

Culminating Phase of Open Source Programmable Logic Controller Software Development Initiative for High School Students, Two- and Four -Year Colleges, and Displaced Workers

Prof. Aleksandr Sergeyev, Michigan Technological University

Aleksandr Sergeyev is currently an Associate Professor in the Electrical Engineering Technology program in the School of Technology at Michigan Technological University. Dr. Aleksandr Sergeyev earned his bachelor degree in Electrical Engineering at Moscow University of Electronics and Automation in 1995. He obtained the Master degree in Physics from Michigan Technological University in 2004 and the PhD degree in Electrical Engineering from Michigan Technological University in 2007. Dr. Aleksandr 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.

Dr. Nasser Alaraje, Michigan Technological University

Dr. Alaraje is an Associate Professor and Program Chair of Electrical Engineering Technology in the School of Technology at Michigan Tech. Prior to his faculty appointment, he was employed by Lucent Technologies as a hardware design engineer, from 1997- 2002, and by vLogix as chief hardware design engineer, from 2002-2004. Dr. Alaraje's research interests focus 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 2013-2014 Fulbright scholarship recipient at Qatar University, where he taught courses on Embedded Systems. Additionally, Dr. Alaraje is a recipient of an NSF award for a digital logic design curriculum revision in collaboration with the College of Lake County in Illinois, and a NSF award in collaboration with the University of New Mexico, Drake State Technical College, and Chandler-Gilbert Community College. The award focused on expanding outreach activities to increase the awareness of potential college students about career opportunities in electronics technologies. Dr. Alaraje is a member of the American Society for Engineering Education (ASEE), a member of the ASEE Electrical and Computer Engineering Division, a member of the ASEE Engineering Technology Division, a senior member of the Institute of Electrical & Electronic Engineers (IEEE), and a member of the Electrical and Computer Engineering Technology Department Heads Association (ECETDHA).

Prof. Scott A Kuhl, Michigan Technological University

Scott Kuhl is an Associate Professor of Computer Science and an Adjunct Associate Professor of Cognitive & Learning Sciences at Michigan Technological University. He received his Ph.D. in Computer Science from the University of Utah in 2009. His primary research interests include immersive virtual environments, head-mounted displays, and spatial perception. A link to his web page can be found at <http://www.cs.mtu.edu/>.

Mr. Bochao Li, Michigan Technological University

Undergraduate in Electrical Engineering, and currently working on Ph.D on Computer Science

Culminating Phase of Open Source Programmable Logic Controller Software Development Initiative for High School Students, Two and Four Year Colleges, and Displaced Workers.

Abstract

Team of undergraduate and graduate students advised by faculty members at Michigan Tech are developing a set of open source and online learning modules, which give students interactive, hands-on experience with programming Programmable Logic Controllers (PLCs) on a standard desktop or laptop computer. The developed software represent different levels of difficulty and students can select the most appropriate version of the module for their knowledge base. It is 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 learning module includes multimedia materials including video, audio, and/or electronic documents that introduce the content presented in the module. While progressing through each module, students can access their comprehension of the presented material using computerized tests testing.

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, instructors will be able to connect with their students and monitor their progress. The learner competency can also be tested by structuring some of the learning modules as games where students can work collaboratively or competitively to solve PLC programming challenges. In this paper, we present the techniques and approaches used for the PLC system development and implementation.

Introduction

With current advancements and reconfigurability of manufacturing, the Programmable Logic Controllers (PLCs) become an integral part of nearly all 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 pertains about 80 % of the market share in the United States. For the past 30 years, the AB PLC solutions have significantly evolved from PLC-2 all the way to PLC 5000 series with several configurations being in between. The most up-to-date AB PLC has endless functionality including, programming with functional blocks, multitasking, and communication capabilities. It is also very common that PLC systems are integrated with robotic solutions to enhance automation processes. As a result, the skills of newly employed industrial workers must include the 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 modern industrial environment.

In recent years, there have been significant changes in engineering education, especially in electrical and computer engineering fields, both in terms of the content and its 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 the engineering education to develop the 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 on the institution, crowded classes, safety problems, recurring maintenance, and etc². Because of resource limitations, the complexity and importance of the subject matter, there is a great need for instructional technologies and methodologies that can allow for 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 students³.

Web-based presentations of introductory PLC concepts are relatively easy to find³⁻⁵. There are several web sites that provide free PLC information such as book chapters, programs for download, and online Q&A^{3,6,7}. The Grand Valley State Michigan Tech 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 web sites provide ladder logic editors free downloads^{3,9} however, not all of these editors simulate program execution. A non-web-based commercial software LogixPro, available for low cost, provides a good PLC simulation environment and comes with various animations for various processes, such as traffic control and batch mixing, I/O configuration. This well design software demonstrates how a ladder logic diagram relates to an automated process^{3,10}, however 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 that 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) Department at Michigan Tech collaborate with 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 and up-to-date PLC skills within the State of Michigan and beyond. This project develops and implements the open source, multi-level, interactive PLC software to be used in high schools, two and four year colleges, as well as to training the displaced workers wishing to improve their knowledge and expertise in the subject matter and to meet the changing needs of the industry. The main objective of this project is developing of three levels of the PLC simulation software so students can select the most appropriate level of difficulty that fits his/her knowledge. For example, it needs 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 includes multimedia materials, including video, audio, and/or electronic documents, which introduce the content presented in the module. Considering

common utilization of Allen Bradley PLCs in industry, it has been chosen as a base platform for the developing simulation software. However, the basic skills that the students learn while completing the learning modules would be easily transferable to other systems.

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 completed, anybody interested in learning more about PLCs can learn from the course materials and then complete interactive problems and games. The website aims to attract both the general public and can also be customized by instructors who wish to use the website in conjunction with a course at a university or community college. Highly customizable software will allow course instructors to use it in various ways: to log in and adapt specific curriculum modules and make them available to the students; monitor the progress of the individual students through the course; and use the system in course labs where students can cooperate to quickly and accurately complete problems and modules.

A team consisting of faculty members from both departments, a graduate student, and several undergraduate students develops the software that powers the website. This project provides a unique learning experience for the students working on the project. Students learn about how to budget time and respond to unforeseen obstacles. The team also gains 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. There are many different tools, which could be used to develop this web-based learning environment. One of the goals of this project is to make the website flexible enough to run on a variety of computers without the need to install any software. Authors selected using the HTML5 standard recommended by the World Wide Web Consortium. The JavaScript programming language has been used to allow users for interaction with the software. To speed up the development process, authors use open source libraries. For simple web component like dialog, tab, button, drag & drop and widgets, the popular and well-documented jQuery library has been chosen for implementation, and utilization of jsPlumb provides the ability to connect web components with style lines. The last capability is useful when drawing a logic diagram or using ladder logic to program a PLC.

To this date, the PLC simulation web-based software is nearly complete. It consists of nine main curriculum modulus and three supplementary modules as shown in Figure 1. The main modulus 1-9 are devoted to general concepts that every student needs to know to become proficient in programming PLCs. Module 10 teach students additional skills of PLC installation, troubleshooting and safety. In module 11, concept of Supervisory Control and Data Acquisition (SCADA) with implementation of Human Machine Interface (HMI) is presented. Lastly, in module 12, students have an opportunity to test their knowledge working with fully simulated water treatment facility and conduct programming of its components and also run and troubleshoot its operation directly from the system HMI.

Module 1 focuses on the binary number system and memory. The interactive aspects of this module focuses on teaching the user how to manipulate this number system. The first portion of this set of games are designed to help the user become familiar with converting base ten numbers

into base two, binary numbers. The game allows the user to input four numbers into the program that they wish to convert into binary. The program checks that at least one number is greater than a determined limit but less than 255. In addition, it checks to make sure that there are no repeats or negative numbers. These verification tools 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 done utilizing the “Dividing by Two” method. When the student gets the correct answers, the process repeats for the remaining numbers. There was some discussion on the method used to teach the students how to convert decimal to binary.

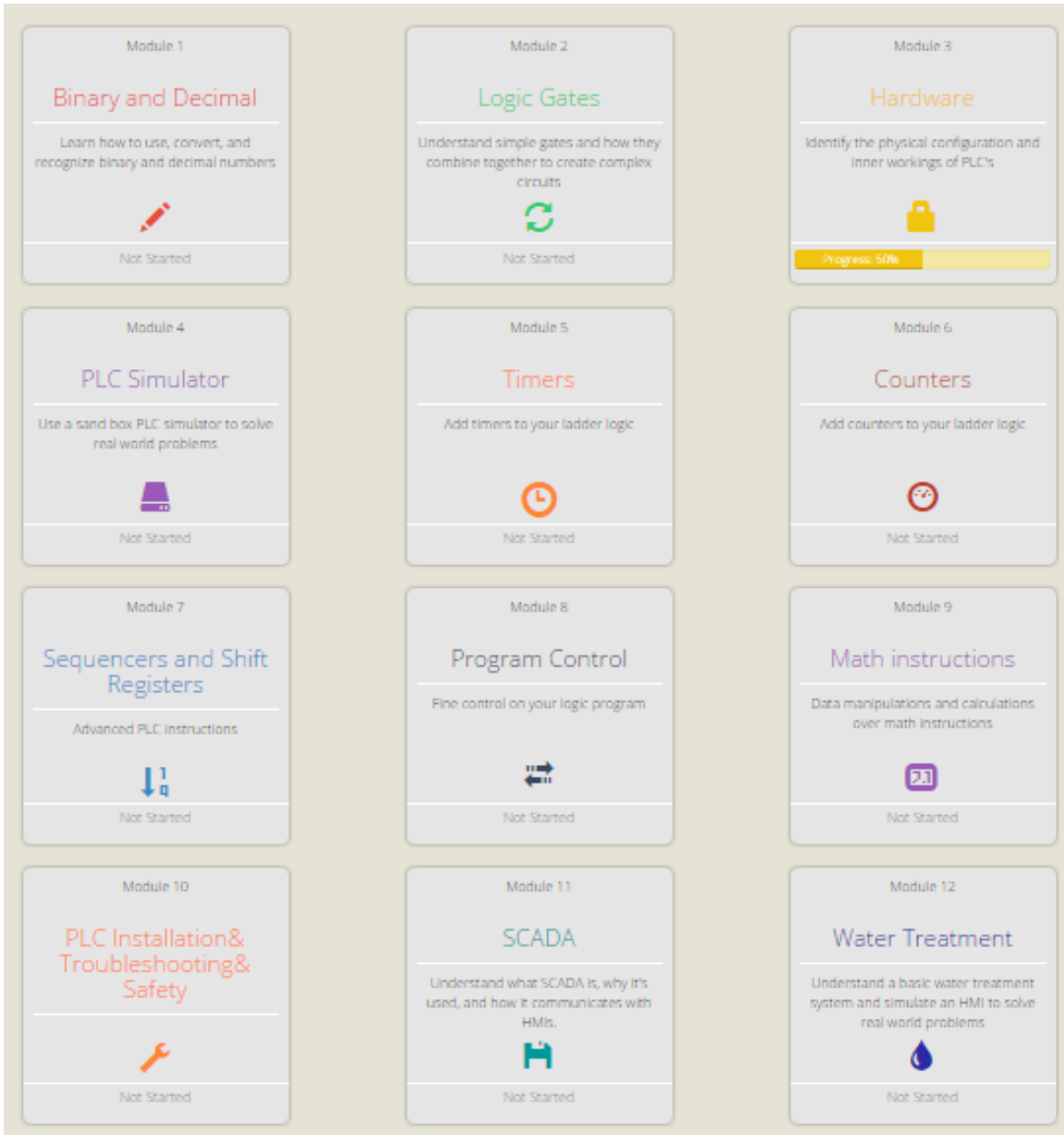


Figure 1: Layout of the PLC web-based simulation software main page

A popular method of conversion, especially in the computer science realm, is to understand the powers of two. For individuals familiar with this method, the process of converting a decimal number to binary is quick, however, it can be confusing for students that are not used to the idea of binary or do not easily remember the powers of two. 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 is 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. Once the memory portion of the game is completed, the user begins the final portion of module 1. Here the user converts the binary form of their selected numbers back into decimal. The general process for this part of the game is determining which powers of two are in the binary number and adding them together to determine the final value. Sample screen shoot of Module 1 layout is shown in Figure 2.

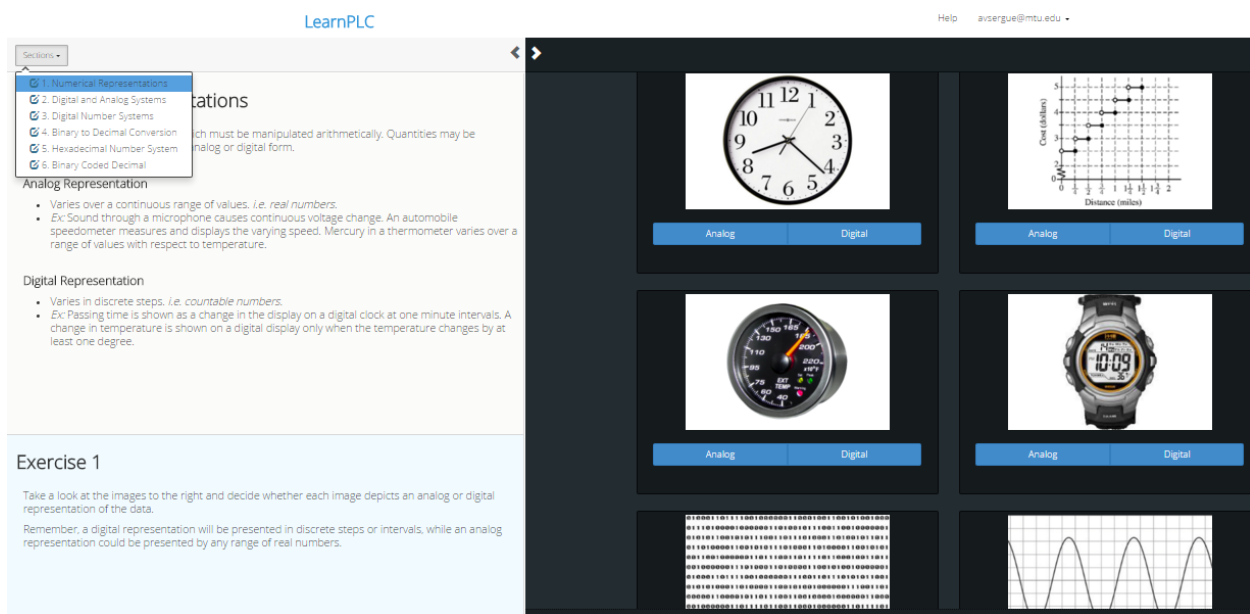


Figure 2: Sample screen shoot of Module 1 layout

Module 2 interactive portion consists of two portions: “test your knowledge” and a logic circuit simulator. The “test your knowledge” portion displays logic circuits to the user and ask them to input the truth table. The circuits are ranked in difficulty and the overall difficulty is weighted as well. 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 is a logic circuit simulator. This simulator has two modes: problem solving and sandbox. In problems 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 they move on to the next problem. The sandbox mode allows the user to create logic circuits of their own choosing. This model encourages learning through experimentation.

This also allows any instructor utilizing the program a basic program to perform a logic circuits lab.

Introduction to Logic Gates

In this module we will study the ways of describing how systems using binary logic levels to make decisions.

Boolean Algebra

Boolean algebra is an important tool in describing, analyzing, designing, and implementing digital circuits.

- Boolean algebra allows only two values: 0 and 1.
- Logic 0 can mean: false, off, low, no, open switch.
- Logic 1 can mean: true, on, high, yes, closed switch.
- There are three basic logic operations: OR, AND and NOT.

Truth Tables

A truth table describes the relationship between the input and output of a logic circuit. The table is composed of one column for each input variable and one column for the output of the logical operation.

Each row contains one possible configuration of 0's and 1's that the input variables could take. In this way the number of entries corresponds to the number of inputs. For example a 2 input table would have 2² or 4 entries. A 3 input table would have 2³ or 8 entries.

Example of a truth table and circuit diagram of an undefined operation.

Exercise 1

Study the three basic logical operations and their corresponding gates on the right. Then continue to the next section.

OR, AND, and NOT Gates

OR Operation

The Boolean expression for the OR operation is $X=A+B$

- This is read as "X equals A or B."
- X will equal 1 when A or B equals 1.
- The OR operation is similar to addition but where A and B are 1, the OR operation produces 1+1 = 1.
- Note that evaluation of the expression stops once a 1 is found. This means that, though counterintuitive, a statement with two 1 inputs will evaluate to true as well.

Truth table and circuit symbol for a two input OR gate.

A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

AND Operation

The Boolean expression for the AND operation is $X=A \cdot B$.

- This is read as "X equals A and B."
- X will equal 1 when A and B equal 1.
- The AND operation is similar to multiplication. The AND operation produces 1 only when A and B are 1.

Truth table and circuit symbol for a two input AND gate. Notice the difference between OR and AND gates.

A	B	x = A · B
0	0	0
0	1	0
1	0	0
1	1	1

NOT Operation

Evaluating Logic Circuit Outputs

The three basic Boolean operations (OR, AND, NOT) can describe any logic circuit. Below we will discuss some common methods to interpret logic circuits and evaluate their output.

Describing Logic Circuits Algebraically

If an expression contains both AND gates and OR gates the AND operation will be performed first, unless there is a parenthesis around the OR expression.

The output of an inverter is equivalent to the input with a bar over it. Input A through an inverter is \bar{A} .

Exercise 2

Review the truth table exercises to the right then press 'Continue' to go to the next section.

Evaluating using a Truth Table

A truth table can be used for not only evaluating output for a certain input configuration, but for all possible input configurations.

First we need to have an algebraic expression for the circuit.

Let's use the circuit diagram we used in 'Evaluating a Circuit Diagram'.

To build the expression, we can use a method similar to the one used to evaluate the output. Start from each input and proceed through each gate. However, instead of writing down each output, we will instead write the resulting algebraic expression. We do this until we get the final Boolean expression.

Once we have the Boolean expression, we can construct the truth table. You should have 4 input columns and 1 final output column. The number of rows, recall from section 1, will be 2⁴ = 16.

You may also have columns for intermediate expressions. These can prove useful when evaluating more complex expressions.

First fill in the input columns with all the different input configurations. Then evaluate the expressions in each column until you reach the final expression.

A	B	C	D	\bar{A}	$\bar{A} \cdot B \cdot C$	A + D	$\bar{A} + D$	X = $(\bar{A} \cdot B \cdot C) \cdot (\bar{A} + D)$
0	0	0	1	1	0	0	1	0
0	0	1	1	1	0	1	0	0
0	1	0	1	1	0	0	1	0
0	1	1	1	1	0	1	0	0
0	1	0	0	1	0	0	1	0
0	1	1	0	1	0	0	1	0
1	0	1	1	0	0	1	0	0
1	0	1	0	0	0	1	0	0

Figure 3: Sample screen shoots of Module 2 learning materials

Module 3 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. Figure 4 shows learning modules of Module 3. The upper picture shows PLC components design and the one below demonstrate the physical systems of tank filling application and how the PLC can be used to control it.

Creating a PLC component diagram.

Each physical PLC can be separated into six distinguishable parts, each with its own purpose, they are:

- Input.** The input receives commands from a source and passes them on to the grouping of parts that handles the controls of the PLC. The input source may be a human operator or another machine.
- Output.** The output receives commands from the internal components of the PLC and performs actions based on given instructions. Actions might range from turning off a motor to sending a wireless signal.
- Power Supply.** The power supply unit supplies power to each individual element of the PLC.
- Central Processing Unit.** The brains of a PLC. The CPU receives information from input and uses the memory and communication modules to interpret the input conditions before passing it along.
- Memory.** The memory module is used by the CPU and communication modules to store and use the information given to the PLC.
- Communication.** Handles data transfer to outside sources, not always included.

Fill in the component diagram:

Exercise 1

Using the descriptions above, complete the flow diagram by dragging the each item to its correct position

Using Input and Output to Build a System

As discussed previously, the input and output modules of a PLC can take many forms. These devices are connected at strategic places in a system and information is fed to and from the computer, allowing a single PLC to control actions of tanks, light systems, motors and more.

Some of the basic devices are as follows:

- Motor** One of the most commonly used items, motors can be configured to drive wheels, spin objects, pull water and complete many more actions. Their speed, power and timing is controlled by the PLC.
- Float.** This device floats above a liquid surface and provides depth readings to the PLC.
- Thermometer.** As expected, a temperature gauge can be placed in a system to report a temperature reading back to the logic controller. This information is often used as a threshold value to determine when action needs to be taken, or as a safety precaution.
- Flow control.** The flow control is a reactive switch controlled by the PLC. It either allows liquid to pass through or stops the flow altogether.

Exercise 3

To the right is a setup with the purpose of heating liquid in a tank. The machine's process is as follows:

- Liquid is pumped into the tank
- A heating element activates and begins warming the water
- Once a set temperature is reached water is pumped out of the tank

Your job is to drag the PLC input and output components onto the tank in the correct location for the process to be completed

You have unlimited attempts to complete the system, feel free to experiment!

Figure 4: Sample screen shoots of Module 3 learning materials

Module 4 introduces the virtual PLC system. Here the scenarios focus on programming in ladder logic. The module starts by asking the user 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.

Module 5 Teaches concepts related to Timers. In this module, the user starts with timer basics learning how to configure PLC timer instruction and further continues with On-Delay, Off-Delay, Retentive and Cascading timers. Each section of this module is accompanied with few exercises.

Module 6 Introduces concept of Counter PLC instruction and its syntaxes followed by specialized Up- and Down-Counter instructions. Concept of cascading counters to achieve higher count limits is also presented in this module. Module culminates with exercises requiring combining Timer and Counter instructions for proper execution of presented to the user scenario.

Module 7 is devoted to the Sequencer and Shift Register instructions. To provide an easy understanding of a subject matter a mechanical sequencer analogy is introduced first. Once concept is comprehended, the PLC Sequencer instruction is introduced along with detailed instructions on how it should be properly configured to use with Ladder Logic. Considering that concepts of Timers and Counters were covered in the previous modulus, exercises that are more comprehensive and require more in-depth understanding of various PLC instructions are introduced to the user.

LearnPLC Help

Sections ▾
◀ ▶

Sequencer Programs


Sequencers require a false-to-true transition in order to operate. This allows for two different types of sequencer programs: event-driven and time-driven. Determining what type of sequencer program to use is a matter of how the program needs to run.

An event-driven sequencer program moves from step to step via a certain event happening. For example, every time a limit switch is enabled, move to the next step of the program.

A time-driven sequencer program is controlled on the basis of a timer, much like a mechanical drum sequencer. A timer instruction is used to enable the progression of the program steps by using the done bit of the timer to create a false-to-true transition on the sequencer instruction rung.

Exercise 3

Identify the type of the sequencer programs at the right side, and make sure you understand how they work.



3. The program we used in last section

4. Another PLC program

Having Trouble Continue

Figure 5: Sample screen shoots of Module 7 learning materials

Module 8 consists of 5 sections covering the following program control PLC instructions: Master Control Relay (MCR), Jump, Subroutine, Immediate Input (IIN), Immediate Output (IOT), Selectable Timed Interrupt, Fault Routine, Temporary End, and Suspend. The rationale behind each instruction is explained first, followed by various application scenarios and exercises for the user to practice and reinforce previously received theoretical knowledge.

Module 9 teaches commonly used in PLC programming mathematical instructions such as addition, subtraction, multiplication, and division. A word-level instructions including square root (SQR), negative (NEG), clear (CLR), binary coded decimals (BCD), and scale data (SCL) are introduced as well. As in previous modulus, the concept of reinforcing theoretical knowledge by means of exercises and case scenarios is extensively used in this module.

Module 10 covers important concepts of PLC installation, troubleshooting and safety. It starts off explaining requirements and proper codes to follow while installing the PLC in industrial environment. Considering universal nature of the PLC in terms of receiving and sending various AC and DC signals, which may be susceptible to electromagnetic interference, models of electrical noise and leaky inputs and outputs, are explained and followed by proper grounding, voltage variations and surges. Also, concepts of program editing, monitoring, as well as necessary steps to properly commission the system are explained. The module culminates by outlining preventive maintenance and troubleshooting procedures commonly employed in PLC industry

Module 11 is devoted to the following subjects: human machine interface (HMI), supervisory control and data acquisition (SCADA), data communication, and networking schemes. The last discusses the following: Data Highway, Serial Communication, DeviceNet, ControlNet, EnterNet/IP, Modbus, Fieldbus, and Profibus. Even though this module is more on the theoretical side, the user is still required to complete several exercises and check the subject comprehension by answering several question.

Module 12 is a specialized module devoted to the water treatment facility and how PLC and SCADA is used for its proper operation. In the introductory part of this module, the user study a basic water treatment system and its parts before presenting an interactive HMI simulator with scenarios that simulates the water treatment process. The basic water treatment system shown in Figure 6 has three parts: a Pump House, Filter Tanks, and a Water Tower. The pump house is the first stage. It draws water from a river and passes it to the filter tanks. A filter tank receives the water from the pump house and uses a filtering system to clean the water. There are always at least two filter tanks in a water treatment system. This is so that filtration can continue even if one of the tanks is malfunctioning. After the water is cleaned, it is pumped out to the water main with some being diverted to a water tower. The water tower stores the clean water to maintain water pressure in the water main. This pressure should run about 50 to 100 psi with about 600 gpm, gallons per minute. Upon studying the structure of the water treatment facility, the user is required to match the names of the three parts of a water treatment system to the diagrams and provided descriptions. Once this introductory task is successfully completed, the user is introduced to details of the system components and their operation. The system components are: Pump House, Filter Tank, Backwash, Water Tower. While studying the system components, the user is tasked to correctly answer sort test questions and complete some short simulation assignments. After the comfortable level of operating these system components is achieved, the user is provide with two exercises: Backwash with HMI and Backwash with Error HMI. In the first assignment, the user is asked to complete to complete water tank backwash procedure using HMI according to the following instructions:

“During your duty at the water treatment plant you notice the outflow pressure is low. You must perform a baskwash on the system and return it to normal. Turn on and off the pumps and valves

in the correct order to begin the backwash. Once the backwash animation begins, wait a few moments and then click on the dirty water drain pipe to 'clear' the debris. Turn on and off the pumps and valves again until water is filtering through the system normally again. If a step is missed or performed in the wrong order, the system will reset and you must begin again.

When you have accomplished this simulation, press 'Submit' and then 'Continue'”

During the second exercise, the user needs to perform the same backwash procedure but the HMI being in the faulty state which must be analyzed and fixed. The faults can be related to two causes: the physical part is broken or an HMI encountered. In each case must be repaired before use of HMI and backwash procedure can be performed. Below is the task received by the user:

“During your duty at the water treatment plant you notice the outflow pressure is low. You must perform a backwash on the system and return it to normal; however, during the process, a fault occurs. During this simulation, either the HMI or a random physical part will break and turn red to indicate that it is not working. In order to correct this can continue, you can press either the red "Send an Engineer" button or the dark blue "Refresh the HMI" button. If you refresh the HMI and the corresponding button is not green, then you must send the engineer to fix it.

If you send the engineer and the corresponding button is not green, then you must refresh the HMI to fix it. Once the buttons light green, you can continue as normal.

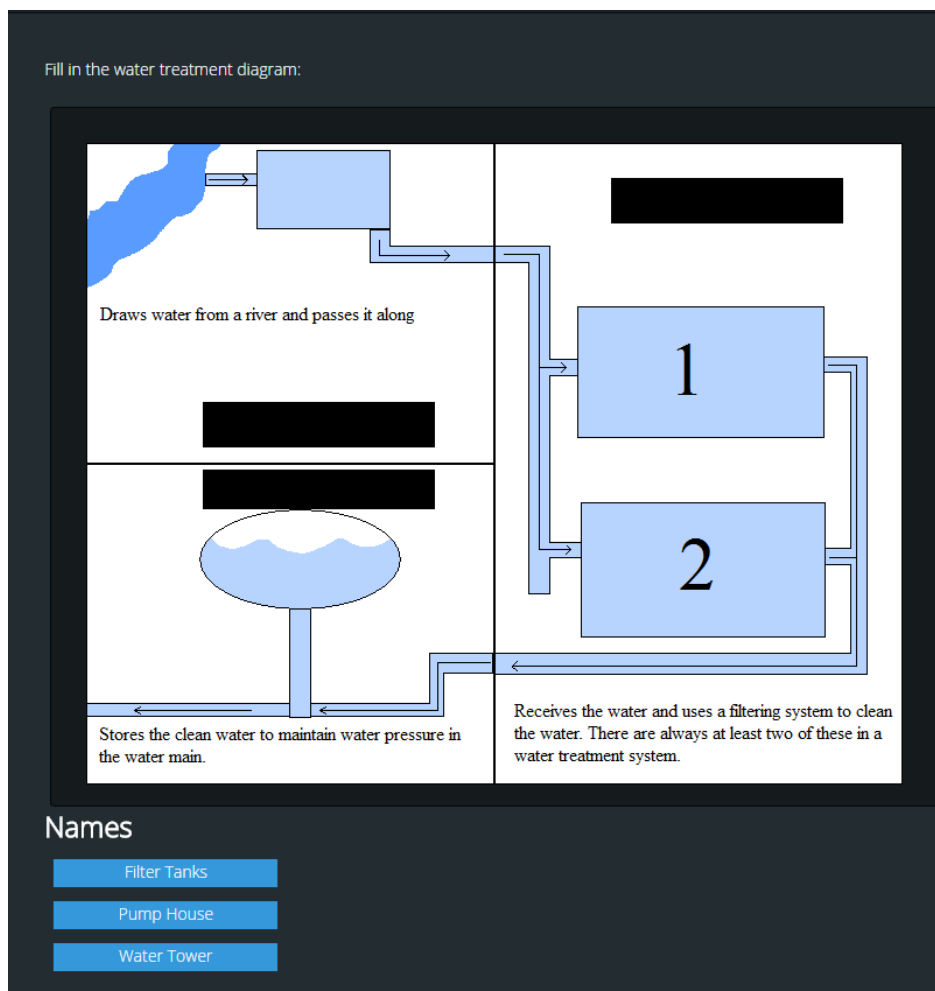


Figure 6: Exercise for water treatment system.

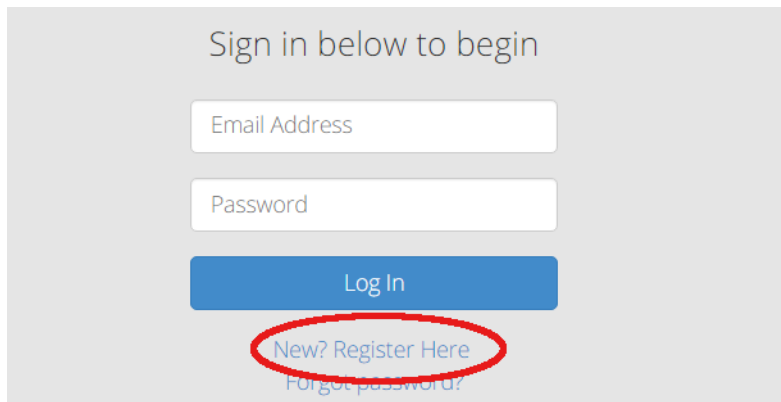
Website Registration and Software Navigation

The developed software represent different levels of difficulty and students can select the most appropriate version of the module for their knowledge base. It is 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 learning module includes multimedia materials including video, audio, and/or electronic documents that introduce the content presented in the module. While progressing through each module, students can access their comprehension of the presented material using computerized tests testing.

The learning system hosted by Michigan Tech will be freely available for anybody around the world to use over the Internet. Below, are the steps needed to successfully register and use this software package:

URL: <https://plc.csl.mtu.edu>

1. Create a user account
 - a. Click on Register Here



- b. Use your Email to register a **student** account with **Course Code left blank**, there is no benefit to register as an instructor.
 - c. Go to your Email, and click on the link of the confirmation email to activate your account.
 - d. Go back to the website <https://plc.csl.mtu.edu>. Login to your account.
2. Learning Modules
 - a. To access each module, simply click on the module blocks.
 - b. A module is divided into multiple sections. Each section page consists three areas as shown in Figure 7.
 - c. You can click on the small arrows to expand either **reading** area or the **exercise space**.

LearnPLC Help test@plc.com

Sections

Numerical Representations

Physical systems use quantities which must be manipulated arithmetically. Quantities may be represented numerically in either analog or digital form.

Analog Representation

- Varies over a continuous range of values, i.e. *real numbers*.
- Ex: Sounds through a microphone causes continuous voltage changes. An automobile speedometer changes with speed. Mercury in a thermometer varies over a range of values with respect to temperature.

Digital Representation

- Varies in discrete steps, i.e. *countable numbers*.

Exercise 1

Take a look at the images to the right and decide whether each image depicts an analog or digital representation of the data.

Remember, a digital representation will be presented in discrete steps or intervals, while an analog representation could be presented by any range and precision of real numbers.

Readings

Exercise Space




Analog



Digital



Digital



Analog

Resubmit Skip Incorrect. You may try again, or skip to the next section

Figure 7: Arrangement of the Learning Modulus

3. Go through modules and learn PLC.

- a. If you are using this for a course, please follow the instruction of your course instructor.
- b. Recommended learning steps are:
 - Follow modules and sections. For each section, do the reading first, and then read the exercise description and complete the exercise at the right side.
 - If you encounter any trouble, click **“Having Trouble”** buttons for hints.

4. Bug report.

If you find a bug or a typo. Please write down the **Module number, Section number, Exercise number, with a short description what happened (screenshot is the recommended)**, and send them to te Winn@mtu.edu and bochaol@mtu.edu. We will be greatly appreciate.

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 Electrical Engineering Technology (EET) and Computer Science (CS) Department at the Michigan Tech are collaborating with College to develop a PLC curriculum to help solve the current shortage and future expected growth in control engineering professionals required to have relevant and up-to-date PLC skills within the State and beyond. This project develops and implements the open source, multi-level, interactive PLC software to be used in high schools, two and four year colleges, as well as to training the displaced workers wishing to improve their knowledge and expertise in the subject matter and to meet the changing needs of the industry.

The developed PLC learning system has been tested at Michigan Tech during two introductory PLC class offerings in Fall of 2014 and 2015. The collected from the participating students feedback revealed that the software is very friendly and very easy to use. In addition to these features, students also emphasized that it serves as a good complement to the theoretical and hands-on parts of the course. In addition to the collected from the students feedback, the designers were able to troubleshoot few software related issues and clean up few “bugs”.

The learning system is 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. 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 proposed 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.

References:

- [1] S. L. Toral, F. Barrero, and M. R. Martinez-Torres (2007), Analysis of utility and use of a web-based tool for digital signal processing teaching by means of a technological acceptance model, *Comput. Educ.*, vol. 49, pp. 957–975.
- [2] Z. Aydogmus, O. Aydogmus (2009), A web-Based Remote Access Laboratory Using SCADA, *IEEE Transactions on Education*, vol. 52, #1, pp 126-133.
- [3] S. Hseih, P Hseih., (2005), Web-based Modules for Programmable Logic Controller Education. *Computer Applications in Engineerin Education*, Vol. 13, #4., pp 266-280.
- [4] Siemens Basics of PLCs course [Online]. Available: <http://www.sea.siemens.com/step/templates/lesson.mason?plcs:1:1:1>
- [5] PLCS.net instructional materials [Online]. Available: <http://www.plcs.net/contents.shtml>
- [6] Forum.MrPLC.com [Online]. Available: <http://forum.mrplc.com/index.php?act=Downloads>
- [7] Control.com [Online]. Available: http://www.control.com/control_com/index_html
- [8] PLC Book [Online]. Available: http://claymore.engineer.gvsu.edu/_jackh/books/plcs/
- [9] Global Controls [Online]. Available: http://www.globalcontrols.ca/gc/content_list.asp?mfid=2&catalogid=9&contentid=14
- [10] The Learning Pit [Online]. Available: <http://www.thelearningpit.com/>
- [11] J.J. Blakley and D.A. Irvine (2000), Teaching programmable logic controllers using multimedia-based courseware, *Int J Electrical Eng Educ* , vol. 37, pp. 305-315.
- [12] The Michigan Technological University Admission Office. Retrieved September 13, 2013, from <http://www.mtu.edu/admissions/>
- [13] School of Technology at Michigan Tech <http://www.mtu.edu/technology/>
- [14] Electrical Engineering Technology Department at Michigan Tech

<http://www.mtu.edu/technology/undergraduate/eet/>

[15] Computer Science department at Michigan Tech

<http://www.mtu.edu/cs/department/about/>

[16] Husky Games Development Enterprise <http://www.huskygames.com>