The International Journal of Engineering and Science (IJES)
|| Volume || 10 || Issue || 11 || Series I || Pages || PP 15-20 || 2021 ||
ISSN (e): 2319-1813 ISSN (p): 20-24-1805

Control of a Pumping System Through A PLC And HMI Interface

Yoram Astudillo-Baza[1], Manuel Torres-Sabino[2], Mercedes Lázaro-Gonzaga[3]
[1,2,3] *Instituto Politécnico Nacional, ESIME Zacatenco, Ciudad de México, México*
Corresponding Author: Yoram Astudillo-Baza

ABSTRACT
Development and technological advancement have made it possible to automate tasks or robotize them in order to save costs and time in labor. That is why more and more industries choose to automate their processes and do so through a PLC programmable logic controller, due to its practicality, ease of use and programming.
For this reason, this work presents the design of a control applied to a pumping system through a PLC and with the use of an HMI interface.

KEYWORDS;- *Control, HMI interface, programmable logic controller PLC, pumping system.*

Date of Submission: 03-11-2021 Date of Acceptance: 17-11-2021

I. INTRODUCTION

A PLC or Programmable Logic Controller is an electronic device widely used in industrial automation. With a PLC, the operating logic of machines, plants and industrial processes can be controlled. In them, digital and analog signals are processed and received with which control strategies can be applied. In general, it is possible to find this type of equipment in industrial environments.
There are 2 families of PLCs, the basic version with expandability and the advanced version whose frequency of inputs is higher, as well as the number of inputs and outputs to which it can be extended [1].

The internal structure of a PLC is shown in Fig. 1, it consists mainly of inputs, outputs and memories where the program codes of the tasks to be executed are stored.

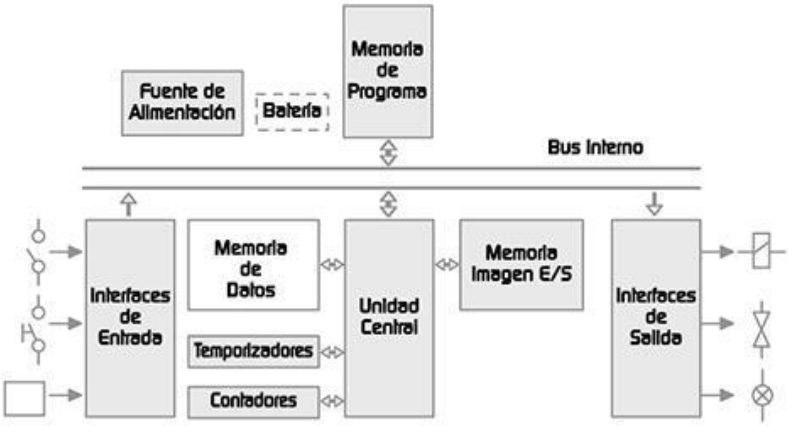


Fig. 1. Architecture of a PLC.

A PLC performs, among others, the following functions: Collect data from input sources through digital and analog sources, make decisions based on pre-programmed criteria, store data in memory, generate time cycles, perform mathematical calculations, act on external devices through analog and digital outputs and communicate with other external systems.

PLCs differ from other automatic controllers, in that they can be programmed to control any type of machine, unlike other controllers that can only control a specific type of machine.

Besides being able to be programmed, they are automatic, that is, they are devices that compare the signals emitted by the controlled machine and make decisions based on the programmed instructions, to keep the operation of said machine stable.

Operation of a PLC

Once started, the processor performs a series of tasks in the following order:

a) When the processor is turned on, it executes a power-on self-check and blocks the outputs. If the check has been successful, the PLC enters the normal operating mode.

b) In the next step, you read the status of the inputs and store them in an area of memory called the input image table.

c) Based on its control program, the PLC updates a memory area called the output image table.

d) Next, the processor updates the status of the outputs by copying the status of the output image table to the output modules. In this way, the status of the output modules of the PLC, relay, triacs, etc. is controlled.

e) Re-execute step b).

Each execution cycle is called a scan cycle, which is normally divided into [2]:

1. Verification of inputs and outputs,
2. Execution of the program.

II. SIEMENS PLC

The Siemens S7 PLC is a PLC capable of performing different automation tasks by controlling different devices or elements necessary for the operation of a process. Various functions can be used to allow more complex programming such as timing instructions, counting, Boolean logic, complex math operations, and control blocks.

In Fig. 2, the main parts of the S7-1200 PLC are shown, which consist of:

1. Power connector.
2. Removable connectors for user wiring.
3. Memory Card slot.
4. Status LEDs for the integrated I / O.
5. PROFINET connector.

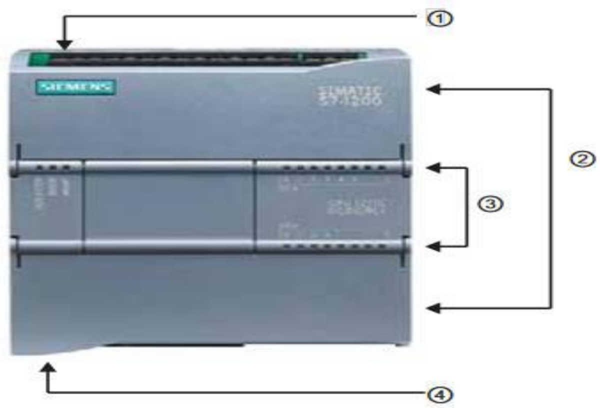


Fig. 2. Parts of the SIEMENS PLC.

The Siemens PLC carries out linear programming, in which the instructions are stored in a block and executed in the order in which they have been stored in the program memory. Upon reaching the end of the program (end of block), the execution of the program begins again from the beginning, this organization is shown in Fig. 3.

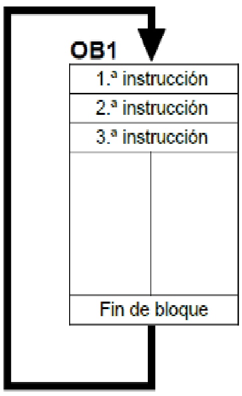


Fig. 3. Block Organization of the SIEMENS PLC.

Within the block organization, there are sub-blocks in each instruction, which in turn contain function blocks and functions with more functions and data blocks. Next, in Fig. 4, shows the blocks handled by the workbench. It is also observed that the type of programming language can be selected, SCL is structured high-level programming; FUP programming refers to programming based on logic blocks and finally KOP programming which is programming through contacts.

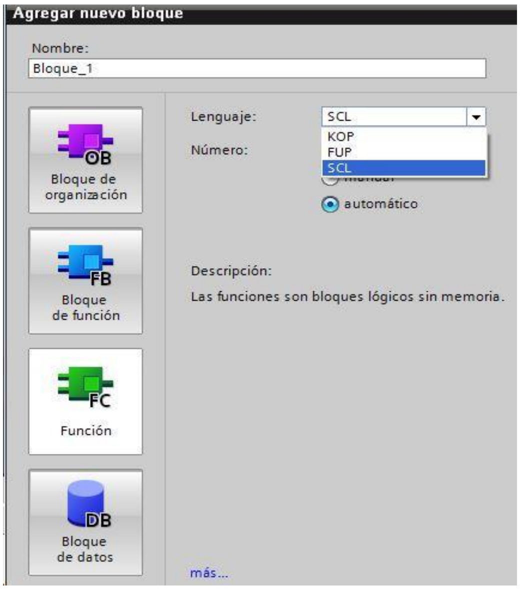


Fig. 4. Types of Blocks of the SIEMENS PLC.

TIA Portal (Totally Integrated Automation Portal)

TIA Portal is the software that is needed to carry out the programming and configuration of the PLC, in the same way; it allows the simulation of how the system would act, as well as the creation of HMI (Human-Machine Interface). The programming in TIA Portal admits the type of data shown in Fig. 5, the first digit indicates the type of record, the second digit the size, and to end the start byte or bit.

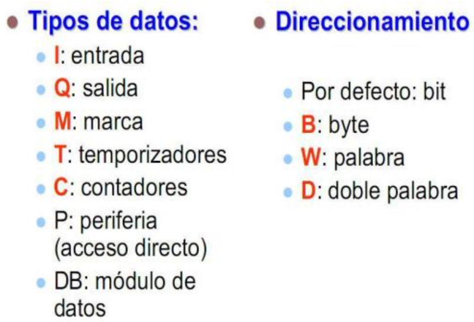


Fig. 5. Data types in the TIA Portal.

Programming Commands

Contacts: There are two types, Normally Open (NO) and Normally Closed (NC). The NO contact closes when the assigned bit is equal to 1. The NC contact closes when the assigned value is equal to 0. It should be noted that Boolean logic is used, where 0 is OFF and 1 is ON.

Coils: The "Coil" output instruction writes a value to an output bit. If the indicated output bit uses the memory identifier Q, the computer turns the output bit on or off in the process image, setting the specified bit to the corresponding current flow state.

Set and Reset: This function is an output activation and deactivation. If S (Set) is activated, the data value of the OUT output address is set to 1. If R (Reset) is activated, the data value of the OUT output address is set to 0.

Flip Flop RS and SR: It is used to build sequential circuits, it has 2 inputs, an R input for reset and an S for set. The difference between RS and SR will be the state of the outputs, that is, if it is RS the states of Set and Reset will be equal to 1 but the output will be equal to 0; if it is SR, the Set and Reset states will be equal to 1 but the output will be equal to 1, that is, activated.

Timers: They are devices that measure the time in which you want to execute an action, there are different types. TP (generates a pulse with a predetermined duration), TON (The on-delay output turns on after a while), TOFF (The output turns off after a while), TONR (The output turns on after time, and the time accumulates over several cycles) and RT (Initializes a timer by erasing the time data from the block).

Comparators: Comparison statements are used to compare two values of the same data type. If the contact comparison is true, the contact is activated.

Convert: The CONVERT instruction that is to convert the value, as its name implies, allows you to convert an element from one data type to another.

Counters: They are used to count program actions. CTU Ascending, increments by 1 when the Boolean value changes from 0 to 1. CTD Descending, decrements by 1 when the Boolean value changes from 0 to 1.

Table 1 shows, in summary, the basic programming commands, as well as the block that represents each element and the type of data it handles.

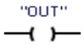
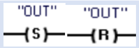
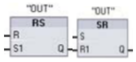
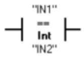
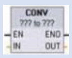
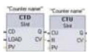
Elemento	Símbolo	Parámetro	Tipo de Datos
Contacto		IN	Bool
Bobina		OUT	Bool
Set y Reset		OUT	Bool
Flip Flop		R, R1, S, S1, OUT, Q	Bool
Temporizador		IN,R,PT,Q. ET	Bool Time
Comparador		Lógica comparativa	---------
Convertidor		IN OUT	Valor de IN Valor de In, convertido al nuevo dato.
Contadores		CU, CD, R, Q. PV, CV.	Bool. Int.

Table 1. TIA Portal Programming Commands.

HMI (Human-Machine Interface)

An HMI is the interface that exists between a machine and the operator; Human Machine Interface. It consists of a digital screen, in which the information from the system to be automated is displayed, and on the digital screen, system actions can be carried out remotely. The remote mode implies being able to operate the system even being far from where it is located, unlike the local mode that activates the system only if the operator activates the starting elements.

The HMIs also include programming, which associates the HMI variables with the input and output variables of the system to be automated. An example of an HMI is shown in Fig. 6, where the operator can remotely execute system actions, such as starting or stopping it, as well as viewing the status of the tanks that fill the bottles.

Fig. 6. HMI example.

Finally, the HMI aims to offer the end operator a friendly and easy-to-use interface that allows them to interact and control the system.

III. TEST PUMPING SYSTEM

System description

There is a pumping system, which is made up of 3 pumps; the operator can select two modes of operation through a switch, which consist of Automatic mode and Manual mode.

In automatic mode, the operating cycle lasts approximately 60 seconds. In each cycle, each of the 3 pumps has to operate alternately for 20 seconds. The duty cycle limit is 4000 cycles.

If the automatic mode is activated, the operation of pump 1 starts, consecutively until reaching pump 3, at the end of this operation, the counter decrements the total number of cycles by 1 to comply with the restriction of the work cycle limit. If for any reason the operator fails and stops in automatic mode, the cycle is interrupted and a restart is carried out from the cycle not completed. Similarly, this operating mode has a reset button, to eliminate the number of cycles to be carried out if the operator requires it.

The manual operation mode is considered to give maintenance to the pumps, in which each pump motor can be started and stopped independently to visualize its individual behavior. The disadvantage of this mode of operation is that each pump is operated independently.

Force and Control connection diagram

The connection diagram, as its name implies, represents the interconnections between the elements of the system. In the Fig. 7, shows the connection diagram, on the left side are the connections of the pump motors to the power supply and on the right side are the connections for the PLC.

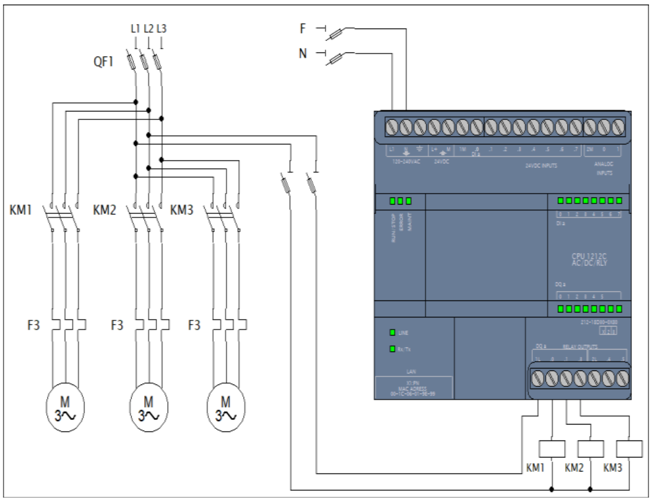


Fig. 7. System connection diagram.

Program Code in TIA Portal

The variables of the program code are shown in Table 2, as well as the type of data handled by each variable. The variable BAG directs the General Start button, BPG to the General Stop button. Each pump has a start and stop button for manual mode named BA1 and BP1 respectively, until reaching pump 3. Similarly, there is a variable that stores the number of cycles, for the restriction of 4000 cycles. Time variables are available so that after 20 seconds the next pump starts to start consecutively. In the same way, the Screen variables are declared for the development of the HMI.

Programming structure

The program has an order and a programming structure; it is mainly composed of a Main, Manual Mode and Automatic Mode.

It is the main function of the program (Main), in this section the other functions are called having the condition of selection manual or automatic mode through the HMI. The Fig. 8 shows the main programming.

Manual mode, in this function, the independent start-up programming of each of the pumps is carried out, respecting the operating condition of one pump at a time to carry out its maintenance, for this reason each pump has its function block, the programming code is shown in Fig. 9.

Automatic mode is the function that stores the programming of the sequential start and stop of the three pumps, respecting the time of 20 seconds of operation. In the same way, the counter is programmed that decreases depending on the cycles carried out, each pump has its programming block, in the Fig. 10 shows the programming in automatic mode.

Variable	Tipo	Dirección
BAG	Bool	%M0.0
BPG	Bool	%M0.1
BA1	Bool	%M0.2
BP1	Bool	%M0.3
BA2	Bool	%M0.4
BP2	Bool	%M0.5
BA3	Bool	%M0.6
BP3	Bool	%M0.7
Numero de ciclos	Int	%MW4
Modo de operacion	Bool	%M1.4
M1	Bool	%Q0.0
M2	Bool	%Q0.1
M3	Bool	%Q0.2
Reset	Bool	%M1.3
Tiempo segundos	Time	%MD1
Tiempo segundos(1)	Time	%MD2
Tiempo segundos(2)	Time	%MD3
Inicio	Bool	%M7.0
Fin	Bool	%M7.1
Programa	Bool	%M7.2
Pantalla1	DInt	%MD8
Aux1	DInt	%MD12
Aux2	DInt	%MD16
Pantalla2	DInt	%MD20
Aux3	DInt	%MD24
Pantalla3	DInt	%MD28

Table 2 Program variables.

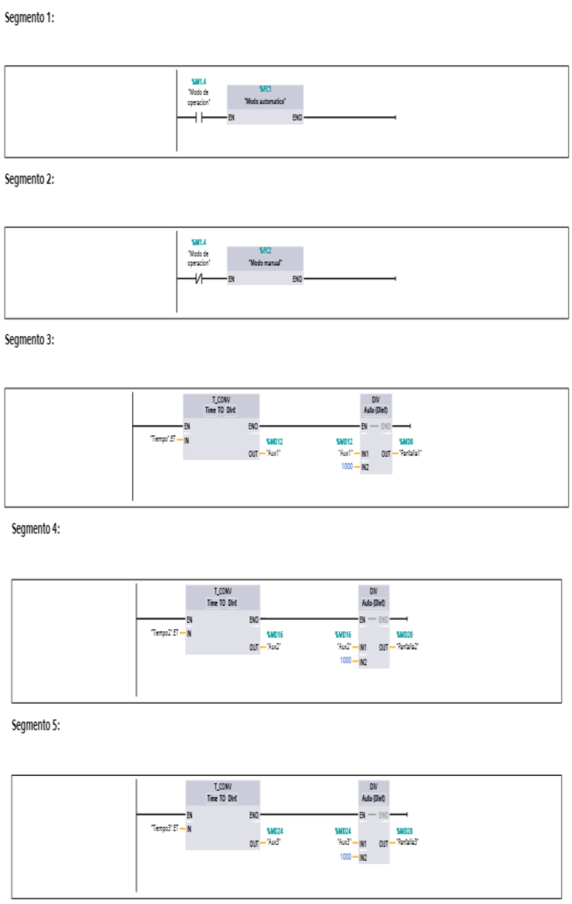


Fig. 8. Main structure.

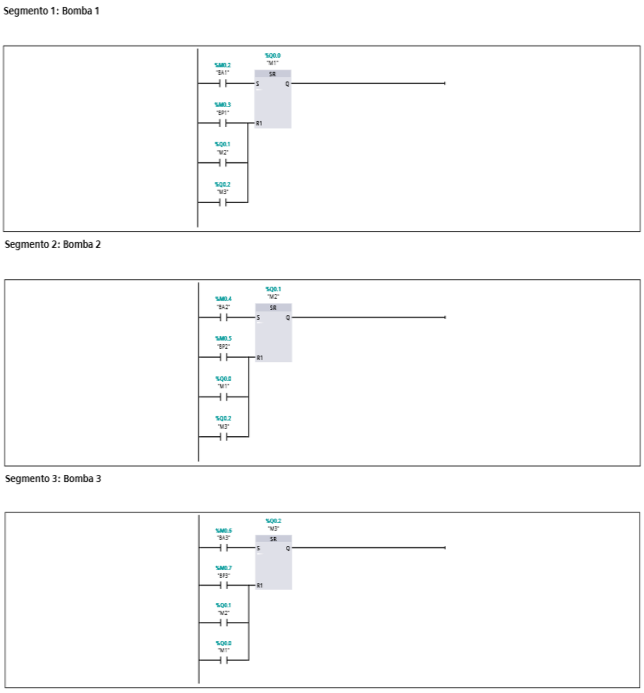


Fig. 9. Structure of manual mode.

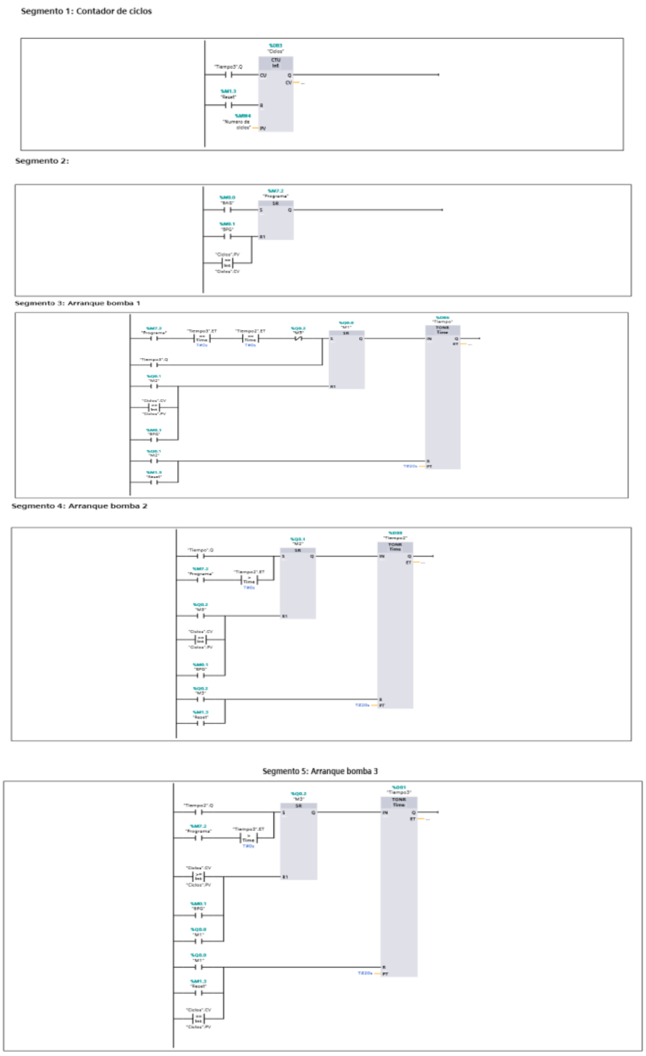


Fig. 10. Structure of automatic mode.

HMI that drives the system

For the realization of the HMI, the outputs and inputs of the pumping system are addressed to the design of the created screen. The interface is shown in Fig. 11, in which the operator by means of the selector button can choose the automatic or manual mode, and the representation of the status of each of the pump motors. It has a general start and stop button for the system, as well as a Reset button that restarts the system. When the pump indicator is green, the pump is running.

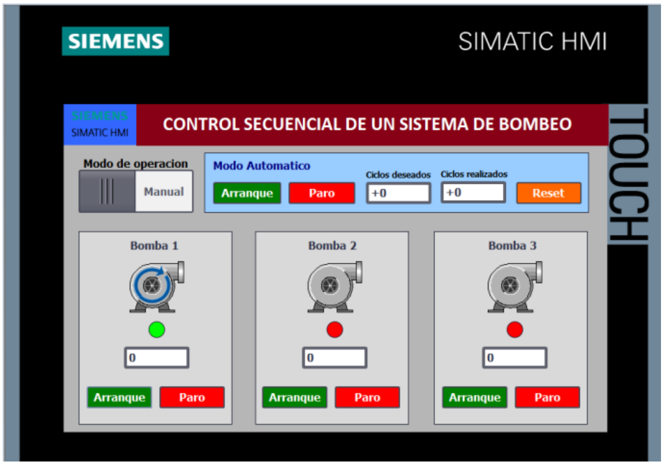


Fig. 11. Drive HMI.

For the automatic mode of operation, there is a panel, as shown in Fig. 12, in which the operator can key in the number of pumping cycles he wishes.

Finally, Fig. 13 shows the automatic mode of operation, for this case 3 cycles were selected, it is shown that pump 2 is activated and runs in second 6, until it reaches 20 seconds, it will be deactivated and the program will proceed to activate pump 3. The cycle will repeat until the counter reaches the desired number of cycles, which are 3.

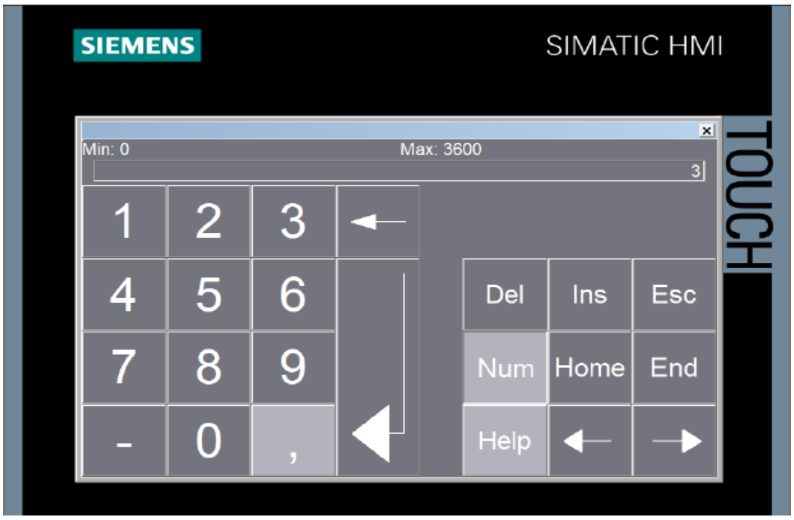


Fig. 12. Selection of the number of cycles.

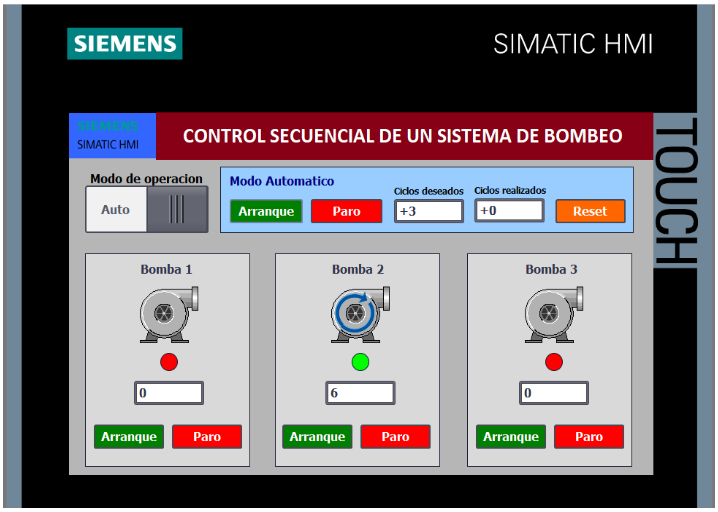


Fig. 13. HMI automatically.

IV. CONCLUSION

It can be concluded that a PLC is a powerful tool that allows the control of machines and processes in industry. Currently, PLCs have advanced in terms of technology, not only through software, but also because the design of HMI interfaces is allowed, where the process control can be carried out and the status of the system to be controlled can be known in a way remote.

It should also be considered that the programming environment used by the PLC is not complex because it is structured programming, which allows the detection of program errors easily.

REFERENCE

[1]. Logicbus. Marzo 2012. Available:http://www.logicbus.com.mx/plc.php
[2]. Prieto, P. Octubre de 2007. Available: Observatorio tecnológico. Recursostic.educacion.es/observatorio/web/gl/component/content/article/502-monografico-lenguajes-de-programacion?start=2.
[3]. https://revistadigital.inesem.es/gestion-integrada/que-es-un-sistema-hmi/
[4]. SIEMENS. (s.f.). SIMATIC S7. Available: https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S71200-MANUAL%20DEL%20SISTEMA.PDF
[5]. ¿CÓMO FUNCIONA UN PLC? Qué es, para qué sirve y características. Available: comofunciona.co.com

Yoram Astudillo-Baza, et. al. "Control of a Pumping System Through A PLC And HMI Interface." *The International Journal of Engineering and Science (IJES)*, 10(11), (2021): pp. 12-20.