC programming by observing student software demonstrations, assigning written reports that document design and program details, and conducting quizzes. These methods require extensive development and oversight, and are difficult to integrate into an overall assessment plan that meets current ABET-TAC accreditation requirements.[2]

Simulator developers have used a variety of technologies to create automation and control devices for education. A microcontroller that communicates with a PC through a serial interface was selected by one team as the preferred simulation hardware tool.[3] Simulator software written in Pascal and C handled both analog and digital signals. Other groups utilized commercial automation software to combine networking, PLC programming, human machine interface design (HMI), and supervisory control and data acquisition (SCADA) topics.[4,5] Some developers recognized the high cost of control hardware and produced systems to simulate PLCs using software alone. Others created multimedia tutorial materials for Web delivery.[6,7]

This paper presents the design and student evaluation of an integrated process simulator and assessment tool for teaching PLC programming. The simulator includes animation and debugging to enhance student learning. Data acquisition (DAQ) boards plugged into PCs with simple external interfaces provide an inexpensive connection between process software and external PLCs. This hardware, coupled with simulator and assessment tools written in a high-level graphical language, create a low-cost integrated system that can engage and challenge students. The assessment test employs embedded program and course objective data fields. A test log file gives detailed analysis for program accreditation and student evaluation.

II. Simulator Software and Hardware

The PLC process simulator is written with LabVIEW™ software. LabVIEW™ combines hardware drivers and high-level data flow programming into an integrated package. LabVIEW™ implements data acquisition and control applications quickly. It is used widely to develop educational applications.[8] This software has tools to facilitate interactive interface development with animation. LabVIEW™ users can customize the basic interface components using graphics programs to create symbols representing industrial process components such as valves, pumps and tanks. Symbol color changes indicate component on/off status changes. State equation models embedded in the program simulate the dynamic response of processes.

The simulator connects to a PLC through a DAQ card installed in a PC and an external hardware interface. Fig. 1 shows the relationship of the software and hardware components. Analog inputs on the DAQ card link to output devices on the user interface screen such as indicator lights, pump and valve symbols, and ladder diagram symbols. Process symbols were customized

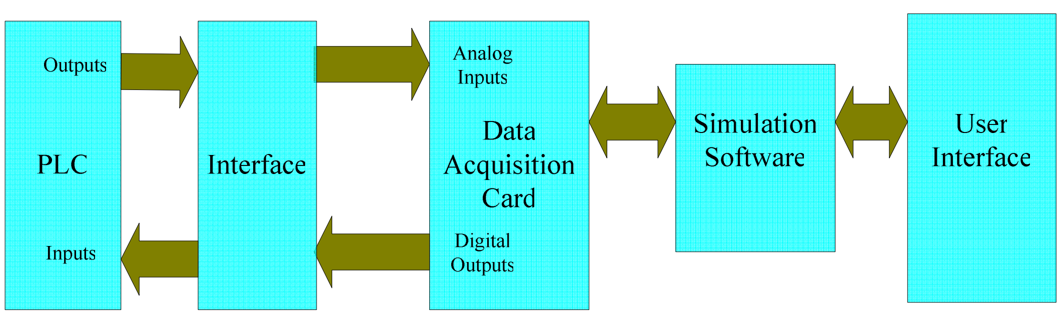

Fig. 1. Simulator Hardware and Software Structure.

from existing LabVIEW™ digital indicators. The PLC applies a voltage to the analog input through the interface. Simulation software converts the analog signal to a logic value for further processing. Process inputs from the user interface map to the digital output port of the DAQ card. These signals connect to the PLC input points through the interface. The process inputs are user-actuated switches or simulated process components such as thermostat contacts and limit switches.

Fig. 2 shows a schematic diagram of a typical hardware interface channel. These circuits link a PLC with relay outputs and dc sinking inputs to a NI-6024E DAQ card. Closing the PLC output contact applies 5 Vdc from the DAQ card to an analog input channel. The simulator uses DAQ analog channels operating in the referenced single-ended mode to maximize the number of outputs available. A load resistor, R_L, gives input channels a bias path to ground when the PLC output is de-energized. Digital outputs from the DAQ card connect to the PLC inputs through an opto-isolator. The PLC supplies the optical transistor from a 24 Vdc source through a current limiting resistor, R_c, in the collector circuit.

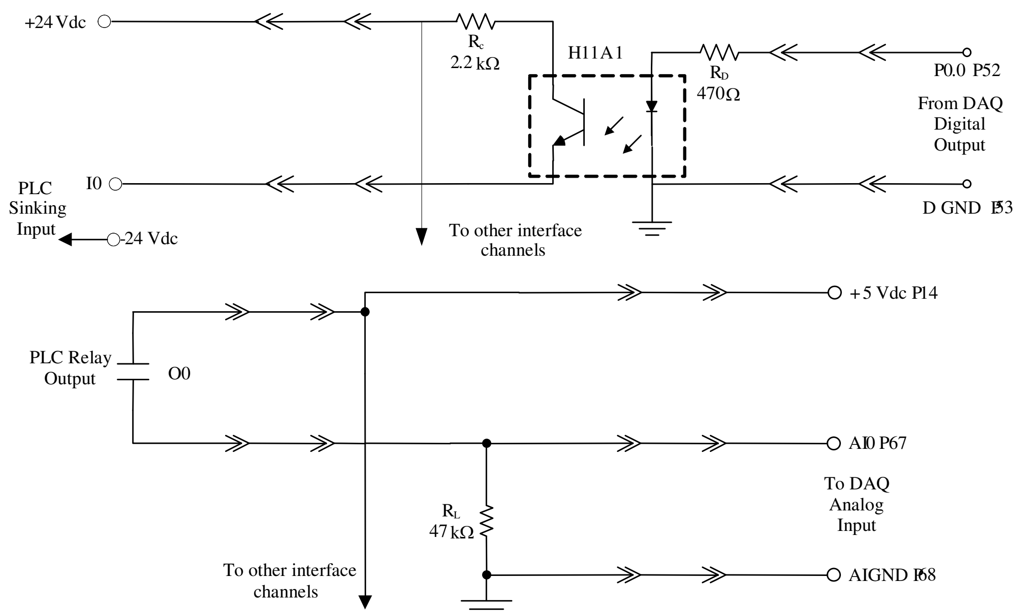

Fig. 2. Typical PLC to DAQ Interface Circuits Showing Input/Output Connections.

A DAQ TTL digital output drives the opto-isolator through the resistor, R_D. A 5 Vdc output from the DAQ card activates the PLC sinking input by energizing the LED causing the optical transistor to conduct. This is a very low-cost interface that is easily modified for other PLC input/output configurations.

Two modules comprise the PLC software, a simulator and an assessment test. Fig.3 shows the relationships of these modules and their major functions. The simulator module contains three exercises that offer different PLC programming challenges. The panel simulator provides simple training device capability with eight switch inputs and 12 indicator light outputs. Students write ladder logic programs using the addresses listed on the simulated switches and lights, and download them to the PLC. The PC simulator connects to the PLC through the external interface. Students observe the results of the program on the simulator.

The three-wire motor control module demonstrates control scheme operation using animated contact and coil symbols. A simple fan animation activates when the motor starts.

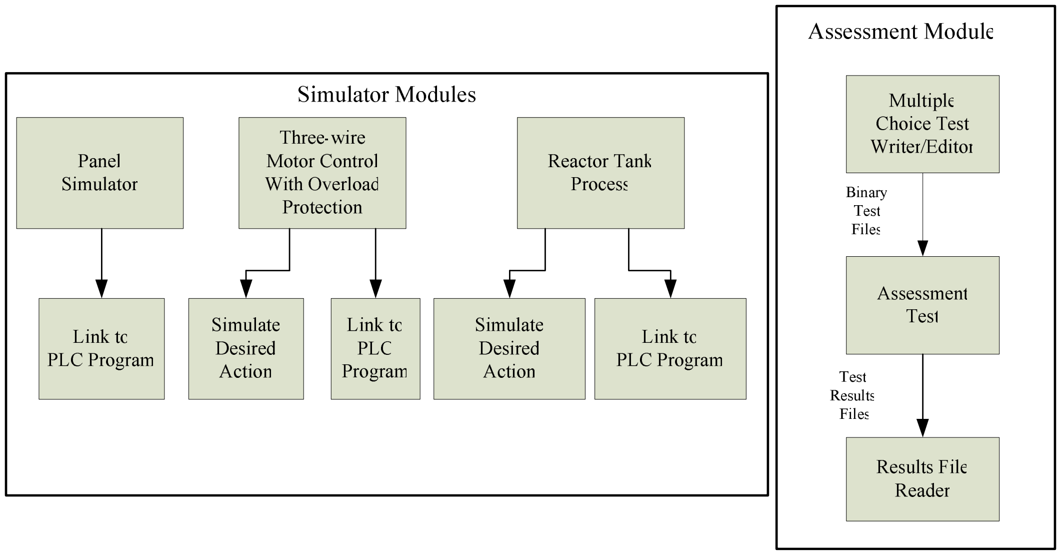

Fig. 3. Simulator and Assessment Software Structure.

This software shows a typical motor control station with on/off momentary contact push buttons and indicator lights. This exercise has two operating modes: simulate and interface. In the simulate mode, pressing the start and stop buttons causes the display to activate in the desired manner. This demonstrates the correct PLC program operation to student users. The interface mode links the module to the PLC for student-developed program testing. The software checks student programs against the required input/output points and displays appropriate error messages if the points are incorrect.

The third exercise combines binary state displays and numerical solution of differential equations to simulate a reaction process, such as brewing or distillation. This exercise has simulation and interface modes like the motor control described above. Fig. 4 shows the reaction process with simulation in progress. In this exercise, suction and discharge valves open, and the pump fills a reactor tank to a preset limit. A thermostatically controlled burner heats fluid for a predefined time. At the end of the time interval, the drain valve opens and the tank empties. The valve, pump, and pipeline symbols change color to reflect their on/off state. Tank fluid color changes on the display as the temperature changes. Strip charts record tank fluid level and fluid temperature over time. The simulation implements hysteresis in the thermostat and level control. Hysteresis is a range of insensitivity in a control device that increases stability and decrease noise. Contact symbols change for both level and temperature control to indicate contact state. Users modify simulation parameters through the control panel before starting the simulation and conduct “what if” experiments to gain better understanding of how the process works.

The reactor exercise software solves first-order differential equations numerically to demonstrate how tank levels change with time and how fluid heats and cools in the tank during the reaction process. Filling the tank is an example of an integral process. Adjusting the pump flow rate and tank diameter illustrates how parametric changes affect system performance. These animations help reinforce lecture topics and textbook examples.[9]

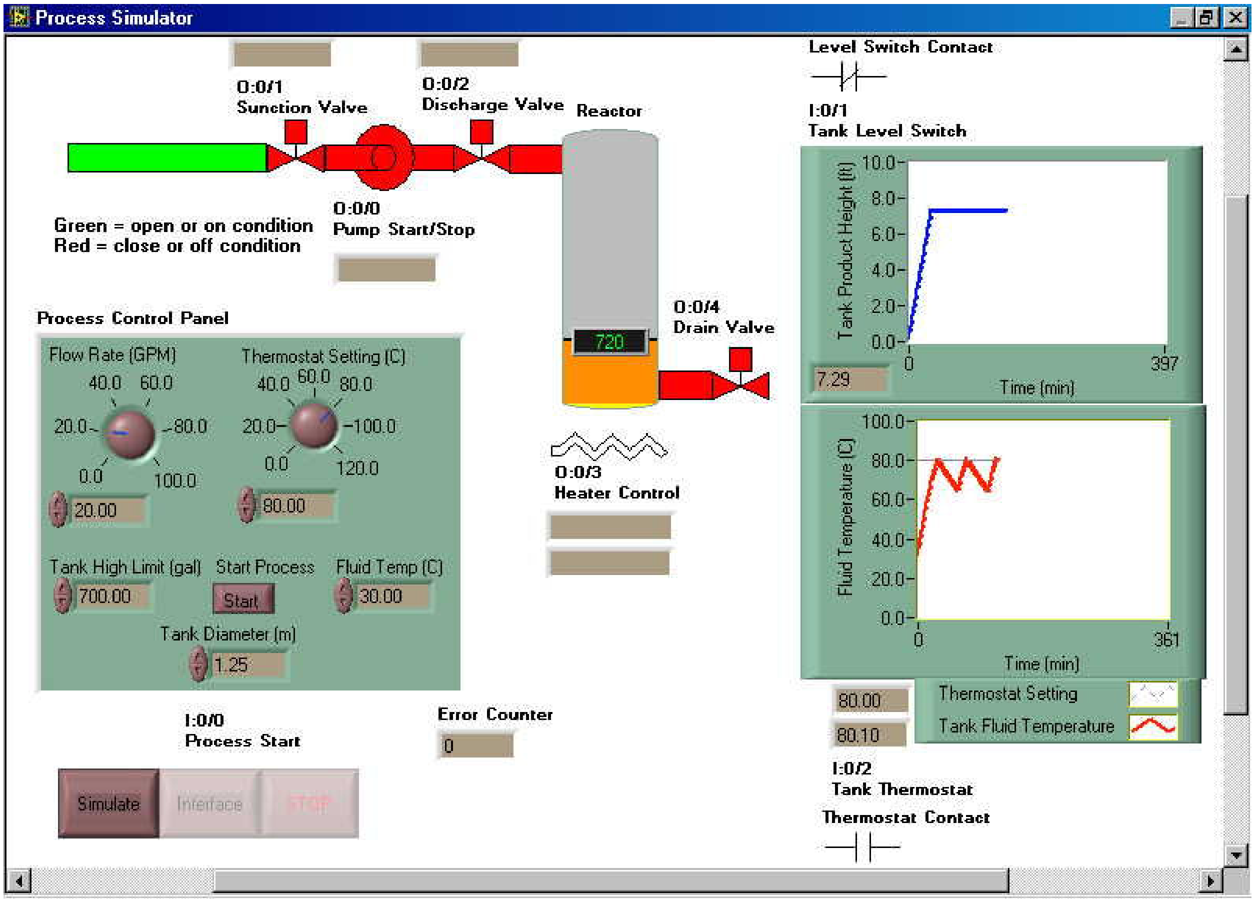

Fig. 4. Reaction Process Screen Display with Simulation in Progress.

In the interface mode, the software links to a student-developed PLC program that controls the process. As in the motor control exercise, the software compares desired process states to the pattern of PLC outputs from the student-developed program, and generates error messages to aid in debugging.

The assessment module consists of three programs. A test writer/editor enables the instructor to develop multiple choice test questions with up to four responses. Questions can be copied from other electronic documents into the writer/editor. The question writer has two fields that link each assessment question to a course and program objective. Populating these fields helps instructors meet student assessment guidelines established by ABET-TAC. The test writer/editor saves the test in binary format for security.

A student must enter their name to select a test. The software verifies that a student has not taken a test previously to prevent duplicate testing. After a student loads a test, control buttons activate, allowing the student to proceed. The test program displays questions and automatically scores them as a student enters responses. A student selects an answer using a radio button array. Pressing a “Check Answer” button determines if s/he has picked correctly. An indicator light glows next to the answer if the selection is correct. A dialog box appears if the selected response is incorrect, and prompts a student to pick again. Only the first response counts toward the test score calculation. A student has three chances to pick the correct answer before the program gives the correct answer. The program records the total number of student responses and displays it as a response percentage. The test has a time limit and will shut down automatically when exceeded. An elapsed time clock on the front panel helps a student manage time during the test.

Fig. 5 shows the assessment test interface with a question loaded. Activating and deactivating control buttons aids user navigation. The program deactivates all buttons except "Test Complete" after the user completes the last question. Clicking on this button followed by "Exit" completes the test, saves a log file, and ends the program.

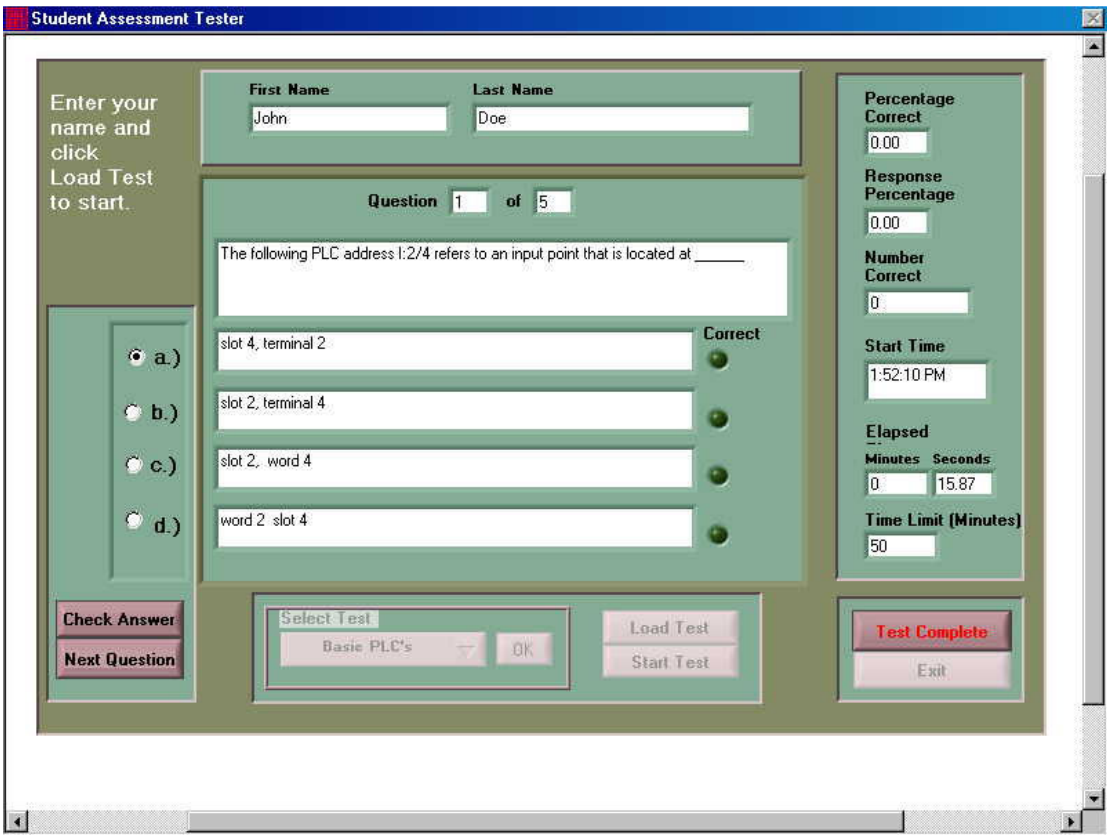

Fig. 5. Assessment Test Interface Showing Controls and Displayed Information.

The test log file is a text file consisting of header information that identifies who took the test, what test was taken, and when it was taken. Summary scoring information follows. A detailed report of student performance on each question comprises the last section. The last section includes: question number, course and program objective, number of responses, elapsed time to complete question, and whether the first response was correct. The assessment test program saves log files to a hidden directory on the PC to maintain privacy and secure the data.

The final part of the assessment module is a log file reader. This program loads log files generated when a student takes a test and formats them for evaluation by an instructor. This log file provides instructors with a tool for evaluating student performance. Including course and program objective fields in the log file allows instructors to use data to satisfy ABET-TAC accreditation criteria.

III. Student Software Evaluation and Sample Results

The software presented in this paper was used as a pilot in a senior-level automation and control course during fall 2008. This course covers modeling of typical processes using differential equations and Laplace transforms. The processes include self-regulating tanks and simplified fluid heating models. The course does not cover PLC programming, so these results do not reflect that aspect of the software. Fifteen students were enrolled in the course; thirteen completed surveys. The class covered these topics using lecture, homework problems, and laboratory experiments. Students worked with the simulator and assessment modules for approximately one week at the end of the semester and responded to a ten-item survey covering the operation and learning value of the modules. Table 1 summarizes survey results. The survey used 1-5 ranking, with 5 representing the highest rank. Student ranked items 5 and 10 highest. They liked the instant feedback of the test, but the relatively low score on item 8 indicates they

Table 1. Student Survey Results

Student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Mean	Standard Deviation
1.) The process simulator helps me visualize the concepts from the lecture.	4	4	4	4	5	5	5	5	4	5	3	4	4			**4.31**	**0.630**
2.)I found the process simulator interesting and engaging.	4	4	5	4	4	5	4	5	4	5	5	4	4			**4.38**	**0.506**
3.) I believe that this process simulator would help me understand similar material from the lecture.	5	4	4	5	5	5	4	5	5	5	4	5	3			**4.54**	**0.660**
4.) The process simulator helped me see how systems change with time.	5	4	3	5	5	5	5	5	5	5	4	3	5			**4.54**	**0.776**
5.) I would like to see more simulations like this.	5	5	5	4	4	5	4	4	5	5	5	5	5			**4.69**	**0.480**
6.) The automated tester/quiz tool had an easy to learn layout	5	4	4	5	3	5	4	4	4	4	4	5	3			**4.15**	**0.689**
7.) I would like to use the automated tester/quiz tool to prepare me for lecture exams	4	3	5	5	3	5	3	3	5	5	5	4	4			**4.15**	**0.899**
8.) I would like to use the automated tester/quiz tool to take lecture exams instead of pencil and paper tests	5	3	5	4	2	3	3	1	3	5	5	1	4			**3.38**	**1.446**
9.) I would like to use the automated tester/quiz tool as part of a self-study course where I can take the test or quiz when I feel prepared.	4	5	4	5	4	4	3	4	5	5	5	4	5			**4.38**	**0.650**
10.) I like it that the automated tester/quiz tool gives me instant feedback on my answers.	5	4	4	5	5	5	5	4	5	5	5	5	5			**4.77**	**0.439**

would prefer traditional testing methods in the lecture. This group would prefer to use the test as a method for preparing for lecture exams or self-study. The survey also indicates that students generally believed the simulator helped them understand time-varying systems and grasp lecture concepts. Item 5 scored high, indicating the group would like to see more interactive simulations. Survey items relating to student interest and engagement are promising.

Fig. 6 lists a typical test log file. The first four lines comprise the file header. This header records the student name, time and date, test name, and test scoring. The test scoring line lists the percentage correct answers, the response percentage, and the number of correct answers. The response percentage measures the number of times a student responded to questions during

John Doe
1/16/2009 9:52:33 PM
Test ID: Practice
80.00 28.75 16.00

Q#	COC	POC	R#	ETQ	Cor
1	1	3	1	7.01	Y
2	1	3	2	24.60	N
3	1	4	1	7.43	Y
4	1	4	1	11.74	Y
5	1	3	2	16.88	N
6	1	3	1	10.36	Y
7	1	4	1	7.33	Y
8	1	4	1	6.68	Y
9	1	4	1	7.12	Y
10	2	3	1	10.16	Y

Fig. 6. Typical Assessment Test Log File Showing Recorded Data

the test. The lowest response percentage is 25% which means each question required only a single response for a student to pick the correct answer. The remainder of the log file consists of detailed question analysis. Each line has columns for question number; course and program objective codes, the number of responses made on the question and elapsed time in seconds that students spend answering a question. The final column records if the first response was correct. By archiving this data for each student, and analyzing it with respect to program and course objectives, instructors can prepare useful information for accreditation review.

IV. Conclusion

This paper presented the design and student evaluation of a pilot program using an integrated process simulator and assessment system for teaching PLC programming. The process simulator uses a simple, low-cost external interface to attach PLCs to data acquisition cards. LabVIEW™ software connects this hardware to process simulations, creating a flexible, low-cost training system. The simulations provide a wide range of programming experiences, and include animation and interactive displays to engage students and promote learning. The assessment module includes a question writing tool, a test with automatic grading and data logging, and a results reader. Adding program and course objective codes to the assessment module generates data for detailed evaluation of student learning and support for accreditation review. Initial student reaction to the system was favorable, although the majority would prefer to continue using written tests in class. The survey indicated that students found the simulations helpful in visualizing time-dependent processes and lecture topics. Because students indicated they would like to see more simulation modules, further development of simulation and assessment software will be incorporated into the course.

Bibliography

1. Famic Technologies Inc., Automation Studio™, Online: www.automationstudio.com
2. Erdogan M. Sener, "Assessment: How Much is Too Much or How Much is Not Enough?", Proceeding of ASEE Conference and Exposition, 2002, Online: http://soa.asee.org/paper/conference/paper-view.cfm?id=17805.
3. João Palma, Luis Lacerda, José Antunes, Filomena Simoes, " A Multipurpose Process Simulator for Automation Engineering Laboratory", IEEE Symposium on Industrial Electronics, July 7-11, 1997, Vol. 3, pp. 931-935.
4. Antonio M. Lopez, Victor M. Gonzalez, Jose M. Enguita, Felipe Mateos, Antonio Robles, "Software Tools for Helping with the Design and Implementation of Automation Projects", Proceeding of 31st ASEE/IEEE Frontiers in Education Conference, Oct. 10-13, 2001, pp. F4C-12-F4C-17.
5. J. Flochova, D. Mudroncik, "Industrial Control Software Teaching at the Department of ACS FEI STU", Online: www.nt.ntnu.no/users/skoge/prost/proceedings/ecc03/pdfs/551.pdf.
6. Sheng- Jen Hsieh, P.Y. Hsieh, "Web-based Programmable Logic Controller Learning System", Proceeding of 32nd ASEE/IEEE Frontiers in Education Conference, Nov. 6-9, 2002, pp. S2D-6-S2D-11.
7. J.C.P Merlos, O. D. Velazquez-Rodriguez, "Development of a PLC Multimedia Tutorial", Proceeding of the 5th Biannual World Automation Congress, 2002, Vol. 13, pp. 195-200.
8. Shuhui Li, Abrar A. Khan, "Developong Digital Measurement and Analysis Laboratory in Circuits and Electronics and Electronics Lab at TAMUK", Proceedings of ASEE Conference and Exposition, 2004, Online: http://soa.asee.org/paper/conference/paper-view.cfm?id=19756.
9. Robert N. Bateson, Introduction to Control System Technology 7th Edition, Prentice Hall, 2002.