Graphical user interface for a PLC programming implemented in an ARM Cortex-M4 microcontroller

Interfaz para la programación gráfica de un PLC implementado en un microcontrolador ARM Cortex-M4

EUSEBIO-GRANDE, Raúl†*, IBARRA-BONILLA, Mariana Natalia, AMARO-BALANZAR, Jovanni and SÁNCHEZ-TEXIS, Fernando

Tecnológico Nacional de México, Instituto Tecnológico Superior de Atlixco, México.

ID 1st Author: *Raúl, Eusebio-Grande* / **ORC ID:** 0000-0001-7062-3244, **CVU CONACYT ID:** 1000358

ID 1st Co-author: *Mariana Natalia, Ibarra-Bonilla* / **ORC ID:** 0000-0001-7123-9105, **CVU CONACYT ID:** 237756

ID 2nd Co-author: *Jovanni, Amaro-Balanzar* / **ORC ID:** 0000-0002-2646-7659

ID 3rd Co-author: *Fernando, Sánchez-Texis* / **ORC ID:** 0000-0001-8964-1039, **CVU CONACYT ID:** 233888

DOI: 10.35429/JIE.2022.16.6.28.35 Received January 25, 2022; Accepted June 30, 2022

Abstract

This paper presents the design of a Programmable Logic Controller (PLC) using an ARM Cortex-M4 microcontroller, model STM32F407, which allows integrating the OS-Micropython operating system. To controlling the PLC hardware, the development of a graphical or visual type-programming interface is presented that is easy to use for the operator and incorporates power control functions, such as control of digital inputs/outputs, analog channels readings, timers and counters. The interface communicates with the PLC microcontroller through the UART serial protocol. The firmware microcontroller and the graphical interface were programmed in Python language. Two validation tests for the interface are presented, that probe the correct performance. This work is part of one of the stages of development of a clay sifter- lump-breaker-type machine, which will be operated by ceramic producers in the Cohuecan region in the state of Puebla, Mexico, where it is expected to automate the pottery process and influence the economic reactivation of the region.

Interface, PLC, ARM microcontroller, Python

Resumen

El presente trabajo presenta el diseño de un controlador lógico programable (PLC) utilizando un microcontrolador ARM Cortex-M4, modelo STM32F407, el cual permite integrar el sistema operativo OS-Micropython. Para controlar el hardware del PLC se presenta el desarrollo de una interfaz para programación tipo gráfica o visual que sea de fácil uso para el operador e incorpore funciones de control de potencia, tales como control de entradas y salidas digitales, lectura de canales analógicos, temporizadores y contadores. La interfaz se comunica con el microcontrolador del PLC a través del protocolo UART. El firmware del microcontrolador y la interfaz gráfica se programaron en lenguaje Python. Se presentan dos pruebas de validación de la interfaz que comprueban el correcto desempeño. Este trabajo forma parte de una de las etapas de desarrollo de una máquina tipo desterronador-cernidor de arcilla, la cual será operada por los productores de cerámica de la región de Cohuecan en el estado de Puebla, en México, en donde se espera automatizar el proceso de alfarería de la comunidad e incidir en la reactivación económica de la región.

Interfaz, PLC, Microcontrolador ARM, Python

Citation: EUSEBIO-GRANDE, Raúl, IBARRA-BONILLA, Mariana Natalia, AMARO-BALANZAR, Jovanni and SÁNCHEZ-TEXIS, Fernando. Graphical user interface for a PLC programming implemented in an ARM Cortex-M4 microcontroller. Journal Industrial Engineering. 2022. 6-16:28-35.

* Author's Correspondence (E-mail: raul.eusebio@itsatlixco.edu.mx)
† Researcher contributing as first author.

Introduction

The Programmable Logic Controllers (PLC) represent a key piece in industrial automation systems. Nowadays it is common to find automated processes not only in large scale industries, but also small companies are beginning to look for solutions based on automation to improve their production processes. The PLC main function is to execute control processes and sequences of industrial machinery. The basic elements of a PLC are the memory, the Central Processing Unit (CPU), Input and Output modules, Power Supply and Programming device. For decades, the PLC architecture has not changed (Goldschmidt, 2018), the controller hardware is considered a typical embedded system and the software is an operating system with standardized communication protocols (Mellado & Nuñez, 2022). Additionally, the high cost of acquiring a commercial PLC makes it unattractive or unprofitable to implement in small industries. For this reason, it is important to develop alternative solutions for the design of PLC devices. Some solutions have proposed the use of microcontrollers (Sanver, et al., 2018; Asif, et al., 2016; Rida, et al., 2014), including those found on development boards such as the PLC-Arduino, There is also OpenPLC (OpenPLC, 2022), which has support for Raspberry cards (Andrei, et al., 2020), Arduino (Cohenour, 2018), among others. These last two are OpenSource-type projects, however, in Mexico these types of projects are under development and obtaining the cards is not always easy. Despite the progress reported, today it is still very difficult to find qualified personnel capable of programming and implementing an automated system based on microcontrollers, mainly because of the programming and electronics involved. For this reason, this work proposes the design of a PLC based on an ARM microcontroller, which is considered high performance and allows the OS-Micropython operating system integration that allows programming in Python language. The Python language uses functions with a very clean, simple and compact syntax, which is why it is very easy to learn and very suitable for introducing yourself to systems programming; besides that it is easy to combine with other compiled languages, such as C and C++, which are also widely used languages (Langtangen, 2011).

Then, to facilitate programming, an interface for graphical or visual programming of the PLC is proposed, which is developed in Python language. This in order to be used by people with general automation knowledge, without the need to be programming experts.

The paper organization is as follows: Section 2 describes the general architecture of this propose. Section 3 presents the description of the PLC hardware and section 4 presents the description of the graphical user interface for programming of the PLC. Section 5 presents the preliminary results. Conclusions and work in progress are presented in section 6.

System general description

The proposed system is divided into two main stages, the first corresponds to the graphical programming interface, which is developed in Python language and the second corresponds to the PLC electronic board, which is implemented with an ARM Cortex-M4 microcontroller, model STM32F407. The electronic board has input and output modules implemented with power circuits ready to be used with independent power supplies. The communication between the electronic board and the interface is carried out through the UART interface.

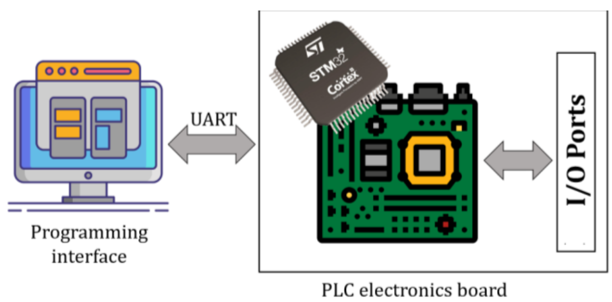


Figure 1 PLC proposal block diagram
Source: Own elaboration

PLC Hardware Description

The PLC proposed design consists of 14 optically isolated digital inputs to receive 12 or 24 VDC signals, 8 independent relay outputs with normally open (NO) and normally closed (NC) operating modes, an expansion port of 16 pins with power supply at 3.3 V that will allow the addition of expansion boards designed for future applications, such as analog-digital converters. The PLC offers a voltage output of 12 VDC at 20W, allowing industrial sensors to be connected directly to the board.

The system can be programmed via an on-board USB port with a micro-USB connector. The relay outputs can work with direct current voltages at 24 VDC, or alternating current at 127-220 VAC. The digital outputs are designed to work with industrial contactors that are ultimately responsible for driving high current loads. An indicator LED is integrated to display the status of each of the inputs and outputs.

ARM microcontroller

The PLC is implemented in an ARM Cortex-M4 microcontroller, model STM32F407 from the ST Microelectronics manufacturer. The microcontroller internal architecture includes a CPU with 100 kB RAM memory, 1 MB Flash memory dedicated to program memory. This microcontroller is considered high-performance because it includes multiple peripherals that are useful for the implementation of the PLC, such as: RS-232 or RS-485 serial communication ports, analog inputs with an analog-digital converter (ADC) with resolution of 12-bit and USB port with a factory integrated Bootloader or on-site programming mode.

Power Digital inputs

The circuit of each of the 14 digital inputs is presented in Figure 2. As can be seen, the input circuit to the optocoupler U5, model PC81713NIPOX, consists of a resistor R4 in series with the status indicator LED, D3. R4 is responsible for limiting the current flowing through the circuit. The output circuit consists of a resistor R6 connected in series to the internal transistor. The output is connected from the collector terminal (4) to the digital terminal of the microcontroller C0. This configuration was designed as an inverted logic, so that in the presence of 12 V at the input, the PLC will detect a logic 0 state. All inputs share a common node or GND, however, this node is isolated from the common node of the microcontroller and the rest of the electronics. The GND node is offered as a screw terminal, this allows the user to use an external source to power the outputs.

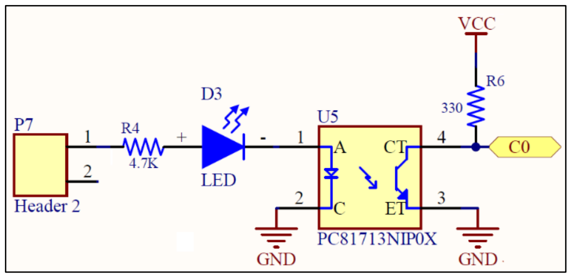


Figure 2 Digital input circuit
Source: Own elaboration

Power Digital outputs

Relay outputs require optical isolation between the microcontroller and the power elements. The circuit of each of the 8 outputs is shown in Figure 3. Terminal D0 corresponds to a microcontroller output pin to which indicator LED D17 and the internal LED of the PC81713NIPOX optocoupler are connected in series. The output circuit uses a 12 VDC source and consists of a series circuit of resistors R40, R41 and transistor (C-E), where the emitter terminal (3) is connected to the base of an external transistor Q1, model MMBT3904LT1G, operating in switch mode. The Collector-Emitter circuit of Q1 is a series circuit of the 12 VDC source and a reverse connected diode D25 for protection of Q1 during the switching process. Also D25 is connected in parallel to the coil of relay K1. The common terminal of each relay is independent, this feature makes the outputs more versatile, because they allow the user to connect different voltage loads to each output.

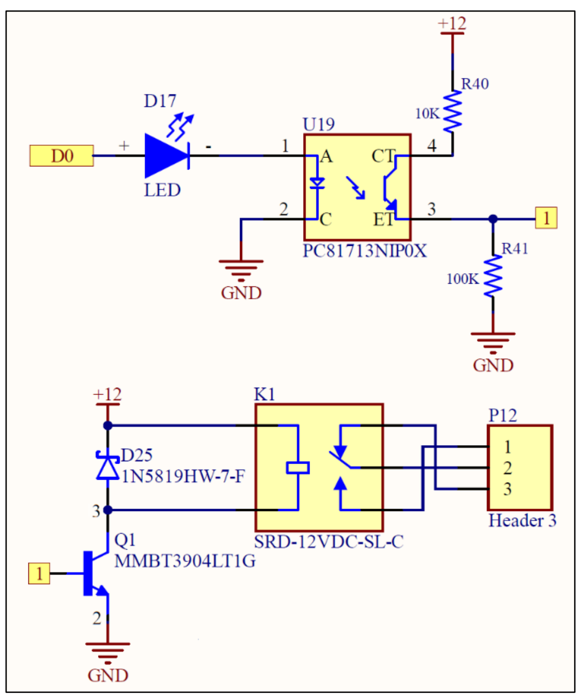


Figure 3 Digital output circuit
Source: Own elaboration

ISSN 2523-0344

Power supply

The PLC internal circuitry is powered by two independent sources of 12 VDC at 20W. Each source is a compact integrated module encapsulated in a protective plastic that allows direct connection to 127-220 VAC. Although the sources have protection against overcurrent and short circuit; at the input of both sources, a fuse protection circuit was implemented, which protects against short circuits, a shunt capacitor-transformer filter circuit that protects against voltage transients or harmonics, and finally, a varistor that protects against voltage spikes. One of the supplies is used to power the microcontroller and the relays. The microcontroller has its own 3.3 VDC switched DC-DC supply. This supply draws power from the main 12 VDC supply mentioned above. The second source is intended as a power supply integrated to the PLC. This source will allow the user to directly power the industrial sensors that are connected to the board, through a screw terminal.

PLC programming interface description.

In addition to the capabilities of the PLC hardware, it is important in a design to consider the means by which the hardware will be controlled. Therefore, most commercial PLC manufacturers implement their own user interface through which programming is carried out. Some of the most common languages are: ladder diagram, Grafcet and blocks. In this work, a graphic or visual programming is proposed so that anyone with general knowledge of automation is able to use it in automation processes. The operation of the interface is divided into two main parts, the programming of the firmware in the microcontroller and the programming of the user interface in the computer. Both systems are programmed in a Python language and the logic is described below.

ARM microcontroller firmware

The microcontroller firmware programming was developed in MicroPython language, implementing functions for the control of digital inputs and outputs, timers, counters, logic functions and analog inputs. Communication with the microcontroller and the programming interface is carried out via UART.

Figure 4 shows a microcontroller programming logic flowchart. The first step is to configure the serial communication with the computer, through a COM port, later it receives the configurations made by the user from the computer interface, to finally execute the functions corresponding to the configuration by the user.

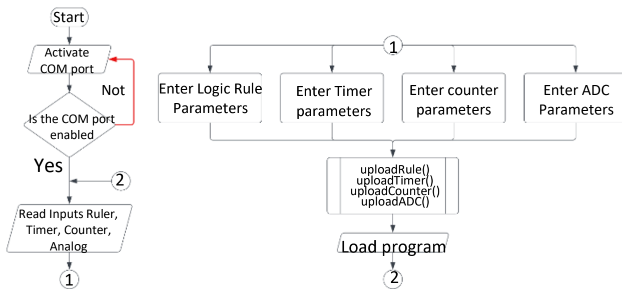


Figure 4 Microcontroller firmware logic flowchart
Source: Own elaboration

Programming interface

The interface with which the PLC is programmed is developed in Python language using the Tkinter library. It is designed as a block-based graphical interface for easy and simple use. Among the configurations available in the programming interface are the following: digital inputs and outputs, timers, counters, logic rules and analog inputs. Figure 5 shows a very general operation flow diagram of the user interface, which is executed from a personal computer (PC).

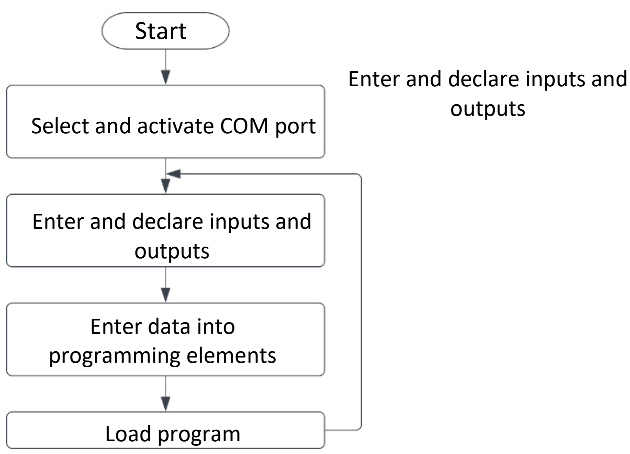


Figure 5 Flowchart of the user interface logic
Source: Own elaboration

ISSN 2523-0344

Results

The proposed work is in progress, then the preliminary results obtained during the development were obtained using a commercial development board that incorporates the same STM32F407 microcontroller (Intesc, 2022) are described below. Figure 6 is a full view of the graphical user interface. It consists of 6 main block parts: 1) Serial communication configuration. 2) Program output terminal. 3) Reading of analog channels. 4) Logic functions 5) Timers and 6) Counters. It should be noted that these functions can be configured by the user according to their needs.

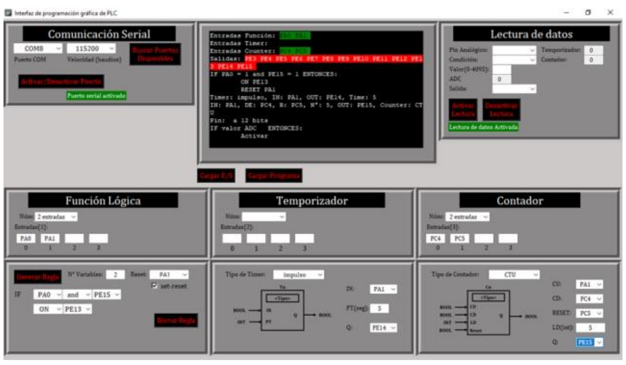


Figure 6 Graphical user interface for PLC programming
Source: Own elaboration

To verify the operation, two practices are described below, one of a small automation process and the second on the reading of an analog signal.

Test 1

Below is a small example of automation of a process which consists of the following steps:

A conveyor belt (represented by a green LED) moves some containers.

1. When the sensor S1 (represented by button 1) detects the container, the conveyor stops.

2. Cylinder A1 extends (represented by a white LED) to open a gate for 5 seconds and fill the container positioned below with some material.

3. Subsequently, an internal counter will be increased to repeat this action 5 more times.

4. There are also two other buttons, one for Start and another for Reset.

Figure 7 shows the configurations for the PLC to execute the actions required by the previously described process. Mainly it is possible to appreciate the configuration of logic functions of AND type, the impulse timer and an ascending counter.

Figure 8 shows the PLC connection diagram and its different components.

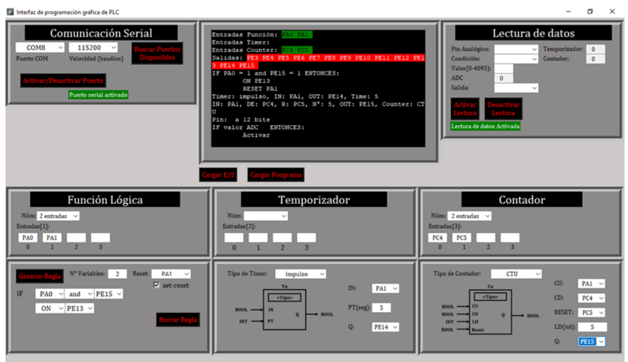


Figure 7 PLC programming for test 1
Source: Own elaboration

The counter and timer values will start at 0 and the only led on will be red, indicating that the counter is not equal to 5. Figure 9 shows the implementation and programming of the counter.

Figure 10 shows the result after pressing the start button, the green led turns on, which represents the motor of the conveyor belt. Figure 11 shows the indicators when the conveyor has been turned off.

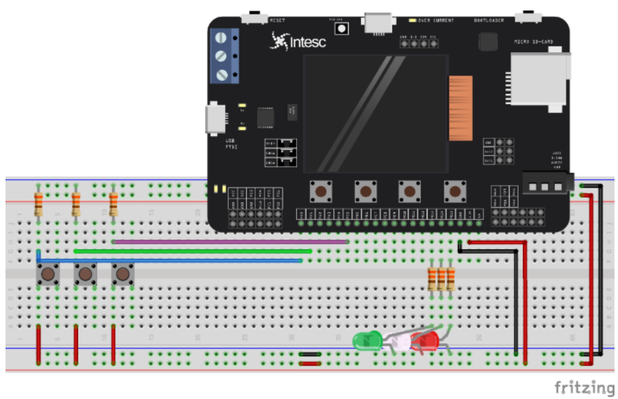
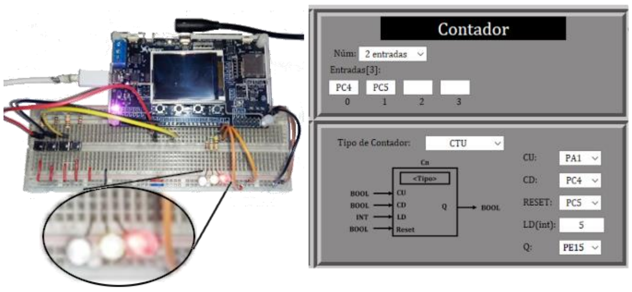


Figure 8 Electrical connections for test 1
Source: Own elaboration

ISSN 2523-0344



a) b)

Figure 9 a) Electrical connections implementation. The red led on indicating that the counter is not equal to 5. b) Counter configuration
Source: Own elaboration

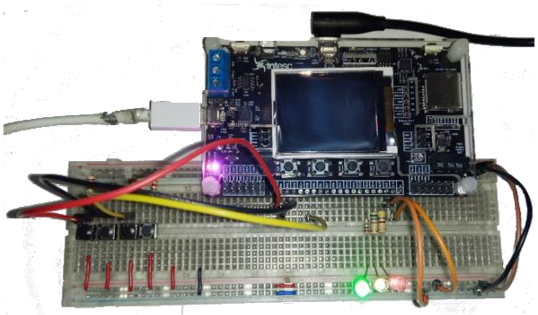


Figure 10 The green led on indicating that the conveyor belt is moving
Source: Own elaboration

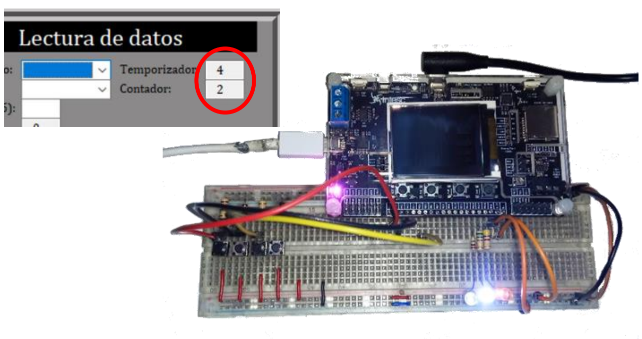


Figure 11 Time and counter indicator on the interface and white led on indicating that cylinder "A" is on, the green led is off indicating that the band is off
Source: Own elaboration

Test 2

Test 2 consists of evaluating the operation of the analog inputs, for this, in Figure 12 we show a connection circuit which consists of an analog potentiometer and a led.

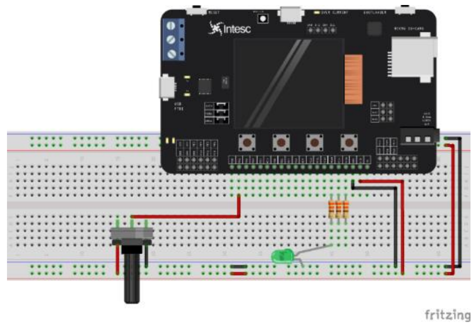


Figure 12 Connection of the potentiometer to the PLC and led
Source: Own elaboration

Figure 13 shows the configuration and visualization of the data of the analog channels. The analog pin section is the name or address of the physical pin of the microcontroller, the condition is the conditional element for some action to happen, the value section (0-4095) is the minimum and maximum value that can be read, remember that the ADC of the microcontroller is 12 bits, ADC in the current reading and finally the output is the digital pin that will be activated or deactivated depending on whether the rule that we have programmed is fulfilled or not.

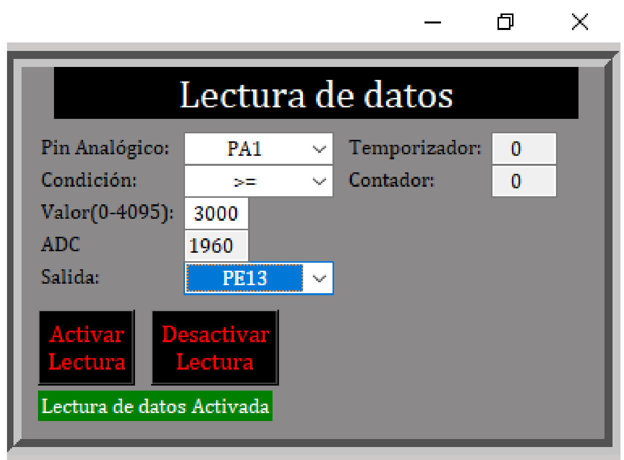


Figure 13 Interface for configuration and display of analog channels
Source: Own elaboration

Figure 14 shows the result obtained when the analog reading is below 3000, a condition that we established for the LED to turn on.

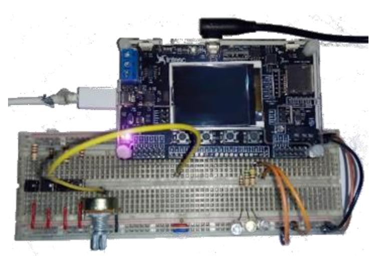


Figure 14 Physical connection of the potentiometer to the PLC and led
Source: Own elaboration

Figure 15 shows the operation of the PLC and its programming by turning on the LED when the potentiometer exceeds the programmed level.

Figure 15 The green led turns on when the potentiometer exceeds the programmed level
Source: Own elaboration

Conclusions

The electronic design proposal for a PLC using an ARM microcontroller with a graphical programming interface was presented. This work is part of one of the design stages of a development project for an industrial-type machine to sift clay, in which a three-phase Power motor will be used and will be operated by ceramic producers in the Cohuecan region in the state of Puebla, in Mexico. For this reason, a graphic control interface was proposed for easy operation and the PLC has relay power outputs that can work with direct current voltages at 24 VDC, or alternating current at 127-220 VAC.

The PLC printed circuit board is in the manufacturing process, so only the performance of the graphical interface was validated, which through two tests could verify its proper functioning.

Future work is to finalize the PLC and perform tests with high power elements.

Acknowledgment

The authors appreciate the support given to PRODEP, for the realization of this work.

Financing

This project is financed by PRODEP through the Support Program for the Strengthening of CA with official assignment M00/2250/2021.

References

Andrei, C. C., Tudor, G., Arhip-Călin, M., Fierăscu, G., & Urcan, C. (2020, November). Raspberry Pi, an Alternative Low-Cost PLC. In 2020 International Symposium on Fundamentals of Electrical Engineering (ISFEE) (pp. 1-6). IEEE. https://doi.org/10.1109/ISFEE51261.2020.9756175.

Asif, M., Raza, A., Sultan, A., & Malik, F. (2016). Design of mini PLC based on PIC18F452 microcontroller using concepts of graceful degradation. University of Engineering and Technology Taxila. Technical Journal, 21(1), 51. Retrieved from: https://www.proquest.com/scholarly-journals/design-mini-plc-based-on-pic18f452/docview/1802563078/se-2

Cohenour, C. (2018, June). An Arduino-based Programmable Logic Control (PLC) Lab Activity for Undergraduate Engineering and Technology (ETM) Students. In 2018 ASEE Annual Conference & Exposition. Retrieved from: https://peer.asee.org/29775. doi: 10.18260/1-2--29775.

Goldschmidt, T., Hauck-Stattelmann, S., Malakuti, S., & Grüner, S. (2018). Container-based architecture for flexible industrial control applications. Journal of Systems Architecture, 84, 28-36. https://doi.org/10.1016/j.sysarc.2018.03.002

Intesc (June 17th, 2022). Ophyra Diseñada con ARM [Archivo PDF]. Retrieved from: https://intesc.mx/wp-content/uploads/2021/08/ManualOphyraRevE.pdf.

Langtangen, H. P. (2011). A primer on scientific programming with Python (Vol. 1). Berlin/Heidelberg: Springer. https://doi.org/10.1007/978-3-662-49887-3.

Mellado, J., & Núñez, F. (2022). Design of an IoT-PLC: A containerized programmable logical controller for the industry 4.0. Journal of Industrial Information Integration, 25, 100250. https://doi.org/10.1016/j.jii.2021.100250.

OpenPLC (June 17th, 2022) Open Source, Open Software Retrieved from: https://openplcproject.com/docs/openplc-overview/

Sanver, U., Yavuz, E., Eyupoglu, C., & Uzun, T. (2018, January). Design and implementation of a programmable logic controller using PIC18F4580. In 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) (pp. 231-235). IEEE. https://doi.org/10.1109/EIConRus.2018.8317072

Rida, M. E., Liu, F., & Jadi, Y. (2014, April). Design Mini-PLC based on ATxmega256A3U-AU microcontroller. In 2014 International Conference on Information Science, Electronics and Electrical Engineering (Vol. 2, pp. 1034-1037). IEEE. https://doi.org/10.1109/InfoSEEE.2014.6947826.