ISBN: 978-628-96613-1-6. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2025.1.1.1262

Design of a Programmable Logic Controller (PLC) Prototype Using Microcontrollers

Fredy Hadid Villanueva de la Ossa[1], Andrés-F Jiménez López, PhD.[2]; Fabián-R Jiménez López, MSc[3]
[1]Faculty of Basic Sciences and Engineering – EISY - iCONBOT – Universidad de los Llanos, Villavicencio, Colombia, fhvillanueva@unillanos.edu.co
[2]Faculty of Basic Sciences and Engineering – Macrypt - Farmtechnology – Universidad de los Llanos, Villavicencio, Colombia, ajimenez@unillanos.edu.co
[3] Engineering Faculty – Research Group I²E – Department of Electronic Engineering – Universidad Pedagógica y Tecnológica de Colombia, Tunja, Colombia, fabian.jimenez02@uptc.edu.co

Abstract*– Process automation is essential for enhancing efficiency and competitiveness in modern industries; however, conventional systems such as traditional programmable logic controllers (PLCs) often involve high costs and complexity. This work proposes the design of a microcontroller-based PLC prototype using open-source software to reduce operational expenses. The prototype aims to provide an affordable automation solution for small and medium-sized enterprises seeking to implement automation without significant initial investments. The design focuses on key aspects such as hardware flexibility, adaptability to diverse applications, and cost-effectiveness. Preliminary results demonstrate the prototype's potential to improve productivity while maintaining economic feasibility. The study highlights the advantages of using open-source platforms and microcontroller technology as viable alternatives to traditional PLC systems, particularly for cost-sensitive industrial applications.

***Keywords*-- Process Automation, Microcontroller-based PLC, Open-Source Software, Industrial Automation, Cost-Effective Solutions.**

I. Introduction

Industrial processes have undergone significant evolution throughout their historical development, driven by the need to adapt to changing global consumption demands and emerging technological innovations [1]. This evolution has led to the adoption of advanced techniques that not only optimize production but also enable companies to respond effectively to new market demands [2].

The impact of this technological transition is not limited solely to operational efficiency but is also reflected in companies' ability to reduce costs associated with resource waste, downtime, and energy consumption, which translates into increased profit margins [3]. Thus, the transformation of industrial processes not only addresses current demands but also constitutes a crucial strategy for ensuring the long-term relevance and sustainability of organizations in an increasingly dynamic market [4].

Despite the significant efforts of industrialization to encompass the entire business sector with innovative automation techniques, small and medium-sized industries often fail to adopt these techniques because, in most cases, they require robust infrastructure that emerging companies cannot afford [5]. In their pursuit of these improvements, they seek more economical but less reliable alternatives, which can sometimes be detrimental to production and pose risks to personnel responsible for equipment operation [5].

Automation is defined as the use of controlled technologies and systems to perform tasks or processes with minimal or no human intervention [6]. This includes the use of electronic devices [7], software [8], and machinery, which enable improved efficiency, reduced errors, and optimized productivity across various business sectors [9]. Its primary objective is to increase the precision and speed of operations, freeing individuals from repetitive or complex tasks so they can focus on higher-value activities [10].

In the industry, programmable logic controllers (PLCs) stand out as essential devices for the automation of programmable processes [11]. Designed to execute logical sequences automatically, PLCs facilitate the supervision and management of systems through continuous scan cycles, which include reading inputs, executing the program, and updating outputs, ensuring a rapid and precise response to environmental changes [12], [13]. Their robustness, reliability, and ability to operate under adverse conditions make them a versatile solution [14]. In addition to controlling and automating industrial tasks, such as sensor monitoring and actuator activation, they are notable for their programming flexibility using graphical languages like ladder diagrams and function block diagrams, enabling efficient configurations and modifications without altering the hardware. Their integration with other systems optimizes production processes and facilitates the control of critical variables such as temperature, pressure, and flow in complex infrastructures.

Despite their numerous advantages, PLCs also present certain limitations that must be considered when selecting them as automation mechanisms. One of the main weaknesses is their high initial cost for companies, especially in applications requiring multiple modules or large-scale systems [15]. PLCs can also be less flexible in terms of customization compared to other microcontroller-based solutions, as their architecture and software are more oriented toward standard industrial applications [16]. The complexity of advanced PLC programming, which requires highly trained personnel, increases operational costs [17]. Additionally, hardware update times and dependence on specific suppliers can delay system modernization and limit the adoption of emerging technologies in automated industrial processes [18].

In response to the challenges outlined, microPLCs—programmable logic controllers based on microcontrollers—have emerged as a more compact solution than traditional PLCs, designed for smaller-scale applications requiring more economical and flexible solutions [19]. MicroPLCs offer advanced functionalities with the low cost and adaptability of microcontrollers, making them ideal for industries seeking efficient automation [20].

MicroPLCs, based on microcontrollers, emerged in the 1990s as a compact and more affordable solution than traditional PLCs, leveraging advancements in microcontroller technology to automate smaller-scale processes [21]. Initially, they were implemented in simple industrial applications where conventional PLCs were too complex and costly. Over time, microPLCs evolved and acquired greater processing capabilities, enabling them to tackle more complex tasks in sectors such as manufacturing, building automation, and energy management [22]. Currently, their flexibility, low cost, and ability to integrate with industrial networks and emerging technologies like the Internet of Things (IoT) have made them a popular choice across various industries [23].

This paper proposes a microcontroller-based PLC design that ensures the automation of industrial processes using affordable materials. This prototype addresses several challenges of small and medium-sized companies, which require automated systems to increase productivity or improve the quality indices of their final products [24]. This automaton will also programmed with open-source software, which will further reduce operating costs, as specialized software for commercial PLCs significantly increases the implementation costs of these licensed automatons [25], [26]. The scope of this work is limited to system design, leaving implementation and validation for future stages of development.

II. METHODOLOGY

This project utilized the deductive-inductive scientific methodology for its development. The development of the microPLC prototype for industrial applications followed the V-Methodology, a strategy that ensures continuous verification and validation throughout each phase of the project lifecycle. This approach is particularly well-suited to the project, as it requires precise integration of hardware and software to create an efficient, affordable, and adaptable automaton tailored to the needs of small and medium-sized enterprises.

OpenPLC is an open-source platform designed for programming and simulating programmable logic controllers (PLCs), adhering to the IEC 61131-3 standard. It supports languages such as Ladder Logic and Structured Text and is compatible with hardware like Raspberry Pi, Arduino, and ESP32, facilitating the development of low-cost automation solutions. Thanks to its flexible and accessible approach, OpenPLC has been adopted in numerous industrial and research applications, promoting the transition to Industry 4.0 [26].

A programmable logic controller (PLC) can be tasked with controlling a physical system, such as the temperature of an oven or the speed of a motor. This physical system can be described by a differential equation that relates the input and output.

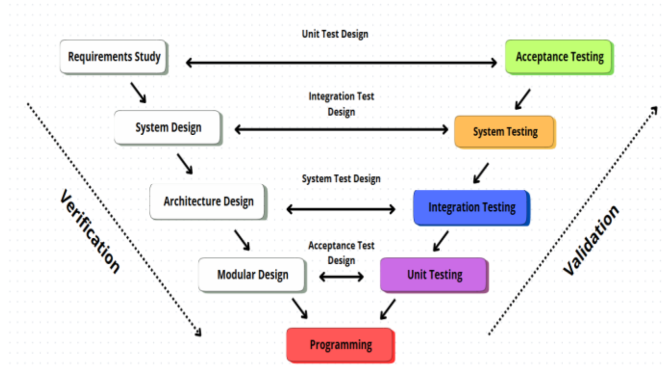

Fig. 1. V-Methodology Diagram. Source: Author.

For example, if the controlled system is a first-order system, it can be described by the differential equation (1):

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t) \quad (1)$$

Where:

- $y(t)$ is the system's output variable (e.g., temperature or speed).
- $u(t)$ is the input signal controlled by the PLC (e.g., the on/off signal for a motor or heater).
- τ is the system's time constant (related to the process's inertia or delay).
- K is the system's gain (related to how quickly the system responds to the input).

The microPLC can implement feedback control to adjust the input $u(t)$ based on the error between the desired value $y_{ref}(t)$ and the actual value $y(t)$. The error is defined as (2):

$$e(t) = y_{ref}(t) - y(t) \quad (2)$$

A basic control that the PLC could implement is proportional control (P) (3):

$$u(t) = K_p\, e(t) = K_p\,(\, y_{en}(t) - y(t)\,) \quad (3)$$

Here, K_p is the proportional gain constant, which determines how aggressively the PLC adjusts the input $u(t)$ to correct the error. If the microPLC uses a PID (Proportional-Integral-Derivative) control, the control equation becomes more sophisticated (4):

$$u(t) = K_p\, e(t) + K_i \int_0^t e(\tau)\, d\tau + K\_d\, {}^{de}(t)/{}^{dt} \quad (4)$$

Where:

- K_p: Proportional gain.

- K_i: Integral gain, which eliminates accumulated error over time.
- Kd: Derivative gain, which anticipates the rate of change of the error.

This PID model allows the microPLC to control the system more precisely, reducing oscillations and improving response time. Using the Laplace transform, the differential equation describing the system can be converted into a transfer function, facilitating system analysis and control design. Applying the Laplace transform to the first-order system equation (5):

$$\tau s\, Y(s) + Y(s) = K\, U(s) \tag{5}$$

Rearranging to obtain the system's transfer function (6):

$$G(s) = Y(s) / U(s) = K / (\tau s + 1) \tag{6}$$

Proportional control in the Laplace domain is expressed as (7):

$$U(s) = K_p\, (Y_{re}f(s) - Y(s)) \tag{7}$$

By combining the system with the controller, the behavior of the closed-loop system can be analyzed, and K_p, K_i, y Kd can be designed to achieve the desired responses (stability, response speed, etc.).

As shown in Fig. 2. Block Diagram of a Programmable Logic Controller. Source: Authors, the general description of a PLC starts with the input module sending signals to the CPU, where data is processed. The output module then controls external devices. The power supply provides energy to all modules, and a programming device is connected to configure the PLC. Input and output connections are protected by optical isolation.

Fig. 3 highlights the similarity between Ladder Logic in PLCs and C programming in microcontrollers. In both systems, the code begins with an initialization phase, followed by an infinite loop where instructions (rungs in Ladder Logic or statements in C) are executed repeatedly. Both methods execute instructions one at a time, from top to bottom, ensuring continuous process control. The concept of "initialization rungs" in PLCs, which execute only once, is similar to the initialization block in microcontroller programs before entering the infinite loop [27].

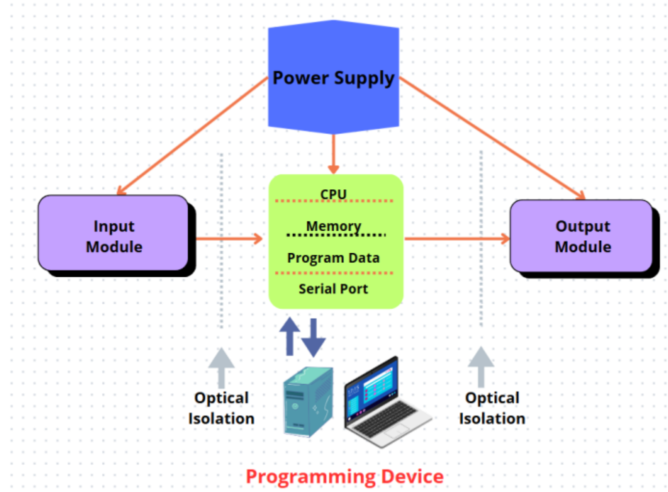

Fig. 2. Block Diagram of a Programmable Logic Controller. Source: Authors.

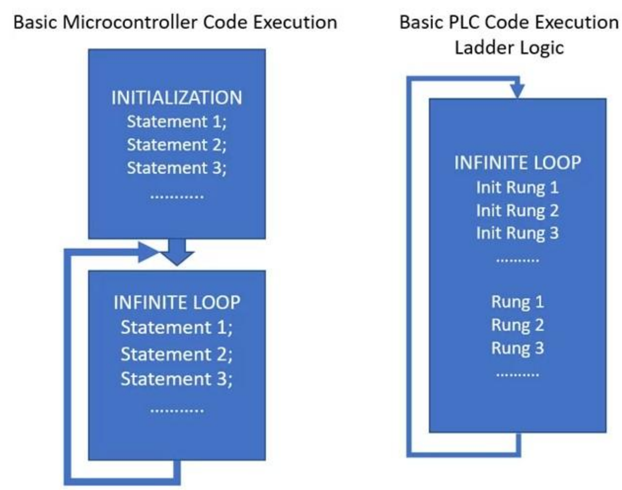

Fig. 3. Comparison of Microcontroller and PLC Programming. Source: [27].

A. Electronic Components

For the development of the microPLC, the ESP32 microcontroller was used as the system's brain due to its ability to process digital and analog signals and its integrated Wi-Fi and Bluetooth connectivity. This allows the microPLC not only to manage connected devices but also to communicate wirelessly with other systems. Additionally, the ESP32 features multiple communication interfaces, such as UART, SPI, I2S, and I2C, facilitating connections with external sensors and actuators.

Optocouplers were used to isolate the control signals generated by the ESP32 from power devices, such as relays and motors. This isolation is essential to protect the microcontroller from voltage fluctuations or interference from high-power devices. Furthermore, they provide greater safety and reliability in system operation. Solid-state relays (SSRs) are electronic devices used to control the current in a power circuit without mechanical components. Unlike traditional electromechanical relays, SSRs operate using semiconductors such as thyristors or triacs to switch current. The SSR consists of three main parts:

1) Input Circuit *(Control Signal):* Receives a low-voltage signal, typically DC, which activates the switching in the power circuit.
2) Isolator *(Optocoupler):* Isolates the input circuit from the power circuit using an optoelectronic device.
3) Power Circuit *(AC or DC):* Uses components such as triacs or thyristors in AC applications and MOSFETs in DC applications.

Current switching in an SSR varies depending on whether it is AC or DC. For AC, the triac activates when the gate voltage VG exceeds a threshold V_{th} (8):

$$V_G > V_{th} \Rightarrow I_{AC} \neq 0 \tag{8}$$

The load current $I_{AC(t)}$ is defined as (9):

$$I_{AC(t)} = \frac{V_{AC(t)}}{R_{load}} \tag{9}$$

In DC applications, MOSFETs are used, with the current described by (10):

$$\boldsymbol{I_D} = \left(\frac{1}{2}\right) * \boldsymbol{k_n} * (\boldsymbol{V_{GS}} - \boldsymbol{V_{th}})^2 \tag{10}$$

SSRs offer advantages such as fast switching speed (on the order of milliseconds), absence of mechanical wear, and galvanic electrical isolation through optocouplers. In the mathematical modeling of SSR behavior, the input-output relationship of an SSR can be modeled as an ideal step function (11):

$$V\_load(t) = V\_AC(t),\ si\ V\_in \geq V\_umbral \quad 0,\ si\ V\_in < V\_umbral \tag{11}$$

Finally, transducers were integrated to convert physical variables (such as temperature, pressure, or flow) into electrical signals processable by the ESP32. This facilitates the feedback necessary for control systems and enables monitoring.

III. Results

After reviewing the references and analyzing existing systems, a microPLC prototype was designed using the ESP32 microcontroller. This design considered the interconnection of analog and digital inputs and outputs, selecting the most suitable components for each type of signal. The necessary hardware configurations were defined, as well as possible connections with external devices through communication protocols such as Wi-Fi, Bluetooth, UART, SPI, I2S, and I2C. The proposed design offers a flexible and scalable solution, adaptable to various industrial applications.

B. Relationship Between Software Inputs/Outputs and Hardware

Before proceeding with the design, the input and output (I/O) assignments established by the OpenPLC software for the microcontroller pins were considered.

TABLE I
Pin Configuration for ESP32. Source: Author.

ESP32	Pins	Address
Digital Input	17, 18, 19, 21, 22, 23, 27, 32, 33	%IX0.0 - %IX0.7, %IX1.0
Digital Output	01, 02, 03, 04, 05, 12, 13, 14, 15, 16	%QX0.0 - %QX0.7, %QX1.0 - %QX1.1
Analog Input	34, 35, 36, 39	%IW0 - %IW3
Analog Output	25, 26	%QW0 - %QW1

Table 1 shows the pin assignments for the ESP32 as defined by the OpenPLC software. Digital inputs use pins 17 to 33, while digital outputs occupy pins 1 to 16. Analog inputs are assigned to pins 34, 35, 36, and 39, and analog outputs to pins 25 and 26. This configuration ensures efficient integration between the hardware and control signals.

C. Prototype Design

As shown in Fig. 4, the prototype includes a power supply with the LM2596HV, opto-isolated digital inputs, and digital outputs controlled by the ULN2803APG and solid-state relays (SSRs). Analog inputs use a 4-20 mA converter and potentiometers for testing, while analog outputs generate 0-10V signals with the LM358P. Additionally, isolation between AC and DC zones was implemented to protect the system.

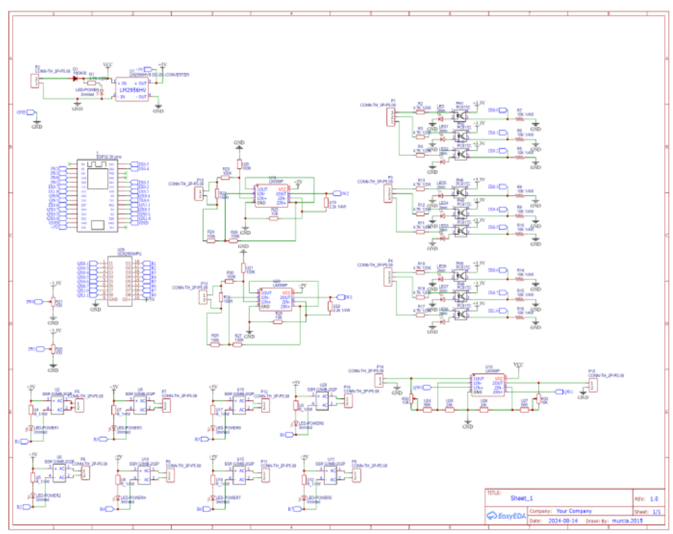

Fig. 4. General Electrical Diagram of the Prototype. Source: Authors.

As shown in Fig. 5, the microPLC design utilizes an LM2596HV regulator to convert high input voltages (up to 60V) to appropriate levels, such as 5V or 3.3V, ensuring a stable supply for the ESP32. A diode is included at the input for reverse polarity protection, preventing damage from incorrect connections. Additionally, an indicator LED with a current-limiting resistor is integrated to visually verify if the system is properly powered. This setup ensures safe and reliable system operation.

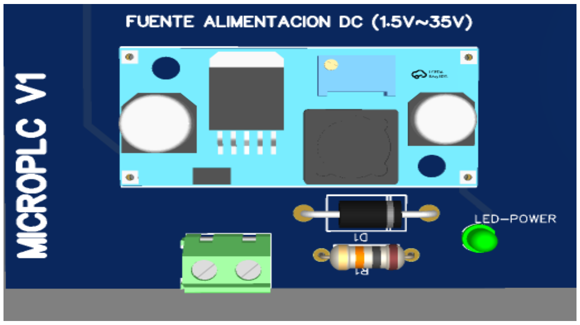

Fig. 5. Power Supply of the Prototype. Source: Authors.

As shown in Fig. 6, opto-isolated digital inputs were integrated to ensure efficient and safe electrical isolation between external devices and the microcontroller. PC817 optocouplers were used, known for their ability to protect the system against voltage spikes and electrical noise from connected devices.

Each digital input is configured to receive signals from external sensors or switches. The PC817 isolates the input signal from the ESP32 by transferring information through internal infrared light, ensuring no direct electrical contact between high and low-voltage circuits. This isolation method is key to protecting the microcontroller and sensitive components from electrical interference.

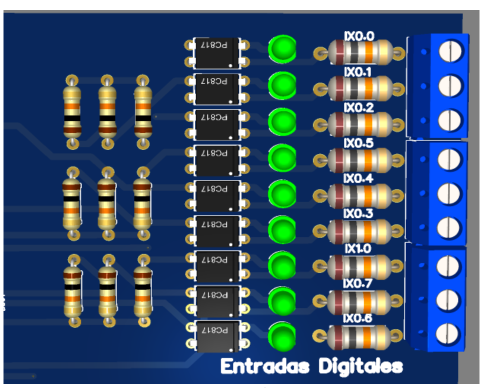

Fig. 6. Opto-Isolated Digital Inputs of the Prototype. Source: Authors.

Additionally, the inputs are equipped with pull-down resistors to ensure a defined logic state when no signal is received, preventing unwanted fluctuations. These resistors, along with protection diodes, ensure stable and reliable digital signals.

As shown in Fig. 7, the prototype includes digital outputs controlled by the ULN2803APG, a Darlington transistor array, and G3MB-202P solid-state relays (SSRs), which allow efficient switching of external AC loads.

The ULN2803APG acts as an intermediary between the ESP32 and the SSRs, handling higher currents without compromising the integrity of the microcontroller's output pins. This transistor arrangement allows the ESP32's low-voltage signals to safely control the SSRs, which switch AC loads without mechanical components. The G3MB-202P SSRs are used to control devices such as motors or lighting systems with greater durability and efficiency.

Furthermore, a separation between AC and DC ground planes was implemented in the PCB design. The AC section is not connected to any ground plane (GND), preventing potential interference and ensuring safe operation. This separation of ground planes on the top and bottom layers of the PCB helps isolate high-power AC signals from low-voltage DC logic components, improving system safety and reliability.

Each digital output also includes indicator LEDs that provide a clear visual signal of the relay status, showing whether an external load is active. Current-limiting resistors are present to protect the LEDs and ensure their proper operation.

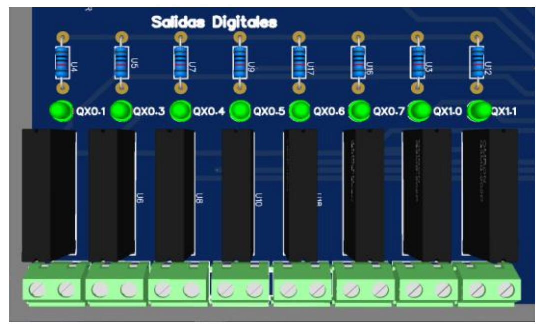

Fig. 7. Digital Outputs with Solid-State Relays in the Prototype. Source: Authors.

As shown in Fig. 8, the design includes two analog inputs configured to receive standard 4-20 mA signals, typical in industrial environments. These inputs use the LM358P operational amplifier to convert current signals into voltage for processing by the ESP32. The 4-20 mA converter allows interaction with industrial sensors, ensuring precise and stable data transmission for variables such as pressure or temperature. The LM358P ensures proper signal conditioning, providing reliable readings for the analog inputs.

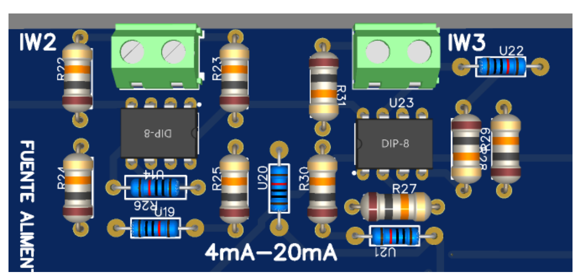

Fig. 8. Analog Inputs with 4-20 mA Converter. Source: Author.

Two analog inputs using potentiometers were included for testing and manually adjusting input values (Fig. 9). The potentiometers allow the input voltage to be varied on these channels, simulating analog signals to verify the correct operation of the system's analog inputs. These potentiometers are connected directly to the ESP32's analog input pins, enabling the system to read and process manually adjusted voltage values. This approach was implemented as a testing tool, allowing the evaluation of the microPLC's response to different signal levels and ensuring the analog inputs function correctly before connecting external sensors.

Fig. 9. Analog Inputs with Potentiometers in the Prototype. Source: Authors.

Two analog outputs configured to generate 0-10V signals were included, based on the LM358P operational amplifier (Fig. 10). These outputs are designed to control external devices requiring adjustable voltage, such as speed drives or actuators.

The LM358P is used to condition and amplify the ESP32's output signals, adapting the low-voltage signal to a 0-10V range. This is crucial for industrial applications, where controlled devices require this standard voltage range. The operational amplifier ensures stable and precise output across the voltage range, providing reliable and efficient control.

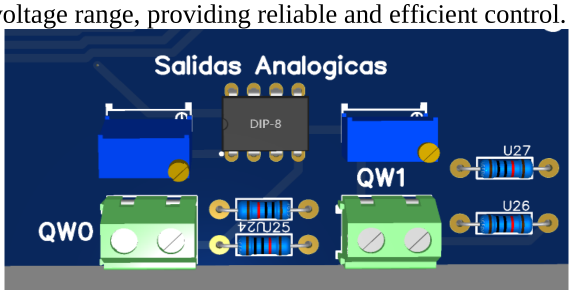

Fig. 10. Analog Outputs. Source: Authors.

These two analog outputs are designed to interact with equipment requiring high-precision analog inputs, enabling easy integration with industrial control systems. Additionally, the design ensures proper separation and isolation between input and output signals, guaranteeing signal integrity.

Finally, as shown in Fig. 11, the final design of the microPLC is presented in 3D format. This device is based on the ESP32-WROOM-32D microcontroller, which serves as the processing core and enables connectivity with both digital and analog inputs and outputs.

D. Comparative Cost Analysis of the Prototype

Table II details the estimated materials cost in the design and construction of the prototype. This list functions similarly to a Bill of Materials (BOM) and represents an approximate cost based on consultation with both local and international suppliers to visualize a reference price [28], [29], [30]. This cost reflects only the value of the electronic and hardware components and does not include expenses associated with assembly, programming or development.

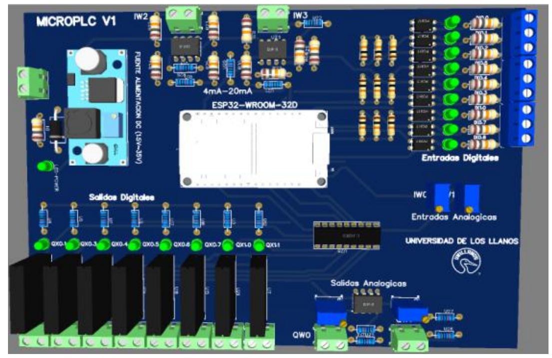

Fig. 11. MicroPLC Prototype for Industrial Applications. Source: Authors.

TABLE II
Pin Configuration for ESP32. Source: Author.

Name	Quantity	Price Unit (USD)
ESP32 30 pins	1	6.75
1N5408	1	0.095
Led(3mm)	18	0.024
CONN-TH_3P-P5.08	3	0.2
CONN-TH_2P-P5.08	13	0.26
PC817C	9	0.17
4.7K 1/2W.	10	0.011
10K 1/4W	9	0.011
LM2596HVS DC-DC	1	2.81
SSR G3MB-202P	8	1.25
R_1/4W	8	0.011
100	2	0.179
100R	2	0.011
100K	4	0.011
100k	2	0.011
10K	2	0.011
10K	2	0.179
LM358P	3	0.239
2.2k 1/4W	2	0.011
100K	2	0.011
56K	2	0.011
24k	2	0.011
ULN2803APG	1	0.548
	Total	28.073

The high initial cost of traditional PLCs is a significant barrier to automation in small and medium-sized enterprises. A commonly used entry-level/mid-range industrial PLC, such as the Siemens LOGO! DC-DC PLC, with comparable I/O capabilities (Figure 12), has a market price of approximately USD 240.76 [32]. This price highlights the cost-effectiveness of the microcontroller-based prototype.

In addition to the hardware cost, the effective implementation of a traditional PLC, such as the Siemens LOGO!, requires licensed software for programming, configuration, and commissioning. For illustrative purposes, the LOGO! Soft

Comfort V8.4 software, specifically required to program this device, represents an additional cost with an approximate market price of USD 45.16 [33], increasing the total cost.

Fig. 12. PLC LOGO! model Source: [31].

IV. CONCLUSIONS

The design of the microPLC prototype, based on the ESP32 microcontroller, successfully achieved efficient integration of electronic components, ensuring proper interconnection between analog and digital inputs and outputs. However, the design process encountered challenges in configuring the input and output pins, which required careful assignment based on the specifications of the OpenPLC software. This highlights the importance of meticulous planning in hardware-software integration for industrial automation systems.

The cost analysis confirms the economic viability of the proposed prototype as an affordable solution for small and medium-sized enterprises. A substantial difference in initial investment is evident: approximately USD 28.073 for the prototype, compared to approximately USD 240.76 for the hardware of a traditional PLC like the Siemens LOGO! DC-DC, plus USD 45.16 for its essential licensed software. This significant cost reduction is achieved primarily through low-cost hardware and the adoption of open-source software such as OpenPLC. By complying with the IEC 61131-3 standard, OpenPLC eliminates the need for expensive proprietary licenses that significantly increase the implementation costs of commercial PLCs.

The power supply system, utilizing the LM2596HV regulator, demonstrated reliable performance in converting high input voltages to appropriate levels for the ESP32, maintaining stable power delivery. The inclusion of a protection diode and an indicator LED facilitated error identification and provided a clear visual signal of the system's power status. Nevertheless, future iterations could explore optimizations in regulator efficiency to enhance overall energy performance.

The opto-isolated digital inputs, implemented with PC817 optocouplers, provided effective electrical isolation, safeguarding the system against voltage spikes and electrical noise. This design feature significantly improved system reliability in industrial environments. However, incorporating additional testing mechanisms to verify signal stability could further enhance system performance and robustness.

Digital outputs, controlled by the ULN2803APG Darlington transistor array and solid-state relays (SSRs), exhibited efficient switching of alternating current (AC) loads. The integration of indicator LEDs proved essential for monitoring the status of each output. For applications requiring higher power handling, additional overcurrent protection mechanisms could be considered to further improve system durability and safety.

For analog inputs, the 4-20 mA converter coupled with the LM358P operational amplifier ensured precise and stable readings from industrial sensors, aligning with industry standards. The inclusion of potentiometers for manual testing allowed straightforward verification of analog input behavior. While effective, expanding the number of input channels could enhance the system's monitoring capabilities, particularly in complex industrial setups.

The 0-10V analog outputs, also based on the LM358P, demonstrated effective control of external devices such as actuators and speed drives. The output voltage range remained stable, making the system suitable for integration into automation frameworks. However, for applications demanding higher precision, exploring operational amplifiers with lower offset and greater accuracy could yield improved performance.

The isolation between AC and DC ground planes was critical in preventing electrical interference and ensuring operational safety, particularly in high-power applications. This design choice enhanced system protection and enabled stable performance in challenging industrial environments. Future iterations could investigate additional filtering techniques to further improve signal integrity and system resilience.

ACKNOWLEDGMENT

This research was conducted by Macrypt and EYSI research groups at the Universidad de los Llanos, in collaboration with the I2E research group at the Universidad Pedagógica y Tecnológica de Colombia. The study is funded by the Universidad de los Llanos (DGI) - Colombia, under the project: Autonomous fruit harvesting system immersed in virtual robotic experimentation platform (2025 FCBI).

REFERENCES

[1] Sima, I. G. Gheorghe, J. Subić, and D. Nancu, "Influences of the Industry 4.0 Revolution on the Human Capital Development and Consumer Behavior: A Systematic Review," Sustainability, vol. 12, no. 10, pp. 4035, May 2020. doi: 10.3390/SU12104035.

[2] L. A. Rodríguez, "Análisis del alcance de las herramientas Agile en la Industria 4.0 (paradigmas 7-12)," Tesis de Grado, Universidad Pontificia Comillas, Madrid, España, 2024.

[3] M. Javaid, A. Haleem, R. P. Singh, R. Suman, and E. S. Gonzalez, "Understanding the adoption of Industry 4.0 technologies in improving environmental sustainability," Sustainable Operations and Computers, vol. 3, pp. 203–217, Jan. 2022. doi: 10.1016/J.SUSOC.2022.01.008.

[4] M. Z. Yaqub and A. Alsabban, "Industry-4.0-enabled digital transformation: Prospects, instruments, challenges, and implications for business

strategies," Sustainability, vol. 15, no. 11, pp. 8553, May 2023. doi: 10.3390/SU15118553.
[5] J. D. León Callupe, "Diseño e implementación de un PLC basado en Raspberry Pi & Node-RED para aplicaciones industriales IoTT," Tesis de Grado, Universidad Ricardo Palma, Lima, Perú, 2021.
[6] N. Agudelo, G. Tano, and C. A. Vargas, "Historia de la automatización," Tesis de Grado, 2024.
[7] C. Stolojescu-Crisan, C. Crisan, and B. P. Butunoi, "An IoT-Based Smart Home Automation System," Sensors, vol. 21, no. 11, pp. 3784, May 2021. doi: 10.3390/S21113784.
[8] I. Stanislav, "The impact of automation on tourism and hospitality jobs," Information Technology & Tourism, 2020. [En línea]. Disponible: https://sci-hub.se/https://doi.org/10.1007/s40558-020-00175-1. [Consultado: 11-ago-2024].
[9] S. Gallegos, M. Angel, and M. C. Forero, "Implementación de la automatización de procesos robóticos (RPA) para mejorar la productividad de la empresa Covisian Perú S.A.," Tesis de Grado, 2024.
[10] E. Cardona and J. R. Gratacós, "Automatización Robótica de Procesos con Inteligencia Artificial," Memoria, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona, Feb. 2023.
[11] E. R. O. Yagua, "Análisis y diseño de un sistema automatizado para el proceso de evaporación en la elaboración de azúcar," Tesis de Grado, 2024.
[12] C. D. M. Molina, "Diseño y simulación de un sistema automatizado para el proceso de mezclado en plantas de producción de bebidas carbonatadas," Tesis de Grado, 2024.
[13] S. D. Mejia Idarraga, H. A. Mendoza Montoya, and L. F. Cabezas Burbano, "Desarrollo de prototipo de sistema de riego automatizado para cultivo de plantas medicinales localizado en la granja agroecológica UCEVA," Tesis de Grado, 2022.
[14] L. Ribeiro and T. B. Sheridan, "Challenges for industrial cyber-physical systems: Security, design, and human interaction," Annual Reviews in Control, vol. 45, pp. 18–28, 2018. doi: 10.1016/j.arcontrol.2018.04.018.
[15] V. Kumar and P. Mehrotra, PLC Programming for Industrial Automation, 2nd ed. New York: McGraw Hill, 2019.
[16] S. Bennett, A History of Control Engineering 1930-1955, Institution of Engineering and Technology (IET), 2005.
[17] G. McMillan and D. Cooper, Process/Industrial Instruments and Controls Handbook, 6th ed. New York: McGraw Hill, 2019.
[18] J. Williams, "Emerging Trends in Industrial Automation: MicroPLC and Beyond," Journal of Industrial Automation, vol. 24, pp. 113–125, 2018.
[19] D. Johnston, "The rise of MicroPLCs in Industrial Automation," Automation Today, vol. 18, pp. 30–35, 2020.
[20] C. Martinez and J. P. Rodriguez, "Evolution of MicroPLC in Modern Industrial Automation," IEEE Transactions on Industrial Electronics, vol. 57, pp. 3207–3216, 2010. doi: 10.1109/TIE.2010.2048837.
[21] M. Rodriguez and J. Hernandez, "MicroPLCs and Their Role in the Modernization of Industrial Automation," Industrial Systems Journal, vol. 43, pp. 76–84, 2019. doi: 10.1007/s10664-019-09779-8.
[22] V. Patel, "IoT Integration with MicroPLCs for Small-Scale Industrial Applications," International Journal of Automation and Computing, vol. 18, pp. 115–123, 2021. doi: 10.1007/s11633-020-1271-4.
[23] W. A. C. Remolina, M. A. A. Robayo, and C. A. Verdugo Duran, "Desarrollo de un PLC con Microcontrolador ATmega328p como Parte de la Integración en Sistemas para la Automatización de Tareas en la Gestión Integral Académico Industrial," Revista de Investigación UTS, Mar. 2020. [En línea]. Disponible: http://repositorio.uts.edu.co:8080/xmlui/handle/123456789/682.
[24] J. Palomino, "Development and Validation of a Less Expensive and Portable PLC Module for Students Training in Industrial Automation," ResearchGate, 2022. [En línea]. Disponible: https://www.researchgate.net/publication/370603355. doi: 10.53314/ELS2226039P.
[25] F. C. Valencia, J. A. Domínguez Chávez, and R. Castañeda Sheissa, "Diseño y construcción de un PLC didáctico para el laboratorio de electrónica," Tesis de Grado, Universidad Veracruzana, 2021. [En línea]. Disponible: https://www.uv.mx/instru/files/2021/01/Tesis_Francisco_Cast.pdf.
[26] T. R. Alves, M. Buratto, F. M. de Souza, and T. V. Rodrigues, "OpenPLC: An Open Source Alternative to Automation," in Proc. IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain, 2014, pp. 1-4. doi: 10.1109/ETFA.2014.6970342.
[27] NeuronicWorks. (2024). *PLC versus Microcontroller Programming | NeuronicWorks*. Retrieved August 11, 2024, from https://neuronicworks.com/blog/plc-versus-microcontroller/
[28] "Products," Ferretronica, (2025). [Online]. Available: https://ferretronica.com/products
[29] "Homepage," Vistronica, (2025). [Online]. Available: https://www.vistronica.com/
[30] Mouser Electronics, (2025). [Online]. Available: https://co.mouser.com/
[31] "PLC LOGO! 12/24RCE-Siemens." Didácticas Electrónicas I+D, (2025). [Online]. Available: https://www.didacticaselectronicas.com/shop/6ed1052-1md08-0ba1-plc-logo-12-24rce-siemens-19373#attr=
[32] Mouser Electronics. "Siemens 6ED1052-2HB08-0BA2 LOGO! 12/24 RCE Logic Module," (2025). [Online]. Available: https://co.mouser.com/ProductDetail/Siemens/6ED10522HB080BA2?qs=jRuttqqUwMSJi0UY0xHKrw%3D%3D
[33] RelePro. "Siemens LOGO!Soft Comfort V8 (6ED1058-0BA08-0YA1)," (May 1, 2025). [Online]. Available: https://relepro.com/modulos-logicos-logo-8/4581/siemens_logosoft-comfort-v8_6ed1058-0ba08-0ya1.html