SEQUENTIAL CONTROL OF A SIMULATED PROCESS OF 5 HYDRAULIC CYLINDERS THROUGH A PLC.

Authors: David Steven Galvis Arévalo 1802584, Johan Andres Sebastian Gómez Espinosa 1802322, Eric Xavier Buitrago Forero 1802410.

Abstract: This report presents the development of a sequential process implemented in a PLC in which phase curves made by actuators are recreated (cylinders) that contain real physical characteristics and that are simulated in the Automation Studio software. Through the use of timers and counters, the profiles associated with each of the cylinders are fulfilled, the timers allow the time between segment by segment of each profile to be fulfilled, the position changes the same and the counters allow these cycles to be carried out. repeat the number of times indicated.

Finally, when this procedure is carried out, the phase curves are obtained in graphs for each of the actuators at the time that the user wishes, working simultaneously as an industrial application process in which different actuators are synchronized and this thanks to the logic controller programmable.

Abstract: *In this paper is presented the development of a sequential process programmed in PLC which recreates given phase curves made by actuators (cylinders) and with physical features simulated in Automation Studio. Through use of timers and counters the cylinder's profiles can be done. The timers allow the user to have the right time between segments of each profile and changes of position of itself, and the counters allow the process to repeat the cycles that the user indicates.*

Finally when the process is done there are presented the phase curves graphs for the 5 actuators. It works simultaneously as aN industrial process where actuators are synchronized and this by using the programmable logic controller.

Introduction:

Sequential processes propose states (operations to be carried out to transform the raw material into a product) and transitions (information related to sensors or logical elements such as timers or counters) in an ordered sequence that identifies the dynamic evolution of a controlled process. [1]

Counters are used when it is necessary to count the number of times a contact is activated. Among the functions of the PLC are counting and have the appropriate circuits for this purpose. There are up and down counters, it means that it counts up or down and when they reach the desired value, the counter contact changes state.

Timers allow the connection or disconnection of a contact after a certain time, when you want to achieve delay times greater than those that can be obtained with a single timer, several timers are connected to each other, which is known as a cascade connection . [2]

A programmable logic controller is a digital electronic device that uses programmable memory to store instructions and perform logic, sequencing, timing, counting, and arithmetic functions for machinery and process control.[3]
]

Objectives

- General

*Design a software solution for a sequential process using the PLC.

- Specific

*Identify the necessary actuators according to the profile that each one must carry out.

*Use a program in the PLC that allows to develop a sequential process using timers and counters.

*Develop an environment in Automation Studio that allows simulating the previously established process.

Development

Having the profiles and the load for each actuator, one must then look in commercial catalogs for cylinders that meet the specifications such as speed, load, thrust force and recoil.

Materials:

- PLC Siemens S7-1500
- Realtek RTL8188EU Wireless LAN 802.11n USB 2.0 Network Adapter.
- TIA Portal V13
- OPC IBH
- Automation Studio 6.1

Procedure

The phase curves to be recreated for the corresponding work group

GROUP 2

Grupo	Secuencia	Perfiles de Carga				
		A	B	C	D	E
2	D(XE(XB(XA(XC))))	1	5	2	3	4

are identified. The previously taught sequence D(xE(xB(xD(xC)))) is carried out where < x > is the number of repetitions requested by the user.

Where the phase curves of the profiles to be made are in their respective order from figure 1 to 5.

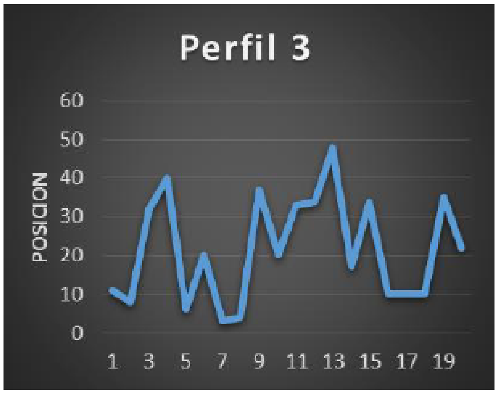

Figure 1: Profile 3, assigned to stage D

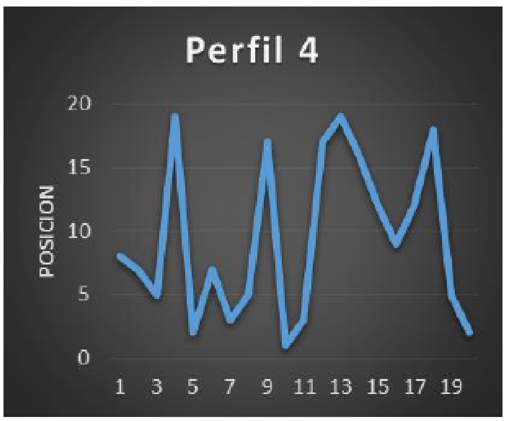

Figure 2: Profile 4, assigned to stage E

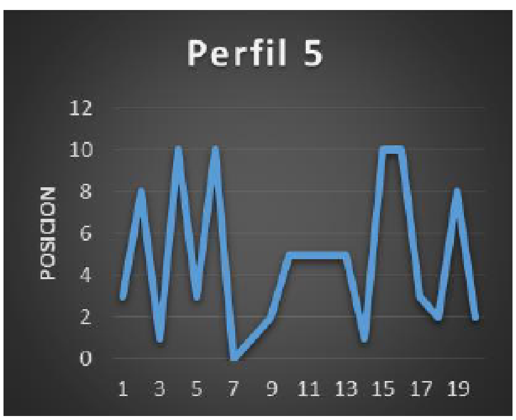

Figure 3: Profile 5 , assigned to stage B

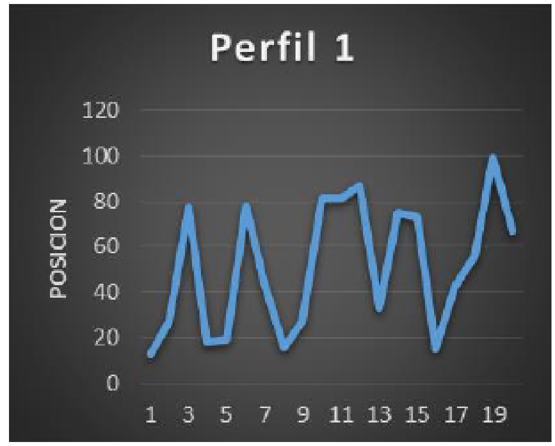

Figure 4: Profile 1, assigned to stage A

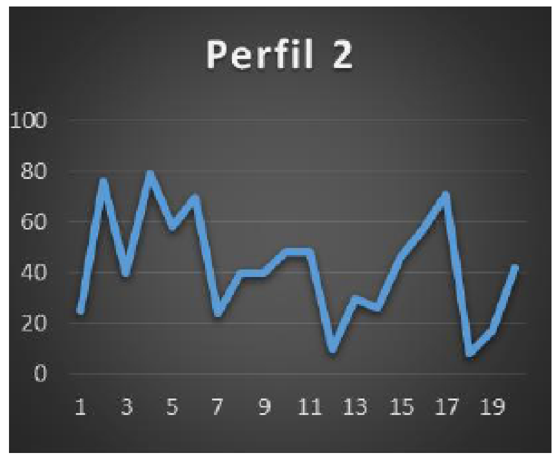

Figure 5: Profile 2, assigned to stage C

The load for each of the cylinders must be taken into account, based on table x.

A	B	C	D	E
14.4Tn	18.6Tn	8Tn	3.45Tn	12Tn

Table 1: Load for the actuators

The cylinders capable of developing the process and their most relevant characteristics are then shown.

Figure 6: LARZEP double-acting hydraulic cylinders

For all the cylinders, first the main characteristic that was identified was the load that it is going to support, the thrust and recoil forces and the maximum and minimum speeds that it can reach.[4]

Cil	Capacidad (Tn)	Carrera (mm)	Modelo	Capacidad (kN) Push - Pull	Volumen (cc)	Peso (kg)
D	20	160	D02016	214 - 53	499	11,7
E	100	160	D10016	911 - 189	2123	51,8
B	150	50	D15005	1380 - 370	1005	66
A	150	150	D15015	1380 - 370	3015	90
C	350	150	D35015	3368 - 1014	7359	247

Table 2: Reference and characteristics of the actuators

When calculating the necessary force, it was necessary to calculate the maximum and minimum speeds, for this the following table was used

Velocidad del émbolo (mm/s)	Factor de carga máximo
8 a 100	70%
101 a 200	30%
201 a 300	10%

Table 3: Load factor as a function of speed.

According to the established x, the load factor was found and the cylinder containing the necessary characteristics was found.

TIA Portal program

In this part of the main we initialize all the blocks, which we need to be fulfilled cyclically since the main is a repetitive cycle.

*Main

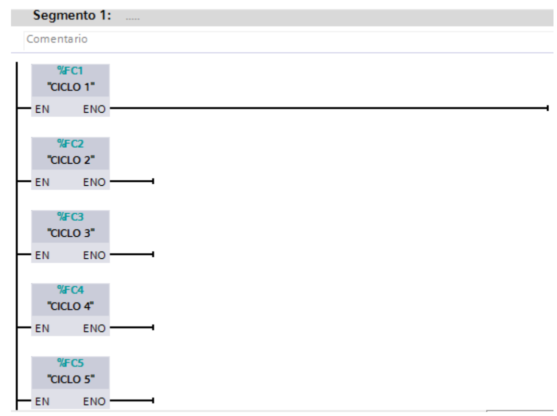

Figure 7: All cycles are initialized at the same time.

In segment 1 of the main we initialize all the parts of the cycle, being cycle 1 the envelope and cycle 5 the one that repeats the most.

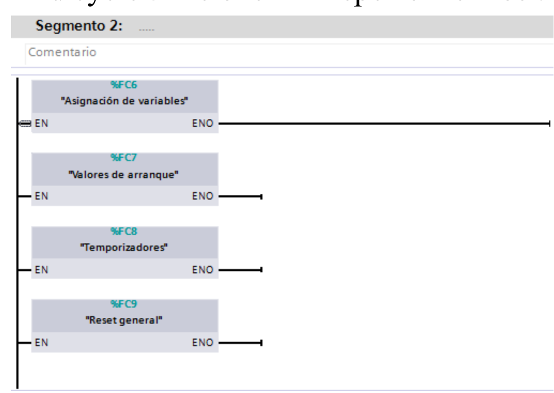

Figure 8: Timer, counter and variable assignment blocks are initialized.

In segment 2 of the main we initialize the functions such as variable assignment, start values, timers, General Reset.

How does each block work?

Assignment of variables: the block of variables of the plc is equalized with the variables of the data block with the function that if there is any change in the program, a complete change in the program does not have to be made, but only with the renaming of a variable.

Start variables: it is a block in scl language and we use it to give the start values to the throttles and the times of each cycle.

Timers: we put all the counters for each cycle since it changes its time therefore they tend to make the times different.

General reset: when it completes its program, it resets counters for each part of the cycle and returns to 0 to restart its complete cycle.

***Block: Variable assignment function:**

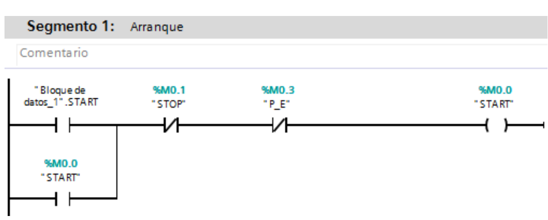

Figure 9: Start of the program.

In segment 1 of the variable assignment function block, it is the start of the system where we match the value of the automation to the start memory of the plc, it has a retention in its operation and the only thing that makes it stop is that it stops.

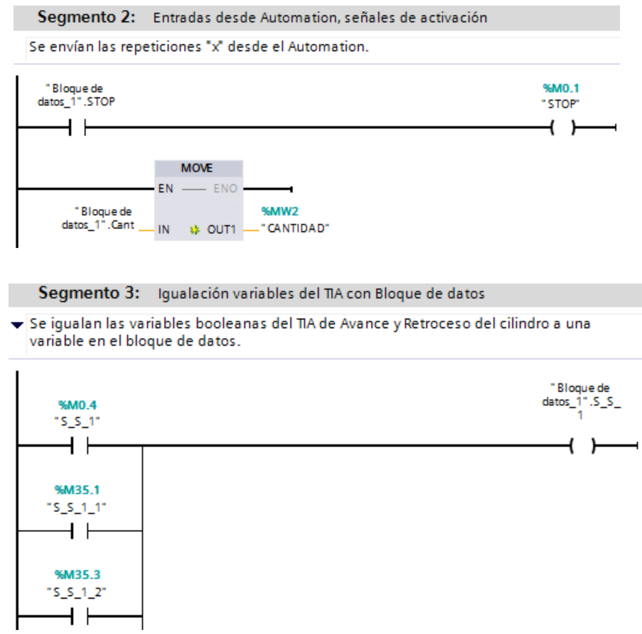

Figure 10: Matching system variables with data block variables.

In segment 2 we see the equalization of the automation stop with the plc stop and we also see the equalization of the number of repetitions per cycle that we bring to a plc memory.

In these first segments we bring automation values to the plc which are the ones we are interested by bringing the following segments we match each direction of each cycle of each stem and the value of each throttling.

In segment 3 we realize that the direction that we send to the automation of each part of each cycle of the profile depends on the unit of time that each one handles, so we handle several auxiliaries that send us the signal to the data block since we cannot call the same memory in various parts of a function to send the same sense several times.

***Block: General reset.**

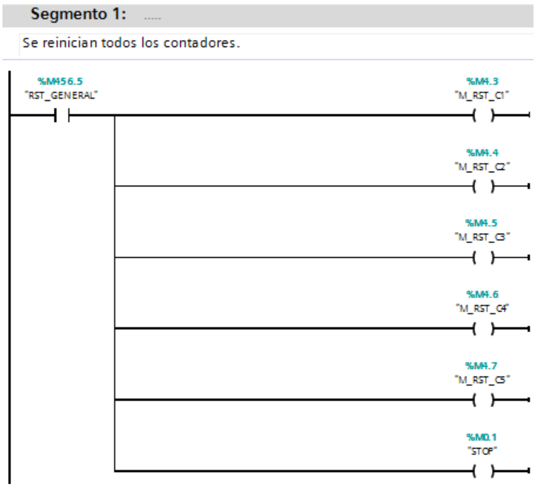

Figure 11: Reset all counters.

In the general reset we give that when the envelope completes its cycle, it means that it has already completed the entire process, which activates a memory called “RST_GENERAL” that in this function resets the memories of the counters of each cycle and turns on the stop that stops the whole process waiting for a new start signal to be given.

***Block: Timers.**

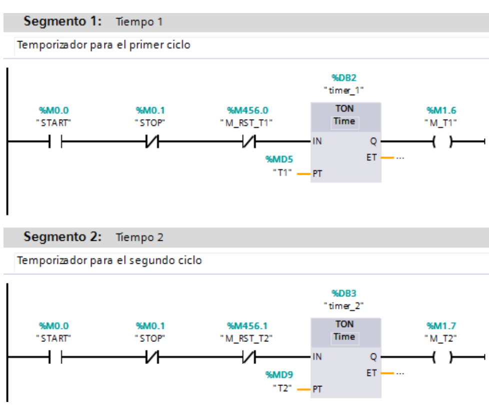

Figure 12: Timers for cycles 1 and 2

To maintain the direction signal and the throttling value we have a block where we have a timer that counts each part of each cycle

The same procedure is carried out for the remaining 3 cycles and thus declares the times for the 5 cycles.

***Block: Start values.**

```
//Valores de tiempos
//tiempo perfil mas bajo hasta la embolvente
// 15 segundos es el cambio de posición del vástago para el primer cilindro
"T1" := 15000;
"T2" := "T1" / "CANTIDAD";
"T3" := "T2" / "CANTIDAD";
"T4" := "T3" / "CANTIDAD";
"T5" := "T4" / "CANTIDAD";
// Valores de apertura de la valvula de estrangulación ciclo 1 envolvente
"EST_V_S_1_1" := (0.0067);
"EST_V_S_1_2" := (0.0035);
"EST_V_S_1_3" := (0.0213966666666667);
"EST_V_S_1_4" := (0.00926333333333333);
"EST_V_S_1_5" := (0.0227);
```

Figure 13: Throttling values.

Figure x shows the starting values with which the program starts, it contains variables which are the throttling percentage of the valves for each of the stages of each cycle, these data are sent through a block < MOVE> equaling the data block variable declared in Automation.

***block .**

1	Static		
2	START	Bool	false
3	STOP	Bool	false
4	Cant	Int	1
5	EST_S_1	Real	0.0
6	EST_S_2	Real	0.0
7	EST_S_3	Real	0.0
8	EST_S_4	Real	0.0
9	EST_S_5	Real	0.0
10	S_S_1	Bool	false
11	S_B_1	Bool	false
12	S_S_2	Bool	false
13	S_B_2	Bool	false
14	S_S_3	Bool	false
15	S_B_3	Bool	false
16	S_S_4	Bool	false
17	S_B_4	Bool	false
18	S_S_5	Bool	false
19	S_B_5	Bool	false

Figure 14: Variables of the data block

These variables are also declared in Automation using the OPC Client, in order to be able to simulate the process, ***<Start>*** is the start of the cycles, the cylinders are activated by performing the corresponding profiles. The **<Stop>** has the function of pausing the counters and therefore the entire process. ***<Cant>*** It is responsible for sending from the Automation the number of repetitions that the user wants, it is the variables x of the sequence. The variables declared ***<EST_S_#>*** are the throttling value that each valve needs to meet a certain speed, 5 variables are created for the 5 cylinders, and these vary their value over time. And finally the variables ***<S_S_#>*** and ***<S_B_#>*** are of boolean type and mainly activate the solenoids so that each cylinder advances or retreats, thanks to the use of 4/3 valves implemented in Automation Studio.

***Block: CYCLE 1 and 2 ...5**

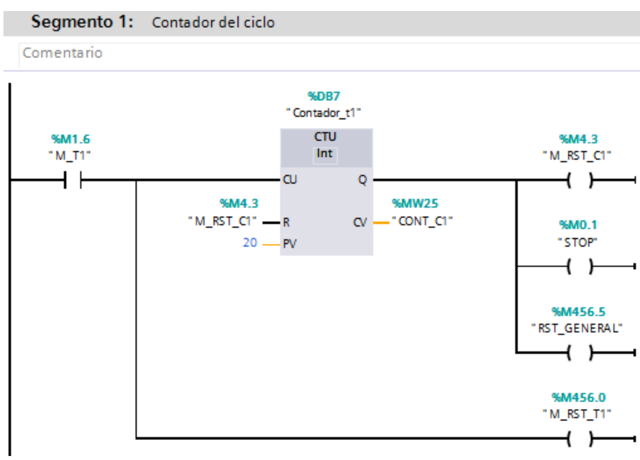

Figure 15:

Cycle counter Each cycle has a counter that contains the 20 segments of each profile, and it is activated when the cycle timer also activates.

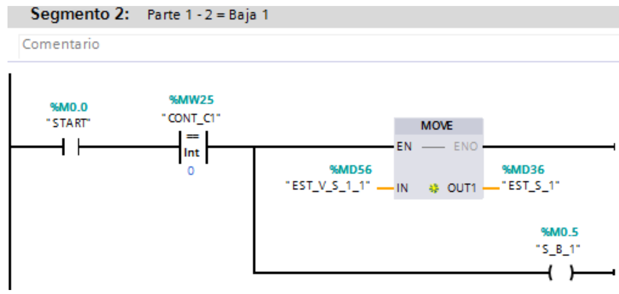

Figure 16: Sending data and comparison

Figure x shows the general start at the beginning of the cycle, which causes all the profiles to be carried out simultaneously, the contact after this is the comparison between the value of the counter and the segment in which it is located. When activated, it allows the “MOVE” block to send the throttling percentage of the valve to the Automation through the OPC.

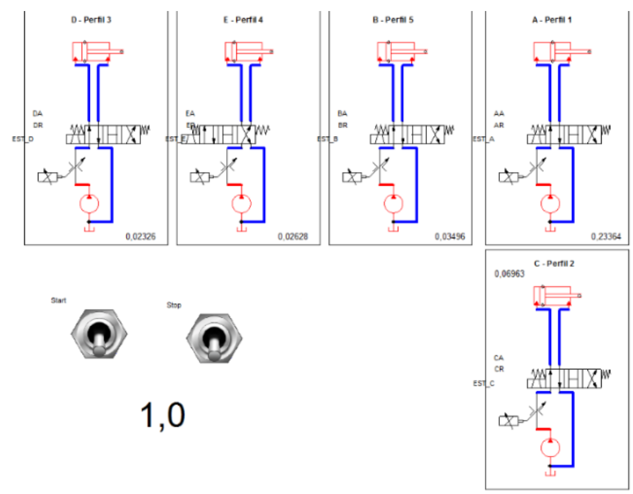

Figure 17: Diagram of the process in Automation Studio

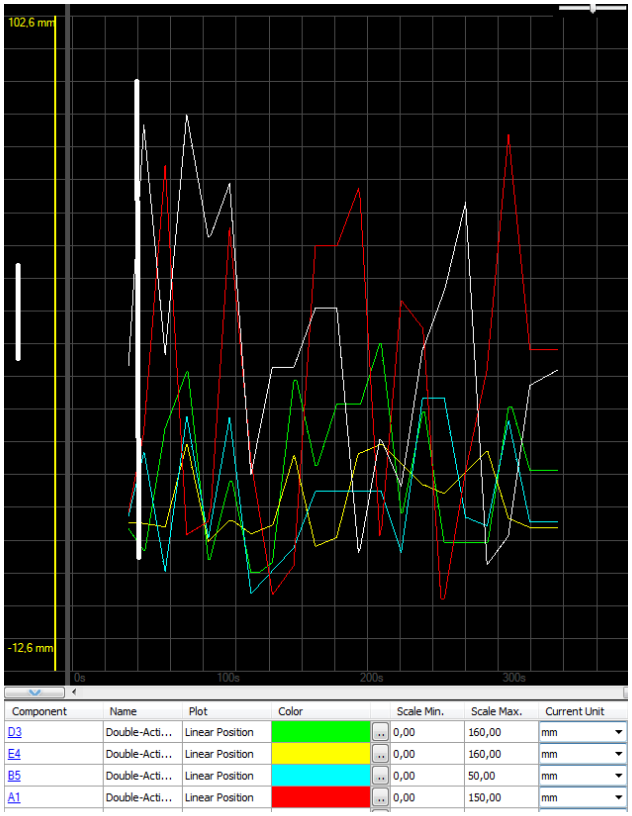

Figure 18: Phase curves of the cylinders

Conclusions:

*After having found the corresponding cylinder, it was concluded that it is one of the multiples that must be taken into account for a simulation successful since it is also necessary to know the circuit to be made and its other components such as the directional valves, the throttles and the compressor.

*At the time of carrying out the program, it was observed that a variable cannot be defined in several segments because a conflict is generated and the variable is always defined at 0, however the desired result is achieved with the overlapping of variables.

*To develop a simulation close enough to reality, it is necessary to take into account each value of each component.

References:

[1]P. Ponsa, A. Granollers, Design and industrial automation, Universitat Politècnica de Catalunya, Material and readings of industrial design, pp 4.

[2]W. Bolton, Mechatronics: electronic control systems in mechanical and electrical engineering, Editorial Alfaomega, 2 ed, pp 436 - 439.

[3]W. Bolton, Mechatronics: electronic control systems in mechanical and electrical engineering, Editorial Alfaomega, 2 ed, pp 423 - 425.

[4] LARZEP double-acting cylinders:

https://www.larzep.com/es/productos/cilindros-hydraulic/d-double-effect