Open Source, Multilevel, Interactive Programmable Logic Controller Software Development for High School Students, Two- and Four-Year College Students, and Displaced Workers

Aleksandr Sergeyev
Michigan Technological University
avsergue@mtu.edu

Nasser Alaraje
Michigan Technological University
alaraje@mtu.edu

Kurt Kalenauskas
Michigan Technological University
kuhl@mtu.edu

Scott Kuhl
Michigan Technological University
kdkalena@mtu.edu

Bochao Li
Michigan Technological University
bochaol@mtu.edu

Abstract

Two programs in electrical engineering technology and computer science at Michigan Technological University are collaborating in a joint effort to develop a set of open source and online learning modules that give students interactive, hands-on experience with programming programmable logic controllers on a standard desktop or laptop computer. Faculty from both departments lead a group of undergraduate and graduate students from both units to effectively work on this joint effort. Three levels of the software are to be built so the future users can select the most appropriate version of the module based on their knowledge of the subject matter. It will be flexible enough to get high school students interested in programming PLCs, to train displaced workers seeking a certificate, to train technicians enrolled in two-year degree programs, or to teach more advanced concepts of PLC programming in four-year university programs. Each level of the learning modules will include multimedia materials, including video, audio, and/or electronic documents, which introduce the content presented in the module. Upon completion of each level, students will take the computerized comprehensive exam, testing all aspects of the material.

The learning system hosted by Michigan Tech will be freely available for anybody around the world to use over the Internet. When the system is used in conjunction with a class,

ISBN 978-1-60643-379-9

instructors will be able to connect with their students and monitor their progress. Learner competency will be tested by structuring some of the learning modules as games, where students can work collaboratively or competitively to solve PLC programming challenges.

This three-year project devoted to the design and software implementation is in its first phase. In this paper, we present the preliminary techniques, approaches as well as the overall project vision on effective PLC learning/gaming system development and implementation.

Introduction

With current advancements and reconfigurability of manufacturing, programmable logic controllers (PLCs) have become an integral part of nearly all of today's industrial processes. A PLC is a digital computer used for automation of electromechanical processes and is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. There are few major PLC makers such as Allen Bradley (AB), Siemens, Modicon, ABB, Mitsubishi, GE, Omron, Bosch, Fuji, and Toshiba, but AB maintains about 80 % of the market share in the United States. For the past 30 years, AB PLC solutions have significantly evolved from PLC-2 all the way to PLC 5000 series, with several configurations in-between. The most recent AB PLC has endless functionality, including programming using functional blocks, multitasking, and communication capabilities and therefore is the most widely employed by the industry. It is also very common that PLC systems integrate with robotic solutions to enhance automation processes. As a result, the skills of newly employed industrial workers must include knowledge of PLC and robotic systems, as well as how to integrate these two systems together in one efficient automated process suiting the requirements of the modern industrial environment.

In recent years, there have been significant changes in engineering education, especially in electrical and computer engineering education, both in terms of content and delivery. With the advent of computers, learning through computer-based environments has dramatically increased [1, 2]. The high demand in engineering professionals equipped with relevant and up-to-date PLCs skills drives engineering education to develop an alternative to the standard in-class instruction approaches. Traditional approach of teaching PLCs assumes the training to be done on actual equipment. Theory and exercises are integrated into a course to improve and perfect student skills. The conventional way of performing an experiment is to be physically present in the laboratory. Students work in groups of two to three in a laboratory and receive help from an instructor. While the traditional way of teaching PLCs is very effective, the disadvantages of traditional labs include the need for a qualified person/teacher, the demand for institutional space, time constraints on the students and the institution, crowded classes, safety problems, recurring maintenance, etc. [2].

Because of resource limitations and the complexity and importance of the subject matter, there is a great need for instructional technologies and methodologies that can allow greater access to PLC educational resources and communicate complex concepts in ways that help engineering students to learn more effectively and pique the interest of prospective students [3].

Web-based presentations of introductory PLC concepts are relatively easy to find [3-5], and several websites provide free PLC information such as book chapters, programs for download, and online Q&A [3, 6, 7]. Grand Valley State University has a free downloadable e-book [3, 8]. However, tutorials that provide opportunities for practice with feedback are harder to find, and tutorials that adapt practice and feedback to the needs of individual learners appear to be non-existent. Some websites provide ladder logic editors for free download [3, 9]; however, not all of these editors simulate program execution.

Two non-Web-based systems are notable for commercial software. LogixPro, available for low cost, provides a good PLC simulation environment and comes with animations for various processes, such as traffic control and batch mixing, I/O configuration. This well designed simulation software shows how a ladder logic diagram relates to an automated process [10, 3]. On the other hand, LogixPro does not provide the basics on fundamental knowledge, such as Boolean algebra, digital gates, ladder notation, wiring, and syntax of individual instructions.

Blakley and Irvine report development of PLCSIM, a PLC simulator designed to assist in teaching ladder diagram programming, and PLCTUT, a multimedia teaching package uses audio, video, and animations to teach about PLC hardware and programming [3, 11]. However, their systems are tightly linked to a particular brand of PLC, the Toshiba EX20PLUS.

The Electrical Engineering Technology (EET) and Computer Science (CS) Departments at Michigan Tech is ready to take the challenge and collaborate with Bay Community College to develop a PLC curriculum to help solve the current shortage and future expected growth in control engineering professionals required to have relevant PLC skills within the state and beyond. This project will develop and implement the open source, multi-level, interactive PLC software to be used in high schools, community and four-year colleges, as well as training displaced workers wishing to improve their knowledge and expertise in the subject matter and to meet the changing needs of industry.

We will work with a faculty expert on PLCs and build three levels of software so that students can select the most appropriate version of the module for their situation. For example, it will be flexible enough to be used to get high school students interested in programming PLCs, could be used to train displaced workers seeking a certificate or two-year degree, or be used in more advanced courses that are part of a four-year degree. Each level of the learning module will include multimedia materials—video, audio, and/or electronic documents that introduce the content presented in the module. We plan to focus our implementation on programming Allen Bradley PLCs, which are most commonly used in industry. However, the basic skills that the students learn while completing the learning modules would be easily transferable to other systems.

University Overview

Michigan Technological University is a public university committed to providing a quality education in engineering, science, business, technology, communication, and forestry. In Fall 2013, total enrollment was 6,979 students, including 1,333 (19.1%) graduate students. Over 65% of Michigan Tech students are enrolled in engineering and technology programs [12].

The School of Technology [13] offers programs covering the entire spectrum of technology. The School of Technology awards bachelor's degrees in computer network & system administration, construction management, electrical engineering technology, mechanical engineering technology, and surveying engineering.

The EET program [14] at Michigan Tech offers a Bachelor of Science in Electrical Engineering Technology. The program is applications-oriented and focuses on preparing graduates for entry into the workforce upon graduation. Graduates of the program are electrical engineering technologists with career options in microcontroller applications, robotics, industrial automation, instrumentation, and control.

The Computer Science Department [15] at Michigan Tech has had a long-standing reputation of outstanding educational programs enabling students to grow with and adapt to rapidly changing technologies. The diversity of courses offering includes computer science, computer systems science, software engineering, and support for computer-related degree programs across Michigan Tech campus.

Open-Source Software Development Model

This project aims to develop a free and open-source website that introduces students to PLCs and SCADA systems. Once complete, anybody interested in learning more about PLCs can learn from the course materials and then complete interactive problems and games. The website will aim to attract the public and can be customized by instructors who wish to use the website in conjunction with a course at a university or community college. For example, course instructors will be able to log in and customize which curriculum modules are available to the students, monitor the progress of their individual students through the course, and use the system in course labs where students can collaborate or compete to quickly and accurately complete problems and modules. We also plan to collect feedback about the course and make improvements. Since we plan to make the software open source, other institutions will be able to make significant customizations and contribute back to us or to take those customizations and create their own version of the website.

The software that powers the website will be developed by a team of consisting of a faculty member, a graduate student, and multiple undergraduate students. This project will be a unique learning experience for the students working on the project. Students will learn about how to budget time, respond to unforeseen obstacles, and satisfy the people who are using their product. The team will also gain experience using agile software development methods so that they can rapidly respond to changes and feedback. The multidisciplinary nature of the overall project involves giving computer science and engineering students the opportunity to work together.

We can choose from many different tools to develop this Web-based learning environment. One of our goals is to make the website that we develop flexible enough to run on a variety of computers without the need to install any software. Therefore, the software will be easy for schools to implement; any computer with a Web browser will be able use the software. We are using the HTML5 standard recommended by the World Wide Web Consortium. We will use the JavaScript programming language to allow users to interact with the software. To speed up the development process, we plan to use open source libraries. For simple Web components like dialog, tab, button, drag & drop, and widgets, we use the popular and well-documented jQuery library. We also plan to use jsPlumb, which provides the ability to connect Web components with style lines. This capability is useful when drawing a logic diagram or using ladder logic to program a PLC.

We are currently working on the preliminary versions of the first two curriculum modules and are simultaneously developing the more complicated software necessary to implement a PLC simulator. The first two modules introduce students to binary/decimal conversion, logic gates, and truth tables. We have also designed some of the software and data structures necessary for the simulator. For the simulator core, an adjacency matrix will store the ladder logic information with each cell representing a connection between two nodes. Each node will have a type (switch, relay, I/O, etc.), current state (open or closed), and function parameters (like the amount of time for a timer node). When we simulate the behavior of the ladder logic, we update the matrix row by row with an algorithm that simulates the behavior of the PLC. The rate we update the matrix is adjustable to ensure that it results in realistic simulation results. When the simulator is complete, we hope to present a PLC sandbox where students can try writing different ladder logic programs and to present students with individual scenarios that they must complete. We plan to present users with a visual representation of the scenario (for example, a garage door opener), so that they can interact with the system and verify its correctness.

Husky Game Development Enterprise

Many of the students developing the software for this project will come from Husky Game Development (HGD) Enterprise [16]. HGD is a course at Michigan Tech that provides students with an environment that approximates a real-world game development studio. HGD differs significantly from traditional courses. In a typical traditional course, students attend lectures, participate in discussions, and independently complete well-defined assignments. In HGD, students self-organize into teams, develop a video game, and then work together to create the video game over the course of two or more semesters. HGD is managed both by the faculty advisor and five student managers who are elected their peers or selected by the faculty advisor. Managers are responsible for evaluating the progress of teams, setting deadlines, developing content for lectures, pursuing industry sponsorships, and supporting previously released games. Although this project is not solely a game, Husky Game

Development has motivated and capable students who will find creative ways to add gaming elements to the website that will engage users.

Three Levels of the PLC Educational Model

To fulfill the knowledge requirements for various layers of education, three levels of the PLC educational model will be developed in the course of this three-year project.

Level 1 (Beginner)

This level will use a game-based approach to target high school students, summer camps, and summer youth programs with the goal of preparing high school students for college-level PLC courses. In this level, students will learn the basic concepts of digital logic and PLC programming using interactive and hands-on game based approaches. When students complete level one, they will have learned the basics of digital logic, Boolean algebra, and game-based basic PLC programming scenarios.

Level 2 (Intermediate)

This section will target community college students, displaced workers, and industry representatives desiring to improve their skill set. The main course will be devoted to learning the intermediate concepts of PLC programming based on industrial, manufacturing scenarios (pick and place, palletizing, etc.). We plan for this level to include several approaches such as game-based "Show Me, Guide Me, Test Me, Figure it Out." This level will begin with a short review of Level 1 material via gaming applications. The main course will be conducted utilizing "Show Me, Guide Me, Test Me, Figure it Out" approach. During the "Show Me" stage, students will observe the demonstration of learning objectives utilizing predefined industrial processes involving PLC programming. During the "Guide Me" stage, students will be guided to interactively practice the skills learned in the demo of the "Guide Me" stage. The "Test Me" stage will be used to test the understanding of learning objectives by working with the proposed simulator. After students successfully complete this stage, the "Figure it Out" mode will allow them to check their understanding of the learning objectives by working on an altered scenario modification to be introduced in the scenario used during the "Test Me" stage, requiring the student to modify the PLC program to accommodate for the changes.

Level 3 (Advanced)

The advanced level will target advanced community college students, four-year college students, displaced workers with a background in PLC, and industry representatives with a background in PLC desiring to advance their skill set. Similar to the previous level, level 3 will use a "Show Me, Guide Me, Test Me, Figure it Out" approach. At the beginning of the level, all of the material from the previous levels will be available to students to review. Unlike level 2, this advanced level will cover program control instructions, data manipulation instructions, math instruction, sequencer and shift register instructions, PLC installation practices, editing, troubleshooting, process control, network systems, and SCADA.

Modules Development Approach

Scenarios, games, and quizzes are critical aspects of the Web-based training program. Together, these items provide reinforcement to lectures, a way to measure knowledge, and to some extent create a virtual lab environment that can be accessed by geographically separated students. These features will be integrated into each module, enhancing the user's experience.

Module 1

The module focuses on the binary number system and memory. The interactive aspects of this module will focus on teaching the user how to manipulate this number system. The first portion of this set of games will help the user become familiar with converting base 10 numbers into base 2, binary numbers. The game will allow the user to input four numbers into the program to convert into binary. The program will check that at least one number is greater than a determined limit but less than 255. It will also check to make sure that there are no repeats or negative numbers. These checks will ensure that the numbers selected will not be too complicated or too easy for the user to convert.

Once the four numbers are selected, the program moves on to the converting portions of the game. The basis of the conversion from decimal to binary is based on the "dividing by two" method. Here the user will divide the number by two and record the quotient and the remainder. Next, the quotient will be divided by two, and again the new quotient and remainder will be noted. This process will continue until the user gets a quotient of 0 with a remainder of 1. From there, the user will put the remainder values in proper order to determine the binary number, adding additional 0s to make an 8-bit number. When the student gets the correct answers, the process will repeat for the remaining numbers.
There was some discussion about the method used to teach the students how to convert decimal to binary. A popular method of conversion, especially in the computer science realm, is simply knowing the powers of 2 and how they go into a number. For individuals familiar with this method, the process of converting a decimal number to binary is quick. This issue is it can be confusing for students that are not used to the idea of binary or do not easily remember the powers of 2. The "dividing by two" method only requires students to understand how to half numbers and understand when they get a remainder.

Once the four numbers are converted, the game will take the converted binary numbers and put them into a randomly sized memory table. Here the user will be asked basic questions on generic memory to test their general understanding. Through the table, they will determine the size or the size of address required for the memory. They will be also asked how many bits, bytes, and words are in the memory.

Once the memory portion of the game is completed, the user will begin the final portion of Module 1. Here users will convert the binary form of their selected numbers back into decimal. The general process for this part of the game is determining which powers of 2 are in the binary number and adding them together to determine the final value. Users will start with the binary number laid out in a table. They will drag powers of 2 over the 1s digits. Once the powers of 2s are properly laid out, the program will convert them into their decimal numbers and ask the user to add the values together and input the correct answers.

Module 2

The interactive portion consists of two portions: “test your knowledge” and a logic circuit simulator. The “test your knowledge” portion displays logic circuits to users and asks them to input the truth table. The circuits are ranked according to difficulty, and the overall difficulty will have a difficulty weight. The program randomly selects circuits from a pool of circuits and populates the quiz with the circuits selected. The difficulty weight keeps the program from selecting too many easy or hard circuits for the selected difficulty.

The second portion of Module 2 will be a logic circuit simulator. This simulator has two modes: problem solving and sandbox. In the problem-solving mode the user has to construct logic circuits from pre-determined toolbox to perform certain tasks. These tasks are determined by user requirements, essentially word problems. The user has to interpret the description, wire the circuit, test it, and submit it for approval. If the circuit is correct, then the user moves on to the next problem. The sandbox mode allows the user to create logic circuits of individual choosing. This model encourages learning through experimentation. This also allows any instructor utilizing the program a basic program to perform a logic circuits lab.

Module 3

This interactive portion revolves around the physical parts of the PLC and input/output devices that are typically used with PLCs. Users have to identify the various parts of a PLC, sensors, and output devices. There is also an assignment where the user has to select the proper device based on a description provided.

Module 4

Module 4 introduces the virtual PLC system. Here the scenarios focus on programming in ladder logic. The module starts by asking users to build programs in ladder logic to mimic logic gates and circuits. Once they have mastered converting logic circuits into ladder logic, they are assigned to solve word problems. All these problems focus on the “examine if open” input, “examine if closed” input, and general outputs. The user learns how to use branches and latch an output on. A very simple sandbox mode is implemented as well.

Future Modules

Additional modules will focus on more advanced PLC topics. Concepts of timers and counters will be introduced. Students will learn about SCADA and how PLCs are integrated in an industrial environment. A fully functioning PLC simulator will eventually be added along with a virtual industrial workstation and a basic water treatment control simulator.

With all these scenarios and quizzes, the program will provide feedback to both the user and the instructor utilizing the Web-based learning module. The program will inform users when they make incorrect choices in their design or answer a question incorrectly. It will provide hints if needed and eventually show the answer if the user cannot figure out how to fix the mistake. Correct and incorrect answers and “show help” selections can be sent back to a database for instructor review.

Conclusion

Academic programs at Michigan Tech are designed to prepare technical and/or management-oriented professionals for employment in industry, education, government, and business. The EET and CS Departments at Michigan Tech are collaborating with Bay Community College to develop a PLC curriculum to help solve the current shortage and future expected growth in control engineering professionals required to have relevant PLC skills within the state and beyond. This project will develop and implement the open source, multi-level, interactive PLC software to be used in high schools, two- and four-year colleges, as well as training displaced workers wishing to improve their knowledge and expertise in the subject matter and meet the changing needs of the industry.

The learning system will be hosted by Michigan Tech and made freely available for anybody around the world to use over the Internet. At Michigan Tech, we plan to integrate the materials into high school outreach programs and into PLC courses. Bay College and other interested instructors or students will be able to use the material independently or in conjunction with a class. When the system is used in conjunction with a class, instructors will be able to connect with their students and monitor their progress. Given the remote location of Michigan Tech, this online PLC education system will allow us to reach a much larger audience. We will solicit feedback from the people who use the system and make improvements based on that feedback. Besides the benefits to the students who use the PLC learning system, the project will also give undergraduate and graduate students hands-on experience working on a large-scale project with multiple people that span different disciplines. We hope that these hands-on experiences will encourage students to think about how they can use their own skills in an entrepreneurial way to improve the lives of others. We also plan to build student competency by structuring some of the learning modules as games where students can work collaboratively or competitively to solve PLC programming challenges. We plan to use metrics, such as the time it takes to solve a particular problem as well as the complexity or efficiency of their solution as game metrics.

This three-year project devoted to the design and software implementation is in its first phase. In this paper, we present the preliminary techniques, approaches as well as the overall project vision on the effective PLC learning/gaming system development, and implementation, but more work is on the way.

References

[1] Toral, S. L., Barrero, F. & Martinez-Torres, M. R.. (2007). Analysis of Utility and Use of a Web-Based Tool for Digital Signal Processing Teaching by Means of a Technological Acceptance Model. *Computing Education*, *49*, 957–975.

[2] Aydogmus, Z., & Aydogmus, O. (2009). A Web-Based Remote Access Laboratory Using SCADA. *IEEE Transactions on Education, 52*(1), 126-133.

[3] S. Hseih, S., & Hseih, P. (2005). Web-Based Modules for Programmable Logic Controller Education. *Computer Applications in Engineering Education, 13*(4), 266-280.

[4] Siemens Basics of PLCs Course. (n.d.). Retrieved from http://www.sea.siemens.com/step/templates/lesson.mason?plcs:1:1:1

[5] PLCS.net Instructional Materials. (1996-2012). Retrieved from http://www.plcs.net/contents.shtml

[6] Forum.MrPLC.com. (n.d.). Retrieved from http://forum.mrplc.com

[7] Control.com. (1999-2014). Retrieved from http://www.control.com

[8] *PLC Book*. (n.d.). Retrieved from http://engineeronadisk.com/V2/book_PLC/engineeronadisk.html

[9] Global Controls. (n.d.). Retrieved from http://www.globalcontrols.net

[10] *The Learning Pit*. (1999-2012). Retrieved from http://www.thelearningpit.com/

[11] Blakley, J. J., & Irvine, D. A. (2000), Teaching Programmable Logic Controllers Using Multimedia-Based Courseware. *Internationl Journal of Electrical Engineering Education* , 37, 305-315.

[12] Michigan Technological University Admission Office. (n.d.). Retrieved from http://www.mtu.edu/admissions/

[13] School of Technology at Michigan Tech. (n.d.). Retrieved from http://www.mtu.edu/technology/

[14] Electrical Engineering Technology Department at Michigan Tech. (n.d.). Retrieved from http://www.mtu.edu/technology/undergraduate/eet/

[15] Computer Science Department at Michigan Tech. (n.d.). Retrieved from http://www.mtu.edu/cs/department/about/

[16] Husky Games Development Enterprise. (2014). Retrieved from http://www.huskygames.com

Biography

ALEKSANDR SERGEYEV is currently an associate professor in the Electrical Engineering Technology program in the School of Technology at Michigan Tech. Dr. Sergeyev earned his bachelor degree in Electrical Engineering at Moscow University of Electronics and Automation in 1995. He obtained the master's degree in Physics from Michigan Tech in 2004 and the Ph.D. degree in Electrical Engineering from Michigan Tech in 2007. Dr. Sergeyev's research interests include high energy laser propagation through the turbulent atmosphere, developing advanced control algorithms for wavefront sensing and mitigating effects of the turbulent atmosphere, digital inline holography, digital signal processing, and laser spectroscopy. Dr. Sergeyev is a member of ASEE, IEEE, SPIE and is actively involved in promoting engineering education.

NASSER ALARAJE is currently the Electrical Engineering Technology program chair and associate professor at Michigan Tech. Dr. Alaraje research interests focuses on processor architecture, System-on-Chip design methodology, Field-Programmable Logic Array (FPGA) architecture and design methodology, Engineering Technology Education, and hardware description language modeling. Dr. Alaraje is a Fulbright scholar; he is a member of American Society for Engineering Education (ASEE), a member of ASEE Electrical and Computer Engineering Division, a member of ASEE Engineering Technology Division, a member of Institute of Electrical & Electronic Engineers (IEEE), and a member of Electrical and Computer Engineering Technology Department Heads Association (ECETDHA).

SCOTT KUHL is an assistant professor of Computer Science and an adjunct assistant professor of Cognitive & Learning Sciences at Michigan Tech. He received his Ph.D. in Computer Science from the University of Utah in 2009. He has been the faculty advisor for Husky Game Development Enterprise since Spring 2010. His research interests include immersive virtual environments, head-mounted displays, and spatial perception.

BOCHAO LI is a Ph.D student of computer science at Michigan Tech. He received a bachelor's degree in Electrical Engineering from Michigan Technological University in 2011. His research interests include immersive virtual environment, human computer interactions, computer graphics.

KURT KALENAUSKAS is an electrical engineering technology undergraduate student in the School of Technology at Michigan Techn. He is specializing in industrial automation and control and has several years of experience in the manufacturing sector, including steel, food, and household goods. Kurt Kalenauskas has also served as an aircraft mechanic for the United States Air Force, serving in both Operation Enduring and Iraqi Freedom.