DEVELOPMENT OF A PLC FOR TEACHING HYDRAULIC AND PNEUMATIC SYSTEMS

10.56238/edimpacto2025.005-007

Raphael Lopes Martins[1], Adriana da Silva Torres[2], Moisés de Matos Torres[3]

ABSTRACT
The objective of this work is to create a programmable logic controller (PLC) for use in the pneumatics bench of the mechanics laboratory of UFVJM. It was created from an ATmega328 microcontroller, 7805 voltage regulators to control the input signals, and relay modules to control the output signals. The PLC has an SD memory card reader to read and interpret a previously made program in a *software*. This *software* was created with the use of Processing®, specifically to program the PLC from a graphical language. The *software* features an intuitive interface with all the input and output symbols needed to create programming logic in the Ladder language. Thus, all the work allows a better understanding of how the electrical and microcontrolled operation of a PLC, created for teaching in the discipline of hydraulic and pneumatic systems, should be. It also envisions the possibility of optimizing the equipment to allow even its industrial application.

Keywords: PLC, Processing, Ladder Language, ATmega328, Voltage Regulator, Relay Module.

[1] Federal University of the Jequitinhonha and Mucuri Valleys
[2] Federal University of the Jequitinhonha and Mucuri Valleys
[3] Federal University of the Jequitinhonha and Mucuri Valleys

INTRODUCTION

With the growth in demand for manufacturing products to meet the needs of consumers, companies needed to design production lines that are able to build their products quickly and with a good quality standard. Industrial automation emerged as an alternative for these companies (ANDRADE, 2018). It was not only able to coordinate the entire production process, but also provided the guarantee of maintaining quality standards, since manufacturing activities were carried out by programmed machines. However, these companies were still very affected by the innovations of their products. Whenever its products were innovated, it was also necessary to modify all or most of its production line (ANDRADE, 2018). These modifications generated intense expenses in the implementation of the new line for the manufacture of the new product models.

The Programmable Logic Controller (PLC) proved to be a great tool for modifying functions of a production line without the need to reformulate control panels that controlled some machines (ANDRADE, 2018). The PLC, as the name implies, is an electronic equipment (*hardware*) that can be programmed for different industrial applications.

According to Andrade (2018), for NEMA (*National Electric Manufacturers Association*) the PLC is a digital electronic device that uses programmable memory to store instructions internally and for the implementation of specific functions, such as logic, sequencing, timing, arithmetic counting, among others.

Given the importance of this electronic device for the industry in general and for the teaching of hydraulic and pneumatic systems (HPS) within educational institutions, the present work proposed to build and test the operation of a low-cost PLC. This was built from a microcontrolled integrated circuit (IC), in which a program containing instructions for its operation can be used. The IC recognizes programming in the C language, however, to adapt to the existing PLC standards, software was developed so that all programming is done in the Ladder language (ANDRADE, 2018).

The CLP produced will be donated along with the *software* of its programming, so that it can be used for teaching in the classes of the SHP subject of the Mechanical Engineering course at the Federal University of the Jequitinhonha and Mucuri Valleys.

OBJECTIVES

The general and specific objectives of this work are presented below.

GENERAL OBJECTIVE

Construction of a PLC from a microcontroller and development of a *software* for its programming in the Ladder language.

SPECIFIC OBJECTIVE

a) Construction of a PLC from a microcotrolator of the Atmega328p model.
b) Development of a circuit for switching a voltage above that supported by the microcontroller.
c) Voltage regulation of input signals. These will be regulated from 24 volts to 5 volts.
d) Elaboration of a program with a graphical interface for PLC programming from a Ladder language.
e) Elaboration of a program in C language, to be stored inside the microcontroller's memory. He should be able to understand and execute what was previously programmed in Ladder in the graphical interface created in the other program.

STATE OF THE ART

In this chapter, the meaning and application of CLP will be presented. In addition, the components that can make up your hardware will be presented. Finally, the *software* used to create a graphic programming and also the PLC programming will be presented.

PLC

Programmable Logic Controller *(PLC) is a digital electronic device like the one shown in Figure 1, capable of being programmed via* software *to perform some specific control task in a machine or in other industrial applications (ANDRADE, 2018).*

The PLC has 3 parts that must be taken into account. Inputs, outputs and programming module (ANDRADE, 2018). The inputs of the PLC represent your sensitive part. They are the ones that receive the signals coming from a system. For PLCs with digital inputs, there are only two possible variables in their operation. They detect only the on or off states, i.e. electronically they detect only 0 or 1. The CLP outputs, on the other hand, are the order executors (ANDRADE, 2018). For example, it is the gates that, after receiving information from the processing of the program, send a pulse to switch the valve of a pneumatic actuator.

According to Andrade (2018), the PLC programming module is composed of CPU, processor and memory. This part corresponds to the brain of the CLP. It is there that the program containing the instructions for interpreting the signals that will arrive at your entrance doors is stored.

The value of a simpler PLC from the Siemens® brand, the same brand of PLC used in the pneumatics laboratory of Mechanical Engineering at UFVJM, is around 2000 reais (PRODUCT. MERCADOLIVRE, 2018).

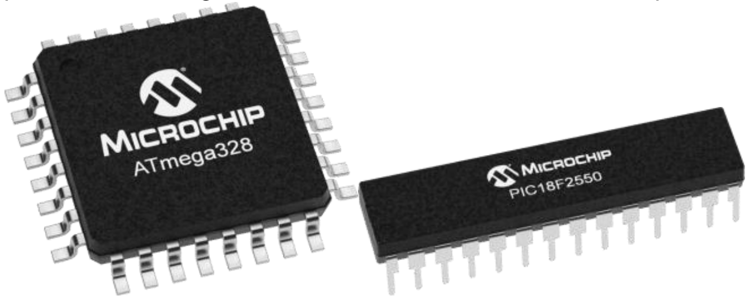
Figure 1: Image of a PLC model.

Source: MERCADO LIVRE, 2018.

MICROCONTROLLERS

Microcontrollers are ICs (integrated circuits), as shown in Figure 2, which can be programmed via *software* to perform a function determined by the programmer (AURELIANO, 2018). There are several types of microcontrollers on the market, such as: 8051, PIC 16F877A, Atemga328P or PIC16F628A. In this work, it was decided to use a microcontroller manufactured by the company Atmel. Atmel's microcontrollers are CIs that have a clear and dynamic programming based on the C language, but with their own library. They are also the most used ICs today, due to their instinctive interconnection with various types of sensors and existing sensor modules (HOME.ROBOTICLAB.EU, 2018).

Figure 2: Representative image of two models of microcontroller chips on the market

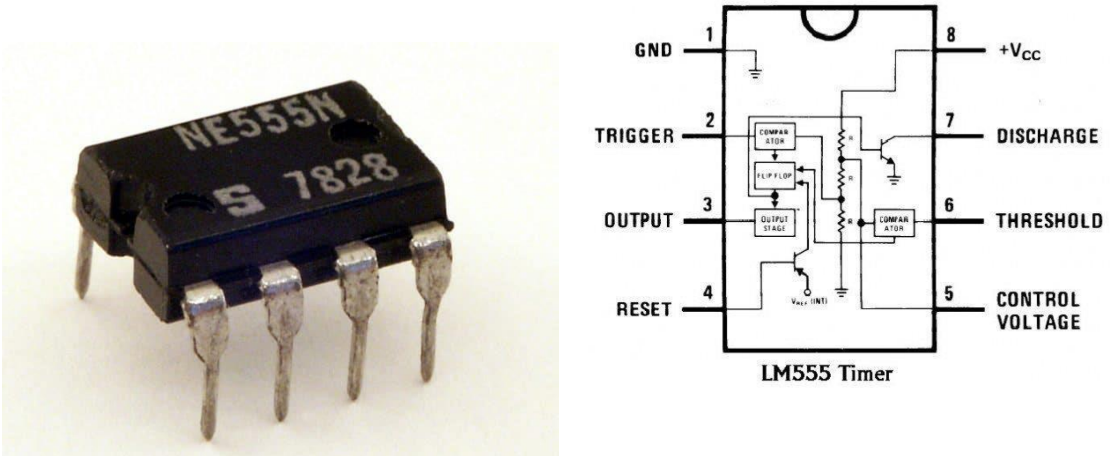


Source: MICROSHIP, 2018.

INTEGRATED CIRCUIT

An integrated circuit, such as the one shown in Figure 3, is an electronic circuit that incorporates miniatures of various electronic components (transistors, diodes, resistors, capacitors) that are "recorded" on a small sheet of silicon (NEWTECK, 2018). The chip is mounted and sealed in a plastic or ceramic shield. From inside the protection, terminals come out that are internally connected to the electronic components by conductive wires.

Figure 3: Representative image of an integrated circuit used to create digital oscillations.



Source: EMBARCADOS, 2018, AUTOCOREROBOTICA, 2018.

ARDUINO BOARD

Arduino is an embedded board that facilitates the programming of a specific microcontroller. Examples of Arduino boards are shown in Figure 4. Generally, the microcontrollers used in the Arduino board are IC's from the company Atmel. The board's embedded function is expressed in its connections, which are intended to facilitate contact with the IC terminals.

In addition to facilitating contact with the terminals of the microcontroller chip, the

Arduino also has voltage regulators that allow the board to be powered with up to 12 Volts (ARDUINO, 2018). It also has a USB Serial converter chip that allows not only the communication of the computer with the microcontroller, but also processes the passage of the computer's programming to the internal memory of the chip.

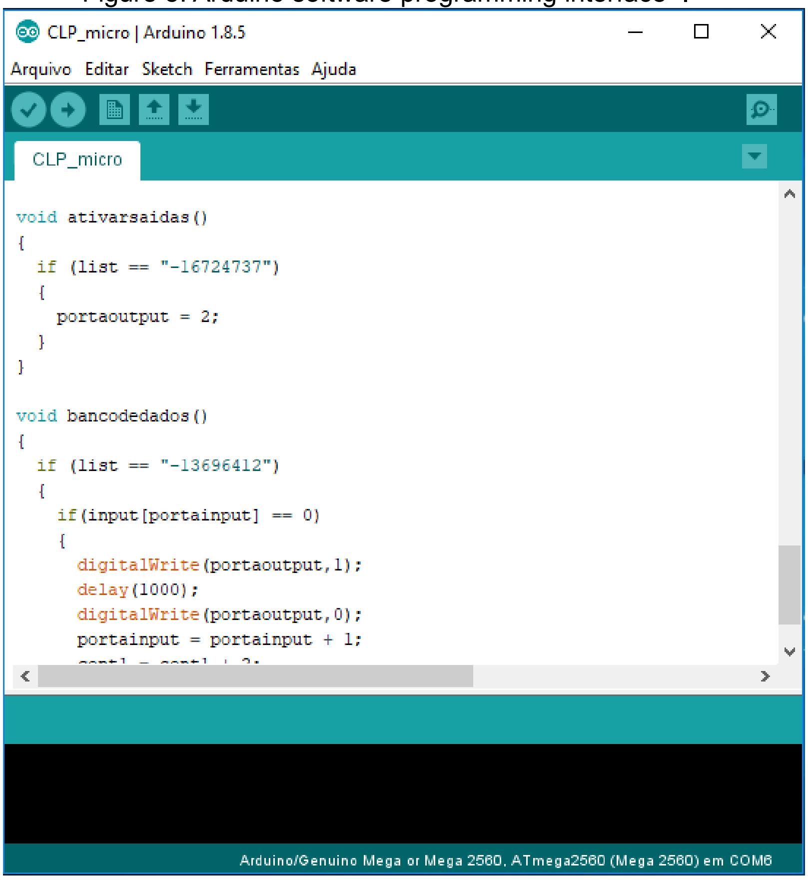
Figure 4: Representative image of an embedded board of the Arduino Pro Mini model of 5 volts and 16MHz.

Source: YOUROBOT-ELECTRONICS, 2018.

ARDUINO SOFTWARE

The Arduino software is a free program manufactured by the Italian company Arduino and is used to program various microcontrollers from the company Atmel. Its programming uses C language, but the program has its own library for its use.

Figure 5: Arduino software programming interface .

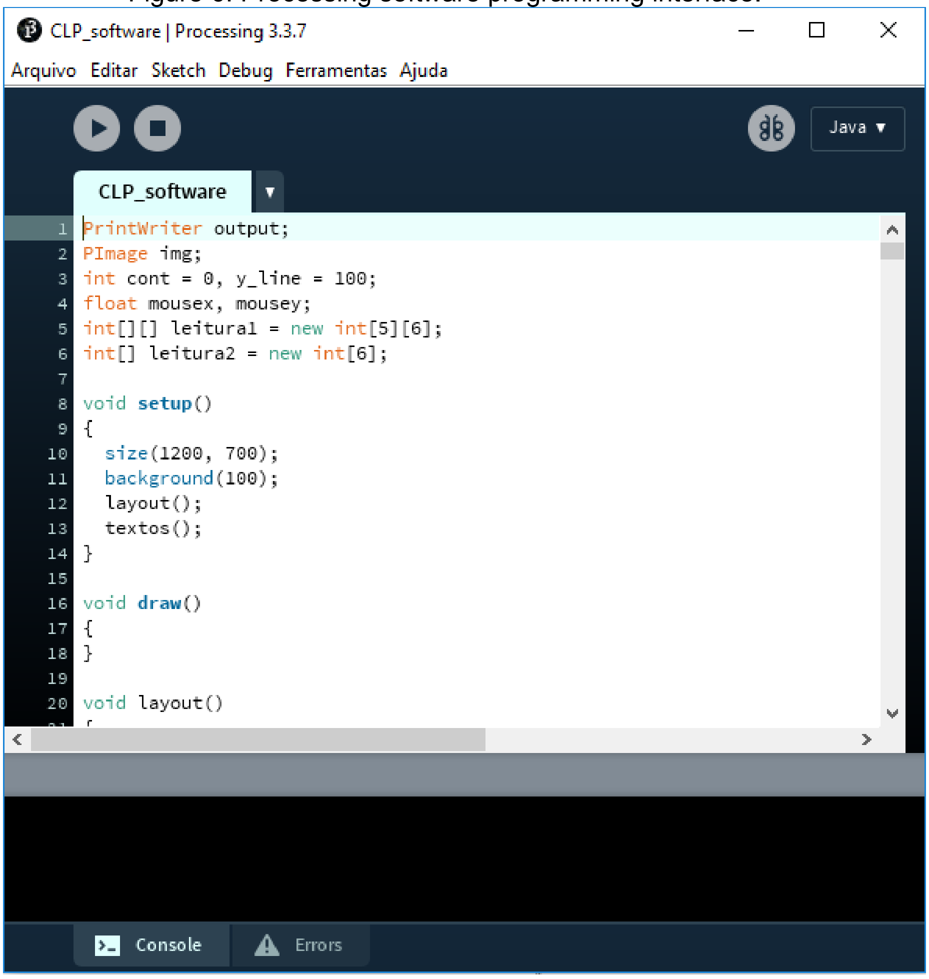


Source: Author himself.

SOFTWARE PROCESSING

Processing is a free program developed by MIT (Massachusetts Institute of Technology) and based on the Arduino software. It serves not only to program Atmel chips, but also to create graphical interfaces for interaction, such as games, control panels and even plot two- and three-dimensional graphics. Its programming is based on C, C++ and Java, however it has its own library for use. Figure 6 shows the program interface.

Figure 6: Processing software programming interface.

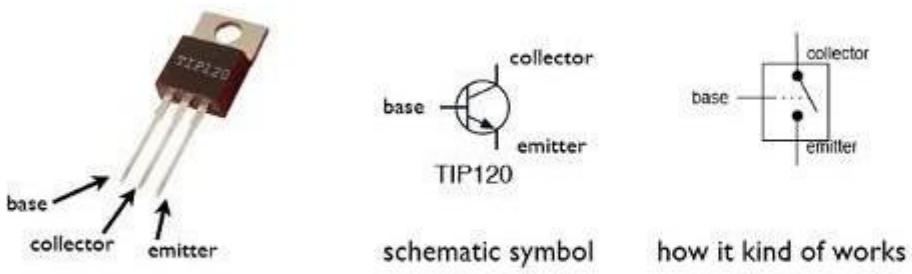


Source: Author himself.

TIP120

The TIP120, as shown in Figure 7, is a medium-power NPN transistor, used for switching and amplification applications (BAUDAELETRÔNICA, 2018). It basically works as an "electronic switch". The voltage required for its activation at the base is 5 Volts, being able to switch a voltage of up to 60 Volts at the collector-emitter.

Figure 7: Electrical symbol and operating model of a TIP120 transistor.

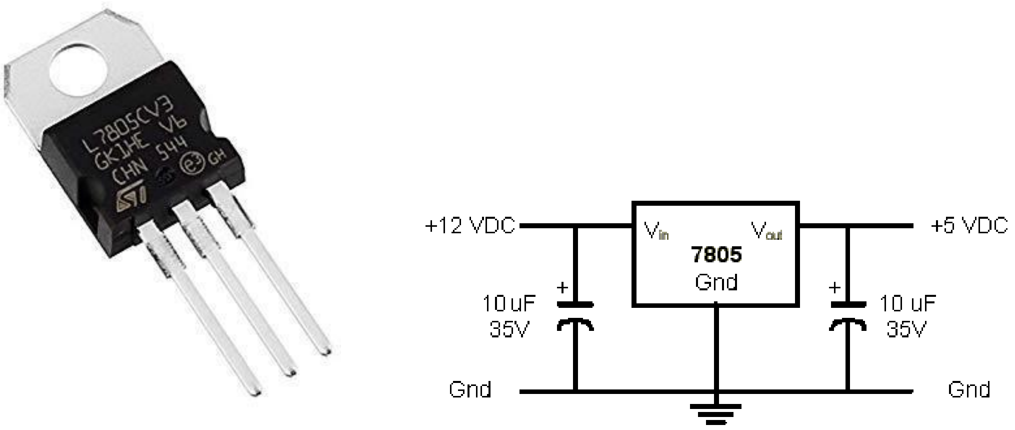


Source: MERCADOLIBRE, 2018.

VOLTAGE REGULATOR 7805

The 7805, as shown in Figure 8, is a voltage regulator capable of regulating input voltages up to 25 Volts, and converting them to 5-volt voltages. It is capable of withstanding a current of up to 1.5 A (SPARKFUN, 2018).

Figure 8: Voltage regulator 7805 and the electrical diagram of its connection

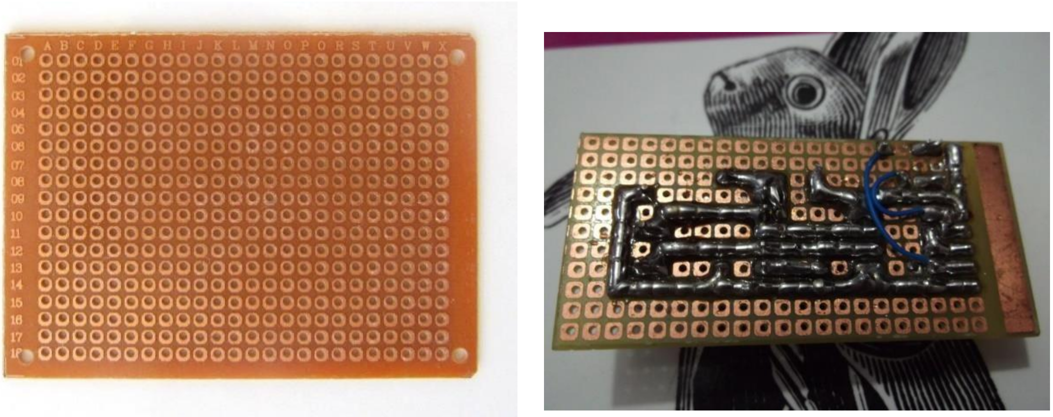


Source: AMAZONA, 2018. QUORA, 2018.

PRINTED CIRCUIT BOARD (PCB) ISLANDED

An islanded PCB is a phenolite board with copper-containing terminals. This board allows prototypes of electrical circuits to be made just by connecting the copper "islands" with tin to form the tracks. That is, trails are formed without the need to corrode the iron perchloride board as a PCB is normally made.

Figure 9: Stranded PCB and the way tin trails are formed on its surface.

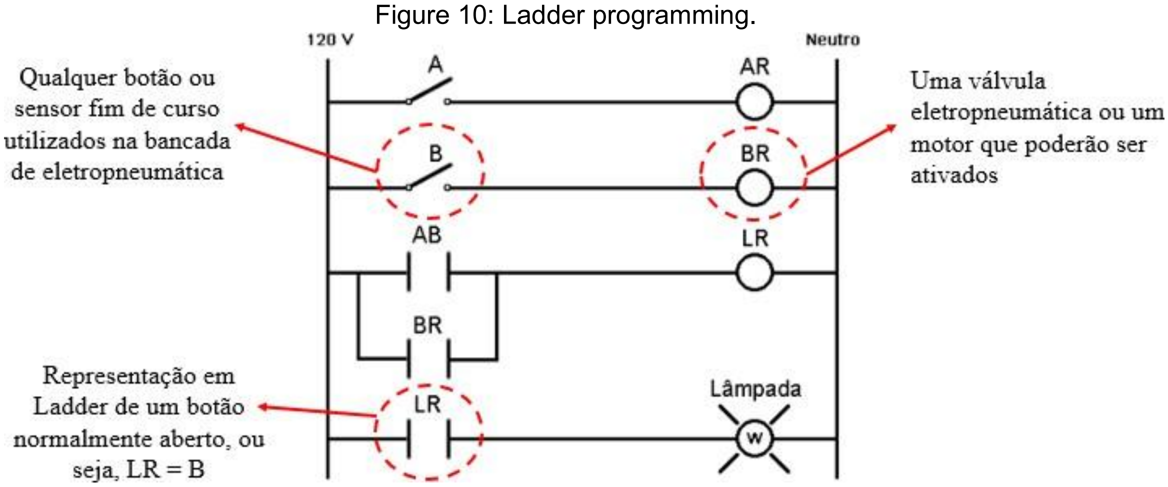


Source: ARDUINOLANDIA, 2018, LABORATORIODEGARAGEM, 2018.

LADDER PROGRAMMING LANGUAGE

The Ladder language, as shown in Figure 10, was the first programming language developed for PLC (SILVEIRA, 2018). It is a language very similar to electrical diagrams, since it originated to replace them. Ladder programming uses relay logic, with switches and coils, thus being simpler and easier to be assimilated by operators with knowledge of electrical circuits and commands (ENSINANDOELÉTRICA, 2018). It is composed of several horizontally arranged circuits, with the coil at the right end, fed by two lateral vertical bars. Because of this format, it is called *ladder*, which means ladder in English.

Figure 10: Ladder programming.



Source: CITISYSTEM, 2018. Adapted.

MATERIALS AND METHODS

In this chapter, the materials used to build the PLC will be presented. The methods used to assemble the electronic part and the *software* responsible for programming the equipment in a graphic language will also be presented.

MATERIALS

In this section, the materials used for the development of the PLC are presented.

a) Arduino Pro Mini 16MHz 5 Volts Board;
b) Isolated phenolite plaque;
c) SD Card Module;
d) 220 ohm resistors;
e) Transistor TIP120;
f) Voltage regulator LM7805;
g) Male vs. male jumpers;
h) Female x female jumpers;
i) MDF box;
j) Terminal blocks for banana pin fitting

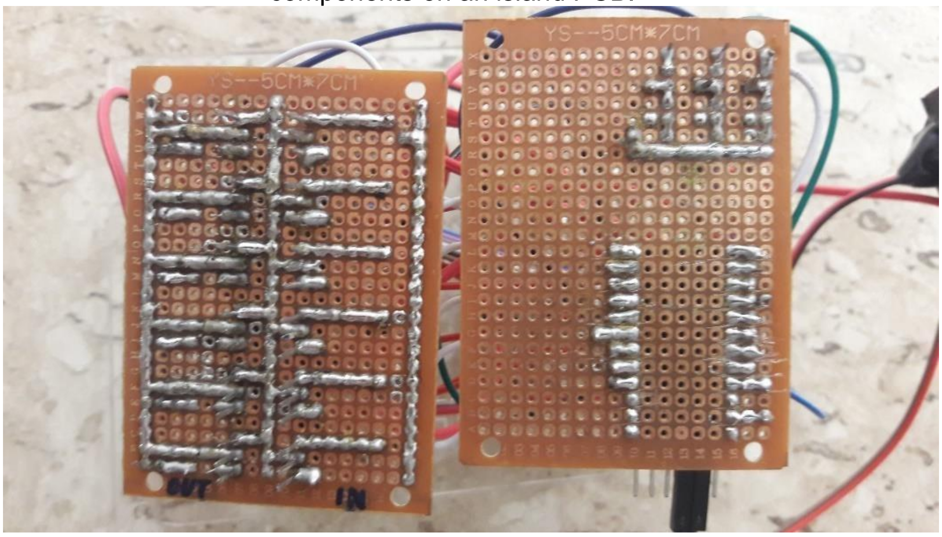
METHODOLOGY

In this section, the methodology used for the construction of the PLC is presented. In addition, it shows how it was created and how the software can be used for its programming.

Construction of the electronics

To start the development of the PLC, isolated phenolite plates were used to assemble the prototype of the circuit, as can be seen in Figure 11. On the phenolite board, the Arduino pro mini board was soldered.

Figure 11: Representative image of the development of the PLC of this work from the soldering of components on an island PCB.

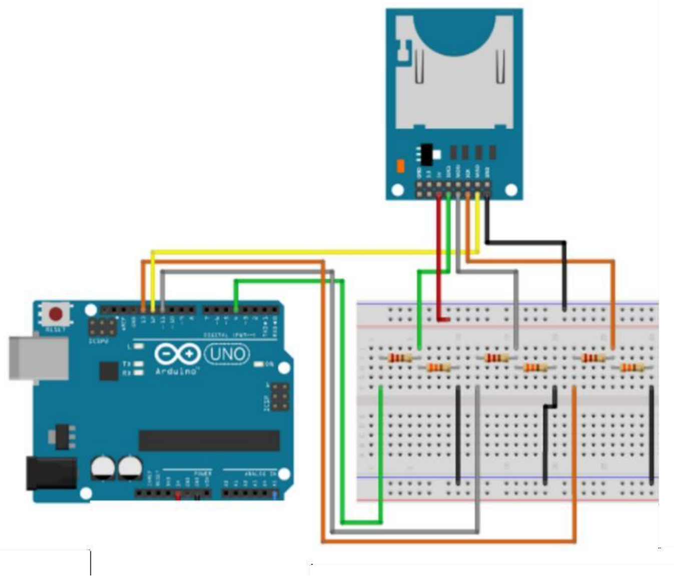


Source: Author himself.

Then, the connections of the SD Card module with the Arduino board were made. This module will allow information inside a mini SD card to be read and later interpreted by the programming.

Some pins of the SD Card module support a maximum of 3.3 Volts. Since the Arduino board provides a higher voltage, which is 5 Volts, it was necessary to create a small voltage divider circuit to reach the required voltage. Figure 12 shows the diagram of the module connection:

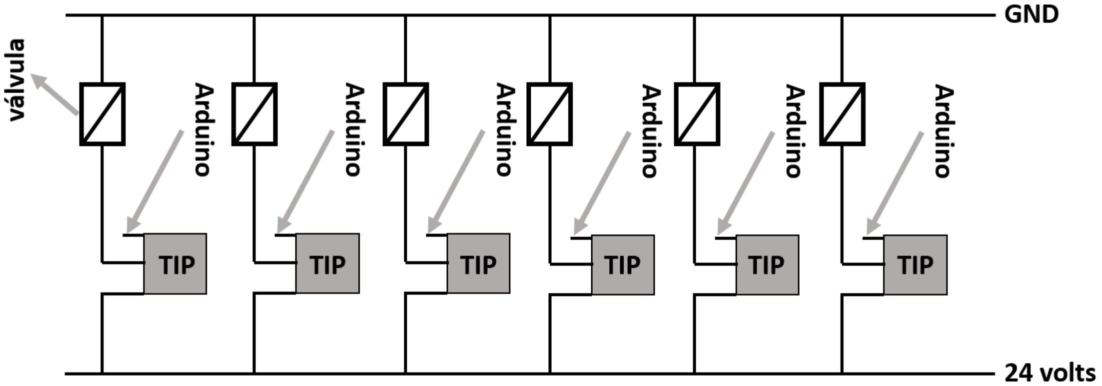
Figure 12: Electrical way of connecting the PLC SD card module of this work.



Source: FILIPEFLOP, 2018.

The electropneumatic and electrohydraulic valves of the hydraulic and pneumatic system benches require 24 volts to be activated. However, the microcontroller of the Arduino board provides only 5 Volts. To provide the activation of these valves, a circuit was developed for switching the 24 Volts from the use of TIP120 transistors. These transistors basically work like switches. From a signal of only 5 Volts that arrive at their base, they are capable of driving an electrical circuit of up to 60 Volts. Therefore, they are sufficient for our project. Figure 13 shows the design for switching using these TIP120 transistors.

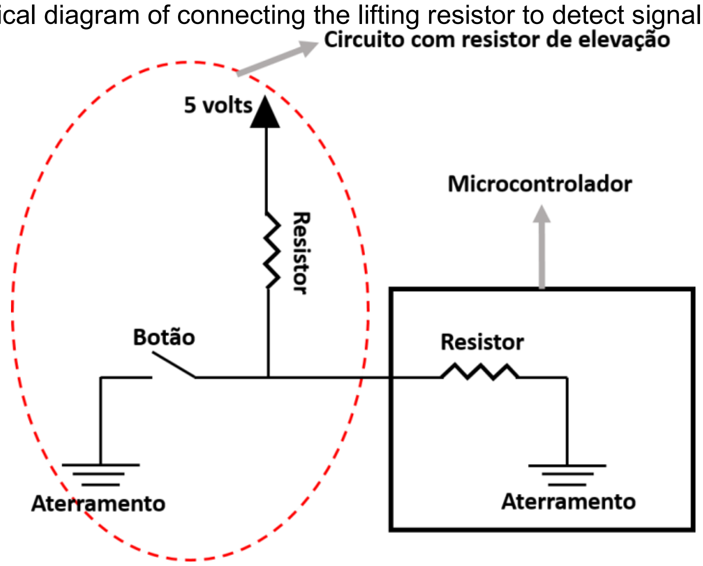
Figure 13: Connection scheme of the TIP120 transistors connected within the PLC of this work



Source: Author himself.

To detect the PLC inputs, a technique was adopted to monitor the signal input in the microcontroller, called elevation resistor or *pullup resistor*. This technique aims to use a resistor, a button and the grounding of the chip itself to detect the arrival or not of electrical signals at the digital ports of the microcontroller. Figure 14 shows an electrical schematic of how to use this lifting resistor:

Figure 14: Electrical diagram of connecting the lifting resistor to detect signal at the PLC inputs.

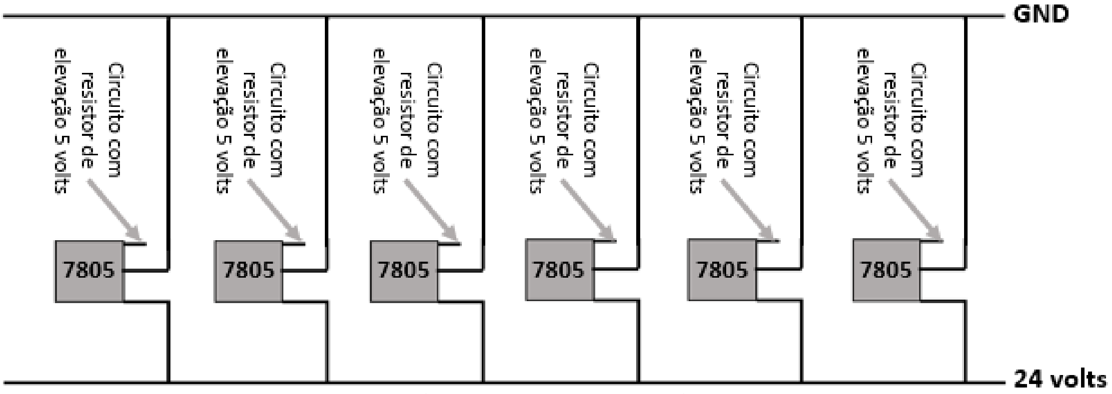


Source: Author himself.

As can be seen in Figure 14, when the button is pressed, the microcontroller reads 0 Volts. While the button is not pressed, the lifting resistor together with a resistor that already exists inside the microcontroller, together form a voltage divider, which allows the reading of a voltage value above 0 Volts by the chip's digital pin. Thus, the microcontroller is able to detect when a button is pressed or not. It should be noted that this button represents any digital sensor, used on the pneumatics bench of the Mechanical Engineering laboratory, which can be activated.

Although the lift resistor circuit was assembled in the same way as shown in the schematic in Figure 14. To reach the 5 Volts input of the circuit, first the 24 Volt voltage had to be regulated. The role of voltage regulation fell to the 7805 voltage regulator. In all, six of these regulators were used to detect six different inputs of the PLC. The circuit assembled for voltage regulation was schematized as shown in Figure 15:

Figure 15: Representation of the connection scheme of the 7805 voltage regulators within the PLC of this work.

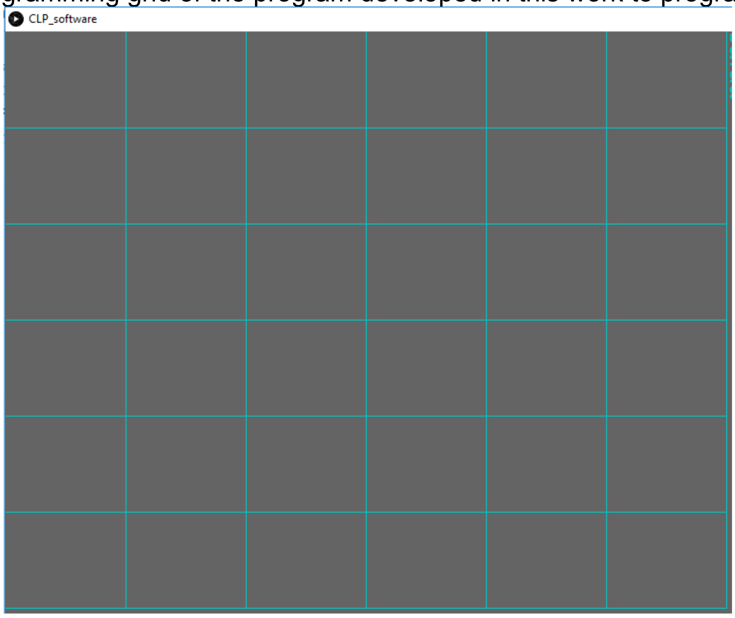


Source: Author himself.

Creation of the *software* to program the PLC from a Ladder language

To create a *software* with a graphical interface for programming in Ladder, the *Processing software* was used. In it, the logic for the presentation and operation of the entire program was programmed in C and C++ language. In the interface of the created program there is a grid with several squares on the left. It is in these squares that the user must assemble the programming logic in Ladder.

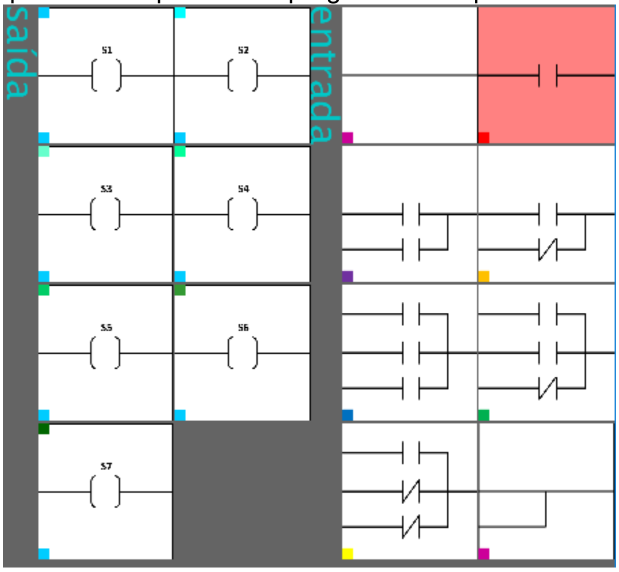
Figure 16: Programming grid of the program developed in this work to program the PLC



Source: Author himself.

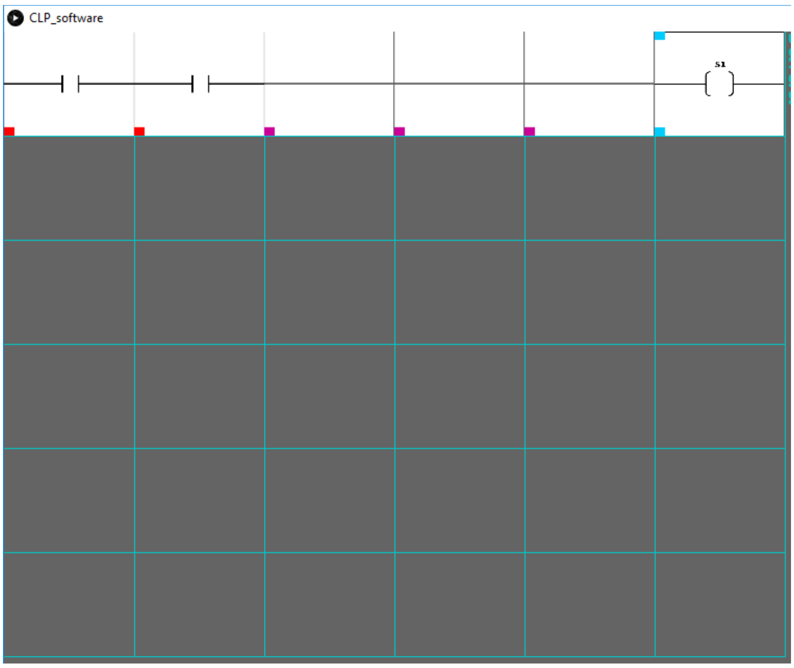
On the right side of the interface, as can be seen in Figure 17, are the symbols for the inputs and outputs. These symbols can be selected on the right and positioned on the left grid, as shown in Figure 18. The selection of inputs and outputs is confirmed by a red coloring that is over the item that was clicked.

Figure 17: Symbols of inputs and outputs of the program developed in this work to program the PLC



Source: Author himself.

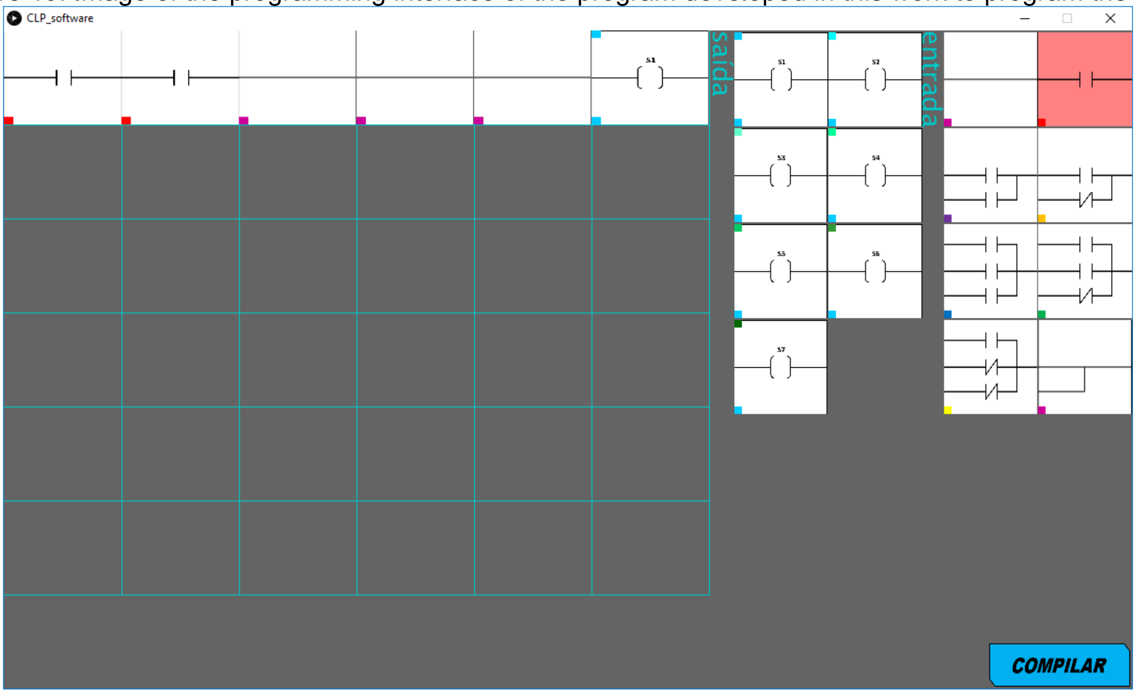
Figure 18: Image of the programming grid of the software developed in this work with a LADDER programming line assembled.



Source: Author himself.

After the symbols are selected and positioned to assemble the Ladder programming, click on the "compile" button to store the programming made in a file called "clp.txt". Figure 19 shows the complete programming interface with the "compile" button.

Figure 19: Image of the programming interface of the program developed in this work to program the PLC



Source: Author himself.

Creation of the *software* for the microcontroller to interpret the information contained in the "clp.txt" file

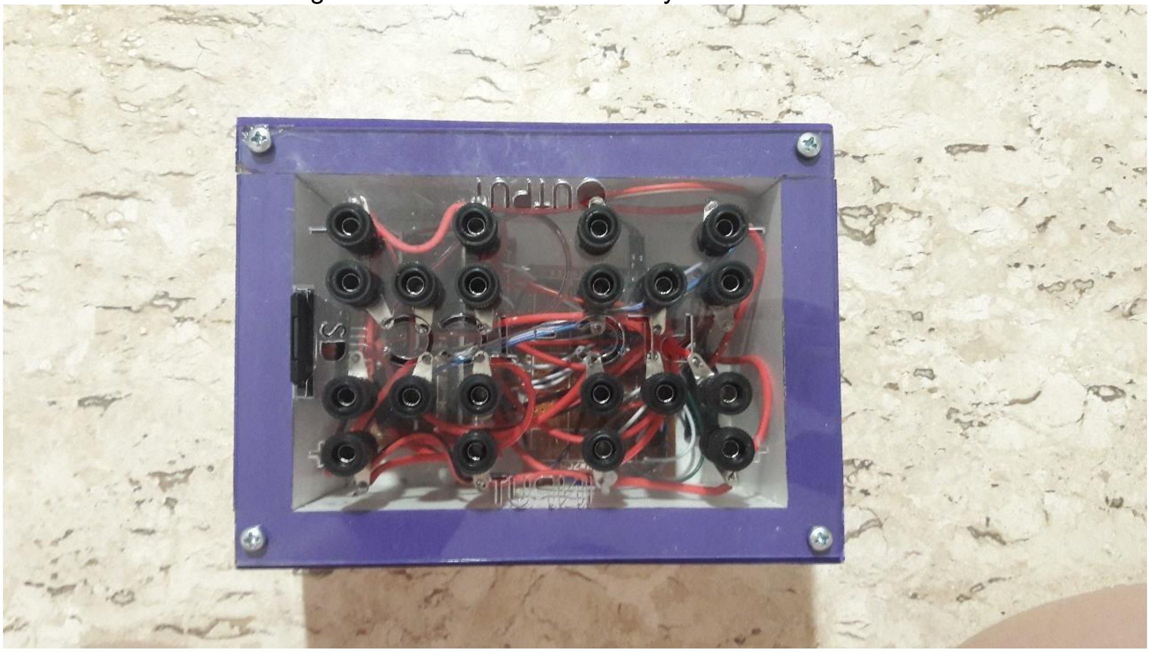
The software that was compiled to be stored inside the microcontroller, was created with the help of the Arduino program. In it, a program in C language was developed to interpret the information that was previously generated in the graphical interface and stored within the "clp.txt" file. This file will then be placed inside a memory card to be interpreted by the microcontroller with the aid of the SD card module.

4Construction of a box to store the electrical circuit

To store and protect the electrical circuit, a white MDF box was created with the following dimensions: 160 mm high by 120 mm wide and 70 mm deep. These dimensions are compatible with the place where the panels of the hydraulic and pneumatic system bench of the UFVJM mechanics laboratory are located.

A design was also created for the front panel of the PLC, delimiting the places to position the terminals and the SD card module. This design was sent to a printer who laser cut it on a 2 mm thick acrylic plate. As shown in Figure 20.

Figure 20: PLC mounted inside your mdf box.



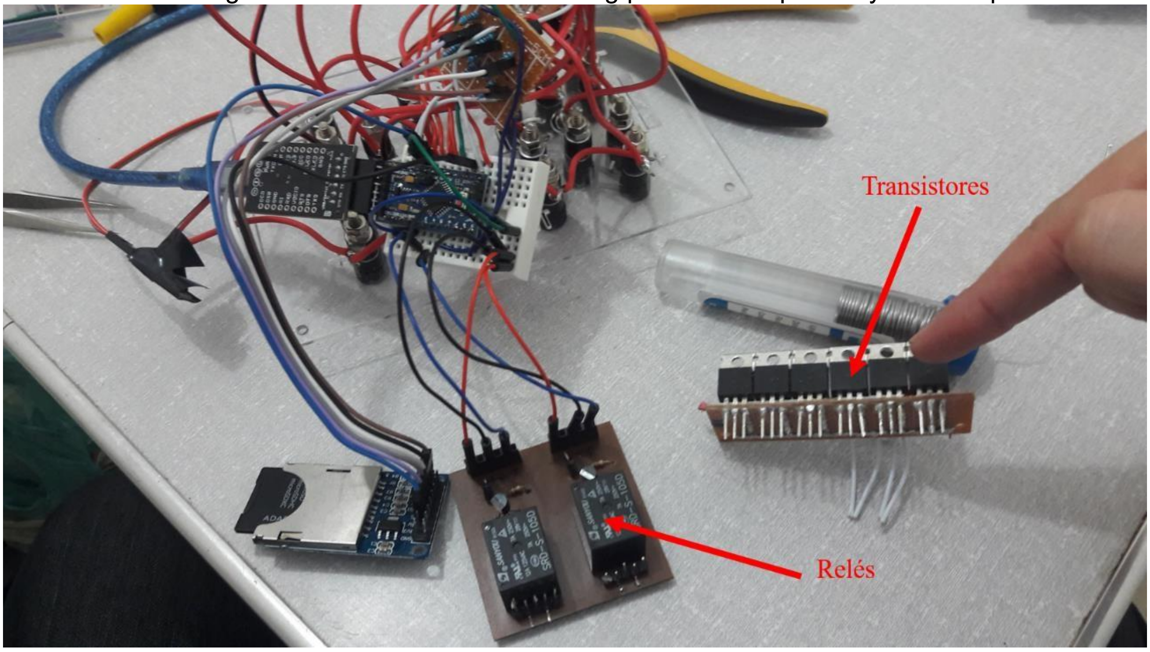
Source: Author himself.

RESULTS AND DISCUSSION

All tests were carried out on the pneumatics bench of the mechanical engineering laboratory at UFVJM. It was verified in the first test that the PLC did not perform the functions contained in the "clp.txt" file. The PLC was opened to analyze if it was a circuit failure, but it was not. It was discovered that the information contained inside the memory card was not read due to a problem with the "SD.h" library of the Arduino software. The problem was solved when the code responsible for the initial configuration of the SD card module was all shifted to the end of the "void setup()" function.

In the second test, it was found that the pneumatic valves did not switch when the PLC sent a command. It was found through the measurement of the voltage with a multimeter that it was reaching only 4.5 volts at the valve. This occurred due to inconvenient contacts of the tin trails that suffered a corrosive process. To prevent the corrosive problem from continuing to affect the circuit with transistors, these were replaced by six relay modules.

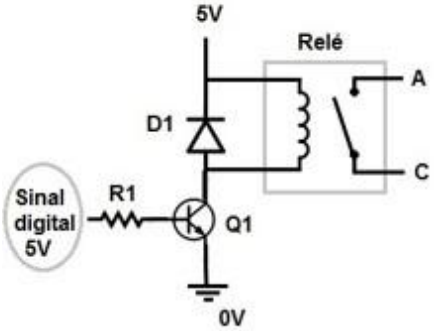
Figure 21: TIP120 transistors being pulled out to put relays in their places.



Source: Author himself.

Eventually, to activate a relay by the microcontroller, it was necessary to assemble a safety circuit, such as the one shown in Figure 22, containing a BC337 transistor, a 1N4007 diode and the relay to be used.

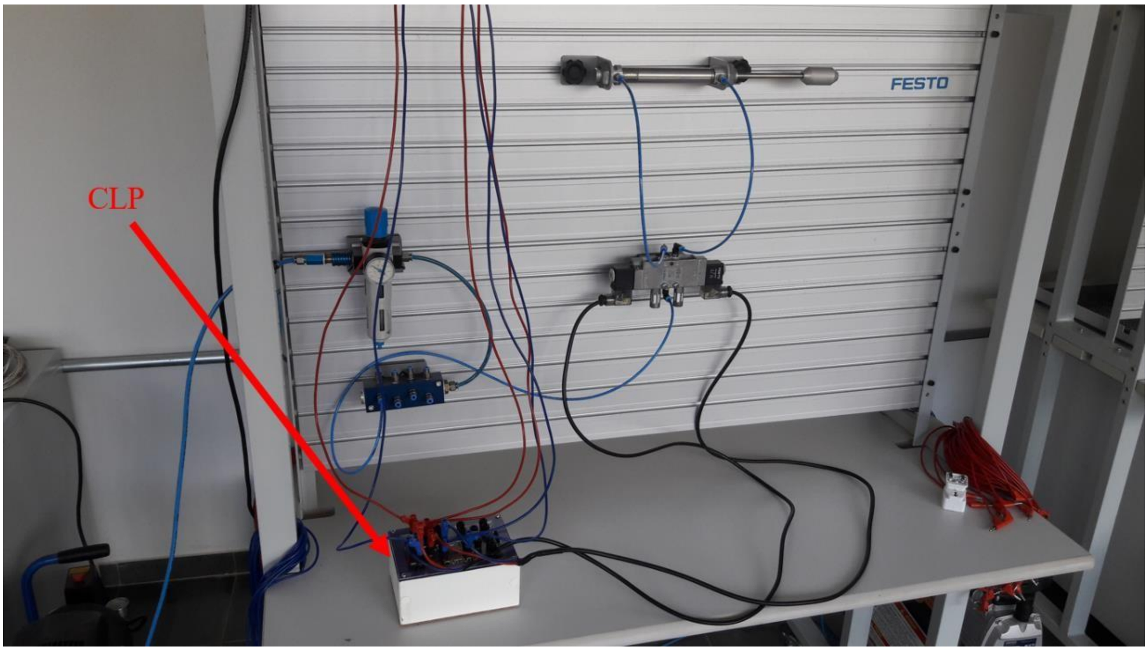
Figure 22: Electrical schematic for triggering a relay from a 5-volt signal coming from the microcontroller.



Source: ENGENHEIRANDO, 2018. Adapted.

Finally, the third test was carried out, using relays to drive the PLC outputs instead of transistors. In this way, when the PLC sent a command to any of its previously programmed outputs, the PLC relay activated the pneumatic valve relay, causing it to switch.

Figure 23: Image of the PLC in operation on the pneumatics bench of the mechanics laboratory of UFVJM.



Source: Author himself.

CONCLUSION

It was concluded that the PLC produced worked in the third test, since it was able to activate the pneumatic valves. The fact of activating the valves means that the microcontroller was able to interpret what was written inside the SD card and then perform functions sending signals to activate the relay modules.

The implication of this is that a low-cost programmable logic controller has been created, which can actually be programmed via the Ladder language to perform electropneumatic valve activation functions. It is low cost because it was spent approximately 150 reais to build all the equipment.

However, much can still be done to improve its functioning. Perhaps these optimizations can be carried out by other students in some scientific initiation project of mechanical engineering at UFVJM.

As far as the software for programming in Ladder is concerned, you can add several new input symbols to diversify the programming combinations to be built. It can also be added *displays* to show and interpret signals coming from analog sensors and not just digital sensors such as push buttons or limit switches, used in conjunction with actuators.

In the microcontroller software, all possibilities for combinations of detected digital signals or combinations of activated sensors must be included in its database. Only then can it be fully trusted to interpret the information that will be present on the memory card.

In the PLC, more inputs can be added during its optimization, which will detect not only digital signals but also analog signals.

Therefore, the realization of this project is very important for the advancement in the frontier of cost reduction for the development of logic controllers for machines. In addition, it will allow more students to come into contact with this type of equipment during their academic training, since the equipment that previously existed in the Mechanical Engineering laboratory did not work properly.

REFERENCES

1. Amazon. (n.d.). *Versatile I 7805 m 7805 5v voltage regulator*. https://www.amazon.in/Versatile-7805-5v-voltage-regulator/dp/B00KHK1C6Y

2. Andrade, C. (n.d.). *O que é CLP: Tudo sobre programação Ladder*. Sala da Elétrica. https://www.saladaeletrica.com.br/o-que-e-clp/

3. Arduino. (n.d.). *Arduino products*. https://www.arduino.cc/en/Main/Products

4. ArduinoLandia. (n.d.). *Placa perfurada fenolite – PCB ilhada*. https://www.arduinolandia.com.br/placa-perfurada

5. Aureliano, A. (n.d.). *Microcontroladores*. Fiozera. https://fiozera.com.br/microcontroladores-914a59cbf7de

6. AutocoreRobotics. (n.d.). *555 – IC timer*. https://www.autocorerobotica.com.br/produto/ne555.html

7. Baudaeletrônica. (n.d.). *Transistor NPN TIP120*. http://www.baudaeletronica.com.br/transistor-npn-tip120.html

8. Embedded. (n.d.). *Conheça o circuito integrado 555*. Embarcados. https://www.embarcados.com.br/circuito-integrado-555/

9. Engineering. (n.d.). *Acionando 110/220V com sinais 5V*. Engenheirando. http://engenheirando.com/eletronica/reles/

10. EnsinoElétrica. (2017, March). *Programação em Ladder*. https://ensinandoeletrica.blogspot.com/2017/03/programacao-em-ladder.html

11. Home.RoboticLab.eu. (n.d.). *Microcontroladores e robótica*. http://home.roboticlab.eu/pt/microcontrollers

12. Garage Lab. (n.d.). *PCB ilhada e solda*. Lab de Garagem. http://labdegaragem.com/forum/topics/pcb-ilhada-e-solda

13. MercadoLivre. (n.d.). *Kit 3x TIP120 NPN (TO-220)*. https://produto.mercadolivre.com.br/MLB-693105077-promoco-kit-3x-tip120-transistor-tip-120-npn-to-220-_JM

14. MercadoLivre. (n.d.). *CLP – CP-WS12/8DI8DO2AI2AO OEM – USB e RS485 Modbus RTU*. https://produto.mercadolivre.com.br/MLB-692335993-clp-cp-ws138di8do2ai2ao-oem-usb-e-rs485-modbus-rtu-_JM

15. MercadoLivre. (n.d.). *S7 1200 1214 DC/DC/DC CPU*. https://produto.mercadolivre.com.br/MLB-904069241-cpu-s7-1200-1214-dcdcdc-_JM

16. Microchip. (n.d.). *ATmega328*. https://www.microchip.com/wwwproducts/en/ATmega328

17. NewTeck. (n.d.). *O que é circuito integrado*. http://www.newteck-ci.com.br/circuitos-integrados.php

18. Quora. (n.d.). *How do I design a fixed power supply using 7805 IC*. https://www.quora.com/How-do-I-design-a-fixed-power-supply-using-7805-IC

19. Silveira, C. B. (n.d.). *Como funciona a linguagem Ladder*. CITI Systems. https://www.citisystems.com.br/linguagem-ladder/

20. SparkFun. (n.d.). *µA7800 series positive-voltage regulators*. https://www.sparkfun.com/datasheets/Components/LM7805.pdf

21. Thomsen, A. (n.d.). *Como gravar dados no cartão SD com Arduino*. FilipeFlop. https://www.filipeflop.com/blog/cartao-sd-com-arduino/

22. YourRobot-Electronics. (n.d.). *Arduino Pro Mini 5V 16MHz*. http://www.yourobot-electronics.com/product/arduino-pro-mini-5v-16mhz-compat%C3%ADvel