International Journal on Management Education & Emerging Technology
ISSN: 2583-9438

Design and Implementation of a Programmable Logic Control (PLC) Trainer Workstation using Open-Source Software and Low–Cost Hardware

Eduardo O. Hernandez, Jr. [1], Darwin D. Banzon[2]
Computer Engineering Department, Dr. Filemon C. Aguilar Memorial College of Las Piñas (DFCAMCLP), Las Pinas City, NCR, Philippines
eduardo.hernandez@dfcamclp.edu.ph

Abstract

This study addresses the high costs and limited accessibility of traditional PLC (Programmable Logic Controller) systems by designing a cost-effective, user-friendly PLC trainer workstation utilizing open-source software and low-cost hardware. The proposed system provides an inclusive platform for learners and developers to explore industrial automation processes while overcoming financial barriers. It integrates modular hardware components, open-source programming tools, and practical simulations, making automation training more accessible and innovative. The system's effectiveness was validated through the training session, revealing significant knowledge improvement among participants, demonstrating its potential as a sustainable and forward-thinking solution for education and industry.

Article Info

Keywords:
Programmable Logic, Control trainer, PLC Training, Open source software, Low cost hardware, Arduino

Date of Submission: 27/3/2025
Date of Review: 02/04/2025
Date of Acceptance: 22/04/2025
IJMEET / Volume 3, Issue 2,2025
ISSN: 2583-9438

INTRODUCTION

Programmable Logic Controllers (PLCs) have replaced hard-wired relay-based control systems in most industries because of their compact size, ruggedness, and, most importantly, their ability to be reprogrammed. Reprogramming a PLC allows changes to be made in the functional operation of a machine system without major physical changes in the control or output system components or wiring. Thus, labor, equipment, and downtime costs are reduced [1].

PLCs have become indispensable in modern industrial automation, offering seamless control, monitoring, and optimization of various processes. Despite their significant contributions to industrial efficiency, the steep costs of commercial PLC hardware and proprietary software present challenges for educators, hobbyists, and small-scale innovators seeking practical exposure to automation technologies.

A PLC is a microprocessor-based controller; it receives analog and digital signal input from input component such as switches and sensors and apply instructions stored in its programmable memory to control outputs to output components such as motors, pneumatic devices and status indicator. It implements functions such as logic and sequence [2][3].

The swift advancement of technology, including new models and innovations in PLC technology, has expanded its application beyond the industrial control domain. Consequently, developing skills through training in PLC cabling, programming, and implementation has become essential for students and individuals interested in industrial automation. Nevertheless, some of the problems is that industrial PLC is an expensive, prebuilt hardware kit also, to acquire programming software and its requisite programming competence is a challenge [4].

The programming languages defined by IEC 61131-3 [5] for PLC is the Ladder Logic (LL), Structure Text (ST), Function Block (FB) and Instruction List (IL) [6]. The PLC programming device can be a handheld device or the personal computer (PC). However, the PC is commonly used for PLC programming because it is readily available and portable. The LL is the most used programming language because it is simple to comprehend and implement [6].

While several researchers have proposed and reported various PLC trainers [7], they often missed a couple or more features such as the fundamental automatic operation of PLCs, the symbols of PLC components accompanied by their descriptions, the hardware connections for input/output components, and the easiness of wiring components during training.

This paper aims to bridge the above gaps by designing and implementing a PLC trainer workstation using open-source software and low-cost hardware. The proposed system emphasizes affordability, accessibility, and user- friendliness, enabling learners and developers to explore and experiment with industrial control processes without the financial burden of traditional PLC setups.

The PLC trainer workstation integrates open-source programming platforms, readily available microcontrollers, and modular hardware components to simulate real-world industrial scenarios. This approach not only makes automation education more inclusive but also fosters a culture of innovation by empowering users to customize and extend the system to meet diverse needs.

By combining theoretical concepts with practical, hands-on experience, the PLC Trainer Workstation also aims to enhance the learning process, equipping users with the skills required to navigate and contribute to the rapidly evolving field of industrial automation. Furthermore, the system aligns with global trends in open innovation, sustainability, and the democratization of technology, making it a forward-thinking solution for education and industry alike.

2 LITERATURE REVIEW

2.1 Elements of the System Architecture

The system architecture of our PLC trainer workstation comprises input devices, a PLC unit, output devices, interface components, a power supply unit, connectivity mechanisms, and safety features. Figure 1 presents a diagram illustrating the system architecture, followed by descriptions of the devices integrated into the workstation.

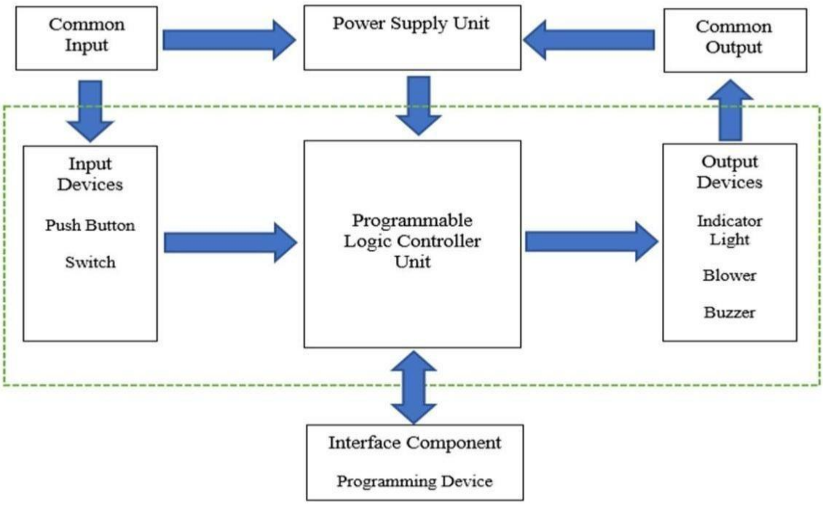

Figure 1. System architecture for the PLC trainer workstation.

Input devices consist of push-buttons and selector switches as in Figure 2. Installed push-buttons are momentary switches that can be used as normally-open (NO) or normally-closed (NC) contacts. Selector switches are used to control and select different electrical circuits or devices.

The PLC unit consists of an Arduino Uno development board (Figure 3) paired with an isolated I/O shield. The Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. The I/O shield features 10 opto-isolated inputs and 8 dry contact outputs operating at 12 Vdc. It protects the main board from electrical surges and interference. The isolated I/O shield used in this work was locally manufactured by e-Gizmo Mechatronics Central. A photo of the installed isolated I/O shield, along with diagram of its major components [7], is shown in Figure 4.

Output devices include indicator lights, buzzers, a blower, a relay, and a timer for external equipment. Figure 5 shows photos of installed output devices.

The interface component consists of a Core i5 notebook computer operating on Microsoft Windows 11, equipped with OpenPLC—an open-source platform designed to program the Arduino Uno using ladder logic. Figure 6 shows the OpenPLC graphical user interface.

The I/O peripherals receive power from a 12 Vdc switch-mode supply (Figure 7), which is derived from a 220 Vac source.

Connectivity is established through USB connection between the notebook computer and Arduino Uno, with OpenPLC communicating with Arduino primarily via the Modbus protocol.

For electrical safety, an emergency stop-button and a circuit breaker are integrated.

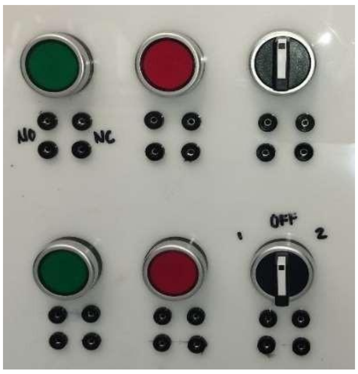

Figure 2. Push-buttons and selector switches as input devices.

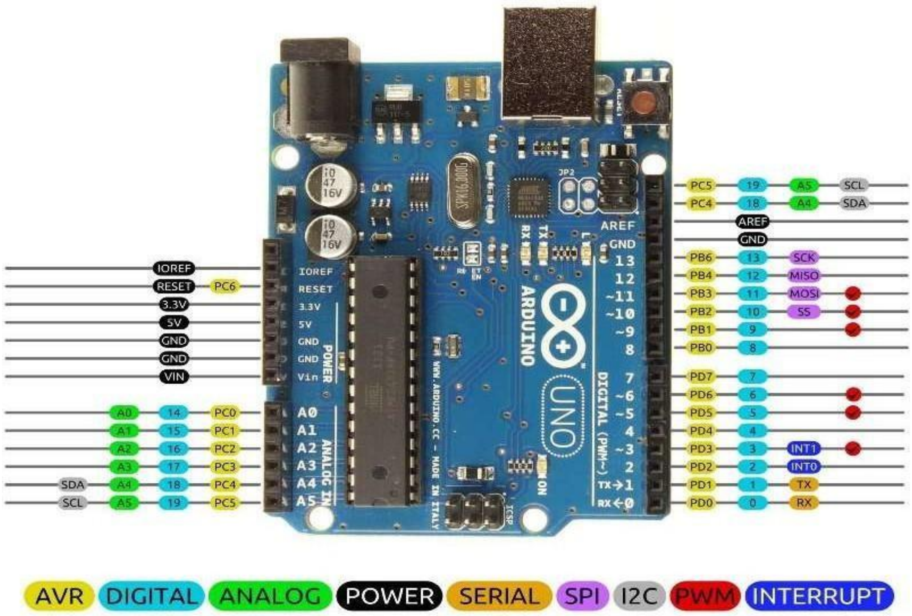

Figure 3. Arduino Uno development board

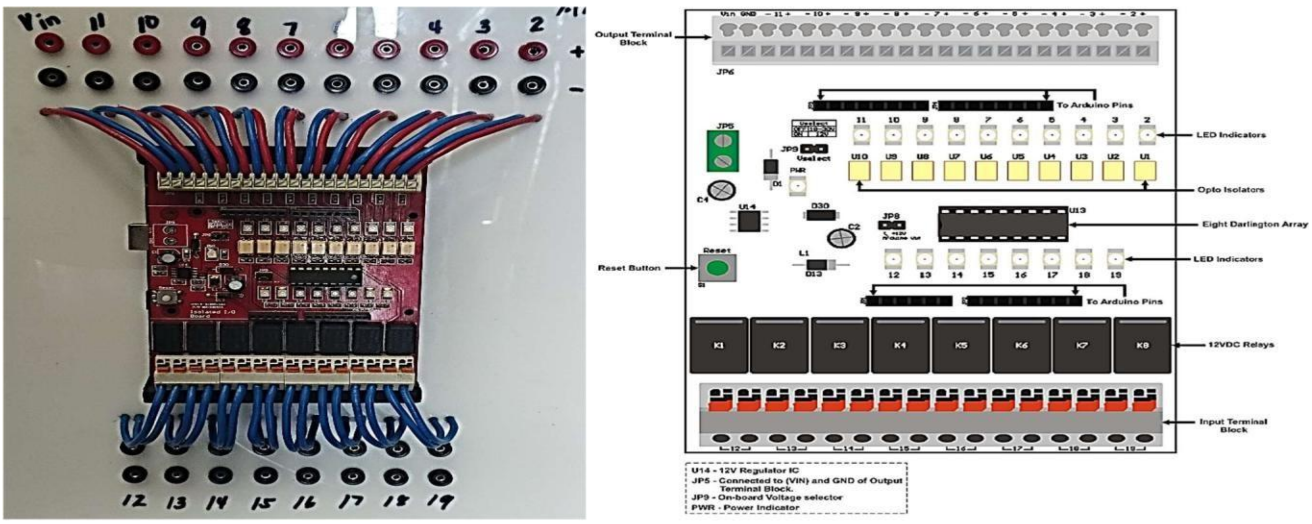

Figure 4. Installed isolated I/O shield and major components with details

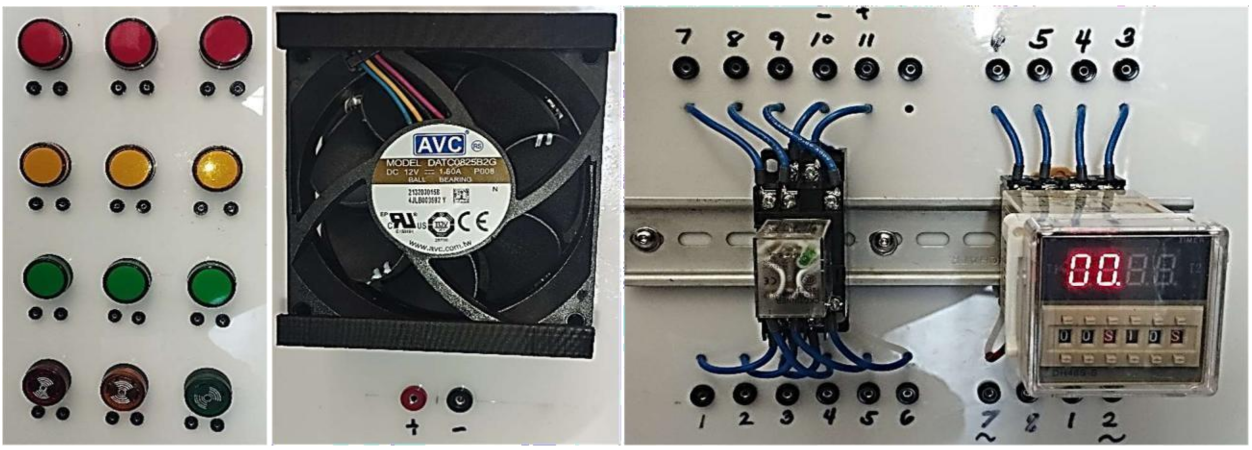

Figure 5. Installed output devices

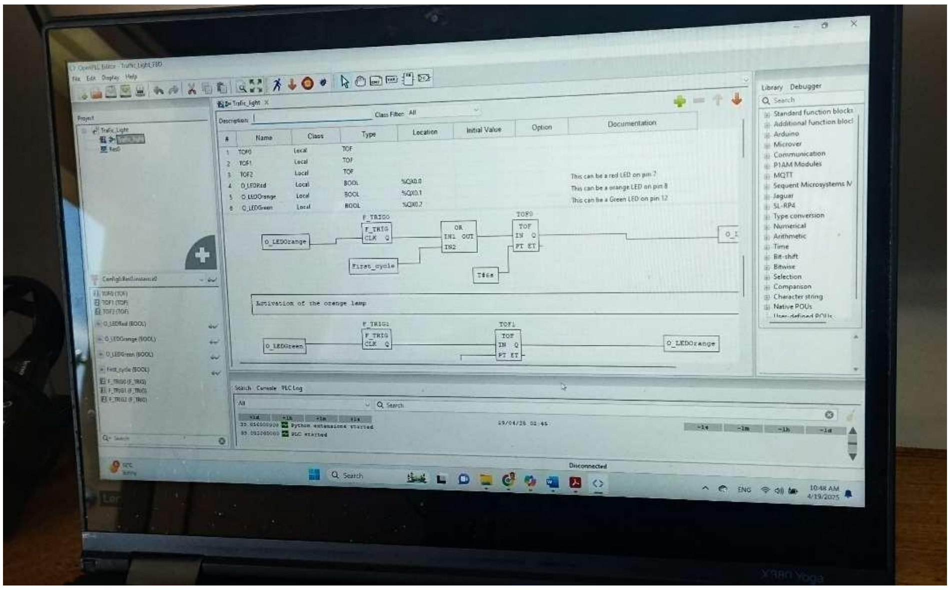

Figure 6. OpenPLC editor graphical user interface

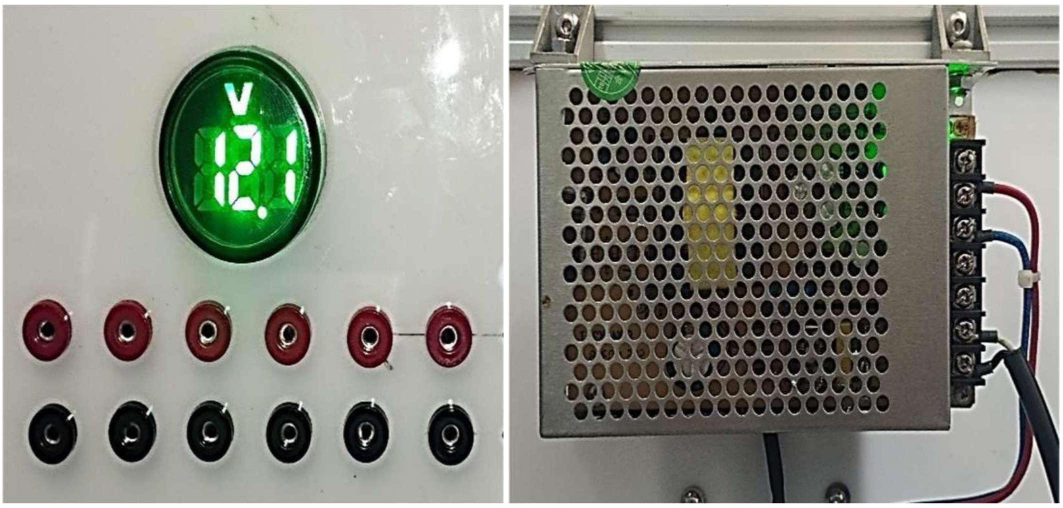

Figure 7. Power supply

2.2 Circuit Design

The PLC trainer workstation circuit corresponds to its system architecture, as illustrated in Figure 2. Figure 8 provides a schematic diagram of the workstation. Apart from the 2-mm female jack chassis panel mount sockets and the voltmeter, no devices are interconnected or hard-wired. All wiring, with the exception of the AC live and neutral lines, must be manually connected using 2-mm male banana plug cable K2 type test leads (Figure 9).

The hardware component and PLC are mounted on a 61 cm x 61 cm white acrylic panel board with aluminum profile frame and stand shown in Figure 10.

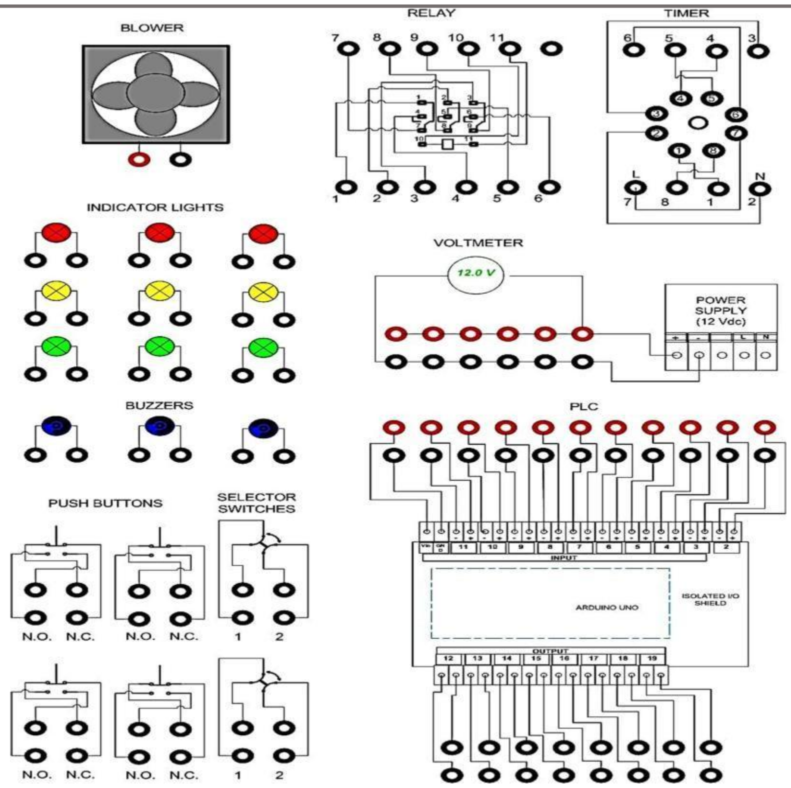

Figure 8. PLC Trainer Workstation Schematic Diagram

Figure 9. 2-mm male banana plug cable K2 type test leads

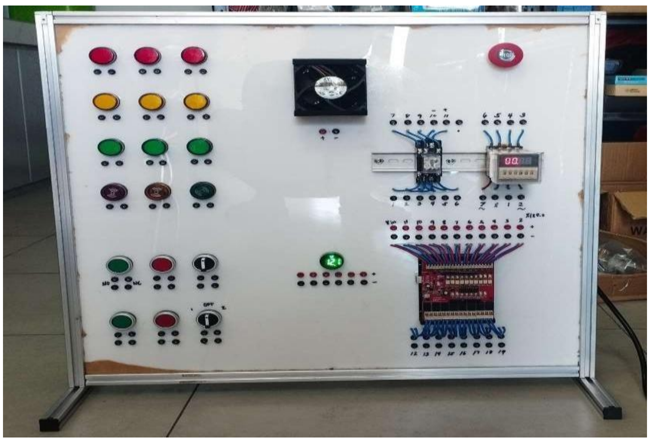

Figure 10. Completed PLC Trainer Workstation

2.3 Programming the PLC

The programming language utilized in this work is the LL, as defined by IEC 61131-3. Its structure resembles a ladder, featuring horizontal rungs and vertical rails. In the LL framework, power flows from the left vertical rail to the right vertical rail via the horizontal rungs. Circuit components are depicted on the horizontal rungs in their default state. Each rung is required to specify a control operation, including at least one input and one output component. The input component is positioned on the left side of the rung, while the output component is situated on the right. Components may appear multiple times within the LL structure. I/O components are addressed according to the specifications provided by the PLC manufacturer. The PLC processes the LL sequentially, reading from left to right and top to bottom.

The researcher utilized the OpenPLC editor to program the PLC. OpenPLC [9, 10], an open-source software developed by Thiago Rodrigues Alves, enables the simulation of PLC functions using a computer and I/O device. It is compatible with Microsoft Windows and Linux operating systems. PLC control programming is achieved by uploading Structured Text (ST) through OpenPLC. The software recommends using PLC Editor as the development environment, which supports five PLC languages: LD, ST, FBD, IL, and SFC. All languages, except for ST, are converted to ST within OpenPLC. The OpenPLC software and its corresponding documentation are available on their website: https://autonomylogic.com/.

PLC applications interact with the external world through Input and Output modules and/or SCADA communication protocols. When designing PLC applications, decide which variables should be attached to I/O and communication modules by labeling the variable with a PLC address.

OpenPLC uses the IEC 61131-3 nomenclature to address input, output and memory locations. The addressing of I/O locations is done through the use of special character sequences. These sequences are a concatenation of the percent sign "%", a location prefix, a size prefix and one or more natural numbers separated by blank spaces.

The following location prefixes are supported:

- I for input
- Q for output
- M for memory

The following size prefixes are supported:

- X for bit (1 bit)
- B for byte (8 bits)
- W for word (16 bits)
- D for double word (32 bits)
- L for long word (64 bits)

For instance, to read the state of the first digital input into a BOOL variable, the variable must be declared at %IX0.0. Similarly, to write the contents of a UINT variable to the second analog output, the UINT variable should be declared at %QW2.

As illustrated in Figure 11, OpenPLC emulates a PLC device using a computer and an I/O device [11]. These I/O devices manage field devices based on commands from the computer. OpenPLC is compatible with various I/O devices, including Raspberry Pi, Arduino and related boards, UniPi Industrial Platform, Modbus Slave Devices, ESP8266, and PiXtend.

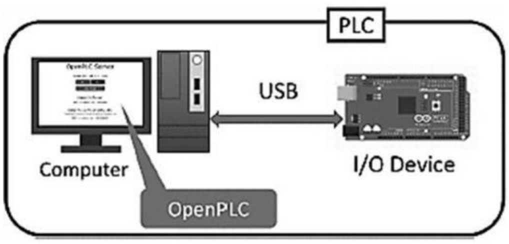

Figure 11. The OpenPLC system

2.4 Cost Analysis

Table 1 outlines the material expenses associated with constructing the workstation, excluding labor, notebook computer, and incidental costs.

Table 1: Costs of materials purchased for the PLC trainer workstation

Item	Quantity	Unit	Rate	Cost, PhP
Arduino Uno R3 board compatible MCU	1	Pieces	500.00	500.00
Isolated I/O shield	1	Pieces	800.00	800.00
Switch-mode power supply, 12Vdc / 7 A	1	Pieces	1,000.00	1,000.00
Pushbutton, 1NO 1NC 10A, momentary switch (spring return)	4	Pieces	130.00	520.00
Selector switch	2	Pieces	150.00	300.00
Emergency-stop button	1	Pieces	100.00	100.00
Indicator lights	9	Pieces	30.00	270.00
Voltage indicator	1	Pieces	200.00	200.00
Buzzers, 12 Vdc	3	Pieces	60.00	180.00
Blower, 12 Vdc	1	Pieces	200.00	200.00
Circuit breaker	1	Pieces	300.00	300.00
Ice cube relay, 12 Vdc	1	Pieces	200.00	200.00
Time delay relay	1	Pieces	400.00	400.00
2-mm male banana plug cable K2 type test leads	20	Pieces	60.00	1,200.00
Female jack chassis panel mount socket for 2mm banana plug	120	Pieces	20.00	2,400.00
DIN rail	1	Meter	300.00	300.00
Terminal block	1	Pieces	100.00	100.00
Electrical cable/wire	1	Lot	500.00	500.00
Fasteners -assorted	1	Lot	500.00	500.00
White acrylic panel board, 61 cm x 61 cm	1	Pieces	700.00	700.00
Aluminum profile extrusion, anodized silver, 2020	6	Meter	300.00	1,800.00
Data cable, USB 2.0, male USB A to male USB B, 1 meter	1	Pieces	100.00	100.00
OpenPLC software	1	Unit	0.00	0.00
			Total	**12,570.00**

2.5 Testing and Validation

The PLC trainer workstation was tested and validated through pre-test and post-test assessments carried out during a training session with computer engineering students from Dr. Filemon C. Aguilar Memorial College of Las Piñas (DFCAMCLP). A one-way analysis of variance (ANOVA) was conducted to determine if the training had a significant impact on the participants' performance.

As part of the validation process, participants applied the knowledge acquired during the session to address an industrial problem. Figure 12 shows participants working with the PLC trainer workstation.

Figure 12. Participants working with the PLC trainer workstation.

Presented below are the pre-test and post-test questions, structured to elicit responses on a scale of 1 to 5.This format facilitates the assessment and quantification of user feedback.

Pre-Test Questions

Knowledge and Understanding

1. How familiar are you with Programmable Logic Controllers (PLCs) and their uses in industrial automation? (1 = Not familiar, 5 = Very familiar)
2. How well do you understand the key components of a PLC system (e.g., inputs, outputs, processing unit)? (1 = Poor understanding, 5 = Excellent understanding)
3. How confident are you in explaining ladder logic programming? (1 = Not confident, 5 = Very confident)
4. How familiar are you with open-source PLC programming platforms? (1 = No familiarity, 5 = Extensive familiarity)
5. How well do you understand the difference between digital and analog inputs/outputs in PLCs? (1 = Poor understanding, 5 = Excellent understanding)

Skills and Experience

6. How confident are you in working with microcontroller boards such as Arduino or Raspberry Pi? (1 = Not confident, 5 = Very confident)
7. How skilled are you at writing and debugging PLC programs? (1 = No skill, 5 = Highly skilled)
8. How well can you wire hardware components like sensors and relays for automation systems? (1 = Poorly, 5 = Very well)
9. How often do you face challenges when programming or setting up PLC systems? (1 = Frequently, 5 = Rarely)
10. How well do you understand how PLCs can be used to solve real-world automation problems? (1 = Poor understanding, 5 = Excellent understanding)

Expectations and Insights

11. How important is it for you to learn new PLC-related concepts during training? (1 = Not important, 5 = Very important)
12. How much do you expect the PLC trainer workstation to improve your knowledge and skills? (1 = No improvement expected, 5 = Significant improvement expected)
13. How beneficial do you think open-source software and low-cost hardware are for PLC training? (1 = Not beneficial, 5 = Highly beneficial)
14. How confident are you in troubleshooting automation systems? (1 = Not confident, 5 = Very confident)
15. How would you rate the level of difficulty you experience when working with PLC-related concepts and skills? (1 = Very challenging, 5 = Not challenging at all)

Post-Test Questions

Knowledge Gained

1. How much has your understanding of PLCs improved after using the trainer? (1 = No improvement, 5 = Significant improvement)
2. How confident are you now in setting up and programming a PLC system? (1 = Not confident, 5 = Very confident)
3. How much has your understanding of ladder logic programming improved? (1 = No improvement, 5 = Significant improvement)
4. How easy was it to use open-source software for programming the PLC? (1 = Very difficult, 5 = Very easy)

Skills and Experience

5. How confident are you now in writing, debugging, and executing PLC programs? (1 = Not confident, 5 = Very confident)
6. How skilled are you at wiring and integrating hardware components with the PLC trainer workstation? (1 = Poorly skilled, 5 = Highly skilled)
7. How effectively did the trainer help you simulate real-world automation scenarios? (1 = Ineffectively, 5 = Very effectively)
8. How well did you troubleshoot errors while using the trainer? (1 = Poorly, 5 = Very well) User Experience

9. How user-friendly did you find the PLC trainer workstation? (1 = Not user-friendly, 5 = Very user-friendly)
10. How satisfied were you with the workstation's features and functionality? (1 = Not satisfied, 5 = Very satisfied)

Feedback and Suggestions

11. How likely are you to recommend the PLC trainer workstation to others? (1 = Not likely, 5 = Very likely)
12. How much do you think the trainer needs improvements or additional features? (1 = No improvements needed, 5 = Significant improvements needed)
13. How does this trainer compare to other learning tools you've used? (1 = Worse, 5 = Much better)
14. How helpful were the educational materials (e.g., manuals, tutorials) provided with the trainer? (1 = Not helpful, 5 = Very helpful)

Reflection

15. How much did using the PLC trainer impact your approach to automation tasks? (1 = No impact, 5 = Significant impact)
16. The selected industrial problem involved designing a control system that implements a holding circuit applied to a traffic light [12]. Refer to Figure 13 that shows the system connection to control the traffic light. When start push button (PB 1) is pressed, red light will turn ON for 5 seconds. After that, green light will turn ON for 10 seconds and red light will turn OFF. The operation will repeat simultaneously until stop push button (PB 2) is pressed to stop the operation. Construct this sequence using PLC ladder diagram.

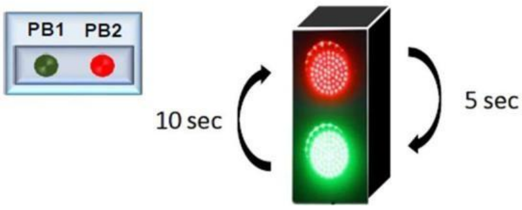

Figure 13. System connection to control the traffic light

3 Results and Discussions

Table 2 displays the results from the pre-test and post-test conducted during the training session, involving two groups: pre-test scores and post-test scores from 12 participants. The average scores for the pre-test and post-test are 44.5% and 80%, respectively. Test results indicated a mean improvement of 35.5% in participant performance.

For the ANOVA analysis, the null hypothesis assumed no difference between the group means, while the alternative hypothesis proposed a significant difference. Figure 14 shows the outcome of the one-way ANOVA test performed using Microsoft Excel.

Table 2. Scores obtained from the pre-test and post-test conducted during the training session

Participant No.	Pre-test Mark (%)	Post-test Mark (%)
CPE01-AF	48	69
CPE02-AM	44	88
CPE03-AA	48	75
CPE04-GM	56	92
CPE05-GA	60	91
CPE06-GS	43	63
CPE07-MK	44	93
CPE08-OJ	64	84
CPE09-PH	56	88
CPE10-PI	25	85
CPE11-RE	51	97
CPE12-TM	47	79
Minimum	25	63
Maximum	64	97
Mean	44.5	80.0

22	Anova: Single Factor							
23								
24	SUMMARY							
25	*Groups*	*Count*	*Sum*	*Average*	*Variance*			
26	Column 1	12	586	48.83333	101.4242			
27	Column 2	12	1004	83.66667	106.0606			
28								
29								
30	ANOVA							
31	*Source of Variation*	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*	
32	Between Groups	7280.166667	1	7280.167	70.17541	0.00000003	4.30095	
33	Within Groups	2282.333333	22	103.7424				
34								
35	Total	9562.5	23					

Figure 14. Results of a one-way analysis of variance (ANOVA) test using Microsoft Excel

Figure 15 illustrates the control system design featuring a holding circuit applied to a traffic light, developed by one of the participants. The program was compiled and uploaded to the PLC memory to execute the traffic light operation. Figure 16 presents a screenshot of the program after the start push-button was activated.

Participants effectively developed a PLC training platform, which they seamlessly applied to the industrial control of processes.

The cost analysis (Table 1) of the developed PLC trainer workstation demonstrated its affordability and cost- effectiveness, with a price of PhP 12,570 (approximately USD 220) only.

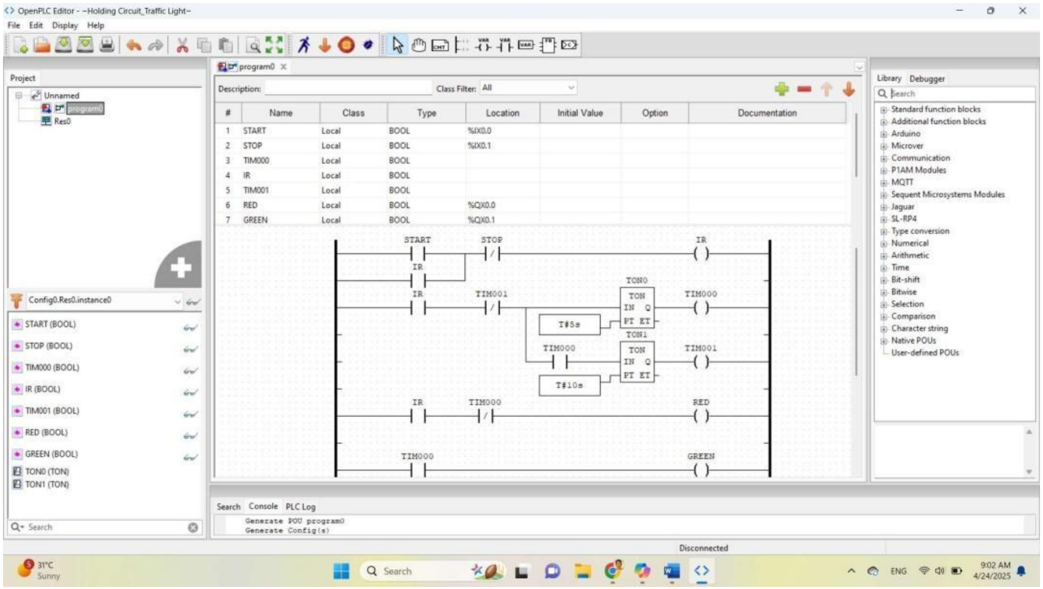

Figure 15. Design of a control system that implements a holding circuit applied to a traffic light made by one of the participants

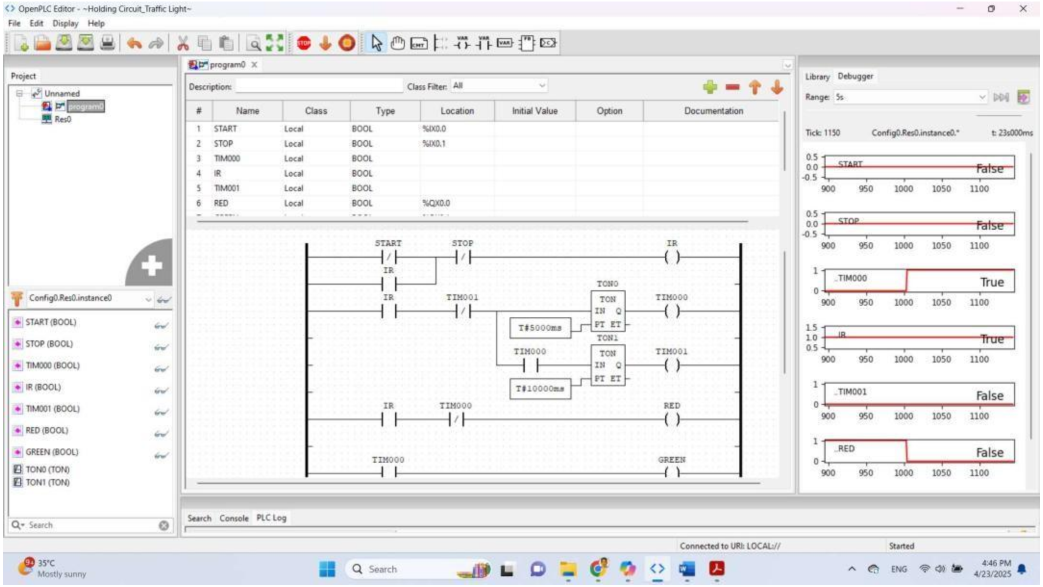

Figure 16. Screenshot of the program after the START push-button was pressed

4. Conclusion and Recommendation

A cost-effective, user-friendly, and interactive PLC training platform has been successfully developed for industrial control processes. Pre-test and post-test results indicate that participants achieved an average knowledge improvement of 35.5% in independently wiring and programming the PLC.

In the one-way ANOVA test, the p-value (0.00000003) is far below the significance level of 0.05, confirming a statistically significant difference between pre-test and post-test scores. With the F-statistic (70.18) greatly surpassing the critical F-value (F crit), the null hypothesis is decisively rejected. These results highlight the effectiveness of the training in significantly improving participants' performance, as evidenced by the marked increase in post-test scores.

Cost analysis indicates that the developed PLC trainer workstation is both affordable and cost-effective. In comparison, branded PLC hardware with similar specifications, combined with proprietary software, could raise the cost by several hundred dollars. To provide context, the cost of a typical PLC trainer workstation varies greatly based on the brand, features, and functionality, typically falling within the range of $200 to $1,500 or higher. Basic models designed for educational purposes are usually more economical, while advanced models with enhanced features and industrial-grade components are considerably more expensive [13].
To maximize the impact of this PLC trainer workstation, it is recommended to:

- Promote the trainer workstation to educational institutions, technical training centers, and small-scale industries to democratize automation learning.
- Continuously develop additional functionalities, including support for advanced programming languages, compatibility with more hardware modules, and integration with IoT systems for expanded applications.
- Encourage community-driven development and open innovation, enabling users to customize and improve the system according to their needs.
- Explore eco-friendly hardware materials and energy-efficient components to align the system with global sustainability trends.
- These recommendations aim to ensure the longevity, adaptability, and wider adoption of the PLC trainer workstation, paving the way for a more inclusive and innovative industrial automation landscape.

Acknowledgment

The authors extend their heartfelt gratitude to the following individuals and organizations who contributed to the success of this research:

- Dr. Filemon C. Aguilar Memorial College of Las Piñas (DFCAMCLP): For offering the platform to test and validate the PLC trainer workstation and for providing valuable backing and encouragement to the authors.
- Thiago Rodrigues Alves: For developing the OpenPLC software, which served as a critical tool in this study.
- First year DFCAMCLP Computer Engineering student participants: For their active engagement, enthusiasm, and insights during the training session and validation process.

This work would not have been possible without their contributions and support, and we sincerely appreciate their role in bringing this project to fruition.

References

1. R. A. Cox, Technician’s Guide to Programmable Controllers, 4th ed., Albany, NY, USA: Delmar, 2001.
2. M. Barrett, "The Design of a Portable Programmable Logic Controller (PLC) Training System for Use Outside of the Automation Laboratory," in 2008 International Symposium for Engineering Education, Dublin City Univ., Ireland, 2008.
3. W. Bolton, Programmable Logic Controllers, 4th ed., UK: Elsevier, 2006.
4. K. Bhise and S. S. Amte, "Embedded PLC Trainer Kit with Industry Application," Int. J. Eng. Sci. Innovative Technol. (IJESIT), vol. 4, no. 3, pp. 1–9, 2015.
5. IEC 61133-1:2013, Programmable Controllers – Part 3: Programming Languages, IEC Standard, 2013.
6. C. D. Johnson, Process Control Instrumentation Technology, 8th ed., Upper Saddle River, NJ, USA: Prentice Hall, 2006.
7. Sumanth B., Neha J., Sneha R., C. K. Prasad, and Dr. K. Viswanathan, "Programmable Controller (PLC) Trainer Kit for Various Power System Applications," Int. J. Innovative Res. Electr. Electron. Instrum. Control Eng. (IJIREEICE), vol. 12, no. 7, pp. 103–112, 2024.
8. e-Gizmo Mechatronix Central, PLC Isolated I/O Shield Hardware Manual, 2011, https://e-gizmo.net/oc/kits%20documents/Isolated%20IO%20Shield/isolated%20io%20shield%20manual.pdf (accessed 2025-04-01).
9. T. R. Alves, M. Buratto, F. M. De Souza, and T. V. Rodrigues, "Open-PLC: An Open-Source Alternative to Automation," in IEEE Global Humanitarian Technol. Conf. (GHTC 2014), 2014.
10. OpenPLC Project. [Online]. Available: https://openplcproject.com. [Accessed: Apr. 1, 2025].
11. S. Fujita, K. Hata, A. Mochizuki, et al., "OpenPLC Based Control System Testbed for PLC Whitelisting System," Artif. Life Robot., vol. 26, pp. 149–154, 2021.
12. N. B. M. Posdzi and T. E. C. Chin, Programmable Logic Controllers (PLC): Programming and Applications, 1st ed., Politeknik Ungku Omar, Ipoh, Perak, Malaysia, 2021.
13. Dolangedu, “What is the cost of a PLC trainer.” [Online]. Available: https://dolangedu.com/blog/what-is-the- cost-of-a-plc-trainer. [Accessed: 24-Apr-2025].