



Joan Olesa Vallespi

# Implementation of a HMI system for a Waste Water Treatment Plant

Dissertation in Electrical and Computer Engineering

15/02/2016



UNIVERSIDADE DE COIMBRA



DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

MAESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

# Implementation of a HMI system for a Waste Water Treatment Plant

Joan Olesa Vallespí

**President of the jury:** António Paulo Mendes Breda Dias Coimbra

**Orientadores:** Manuel Marques Crisóstomo

Lino José Forte Marques

**Vogais:** João Filipe de Castro Cardoso Ferreira

15/02/2015

## ABSTRACT

The objective of this thesis is the automation and simulation of the operation of a Waste Water Treatment Plant (WWTP). Almost all the project will be based on simulations, but one demonstration will be performed in a real device.

This study has been done because the water treatment is a good process to develop an automation system, but the important point of this thesis is the automation system more than the process in itself.

The first part is to introduce the lector in the treatment of wastewaters subject and the second part is to explain the development of the programs and the functioning of each one of them.

The way followed to make this thesis have been divided in seven goals:

1. Understand the functioning of a waste water treatment plant.
2. Design the implementation of the different device that will be necessary to accomplish each process. It means, to choose the type of devices that we will put in every process of the plant.
3. Design all the variables that each device will need. The variables for the PLC and the variables for the HMI.
4. Design a complex program to control entire sensors and actuators that the plant has, using a simulated PLC (Programmable Logic Controller).
5. Design a HMI (Human Machine Interface) system to control all the devices in the simulated plant. The HMI system also serves to get information from the plant.
6. Do a final simulation to confirm the right functioning of the different simulated systems together.
7. Finally, develop a control in real devices. A PLC and HMI program will be designed to control an electric motor.

After the development of the thesis, the principal conclusion is the knowledge acquired in the PLC and HMI systems, above all in the programming.

**KEY WORDS:** W.W.T.P., automation, control, simulation, SIEMENS, S7-300, S7-1200, TP 1500 Basic, KTP600 Basic.

## INDEX

FIGURE LIST .....	6
TABLE LIST .....	6
ABBREVIATIONS LIST .....	6
EQUATIONS LIST .....	6
i. INTRODUCTION .....	7
i.i Objectives and significance of project.....	7
i.ii Definition and characteristics of waste water.....	8
i.ii.i Characteristics of waste water .....	8
i.ii.ii Chemical substances.....	8
i.ii.iii Bacteriological substances.....	9
i.ii.iv Suspended matter and dissolved matter.....	9
i.ii.v Principal parameters.....	10
ii. OPERATION OF A WASTE WATER TREATMENT PLANT .....	11
ii.i Preliminary stage .....	11
ii.ii Primary stage .....	13
ii.iii Secondary stage.....	14
ii.iv Tertiary stage .....	16
iii. SOFTWARE.....	17
iv. HARDWARE.....	18
v. PLC PROGRAM.....	20
v.i Program blocks. ....	20
v.i.i S7-300 Organization Blocks (OB) .....	20
v.i.ii S7-300 Function (FC) .....	21
v.i.iii S7-300 Function Block (FB) .....	23
v.i.iv S7-300 Data Block (DB) .....	24
v.i.v S7-1200 Organization Blocks (OB) .....	24
v.i.vi S7-1200 Function (FC) .....	24
v.ii PLC S7-300 Variables.....	25
v.iii PLC S7-1200 Variables.....	32
vi. HMI PROGRAM.....	33
vi.i Program of TP1500 Basic HMI .....	33
vi.ii Variables of TP1500 Basic HMI .....	33
vi.iii Identification colors of TP1500 Basic HMI .....	34

vi.ii Program of KTP600 Basic HMI .....	35
vii.    POSSIBLE IMPROVEMENTS.....	36
viii.    CONCLUSIONS .....	37
BIBLIOGRAPHY.....	38
ANNEX i.....	39
i.i Element (group) A.....	39
i.ii Element (group) B.....	44
i.iii Element (group) C.....	47
i.iv Element (group) D.....	52
i.v Element (group) E.....	57
i.vi Element (group) F.....	64
i.vii Element (group) G.....	69
i.viii Element (group) H.....	74
i.ix Element (group) I.....	79
ANNEX ii.....	82
ii.i PID [FB1] .....	82
ii.ii Time Counter [FB2].....	83
ii.iii Height simulator [FB3] .....	85
i.iv Mud/Fat simulator [FB4] .....	87
ANNEX iii.....	88
iii.i Principal Menu.....	88
iii.ii Pretreatment 1 .....	89
iii.iii Pretreatment 2.....	90
iii.iv Pretreatment 3.....	92
iii.v Primary treatment 1 .....	94
iii.vi Secondary treatment 1 .....	96
iii.vii Secondary treatment 2 .....	98
iii.viii Tertiary treatment.....	100
iii.ix Visor of curves FCV1/FCV2.....	102
iii.x Visor of curves FCV3/FCV4.....	103
iii.xi Visor of curves FCV5/FCV6.....	104
iii.xii Visor of curves FCV7/FCV8.....	105
iii.xiii Information .....	106
ANNEX iv.....	107
ANNEX v.....	108



**FIGURE LIST**

**Figure i.** Motor to remove big solids in the preliminary stage..... 11  
**Figure ii.** Bottom of the first tank where the muds are settled..... 12  
**Figure iii.** Metal blades of the first tank to remove the fats..... 12  
**Figure iv.** First decantation tank..... 13  
**Figure v.** Generation of muds..... 14  
**Figure vi.** Oxygen injectors..... 14  
**Figure vii.** Secondary decantation tank..... 15  
**Figure viii.** Clean water returning to the Mondego River..... 16  
**Figure ix.** PLC SIEMENS S7-300..... 18  
**Figure x.** HMI SIEMENS TP1500 Basic..... 18  
**Figure xi.** Diagram of signal direction..... 26

**TABLE LIST**

**Table i.** Elements list of the project ..... 26  
**Table ii.** First group variables..... 27  
**Table iii.** Second group variables..... 28  
**Table iv.** Third group variables..... 28  
**Table v.** Fourth group variables..... 29  
**Table vi.** Fifth group variables..... 29  
**Table vii.** Sixth group variables..... 30  
**Table viii.** Seventh group variables..... 31  
**Table ix.** Eighth group variables..... 31  
**Table x.** Ninth group variables..... 32  
**Table xi.** Variables used in the S7-1200 program..... 32

**ABBREVIATIONS LIST**

WWTP: Waste Water Treatment Plant.

PLC: Programmable Logic Controller.

HMI: Human Machine Interface

**EQUATIONS LIST**

**Equation 1.** Mathematic function used along the project..... 82

## **i. INTRODUCTION**

### **i.i Objectives and significance of project.**

The waste water treatment is fundamental to reduce environment wastes that coming from the population or industry and to avoid the incontrollable discharge of waste in our own environment.

In the past, when population of the earth was relatively low, the waste water was directly returned to their riverbed. At that time they did not have chemicals and knowledge about waste water treatments like nowadays.

When the earth population started to grow exponentially because of the industrialization, the waste of water began to be a very big problem to the environment. Then, is when the WWTP appeared, because without this kind of plants our environment could not support the pollution.

The water is one of the most important resources that we have in this planet, is the base of the life, if we want to continue living in this planet we must be careful with our water.

The objective of this thesis, as discussed above, is to make automation and simulation of control on a WWTP. To do this, it is necessary to understand the process very well. That way, it is easier to make a good PLC's program for the plant control.

To make the program, it is indispensable to know all the devices that will be used and their own functioning in order to create the essential variables. The devices (motors, sensors and pumps basically) will be defined after.

Once the program is done, we will proceed with the design of the HMI to have a graphic control and data acquisition of the different variables of the plant devices.

The procedure is the following:

1. Introduction about the different processes that are part of the waste water treatment.
2. Automation with PLC program of all different process that are part of the waste water treatment.
3. Human Machine Interface of all different process that are part of the waste water treatment.

## **i.ii Definition and characteristics of waste water**

The waste waters are all type of waters whose quality has been affected negatively by human influence. The waste water includes:

- Domestic and urban waters.
- Mining or industrial waters.
- Natural waters mixed with the two previous types.

Treating this water is essential for our society; without treatment, pollution problems would be insupportable for the environment. For this reason piping and treatment systems to clean this kind of waters are essential. In big industrialized cities of developed countries they are very common. Also, some industries have their own WWTP.

### **i.ii.i Characteristics of waste water**

All natural water sources contain some other substances, at different concentrations. Rain water has few mg/L of these substances and the sea water has, normally, around 35mg/L of water.

Depending of the physical state of the substances we can differentiate these types:

- Suspended substance in the water
- Colloid substance in the water.
- Dissolved substance in the water.

### **i.ii.ii Chemical substances**

Normally, the waste water is composed by 99% of current water and 1% of suspended or dissolved solids. These solids can be classified into inorganic or organic solids:

- The inorganic solids are composed by: Nitrogen, phosphorus, chlorides, sulphates, carbonates, bicarbonates, arsenic, cyanide, cadmium, chromium, copper, mercury, lead and zinc.



- The organic solids, can be classified in:
  - Nitrogenous (that is contains nitrogen in its molecules): proteins, urea, amines and amino acids.
  - Non-nitrogenous: cellulose, fats and soaps.

### **i.ii.iii Bacteriological substances**

One of the most important reasons to treat the waste waters is the elimination of the pathogenic agents arising from people that are present in excrements, in order to avoid biological contamination. Between these agents we can found:

- Fecal coliform.
- Salmonellas.
- Virus.

### **i.ii.iv Suspended matter and dissolved matter**

Thinking about waste water, we can distinguish between divided in suspended and dissolved matter.

- To extract the suspended matter of the waste water, the most common methods are the physicochemical treatments, mainly, sedimentation and decantation. Normally, the muds that the water contains are extracted by sedimentation, because are denser, and the fats and oils, are extracted by decantation.
- To extract the dissolved matter of the waste water, the most common method is the oxygenation of water to get muds formed from biological components plus organic waste. Then using the decantation, these muds are moved away.

### **i.ii.v Principal parameters**

According to the European Directive, the most important parameters to analyze water waste are:

- Temperature.
- Potential of hydrogen (pH).
- Total Solids in suspension.
- Organic matter.
- Kjeldahl method.
- Ammonias and Nitrates.

## ii. OPERATION OF A WASTE WATER TREATMENT PLANT

The functioning of a Waste Water Treatment Plant will be explained in this section. The WWTP is divided into different stages; in this specific case, the WWTP will be divided in four stages that will be explained below:

### ii.i Preliminary stage

In this stage, the most important thing is to remove the big (more than 2mm) solids to get the correct functioning of the machines along the process. Objects commonly removed during preliminary stage include trash, tree limbs, leaves, branches, and other large objects.

The first part of this preliminary stage is composed by three filters with their metal blades and their respective motors. The first filter picks up the big solid waste, the second, the medium solid waste, and the last one, the little solid particles (less than 2 cm).

In the next image the tools used to remove the solids are shown in the first part of preliminary stage in Figueira da Foz (Portugal) WWTP:



**Figure i.** Motor to remove big solids in the preliminary stage.

The second part of this preliminary stage is composed by one Compressor Pump (CP) to stir the water and separate the muds and the fats dissolved in it. Then, to remove this waste; there are two pumps, one for the muds and the other one for the fats.

In the next figures, the tank where the first physical separation is done is shown.



**Figure ii.** *Bottom of the first tank where the muds are settled.*



**Figure iii.** *Metal blades of the first tank to remove the fats.*

## ii.ii Primary stage

The aim of this stage is to remove the suspended matter in the water. In this case, the first process is the simple decantation by gravity. To do that, a tank where water arrives from the preliminary stage is needed. Normally, in this decantation, 70% of suspended matter is removed. In a normal situation, this process goes in between one or two hours but in this project the process is simulated just during 20 seconds.

Then, to remove this mud, there is one pump. In the next image, first decantation tank is shown in Figueira da Foz WWTP:



**Figure iv.** *First decantation tank.*

### ii.iii Secondary stage

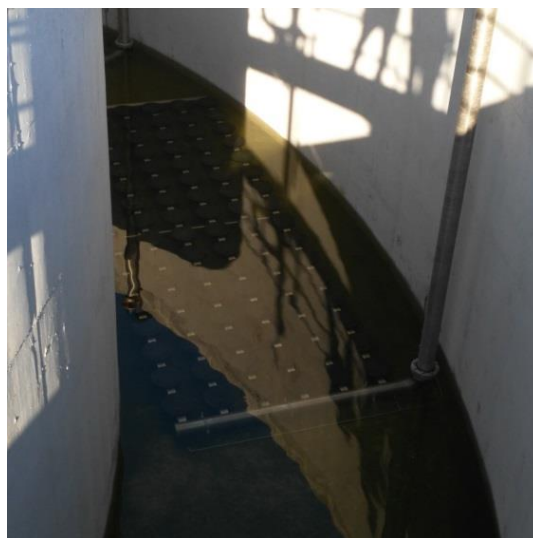
This stage is very important in the process; the aim of the secondary stage is to remove the biological matter dissolved in the water by an oxidation process, followed by sedimentation.

This biological process is a natural control process. It consists in the injection of oxygen to the water. This way the microorganisms can be feed with the organic matter that the water contains. To do that, it has a Side Channel Blower (SCB). During this process the microorganisms grow a lot and create dense lumps. In the next images the tank and the muds generated because of the oxygen injection are shown:



**Figure v.** *Generation of muds.*

And the tank empty where it can see the oxygen injectors:



**Figure vi.** *Oxygen injectors.*



Because of the density of the lumps generated, the lumps go to the bottom of the secondary decantation tank; these lumps receive the name of secondary muds. Then, the secondary muds are picked up by one pump. In the next images, we can see the second decantation tank empty, different parts that compose this system are detailed:



**Figure vii.** *Secondary decantation tank.*

## ii.iv Tertiary stage

The tertiary stage consists in depurate the water for consume. There are several ways to clean the water, but every process has the same purpose, to reduce the ionization of the water before sending it back to the river, to the settlement or to the industry.

Normally, this treatment is based on the cleaning of the rests of detergents, colorants, solvents, odors and tastes. In this case, the process consists in adding chloride in the water before sending it back to the river. To do that, a Chlorine Dispenser Pump (CDP) injects chlorine in the water. In the next image, we can see the clean water returning to the river:



**Figure viii.** *Clean water returning to the Mondego River.*

### iii. SOFTWARE

To accomplish this project, it has been necessary to use specific software from the SIEMENS Company. This software gets the acronym of TIA (Totally integrated Automation), with this software, different types of PLC and HMI from Siemens can be programmed. The TIA software includes:

- **Totally Integrated Automation Portal:** this software is the base of TIA, it is used to update the new software, include the licenses, and make projects with PLC, HMI or SCADA.
- **STEP 7 Professional:** this software is used to make the PLC program. In this project we will use the professional version that allows programming all the SIEMENS PLC's.
- **WinCC Basic:** this software is used to make the HMI program. We could just download just the Basic version because the Professional version it was not available. With the Basic version we can just make a HMI program and that is the reason why this project is made with HMI and not with an SCADA interface.
- **S7-PLCSIM:** this software is used to simulate the programs done with STEP 7 and WinCC. With this version it is not possible to simulate PLC S7-1200/1500.

#### iv. HARDWARE

All the hardware used in the thesis has been simulated for different reasons. Firstly we tried to make some simulations with the PLC S7-1200 of SIEMENS, but the simulations with this device were not available in our software versions. Then, the PLC chosen was S7-300 whose simulations were easier and faster.

To perform the simulations, the CPU 314C-2 PN/DP has been chosen for two reasons, the first is that this CPU have a PROFINET interface to connect with the other hardware devices and the other, is the protocol transport is TCP/IP.



**Figure ix.** PLC SIEMENS S7-300

The HMI screen selected was TP1500 Basic; there are three reasons to choose this screen. The first, because is the biggest in the program (WinCC Basic) and it is better to see the actuators, sensors and their controllers. Another reason is because this device does not have buttons, and the screen is bigger, and finally, it has a PROFINET interface to connect with the PLC chosen.



**Figure x.** HMI SIEMENS TP1500 Basic.

The PLC that the DEEC laboratory has is the S7-1200. This PLC is really used in industry and will be a positive opportunity to work with it. The CPU of this PLC is 1214C AC/DC/Rly whose main characteristics are:

- 14 Digital Inputs (DI) at 24 V.
- 10 Digital Outputs (DO).
- 2 Analogic Inputs (AI).
- 1 Analogic Output (AO)
- PROFINET interface.



**Figure xi.** PLC SIEMENS S7-1200.

The HMI that we can be found in the DEEC laboratory is KTP600 Basic color PN Portrait, this HMI is really used in industry too. The main characters of this device are:

- 5.7" screen.
- 240 x 320 pixels, 256 colors.
- PROFINET interface.



**Figure xii.** HMI KTP600 Basic PN Portrait.

## v. PLC PROGRAM

In this part of the statement, the distribution of the principal PLC program will be explained. The STEP7 has several program blocks. In these blocks is where the program code is written. In this case, all programs have been written in KOP language. All the Programs in S7-300 will be simulated and the program in S7-1200 will be used in a real device.

The program of this thesis is composed by a combination of different program blocks explained below. In **Annex i** and **Annex ii**, we will explain concretely each different program block used to do the simulation, in **Annex iv** the programs used to do the real implementation. Now, we will explain each program block used in the thesis.

### v.i Program blocks.

STEP 7 has four types of program blocks:

- **Organization Blocks (OB)**: these blocks can control the execution of the program. OB can react in front of cyclic events, events controlled by time and, finally, events controlled by alarms too.
- **Function Blocks (FB)**: Function blocks are logical blocks that deposited their values permanently in instance data blocks, so that they remain available after processing block.
- **Function (FC)**: The Functions are logical blocks, without memory. Is more renowned as subroutine.
- **Data Block (DB)**: The Data blocks serve to save different program data.

#### v.i.i S7-300 Organization Blocks (OB)

In this PLC program, three different OBs will be used:

- **Main [OB01]**: This OB is the principal Block in the program, it is generated automatically when it creates a project, and it is used to make a cyclic execution of the rest of the blocks that are in the program.



- **CYC\_INT5 [OB35]**: This OB serves to do a cyclic execution, independent of OB01. That is used to do a PID regulation, because the regulation could have a faster cycle than the OB01. Always that it is necessary to use a closed loop process in the program. In this case, eight PID controllers will be used.
- **COMPLETE RESTART [OB100]**: The function of this OB is to send an initial value, to a determinate variable, when the CPU switches on. In this case, is necessary to send an initial value to the SET POINT and PROCESS VALUE every time that the PID regulation starts.

#### v.i.ii S7-300 Function (FC)

This program has twenty nine FC, not all of the FCs has a different code inside it. In some cases, the actuators are copies of others, with different variables, of course.

Here is a list of the different FCs on S7-300:

- **MDC5 [FC31]**: The function of this FC, is to control the functioning of the Motor Direct Current 5 (MDC5). This motor is used to move up the metal blade in order to pick up the solid waste in the first filter.
- **MDC1 [FC8]**: The function of this FC is to control the functioning of the Motor Direct Current 1 (MDC1). This motor is used to move up the metal blade to pick up the solid waste in the second filter.
- **MDC2 [FC7]**: The function of this FC is to control the functioning of the Motor Direct Current 2 (MDC2). This motor is used to move up the metal blade to pick up the solid waste in the third filter.
- **FCV1/FM1 CONTROL [FC2]**: The function of this FC is to control the regulation of the Flow Control Valve 1 (FCV1) and Flow Meter 1 (FM1). This actuator is the fourth that can be found in the project, it is used to regulate the water flow between the third filter and physical separation of muds and fats.
- **CP1 [FC5]**: The function of this FC is to control the Centrifugal Pump 1 (CP1). This pump is used to inject air in the first tank, doing the first separation between muds and fats.
- **FCV2/FM2 CONTROL [FC3]**: The function of this FC is to control the regulation of the Flow Control Valve 2 (FCV2) and Flow Meter 2 (FM2). This actuator is to regulate the air flow that CP1 injects in the first tank.

- **MP1 [FC4]:** The function of this FC is to control the Mud Pump 1 (MP1). This pump is used to pick up the muds settled at the bottom of the first tank.
- **FP1 [FC9]:** The function of this FC is to control the Fat Pump 1 (FP1). This pump is used to pick up the fats settled on the surface of the first tank.
- **FCV3/FM3 CONTROL [FC6]:** The function of this FC is to control the regulation of the Flow Control Valve 3 (FCV3) and Flow Meter 3 (FM3). This actuator is to regulate the water flow between the first air tank and the second tank (first decantation).
- **MDC3 [FC10]:** The function of this FC is to control the functioning of the Motor Direct Current 3 (MDC3). This motor is used to move the structure that helps to settle the muds in the bottom of the second tank (first decantation tank).
- **MP2 [FC11]:** The function of this FC is to control the Mud Pump 2 (MP2). This pump is used to pick up the muds settled at the bottom of the second tank (first decantation tank).
- **FCV4/FM4 CONTROL [FC12]:** The function of this FC is to control the regulation of the Flow Control Valve 4 (FCV4) and Flow Meter 4 (FM4). This actuator is to regulate the water flow between the second tank and the oxygenize tank (third tank).
- **SCB1 [FC13]:** The function of this FC is to control the regulation of the Side Channel Blower 1 (SCB1). The function of this actuator is to inject oxygen to the water to accomplish the secondary stage.
- **FCV5/FM5 CONTROL [FC14]:** The function of this FC is to control the regulation of the Flow Control Valve 5 (FCV5) and Flow Meter 5 (FM5). This actuator is to regulate the flow oxygen that SCB1 injects in the oxygenize tank.
- **FCV6/FM6 CONTROL [FC15]:** The function of this FC is to control the regulation of the Flow Control Valve 6 (FCV6) and Flow Meter 6 (FM6). This actuator is to regulate the water flow between the third tank (oxygenize tank) and the next one (second decantation tank).
- **MP3 [FC16]:** The function of this FC is to control the Mud Pump 3 (MP3). This pump is used to pick up the muds settled at the bottom of the fourth tank (second decantation tank).
- **MDC4 [FC18]:** The function of this FC is to control the functioning of the Motor Direct Current 4 (MDC4). This motor is used to move the structure that helps to settle the muds in the bottom of the fourth tank (second decantation tank).
- **FCV7/FM7 CONTROL [FC19]:** The function of this FC is to control the regulation of the Flow Control Valve 7 (FCV7) and Flow Meter 7 (FM7). This actuator is to regulate the water flow between the fourth tank (second decantation tank) and the next one (chlorine tank).

- **CDP1 [17]**: The function of this FC is to control the regulation of the Chlorine Dispenser Pump 1 (CDP1). The function of this actuator is to inject chlorine to the water to accomplish the tertiary stage.
- **FCV8/FM8 CONTROL [FC29]**: The function of this FC is to control the regulation of the Flow Control Valve 8 (FCV8) and Flow Meter 8 (FM8). This actuator is to regulate the chlorine flow that CDP1 injects in the fifth tank (chlorine tank).
- **FM9 [FC20]**: The function of this FC is to control the Flow Meter 9 (FM9). The last Flow Meter serves to know the quantity of water that come back to the river.

#### v.i.iii S7-300 Function Block (FB)

This project has four different FBs. The FBs have been used to do different simulations for the different actuators that it can found in the process:

- **HEIGH SIMULATOR [FB3]**: This function block serves to simulate a distance sensor for the MDC1, MDC2 and MDC5, because these actuators are working in function of the distance that is generated by FB3.
- **MUD & FAT SIMULATION [FB4]**: This function block serves to simulate the level of mud and fat that it have in each process. This FB is used in MP1, MP2, MP3 and FP1.
- **PID [FB1]**: This function block serves to simulate a real input. This real input will be a simulation of a water/oxygen/chlorine flow. To do that, a mathematic function to generate this real number has been used. This FB is used in FCV1, FCV2, FCV3, FCV4, FCV5, FCV6, FCV7 and FCV8.
- **TIME COUNTER [FB2]**: This function block is a time counter to control how many time the actuators are in the running state. The actuators with “running state” are MDC1, MDC2, MDC3, MDC4, MDC5, MP1, MP2, MP3, FS1, CP1, SCB1 and CDP1.

#### v.i.iv S7-300 Data Block (DB)

Data block could be created when the FB is used in a FC or in an OB, but it is possible to generate a DB to save variables to do whatever. As FB can be used infinite times, it is necessary to create a different data every time that one FB is used. This project has twenty eight DBs, twenty seven are simulating a FB and one is just to input some necessary variables. The DBs used in the project are listed below:

- DBs generated from **PID [FB1]: PID\_1 [DB8], PID\_2 [DB14], PID\_3 [DB25], PID\_4 [DB10], PID\_5 [DB28], PID\_6 [DB30], PID\_7 [DB31], PID\_8 [DB24].**
- DBs generated from the **MUD & FAT SIMULATION [FB4]: SIMULATION MP1 [DB3], SIMULATION MP2 [DB9], SIMULATION MP3 [DB19], SIMULATION FP1[DB11].**
- DBs generated from the **HEIGH SIMULATOR [FB3]: SIMULATION MDC1 [DB2], SIMULATION MDC2[DB5], SIMULATION MDC3[DB1].**
- DBs generated from the **TIME COUNTER [FB2]: TIME CP1 [DB36], TIME MDC1 [DB44], TIME MDC2 [DB7], TIME MDC5 [DB18], TIME MP1 [DB37], TIME MP2 [DB20], TIME MP3 [DB22], TIME CDP1 [DB4], TIME SCB1 [DB6], TIME FP1 [DB38], TIME MDC3 [DB39], TIME MDC4 [DB21].**

#### v.i.v S7-1200 Organization Blocks (OB)

The PLC program on S7-1200 only has one OB:

- **Main [OB01]:** This OB is the principal Block in the program, it is generated automatically when it creates a project and it is used to make a cyclic execution of the rest of the blocks that are in the program. Also, we have a little program to count the rpm of the electric motor.

#### v.i.vi S7-1200 Function (FC)

The program to control the motors just has one Function:

- **MOTOR [FC1]:** This Function Block serves to make a control on a real motor. Thought the HMI we can input a percentage value to control the tension of the motor.

## v.ii PLC S7-300 Variables

In this part of the project we will see the process used to create the different variables that will be used along of the simulated project. To make the variables, firstly, it is indispensable to divide all the actuators and sensors in different types. The next table will show the classification elements:

ELEMENTS LIST				
Number	Element (name)	Element (type)	Element (group)	Description
1	MDC1	MDC (1)	A	Motor Direct Current
2	MDC2	MDC (1)	A	Motor Direct Current
3	MDC3	MDC (2)	B	Motor Direct Current
4	MDC4	MDC (2)	B	Motor Direct Current
5	MDC5	MDC (1)	A	Motor Direct Current
6	PS1	PS (3)	A	Position Sensor
7	PS2	PS (3)	A	Position Sensor
8	PS5	PS (3)	A	Position Sensor
9	FCV1	FCV (4)	C	Flow Control Valve
10	FCV2	FCV (4)	D, G	Flow Control Valve
11	FCV3	FCV (4)	C	Flow Control Valve
12	FCV4	FCV (4)	C	Flow Control Valve
13	FCV5	FCV (4)	E, H	Flow Control Valve
14	FCV6	FCV(4)	C	Flow Control Valve
15	FCV7	FCV (4)	C	Flow Control Valve
16	FCV8	FCV (4)	D, G	Flow Control Valve
17	FM1	FM (5)	C	Flow Meter
18	FM2	FM (5)	G	Flow Meter
19	FM3	FM (5)	C	Flow Meter
20	FM4	FM (5)	C	Flow Meter
21	FM5	FM (5)	H	Flow Meter
22	FM6	FM (5)	C	Flow Meter
23	FM7	FM (5)	C	Flow Meter
24	FM8	FM (5)	G	Flow Meter
25	FM9	FM (5)	C	Flow Meter
26	MP1	MP (6)	F	Mud Pump
27	MP2	MP (6)	F	Mud Pump
28	MP3	MP (6)	F	Mud Pump
29	FP1	FP (6)	F	Fat Pump
30	MS1	MS (3)	F	Mud Sensor
31	MS2	MS (3)	F	Mud Sensor
32	MS3	MS (3)	F	Mud Sensor
33	FS1	FS (3)	F	Fat Sensor
34	CP1	CP (7)	D	Centrifugal Pump
35	SCB1	SCB (8)	E	Side Channel Blower

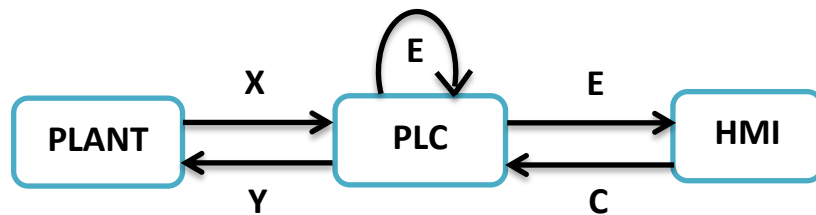
36	OS1	OS (3)	H	Oxygen Sensor
37	CDP1	CDP (7)	D	Chlorine Dispenser Pump
38	CS1	CS (3)	D	Chlorine Sensor
39	PID1 – PID8	PID (9)	I	PID controller to FCVs

**Table i.** Elements list of the project

The **Data type signals** are divided in X, Y, E and C, followed by the identifying and then, doing the Variable (name):

- X: signal which comes from the plant.
- Y: Orders from the PLC to the plant.
- E: States of the elements.
- C: Command or orders from HMI to PLC.

In this image we show how the signals system works:



**Figure xiii.** Diagram of signal direction.

To classify all the variables, the order will be always the same:

**Element (name) +\_+ Variable (name)**

For example: **MDC1 (Element)\_E\_SERVICE (Variable)**

In the superior table we can see the column “Element (type)”, this “type” means the type of variable group that it has in the project; in this case, it has 9 different groups of variables for the different actuators and sensors.



The first element group includes the actuators MDC1, MDC2 and MDC5:

ELEMENTS TYPE 1			
Number	Variable (name)	Variable (type)	Description
1	XRUNNING	BOOL	Running input signal
2	XALARM	BOOL	Alarm input signal
3	C_SERVICE	BOOL	Service command
4	C_OSERVICE	BOOL	Out of service command
5	C_RESET	BOOL	Reset command
6	C_MANUAL	BOOL	Manual Command
7	C_AUTOMATIC	BOOL	Automatic Command
8	C_START_AUTO	BOOL	Start in Automatic mode command
9	C_STOP	BOOL	Stop in Manual mode command
10	C_STOP_AUTO	BOOL	Stop in Automatic mode command
11	C_UP	BOOL	Up command (manual state)
12	C_DOWN	BOOL	Down command (manual state)
13	C_RESET_COUNT	BOOL	Reset of counter command
14	E_RUNNING	BOOL	Running state
15	E_SERVICE	BOOL	Service state
16	E_MANUAL	BOOL	Manual state
17	E_OSERVICE	BOOL	Out of service state
18	E_ALARM	BOOL	Alarm state
19	E_AUTOMATIC	BOOL	Automatic state
20	SECONDS	INT	Seconds running
21	MINUTES	INT	Minutes running
22	HOURS	INT	Hours running

**Table ii.** First group variables.

The second element group includes the actuators MDC3 and MDC4:

ELEMENTS TYPE 2			
Number	Variable (name)	Variable (type)	Description
1	YSTART	BOOL	Start output signal
2	XRUNNING	BOOL	Running input signal
3	XALARM	BOOL	Alarm input signal
4	C_SERVICE	BOOL	Service command
5	C_OSERVICE	BOOL	Out of service command
6	C_START	BOOL	Start command
7	C_RESET	BOOL	Reset command
8	C_STOP	BOOL	Stop command
9	C_RESET_COUNT	BOOL	Reset of counter command

10	E_RUNNING	BOOL	Running state
11	E_SERVICE	BOOL	Service state
12	E_OSERVICE	BOOL	Out of service state
13	E_ALARM	BOOL	Alarm state
14	SECONDS	INT	Seconds running
15	MINUTES	INT	Minutes running
16	HOURS	INT	Hours running

**Table iii.** *Second group variables.*

The third group of elements includes the sensors PS1, PS2, PS5, MS1, MS2, MS3, FS1, OS1 and CS1. The third group will be divided into five different subgroups, the Position group (PS), the Mud group (MS), the Fat group (FS), the Oxygen group (OS) and the Chlorine group (CS). Below, five tables with the three different subgroups:

ELEMENTS TYPE 3			
Subgroup 1: POSITION GROUP			
Number	Variable (name)	Variable (type)	Description
1	HEIGH	REAL	Position of the blade.
Subgroup 2: MUD GROUP			
Number	Variable (name)	Variable (type)	Description
1	MUD_QUANTITY	REAL	Quantity of muds (kg)
Subgroup 3: FAT GROUP			
Number	Variable (name)	Variable (type)	Description
1	FAT_QUANTITY	REAL	Quantity of fats (l)
Subgroup 4: OXYGEN GROUP			
Number	Variable (name)	Variable (type)	Description
1	OXYGEN_QUANTITY	REAL	Quantity of oxygen (ppm)
Subgroup 5: CHLORINE GROUP			
Number	Variable (name)	Variable (type)	Description
1	CHLORINE_QUANTITY	REAL	Quantity of chlorine (ppm)

**Table iv.** *Third group variables.*

The fourth element group includes the actuators since FCV1 to FCV8:

ELEMENTS TYPE 4			
Number	Variable (name)	Variable (type)	Description
1	XALARM	BOOL	Alarm input signal
2	C_AUTOMATIC	BOOL	Automatic command
3	C_MANUAL	BOOL	Manual command
4	C_SERVICE	BOOL	Service command
5	C_RESET_ALARM	BOOL	Reset command
6	C_OSERVICE	BOOL	Out of service command
7	E_ALARM	BOOL	Alarm state
8	E_AUTOMATIC	BOOL	Automatic state
9	E_OSERVICE	BOOL	Out of service state
10	E_MANUAL	BOOL	Manual state
11	E_SERVICE	BOOL	Service state

**Table v.** Fourth group variables.

\* Note: FCV2, FCV5 and FCV8, do not have the Variables C\_MANUAL and C\_AUTOMATIC, because it is not necessary, the E\_MANUAL and E\_AUTOMATIC states will be controlled by CP1, SCB1 and CDPI, respectively.

The fifth element group includes the actuators from FM1 to FM9:

ELEMENTS TYPE 5			
Number	Variable (name)	Variable (type)	Description
1	XALARM	BOOL	Alarm input signal
2	E_ALARM	BOOL	Alarm state
3	XMINS	REAL	Real input value
4	MINS	REAL	Real value in the program

**Table vi.** Fifth group variables.

The sixth element group includes the actuators MP1, MP2, MP3 and FS1:

ELEMENTS TYPE 6			
Number	Variable (name)	Variable (type)	Description
1	YSTOP	BOOL	Stop output signal
2	YSTART	BOOL	Start output signal
3	XRUNNING	BOOL	Running input signal
4	XALARM	BOOL	Alarm input signal
5	C_SERVICE	BOOL	Service command

6	C_RESET	BOOL	Reset command
7	C_OSERVICE	BOOL	Out of service command
8	C_MANUAL	BOOL	Manual command
9	C_AUTOMATIC	BOOL	Automatic command
10	C_START	BOOL	Start in manual mode command
11	C_STOP_AUTO	BOOL	Stop in automatic mode command
12	C_START_AUTO	BOOL	Start in automatic mode command
13	C_STOP	BOOL	Stop in manual mode command
14	C_RESET_COUNT	BOOL	Reset of counter command
15	E_RUNNING	BOOL	Running state
16	E_SERVICE	BOOL	Service state
17	E_MANUAL	BOOL	Manual state
18	E_AUTOMATIC	BOOL	Automatic state
19	E_OSERVICE	BOOL	Out of service state
20	E_ALARM	BOOL	Alarm state
21	SECONDS	INT	Seconds running
22	MINUTES	INT	Minutes running
23	HOURS	INT	Hours running

**Table vii.** Sixth group variables.

The seventh element group includes the actuators CP1 and CDP1:

ELEMENTS TYPE 7			
Number	Variable (name)	Variable (type)	Description
1	YSTART	BOOL	Start output signal
2	XRUNNING	BOOL	Running input signal
3	XALARM	BOOL	Alarm input signal
4	C_SERVICE	BOOL	Service command
5	C_OSERVICE	BOOL	Out of service command
6	C_MANUAL	BOOL	Manual command
7	C_AUTOMATIC	BOOL	Automatic command
8	C_START	BOOL	Start in manual mode command
9	C_STOP	BOOL	Stop in manual mode command
10	C_START_AUTO	BOOL	Start in automatic mode command
11	C_STOP_AUTO	BOOL	Stop in automatic mode command
12	C_RESET	BOOL	Reset command
13	C_RESET_COUNT	BOOL	Reset of counter command
14	E_ALARM	BOOL	Alarm state
15	E_SERVICE	BOOL	Service state
16	E_OSERVICE	BOOL	Out of service state
17	E_MANUAL	BOOL	Manual state
18	E_AUTOMATIC	BOOL	Automatic state
19	E_RUNNING	BOOL	Running state
20	SECONDS	INT	Seconds running

21	MINUTES	INT	Minutes running
22	HOURS	INT	Hours running

**Table viii.** Seventh group variables.

The eighth element group includes just the actuator SCB1:

ELEMENTS TYPE 8			
Number	Variable (name)	Variable (type)	Description
1	YSTART	BOOL	Start output signal
2	YCOOL	BOOL	Cool ok output signal
3	YOIL	BOOL	Oil ok output signal
4	XALARM	BOOL	Alarm input signal
5	XRUNNING	BOOL	Running input signal
6	XCOOL	BOOL	Cool input signal
7	XOIL	BOOL	Oil input signal
8	C_SERVICE	BOOL	Service command
9	C_OSERVICE	BOOL	Out of service command
10	C_MANUAL	BOOL	Manual command
11	C_AUTOMATIC	BOOL	Automatic command
12	C_START	BOOL	Start in manual mode command
13	C_STOP	BOOL	Stop in manual mode command
14	C_START_AUTO	BOOL	Start in automatic mode command
15	C_STOP_AUTO	BOOL	Stop in automatic mode command
16	C_COOL	BOOL	Cool ok (manual mode) command
17	C_OIL	BOOL	Oil ok (manual mode) command
18	C_READY	BOOL	Ready (automatic mode) command
19	C_RESET	BOOL	Reset command
20	C_RESET_COUNT	BOOL	Reset of counter command
21	E_ALARM	BOOL	Alarm state
22	E_SERVICE	BOOL	Service state
23	E_OSERVICE	BOOL	Out of service state
24	E_MANUAL	BOOL	Manual state
25	E_AUTOMATIC	BOOL	Automatic state
26	E_RUNNING	BOOL	Running state
27	E_OILOK	BOOL	Oil ok state
28	E_COOLOK	BOOL	Cool ok state
29	E_READY	BOOL	Ready state
30	SECONDS	INT	Seconds running
31	MINUTES	INT	Minutes running
32	HOURS	INT	Hours running

**Table ix.** Eighth group variables.

Finally, it has another group of variables to control the opening percentage of every Flow Control Valve (FCV).

ELEMENT TYPE 9			
Number	Variable (name)	Variable (type)	Description
1	POSITION	REAL	FCV opening percentage

**Table x.** Ninth group variables.

The simulated program has a total of 452 different variables among states, commands, integer tags, inputs, outputs and internal tags to control all the system.

### v.iii PLC S7-1200 Variables

In this part of the project, the variables used in the program of S7-1200 will be explained. To control the Motor, we just needed few variables. It is not necessary to do a complex division of variables.

Number	Variable (name)	Variable (type)	Description
1	START	BOOL	Start command
2	STOP	BOOL	Stop command
3	INPUT SIGNAL	REAL	Input Percentage of velocity
4	REAL VALUE	REAL	Internal tag to escalation
5	OUTPUT SIGNAL	INT	Output tension value to the motor
6	FORWARD	BOOL	Command forward direction
7	REVERSE	BOOL	Command reverse direction

**Table xi.** Variables used in the S7-1200 program.

## vi. HMI PROGRAM

To accomplish the programs of the HMIs will be used the WinCC software explained in the third point of the memory.

### vi.i Program of TP1500 Basic HMI

The HMI program to control the TP1500 Basic is divided in thirteen images:

- **Principal Menu (1):** This is the menu, from this image the user can go to the others images.
- **Process images (7):** In these images the different parts of the process are shown. To explain the preliminary stage three images have been used, for the primary stage, just one image, for the secondary stage, two more images, and finally, to show the tertiary stage, one more.
- **Visor of curves images (4):** In these images the graphic elements are shown to follow the PIDs development. To show these evolutions there are four images, the first shows the development of PID1 and PID2, the second shows the development of PID 3 and PID4, the third shows the development of PID5 and PID6, and finally, the fourth image, shows the development of PID 7 and PID8.
- **Information (1):** In this image, it is shown the information about the project and about the author.

In the **Annex iii**, it will be explained concretely each part of each different image of the HMI simulated program.

### vi.ii Variables of TP1500 Basic HMI

To do the HMI program, three kinds of variables were needed. In the **Figure xiii**, we can see that the variables which interact with PLC are states (E) or commands (C), but there is another type, the integer tags to control the working of the different actuators.

- **States (E):** In the HMI images, the control panels are shown, every control panel has the states of the actuator. Every “E variable” is equivalent to a state (E\_SERVICE, C\_OSERVICE and etcetera).

- **Commands (C):** In the HMI images, the control panels can be seen, every control panel has buttons with names, and every button serves to connect or disconnect the relay associated to a Command (C\_SERVICE, C\_OSERVICE and etcetera).
- **Integer tags:** In the HMI images, different real inputs can be seen to show different variables of the program (sensors and meters).

### vi.iii Identification colors of TP1500 Basic HMI

The HMI interface has been created with different colors to identify the different states of each actuator or quantities of each meter by graphic way.

Every **actuator** (it considers actuators FCVs, MPs, MDCs, FP, CP, CDP and SCB) of the process has an “identification state color”, the classification of these colors is the following:

- **Grey:** When the actuator is “OSERVICE” state (“Out of service”) or is off.
- **Green:** When the actuator is on “SERVICE” state, or if it has, “RUNNING” state, the square in the actuator put it on green.
- **Red:** When the actuator is on “Alarm state”, in this case, the square is flicking.

Every **flow meter** in the process (from FM1 to FM9) has an “identification flow quantity”, the unit used for the FMs is l/s, could be another, but in metric system is the most common. The colors classification in function of the flow is:

- **Orange:** When the flow is between 0 and 20 l/s.
- **Green:** When the flow is between 21 and 80 l/s.
- **Red:** When the flow is between 81-100 l/s.

\*Note: This identification is completely arbitrary, could be another different, this project is focused in automation, not in the accuracy of the specific processes.



## vi.ii Program of KTP600 Basic HMI

The HMI program to control the KTP600 Basic has only two images:

- **Principal Menu (1):** This is the menu image, from this we can go to the control motor.
- **Control Motor (2):** In this image we can found the different buttons and inputs to control the motor.

## **vii. POSSIBLE IMPROVEMENTS**

To improve this project, there are several things that could have been changed. The principal change that I could have done is the development of a SCADA system to control the system, but unfortunately, we did not have the suitable software to make it. Anyway, the WinCC software allows a Human Machine Interface and a basic control system can be done. Actually, the HMI is like an SCADA but more limited because the presentation and the formats are simpler.

Another thing that could be different in this project is the PLC program application, I would like to test more programs in a real PLC but the time I had was limited. Also, I could use more Function Blocks [FB] instead of Functions [FC], in fact, when we use the FB, the memory used is less and the SCAN time is smaller, but I was not accustomed to use the FB and I decided to accomplish most of the PLC program with subroutines (FC).

Also, I could have done a simulation on the "treatment muds" stage, but I thought that it was too redundant. I thought that was not necessary and I decided to invest the time doing another things.

In general, I am a beginner in the automation world but I think that this thesis is sufficiently complex to be a beginner.

## viii. CONCLUSIONS

After the realization of this thesis I have obtained some conclusions. I have been working in this project during five month approximately and during this time I learned a lot of things that will be important in my professional career.

The first and most important conclusion is the self-learning in the programming of automatic devices. I have never been working with this software before and it was completely new for me. The TIA Portal use blocks to design programs, which are new for me.

Another conclusion would be the necessity that sometimes we have to find information. As a beginner in this world, I had some problems that I repaired searching for information, and above all, spending time treat with the different software used. I have needed to read some manuals of SIEMENS to understand some blocks that the program has. I learned that when you spend the enough time trying to solve a problem, finally you get the solution, perhaps seems unimportant, but for me has been a really relevant thing.

The visual part of the thesis has been made with a HMI system. The program developed by me, would be better in a SCADA, is a pity because with HMI you cannot see all the potential of the program made, anyway the result has been successful.

The conclusion that I obtained during the last weeks of the thesis is related with the communications between the different real devices used. When the real devices are used, you realize of the importance of this part in the automation world. The knowledge of the PROFINET interface will be really useful for me in the future.

For me, make the thesis in English have been a really big achievement too, because I do not used this language too much. Right now, with the Bologna plan, we need to do some credits in English to get the tittle.

As a general conclusion I can say that the development of this thesis has been really useful for me, because I want to focus my professional degree in something related with automatic systems, doing projects with PLC and developing visual control system like HMI or SCADAs.

## BIBLIOGRAPHY

[1] SIEMENS AG, *SIMATIC Programar con STEP 7* [Online]. Nuremberg, SIEMENS AG, 2006 [Data of inquiry: November 2015]. Available on:

<[https://cache.industry.siemens.com/dl/files/056/18652056/att\\_70833/v1/S7prv54\\_s.pdf](https://cache.industry.siemens.com/dl/files/056/18652056/att_70833/v1/S7prv54_s.pdf)>

[2] SIEMENS AG, *SIMATIC WinCC Basic V11.0 SP1* [Online]. Nuremberg, SIEMENS AG, 2011 [Data of inquiry: December and January 2016]. Available on:

<[http://wikifab.dimf.etsii.upm.es/wikifab/images/b/b8/WinCCBasic\\_v11.pdf](http://wikifab.dimf.etsii.upm.es/wikifab/images/b/b8/WinCCBasic_v11.pdf)>

[3] RODRÍGUEZ, Antonio, LETÓN, Pedro, ROSAL Roberto, DORADO Miriam, VILLAR Susana, SANZ Juana M, *Tratamientos de aguas residuales industriales* [Online]. Madrid, Comunidad de Madrid, 2006 [Date of inquiry: October 2015]. Available on:

<[https://www.madrimasd.org/informacionidi/biblioteca/publicacion/doc/VT/VT2\\_Tratamientos\\_avanzados\\_de\\_aguas\\_residuales\\_industriales.pdf](https://www.madrimasd.org/informacionidi/biblioteca/publicacion/doc/VT/VT2_Tratamientos_avanzados_de_aguas_residuales_industriales.pdf)>

[4] DEL VILLAR, Alberto, *Guía de tratamientos avanzados de aguas residuales urbanas* [Online]. Alcalá, Ministerio de Economía y Competitividad, 2011 [Date of inquiry: October 2015]. Available on:

<[http://www.consolider-tragua.com/documentos/guia\\_tratamientos\\_avanzados.pdf](http://www.consolider-tragua.com/documentos/guia_tratamientos_avanzados.pdf)>

# ANNEX i

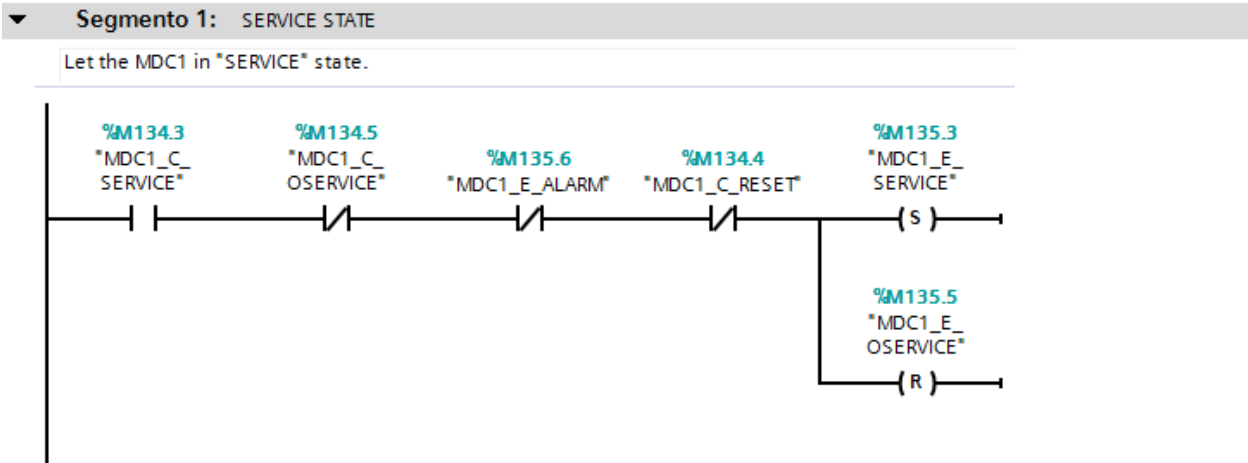
The first annex, as said in the fifth section of the memory, will explain step by step the entire simulated program done along the dissertation. In the **Table i**, the different types of actuators and sensors that it has in the programs can be seen, and how are classified. Well, the fourth column of this table, **Element (group)**, shows the type of subprogram that each element has. It means that every different group have the same program structure, for this reason, explaining the eight types (A, B, C, D, E, F, G, H, I) of subprograms that it have it will explain all the subprograms of the thesis.

Next, each program type step by step will be detailed.

## i.i Element (group) A

The first elements in the **Table i** are the MOTOR DIRECT CURRENT MDC1, MDC2 and MDC5. These motors are used in the preliminary stage to pick up the solid waste in the water. To explain the functioning of these motors the MDC1 will be used as example, but it could have chosen any of the other two.

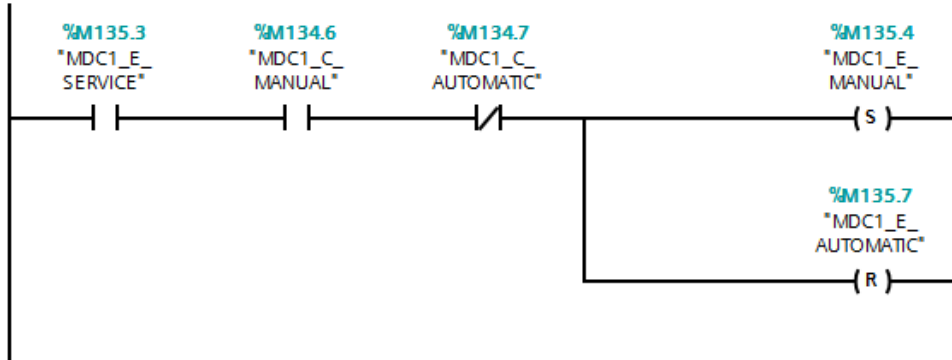
- The first segment is to let the MDC system in "SERVICE" state.



- The second segment is to choose the manual mode of MDC setting the system in “MANUAL” state.

**Segmento 2: MANUAL STATE**

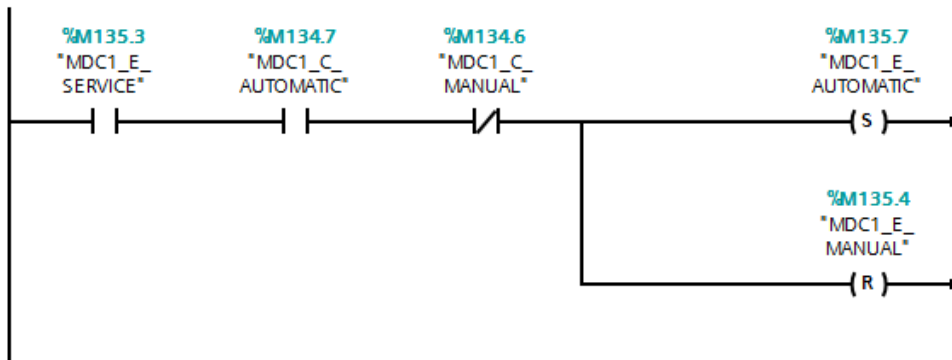
Let's the MDC1 in "MANUAL" state.



- The second segment is to choose the automatic mode of MDC setting the system in “AUTOMATIC” state.

**Segmento 3: AUTOMATIC STATE**

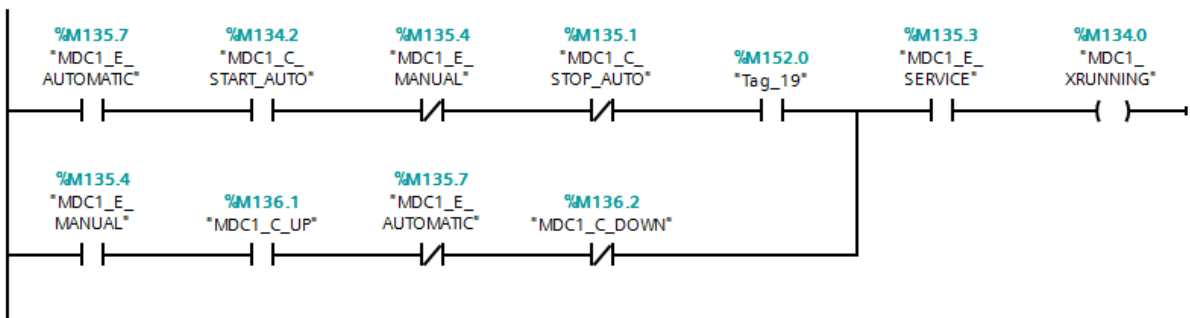
Let the MDC1 in "AUTOMATIC" state.



- The third segment is to send the “START\_AUTO” command (when the system is in “AUTOMATIC” state) or the “START” command (when the system is in “MANUAL” state) to send from the plant to the PLC the “XRUNNING” signal.

**Segmento 4: OUTPUT START SIGNAL**

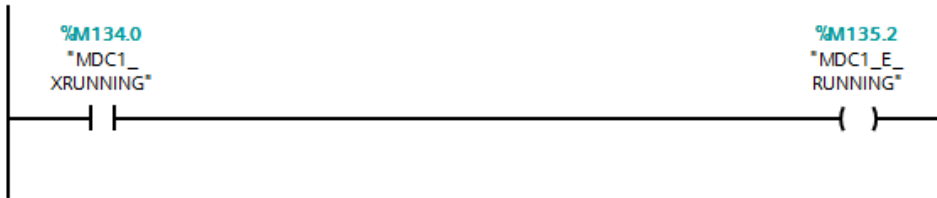
Send de output order of "START" in Automatic/Manual Mode.



- The fourth segment is to simulate the signal that comes from the plant (“XRUNNING” signal) to set the system in “RUNNING” state.

**Segmento 5: RUNNING STATE**

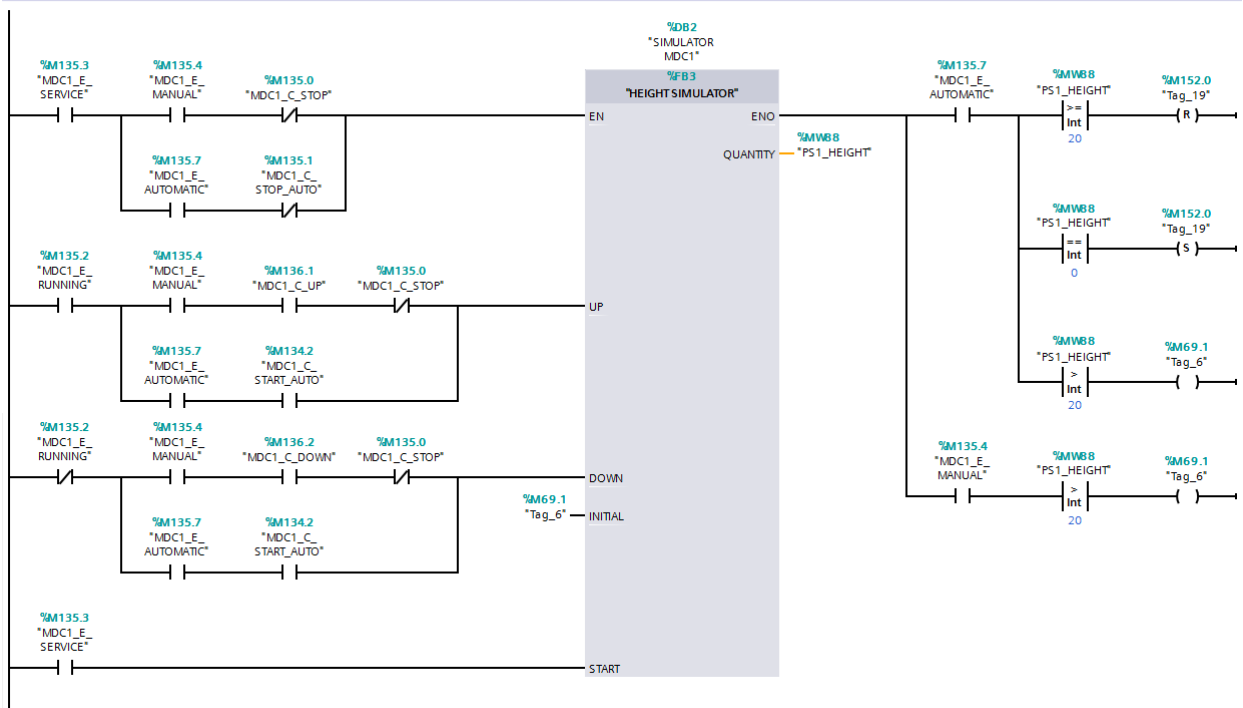
The input signal of ready activate the "RUNNING" state. This state is when the MDC1 is running.



- The next segment is where the height level is generated (by simulation), depending of the work mode of the system (automatic or manual) the simulator (FB3) will work different. When the system is in “AUTOMATIC” state, the actuator will be active every 20 seconds to simulate the ascent or descent of the metal blade which picked up the waste from the water. When the system is in "MANUAL" state, the ascent and descent of the metal blades must be activated manually, “UP”, “DOWN” or “STOP” command can be chosen.

**Segmento 6: SIMULATOR**

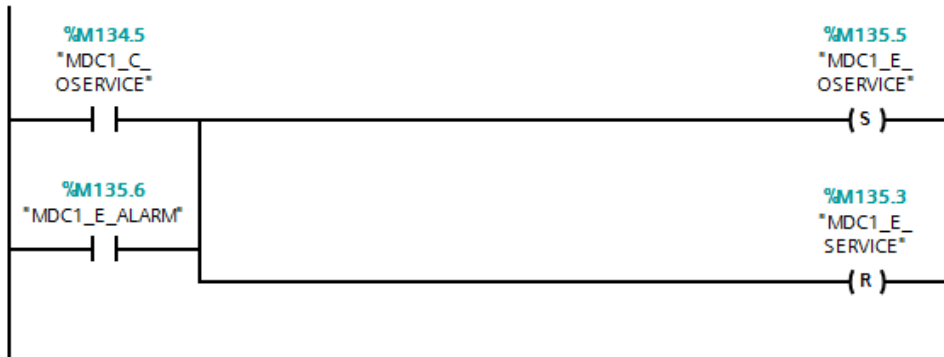
In the FB3 is simulated the going up or the going down of the blades to collect the waste.



- When the MDC sends the “OSERVICE” command or the “ALARM” state is on, the system puts in “OSERVICE” state.

▼ **Segmento 7: OUTPUT OSERVICE SIGNAL**

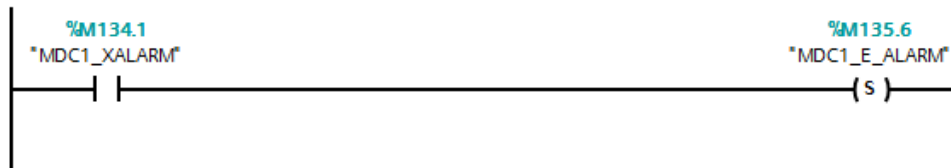
States, orders and signals whe let the MDC1 in "OSERVICE" state.



- The eighth segment is for simulate the alarm signal which comes from the plant and let the system in “ALARM” state.

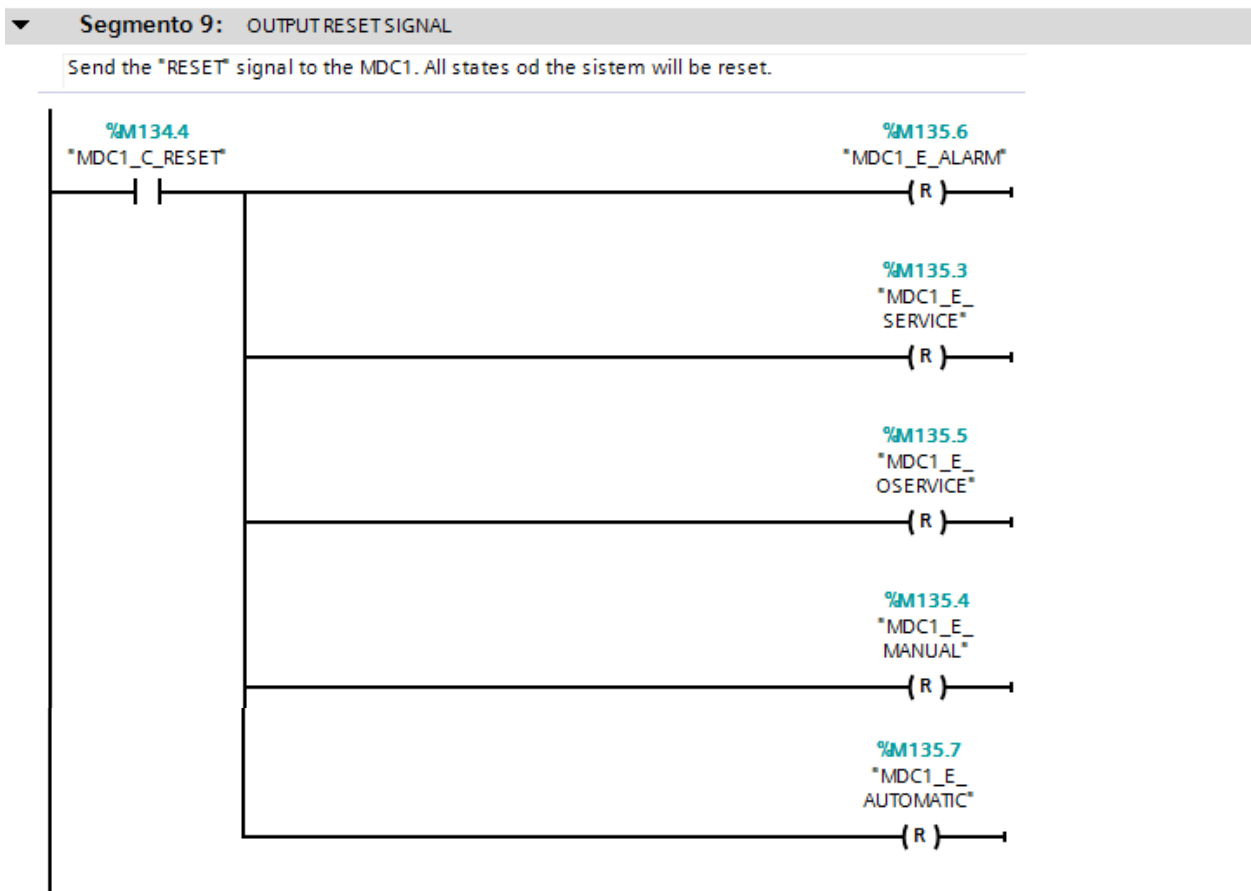
▼ **Segmento 8: ALARM STATE**

The input alarm signal puts the MDC1 in "ALARM" state

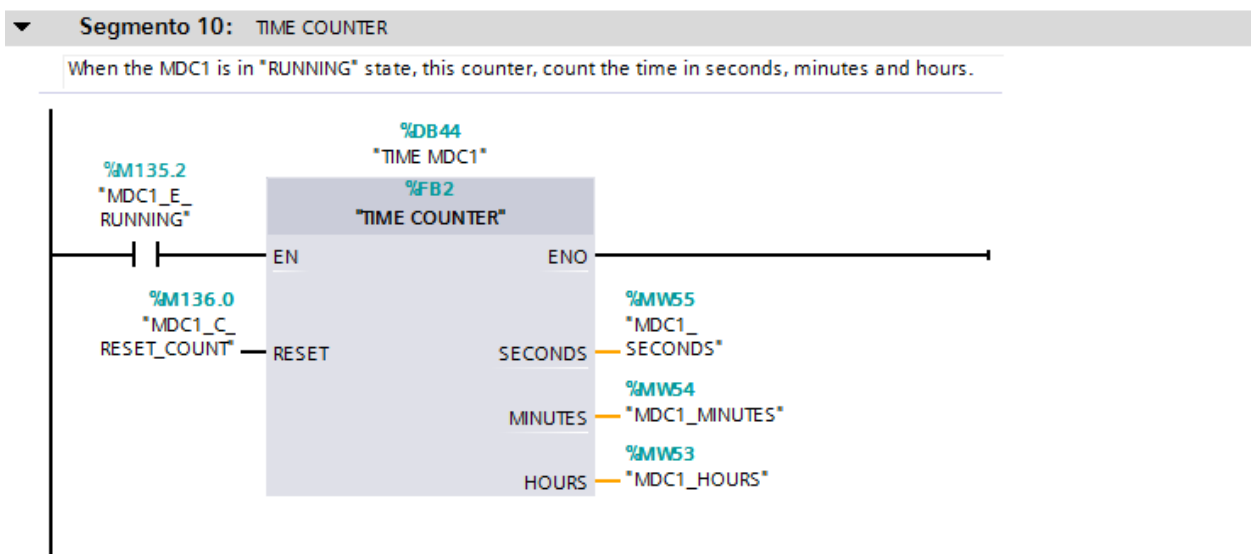




- The ninth segment is to put all the states of the system in reset.



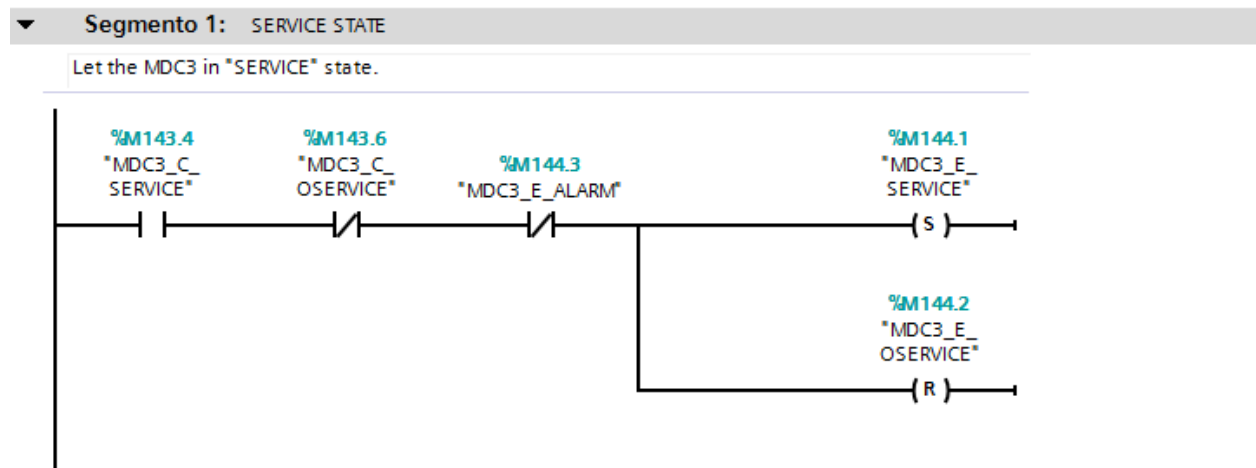
- The last segment is the time counter (TIME COUNTER [FB2]) used to count how many time the MDCs are in "RUNNING" state.



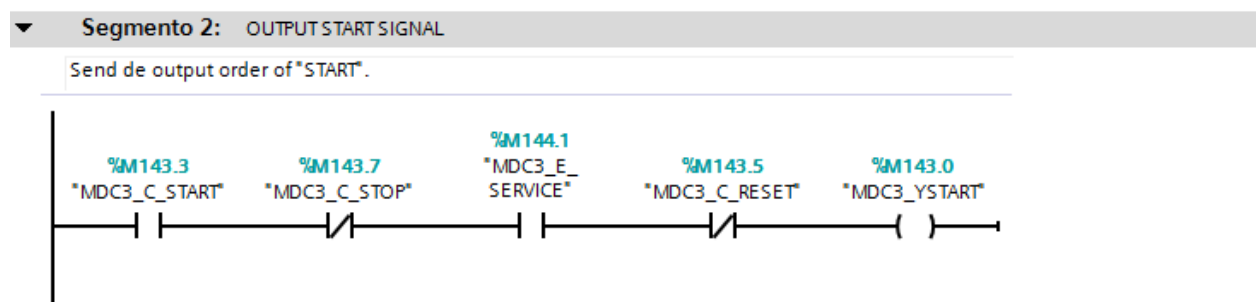
## i.ii Element (group) B

The elements of the second group are the motors that control the tools used to accomplish the primary and secondary decantation, which are MDC3 and MDC4. These motors do not have manual or automatic states because it was not necessary for the function that they will do. To explain these elements the MDC3 will be used as an example.

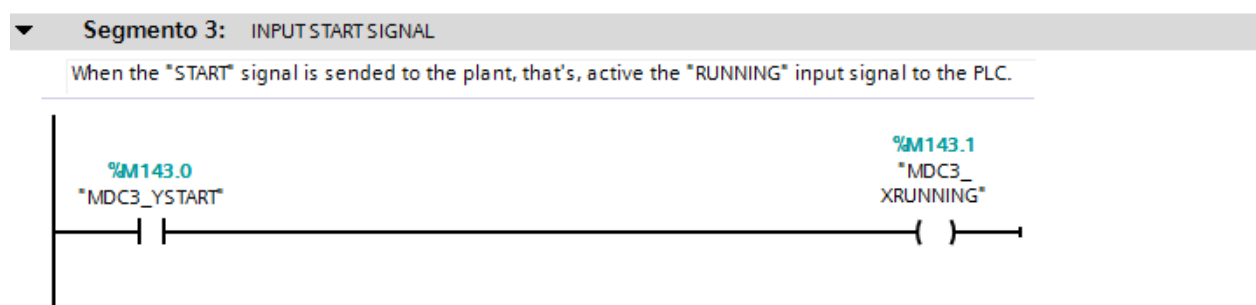
- The first segment is to let the MDC system in “SERVICE” state.



- The second segment is to send to the plant the “START” signal.



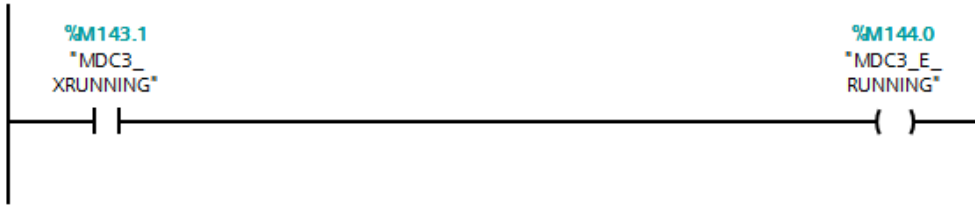
- The next segment is to simulate the input signal “XRUNNING” to the PLC.



- The fourth segment is to simulate when the input signal “XRUNNING” lets the system in “RUNNING” state.

▼ **Segmento 4: RUNNING STATE**

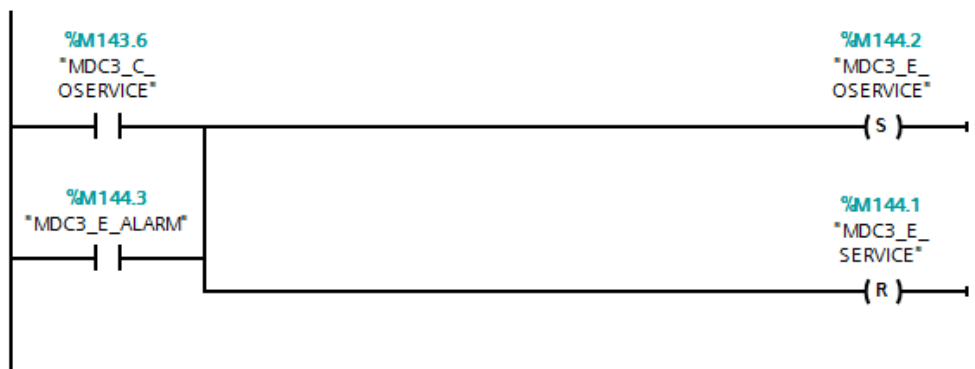
The input signal of ready activate the "RUNNING" state.



- When the “OSERVICE” command is sent or the “ALARM” state is ON, the system is set on “OSERVICE” state.

▼ **Segmento 5: OUTPUT OUT SERVICE SIGNAL**

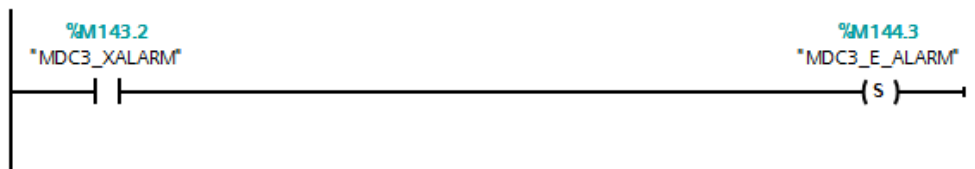
"ALARM" state or "OSERVICE" order, put the system un "OSERVICE" state.



- The sixth segment is to simulate the input signal “XALARM” which lets the system in “ALARM” state.

▼ **Segmento 6: INPUT ALARM SIGNAL**

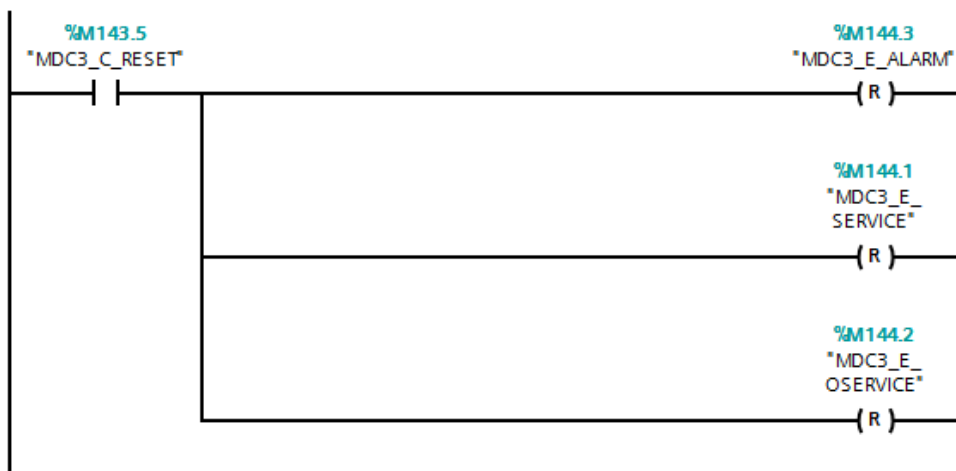
If the signal of alarm arrives from the plant, the MDC3 "ALARM" state is on.



- The ninth segment is to set all the states of the system in reset.

▼ **Segmento 7: OUTPUT RESET SIGNAL**

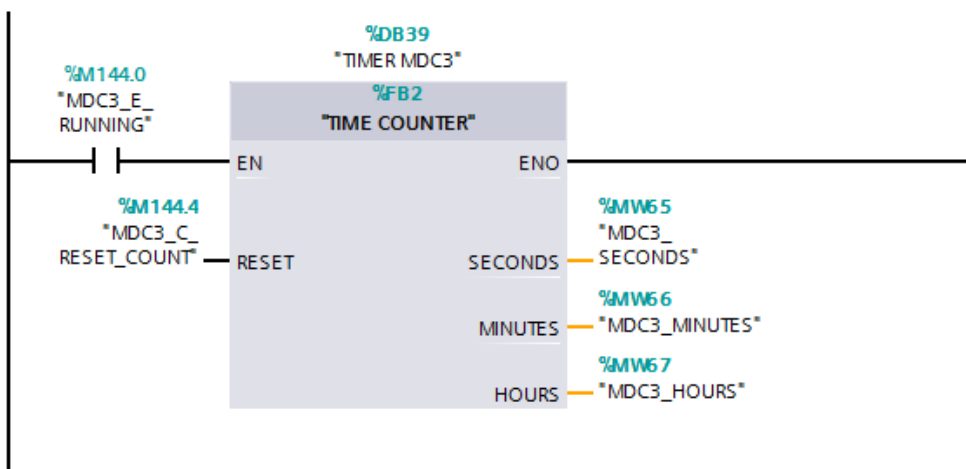
Send the "RESET" signal to the MDC3.



- The last segment is the time counter (TIME COUNTER [FB2]) used to count how many time the MDCs are in "RUNNING" state.

▼ **Segmento 8: TIME COUNTER**

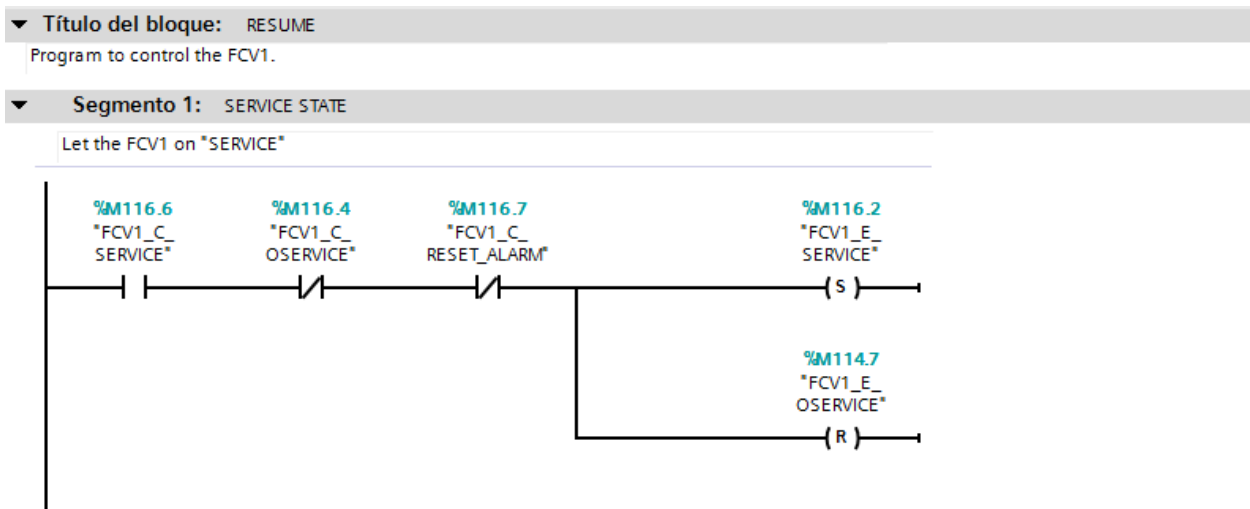
When the MDC31 is in "RUNNING" state, this counter, count the time in seconds, minutes and hours.



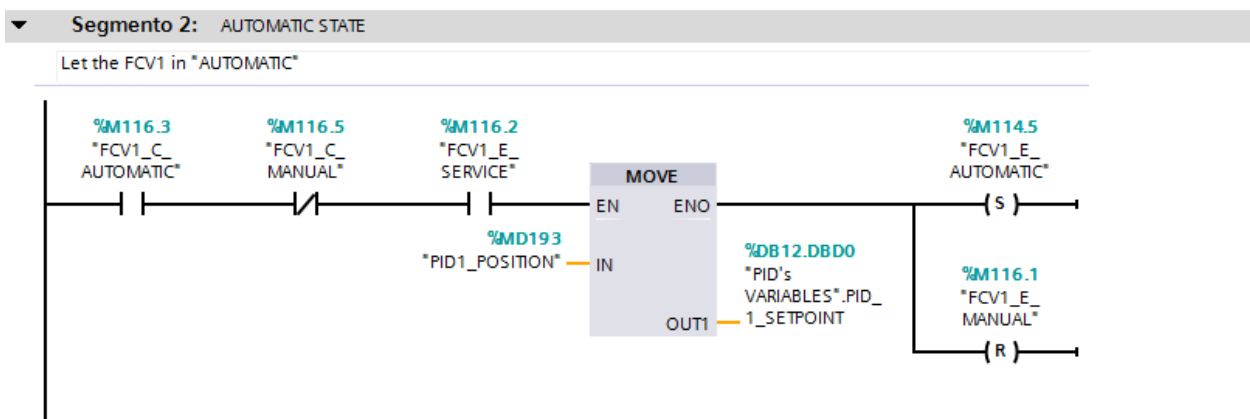
### i.iii Element (group) C

The elements from this group are the first type of FLOW CONTROL VALVES (FCV1, FCV3, FCV4, FCV6 and FV7) and FLOW METERS (FM1, FM3, FM4, FM6 and FM7). The function of these valves is to regulate the flow between the different parts in the process. The program done to the FCV1 will be used for the explanation of the program:

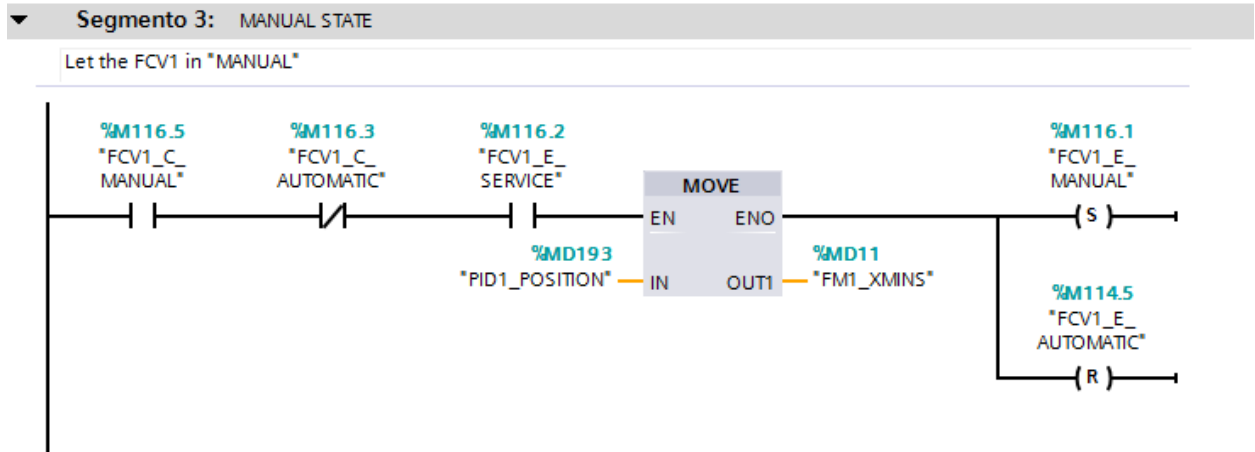
- The first segment is to let the FCV system in “SERVICE” state.



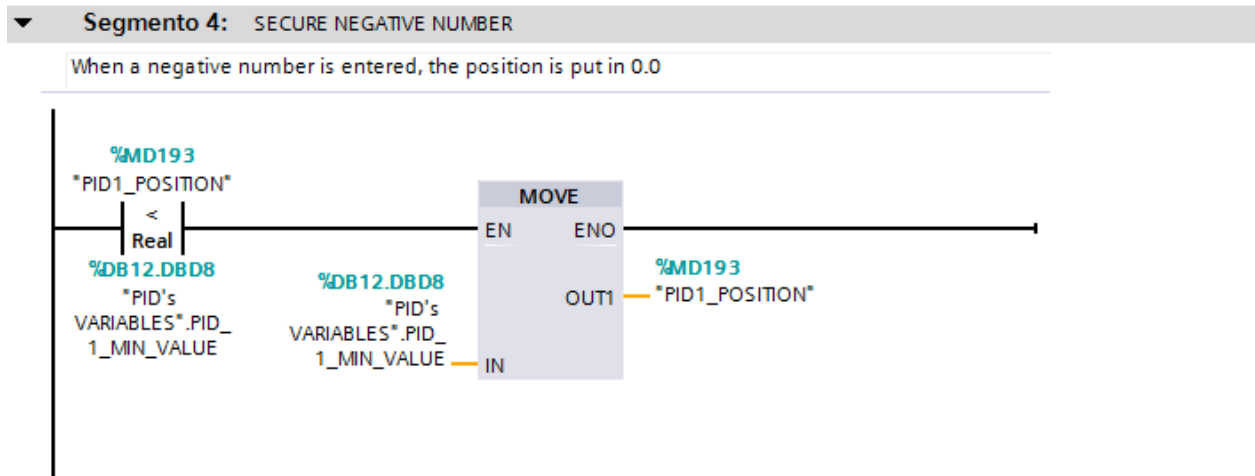
- The second segment is to choose the automatic mode of FCV and to enter the position of the valve (opening percentage).



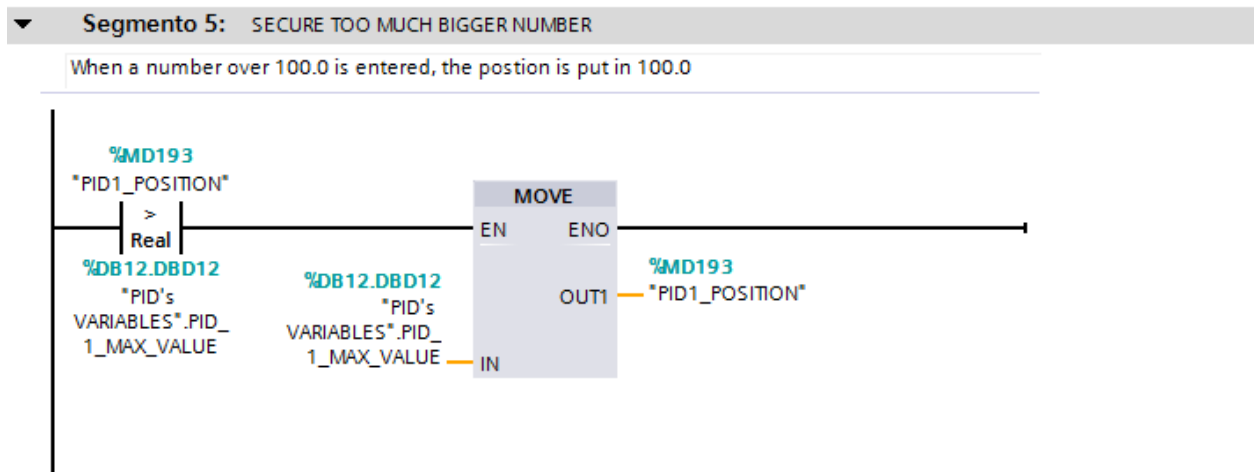
- The third segment is to choose the manual mode of FCV and to enter the position of the valve (opening percentage).



- This segment is to ensure that you will never enter a negative number to the valve.



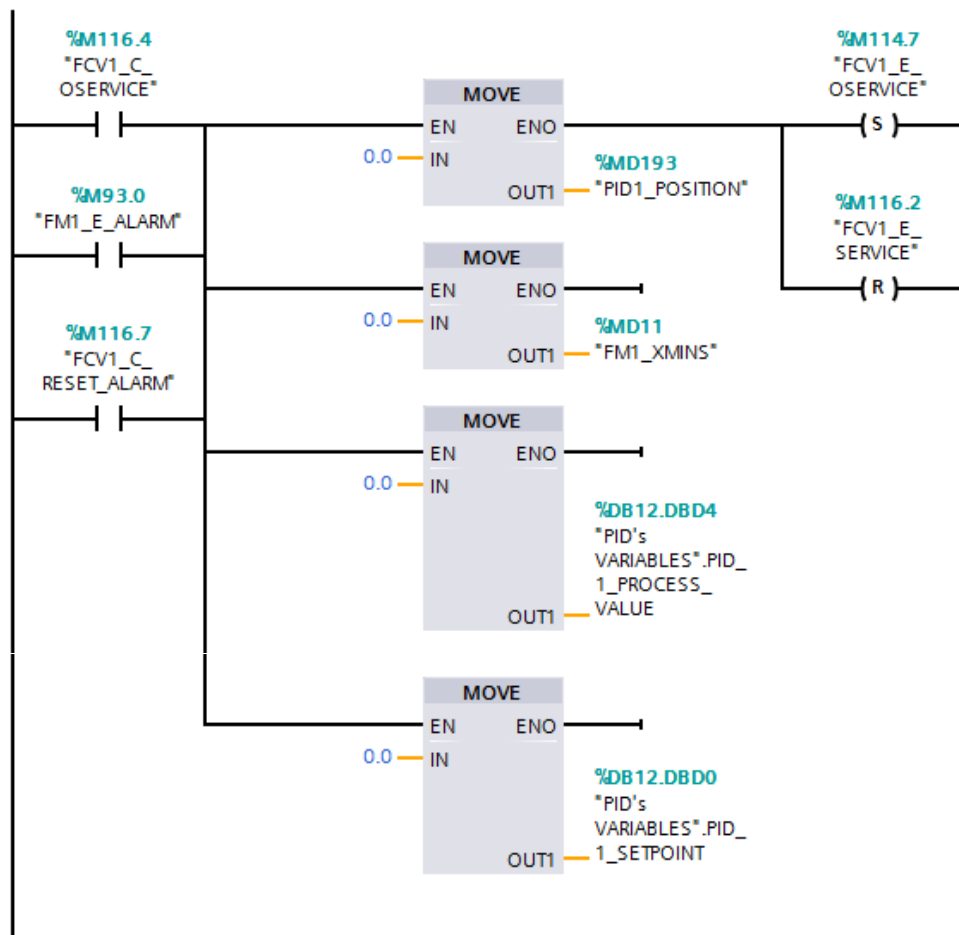
- This segment is to ensure that you will never enter a number bigger than 100 to the valve.



- The sixth segment is to let the system in “OSERVICE” state when the "OSERVICE" command is sent, when the alarm of FVC is active or when the “RESET” command is sent.

**Segmento 6: OSERVICE STATE**

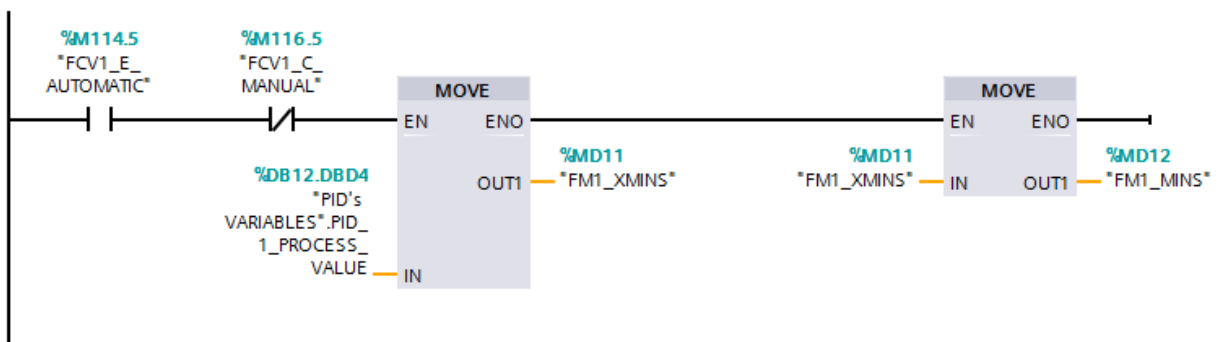
Let the valve position to 0% if the FCV1 when it recives de "OSERVICE" or "RESET" order and if FCV1 is in "ALARM" state.



- This segment is to send the real number generated by PID to the Flow Meter when the system is in automatic mode.

**Segmento 7: MOVE FM1**

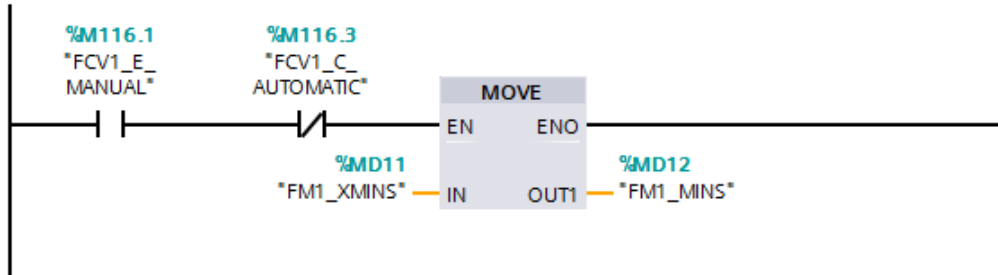
Move the input signal (X) to an internal signal.



- This segment is to send the set point to the Flow Meter when the system is in manual mode.

▼ **Segmento 8: MOVE FM1**

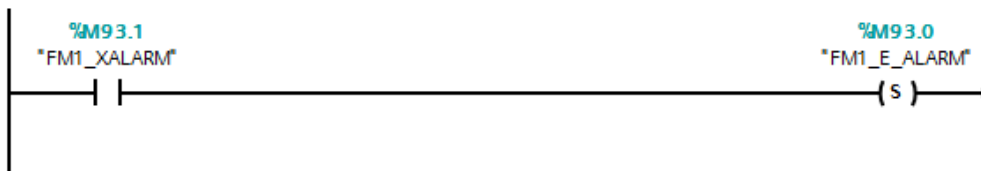
Move the input signal (X) to an internal signal when the system is in manual mode.



- This segment is to simulate an alarm signal of Flow Meter that comes from the plant.

▼ **Segmento 9: FM1 ALARM STATE ON**

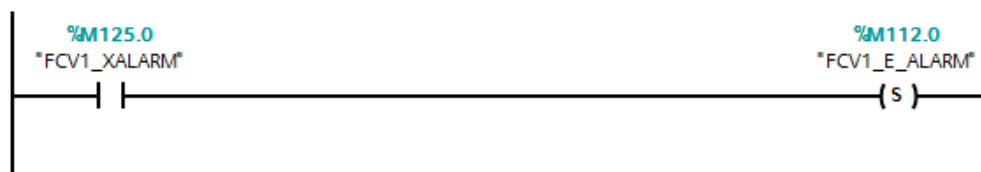
If the PLC receives an alarm signal (X), It puts the FM1 in "ALARM" state.



- This segment is to simulate an alarm signal of Flow Control Valve that comes from the plant.

▼ **Segmento 10: FCV1 ALARM STATE ON**

If the PLC receives an alarm signal (X), It puts the FCV1 in "ALARM" state.

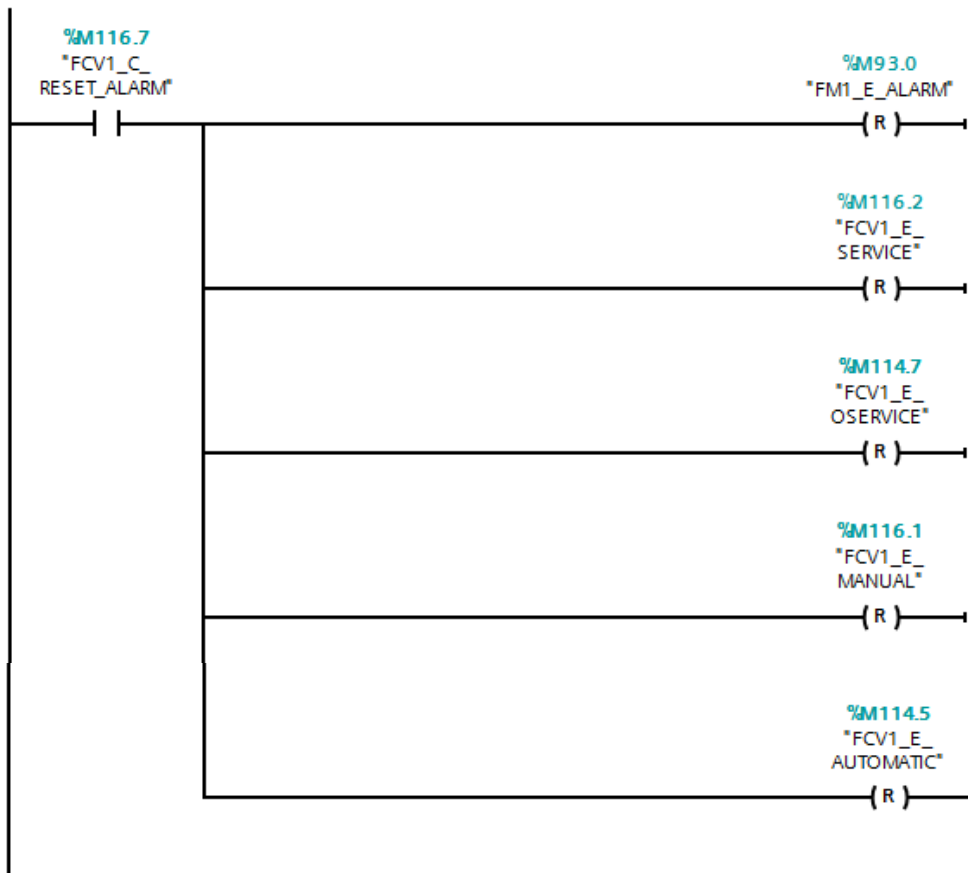




- The last segment is to let all the system in reset.

▼ **Segmento 11: ALARM STATE OFF AND RESET SYSTEM**

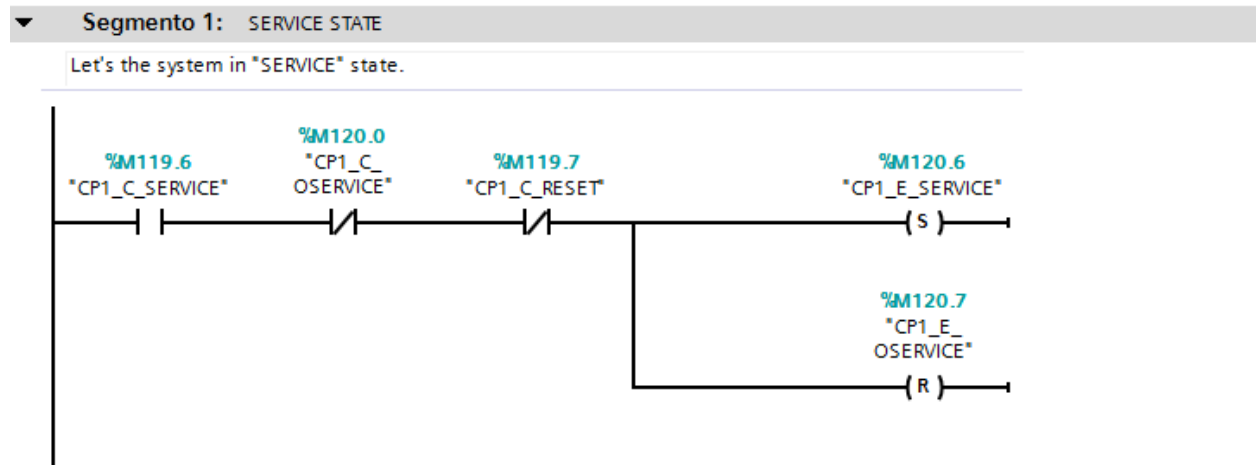
If the "RESET ALARM" order from the system control is activated, all states of the system will be reset.



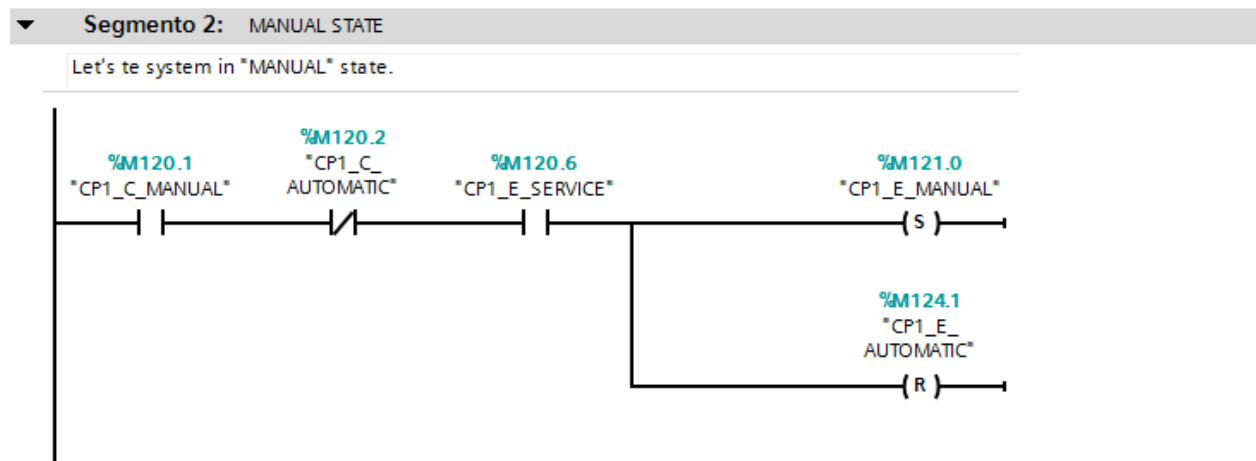
## i.iv Element (group) D

This element is basically the program to control the Centrifugal Pump 1 (CP1) and Chlorine Dispenser Pump 1(CDP1). To control these elements, is necessary the control of the respective FCVs (FCV2 to CP1 and FCV8 to CDP1) and FMs (FM2 to CP1 and FM8 to CDP1).The CP1 controls the air flow that injected in the preliminary stage, and CDP1 controls the chlorine flow injected in the tertiary stage. To show the program it will use the CP1, but it could use the CDP1

- The first segment is to let the system in service state.



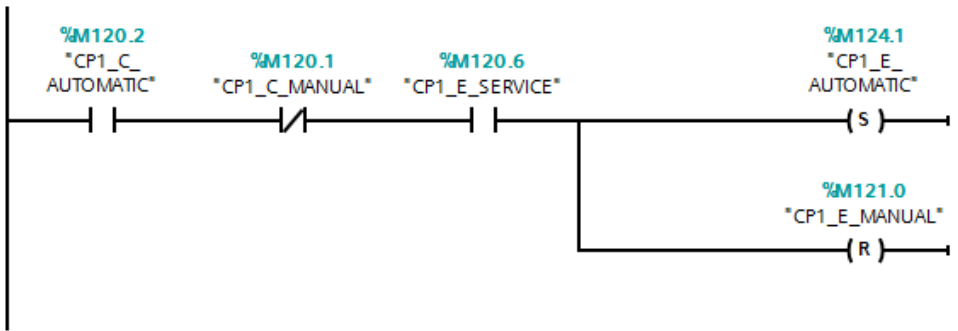
- The second segment is to choose the manual mode.



- The third segment is to choose the automatic mode.

▼ **Segmento 3: AUTOMATIC STATE**

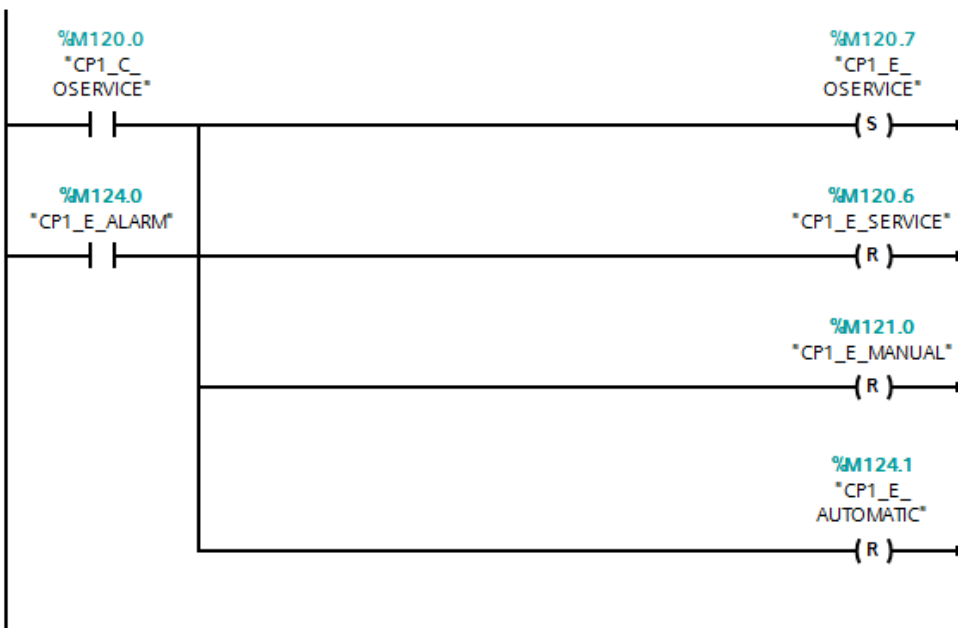
Let's te system in "AUTOMATIC" state.



- The fourth segment is to let the system “OSERVICE” state and without manual or automatic state, that happens when the command “OSERVICE” is sent or when the “ALARM” state is set.

▼ **Segmento 4: OUTSERVICE STATE**

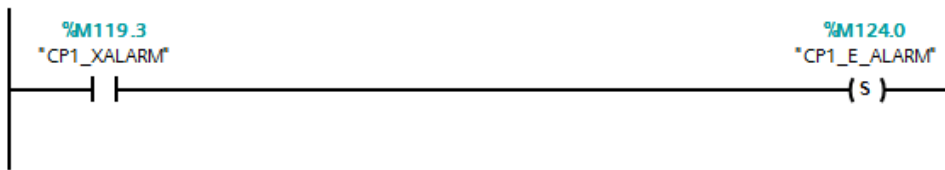
Let's te system in "OUTSERVICE" state.



- The next segment is to simulate an alarm input from the plant.

▼ **Segmento 5: ALARM STATE**

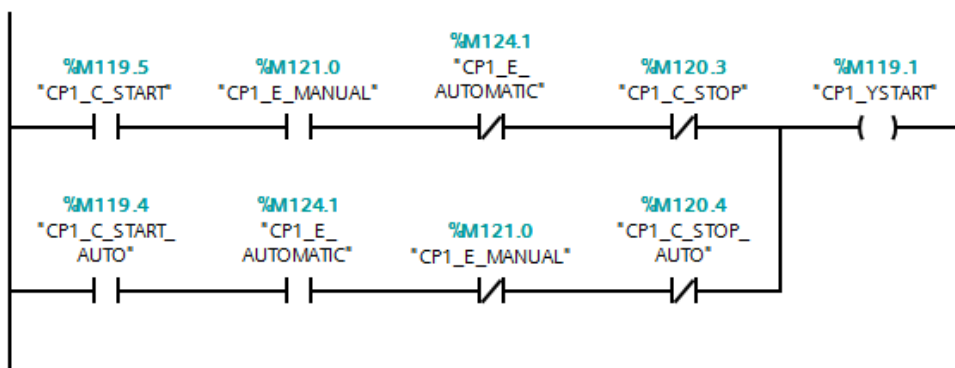
When the input "ALARM" signal is enable, the system puts in "ALARM" state.



- The sixth segment is to put send the "START" order to the plant from the PLC, in automatic or manual mode.

▼ **Segmento 6: OUTPUT START SIGNAL**

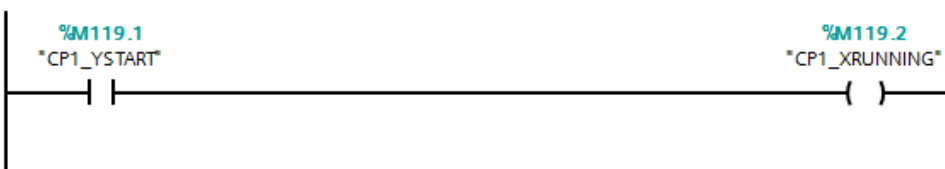
When the output signal of "START" is sended to the plant, the CP1 starts to work.



- The next segments are to send the "RUNNING" order from the CP1/CDP1 (plant) to the PLC. First the PLC sends to the CP1/CDP1 the "START" order, and then, the order from CP1/CDP1 to PLC is sent. This segment it is just to make the simulation.

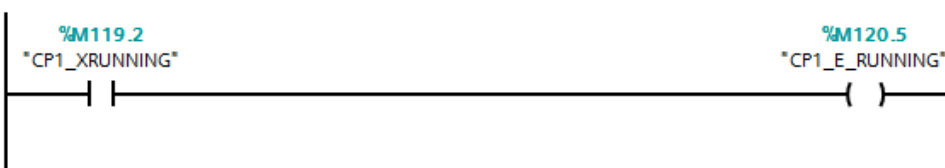
▼ **Segmento 7: RUNNING STATE**

The "STARTS" output signal, enables the input signal of "RUNNING".

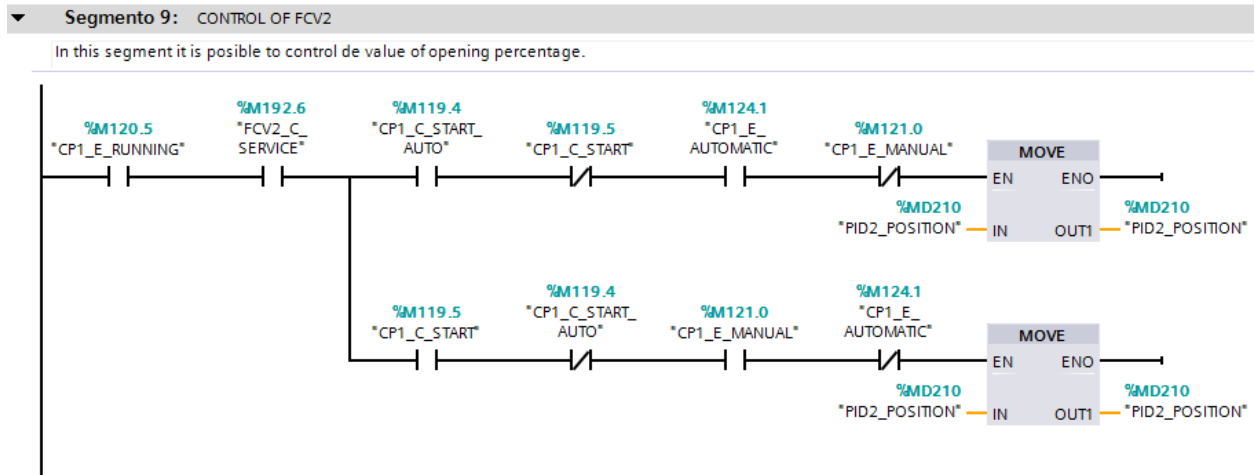


▼ **Segmento 8: RUNNING STATE**

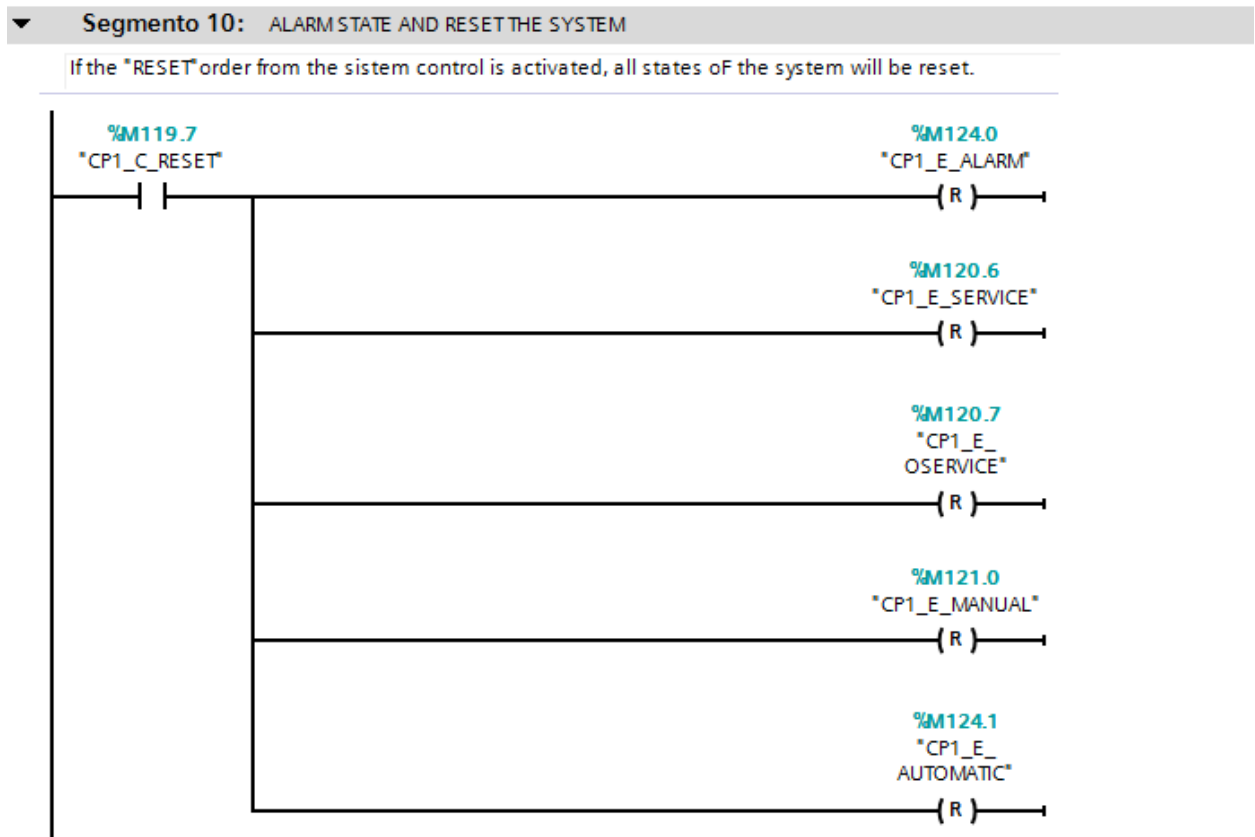
The input signal of "RUNNING" enables the "RUNNING" state, and now, the CP1 is working.



- When the CP1/CDP1 is in “RUNNING” state, it is necessary to set the FCV2/FCV8 in “SERVICE” state to send the Set Point opening of the FCV2/FCV8.



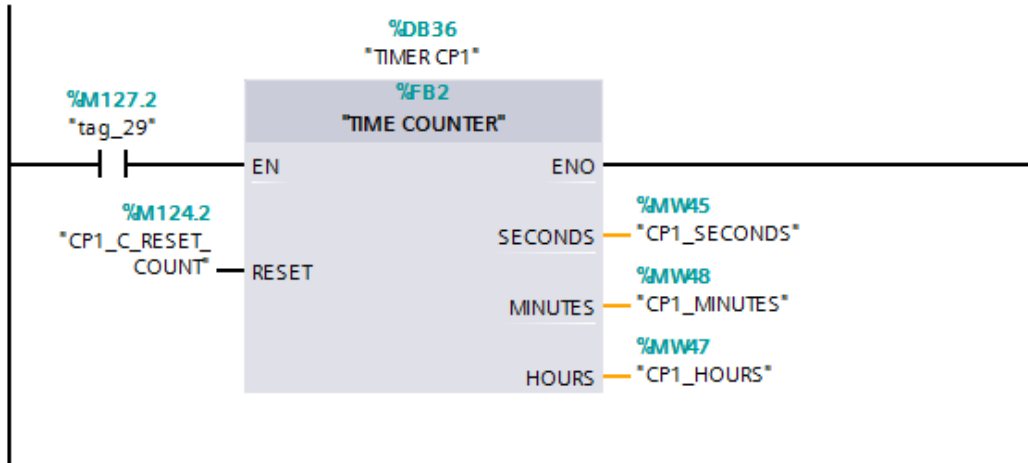
- The tenth segment is to let all the system in reset.



- The last segment is the time counter (TIME COUNTER [FB2]) used to count how many time the CP1/CDP1 is in “RUNNING” state.

▼ **Segmento 11: TIME COUNTER**

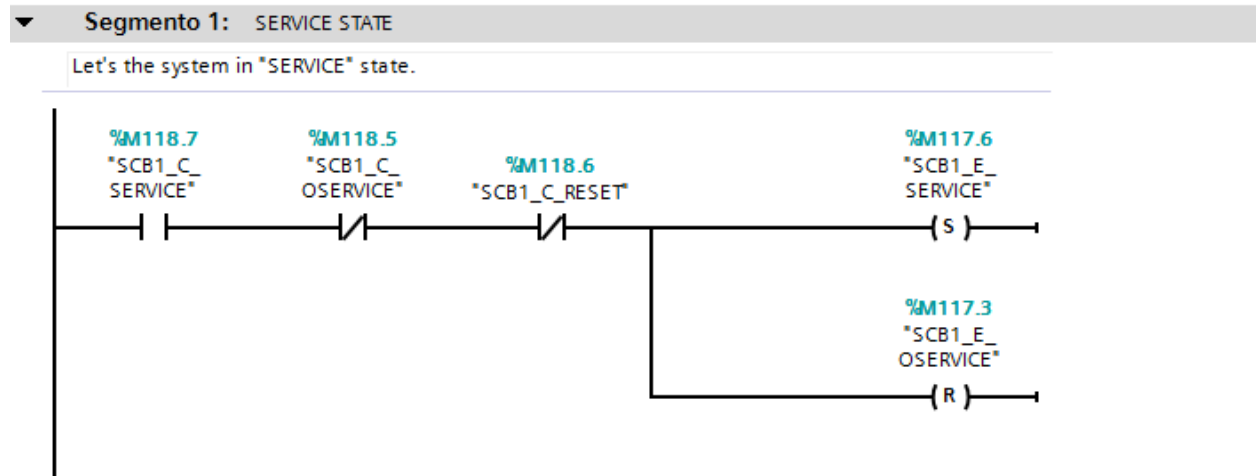
When the CP1 is in "RUNNING" state, this counter, count the time in seconds, minutes and hours.



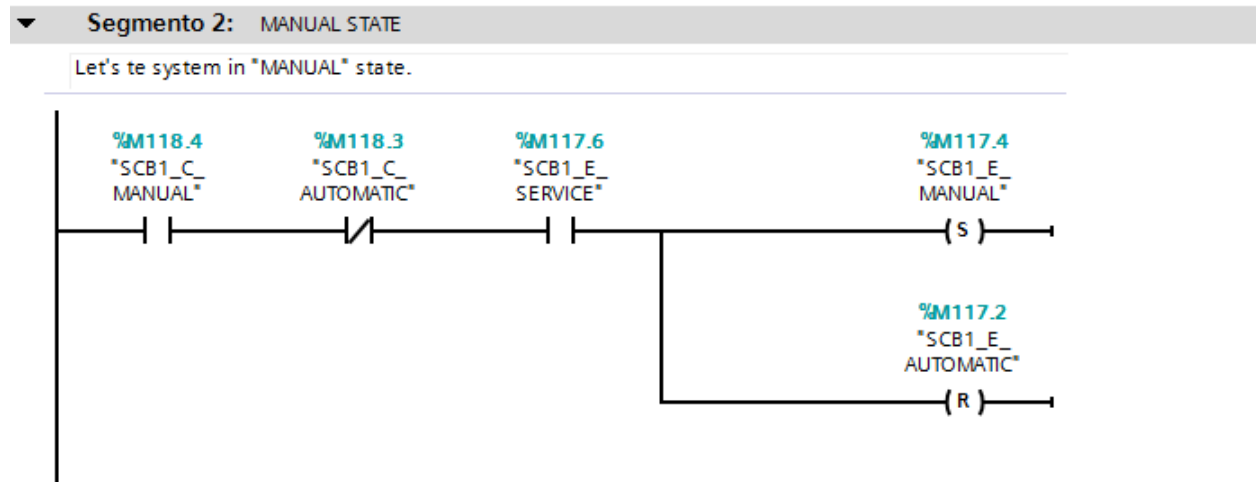
## i.v Element (group) E

The element E is to control the Side Channel Blower 1 (SCB1), to do that is necessary the control of FCV5 because it will control the oxygen flow injected to the water in the secondary stage. It is some similar with the CP1 program, but more complex.

- The first segment is to let the SCB1 system in service state.



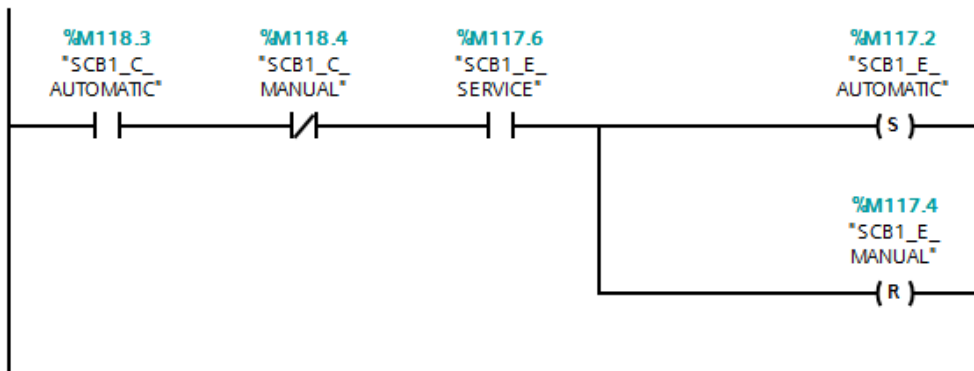
- The second segment is to choose the manual mode of SCB1.



- The third segment is to choose the automatic mode of SCB1.

▼ **Segmento 3: AUTOMATIC STATE**

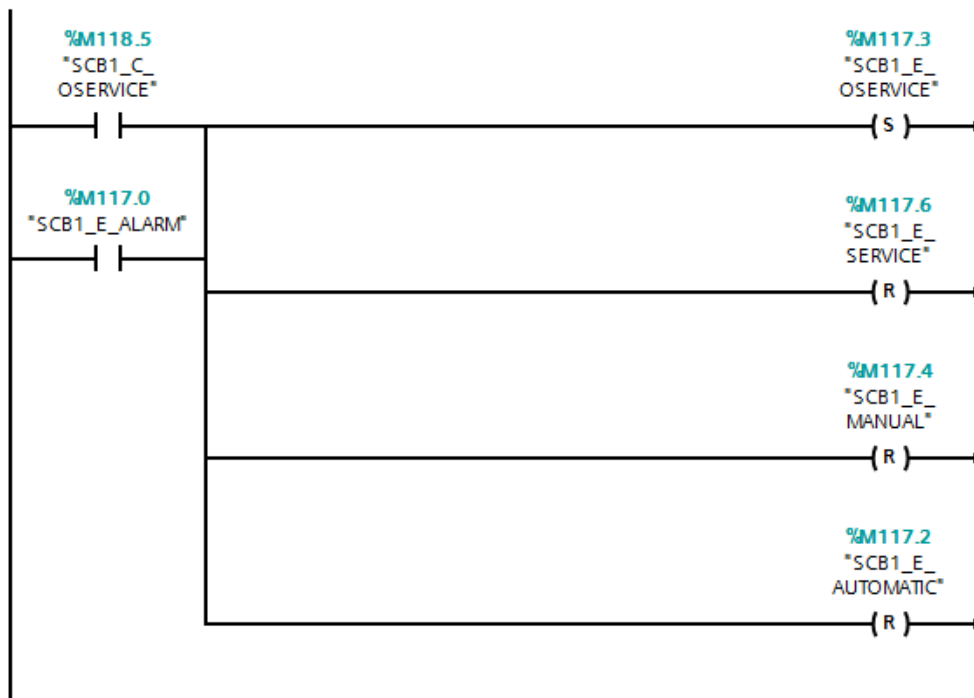
Let's te system in "AUTOMATIC" state.



- The fourth segment is to let the system in “OSERVICE” state and without ”MANUAL” or ”AUTOMATIC” states, that happens when the command “OSERVICE” is sent or when the “ALARM” state is set.

▼ **Segmento 4: OU SERVICE STATE**

Let's te system in "OUTSERVICE" state.

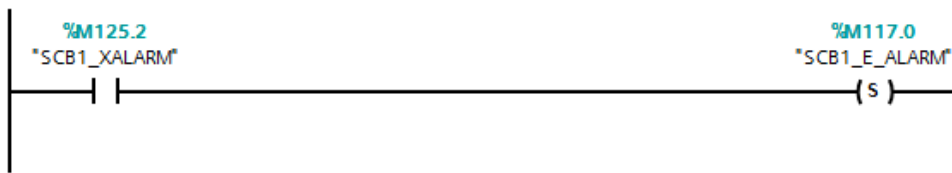




- The fifth segment is to simulate the input alarm from the plant.

**Segmento 5: ALARM STATE**

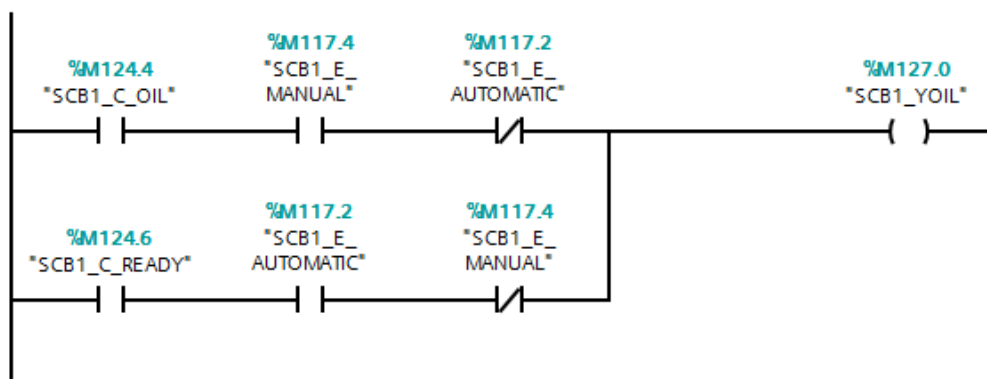
When the input "ALARM" signal is enable, the system puts in "ALARM" state.



- This segment is to send the "oil" signal to the SCB1 in "MANUAL" or "AUTOMATIC" (ready) state.

**Segmento 6: OUTPUT "ON OIL PUMP" SIGNAL**

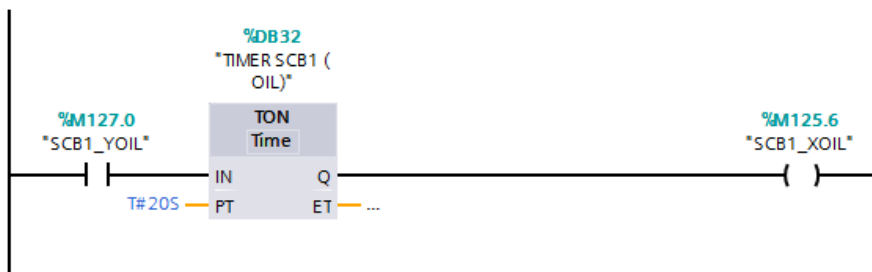
Activation of the oil circuit of the SCB1.



- The seventh and eighth segments are for simulate of the heating of the oil in SCB1. The seventh is to simulate the time and the eighth is to simulate the signal that comes from the plant. When the oil it is ready to work the "OILOK" state is on.

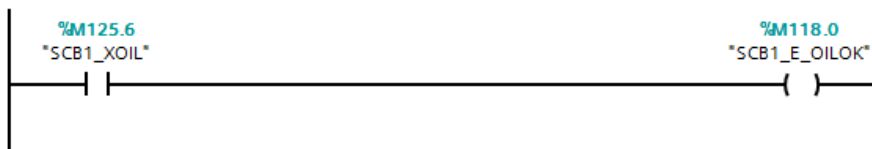
**Segmento 7: INPUT "OK OIL PRESSURE" SIGNAL**

Timer to simulate the heating of the oil system in SCB1.



**Segmento 8: OIL PRESSURE OK STATE**

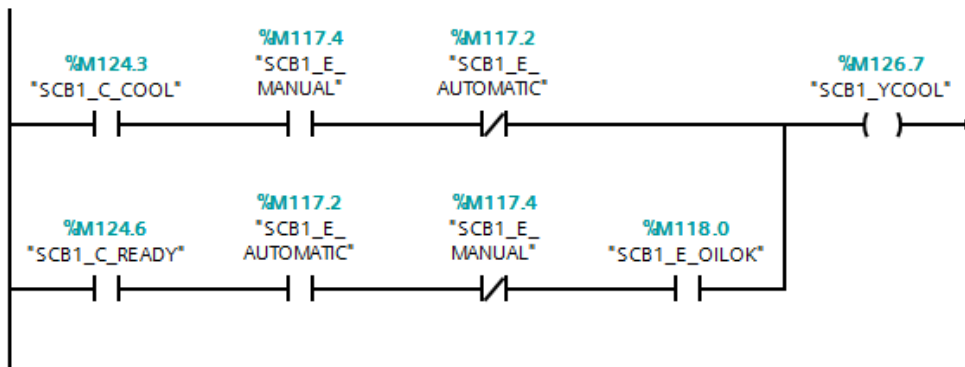
When the input signal of oil arrives to the PLC, It's mean that the Oil system it's ready to work.



- This segment is to send the “COOL” signal to the SCB1 in “MANUAL” or “AUTOMATIC” (ready) mode. When the preparation of the SCB1 is automatic, it needs the “OILOK” state to continue with the cool process.

**Segmento 9: INPUT "OK COOL CIRCUIT" SIGNAL**

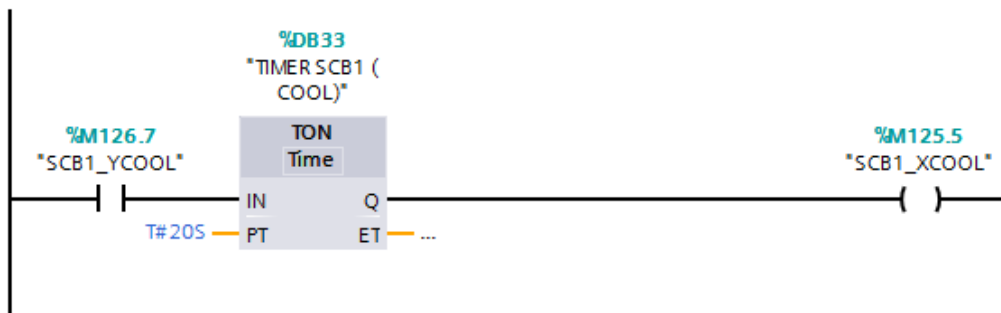
Activation of the cool (refrigeration) circuit of the SCB1.



- The tenth and eleventh segments are for simulate the cooling process of the SCB1. The tenth is to simulate the time and the eleventh is for simulate the signal that comes from the plant. When the cooling system it is ready to work the “COOLOK” state is set.

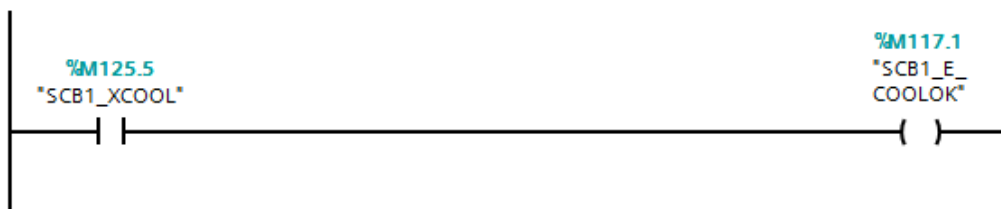
**Segmento 10: INPUT "OK COOL CIRCUIT" SIGNAL**

Timer to simulate the preparation of the refrigeration system of SCB1.



**Segmento 11: COOL CIRCUIT OK STATE**

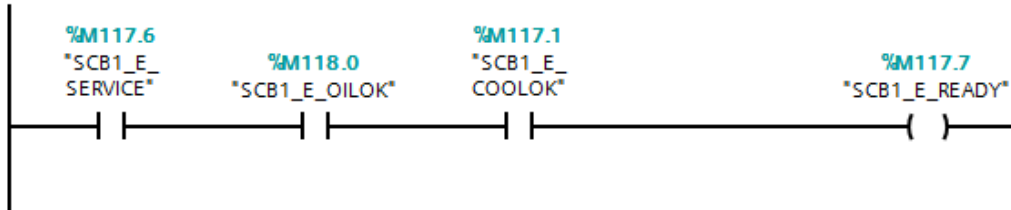
▶ When the input signal of cool arrives to the PLC, it's mean that the refrigeration system is ready to ...



- When the SCB1 system is in “SERVICE” state, “OILOK” state and “COOLOK” state, the system set it in “READY” state. Now, the SCB1 can start to work correctly.

▼ **Segmento 12: READY STATE**

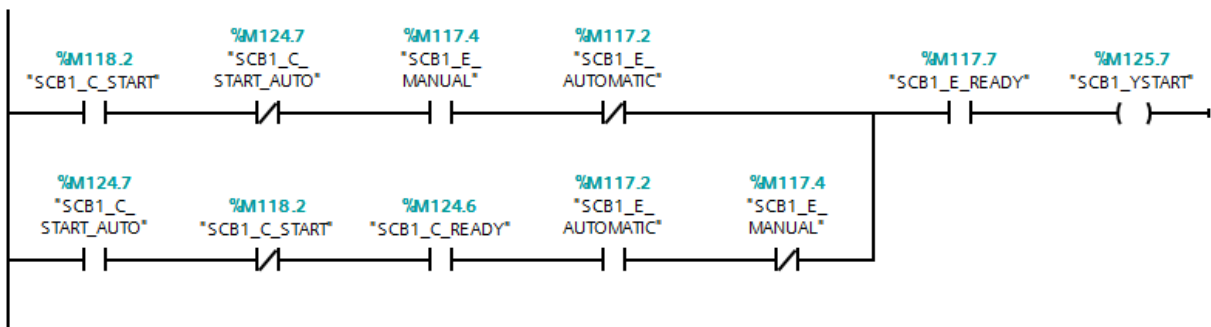
When the preliminar conditions (oil, service and cool) are ready to work, the SCB1 is “READY” to start.



- The next segment is to send the “START” signal to the plant from the PLC, in “AUTOMATIC” or “MANUAL” mode.

▼ **Segmento 13: OUTPUR START SIGNAL**

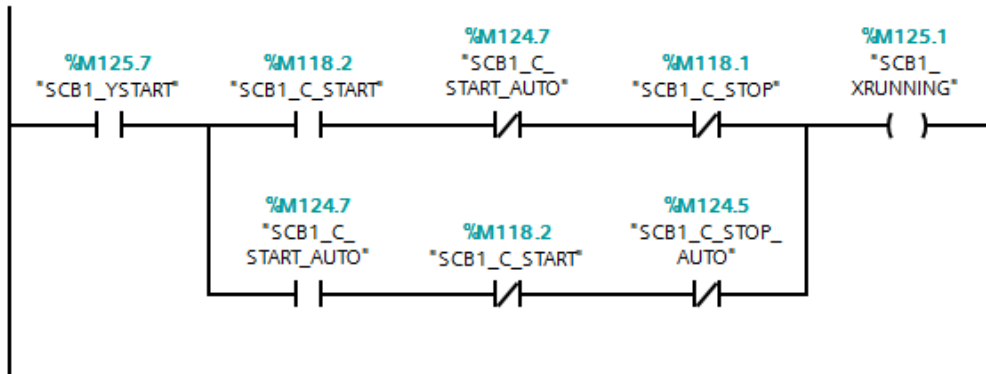
When the system is “READY” and the “START/START\_AUTO” order is sended, It starts to work.



- The next segments are to send the “START” signal from the PLC to the plant (SCB1). Then, the SCB1 send to the PLC the “XRUNNING” signal and finally, the “XRUNNING” signal from the PLC sets the “RUNNING” state of the SCB1.

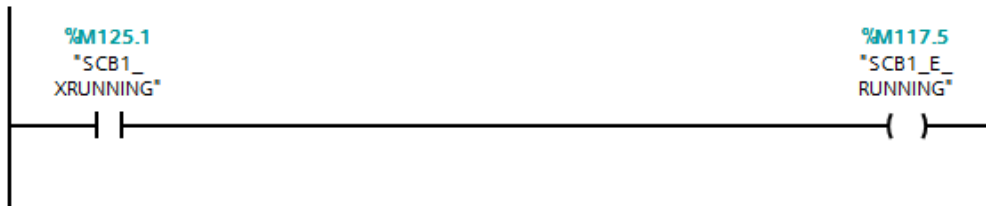
▼ **Segmento 14: RUNNING STATE**

The "STARTS" output signal, enables the input signal of "RUNNING".



▼ **Segmento 15: RUNNING STATE**

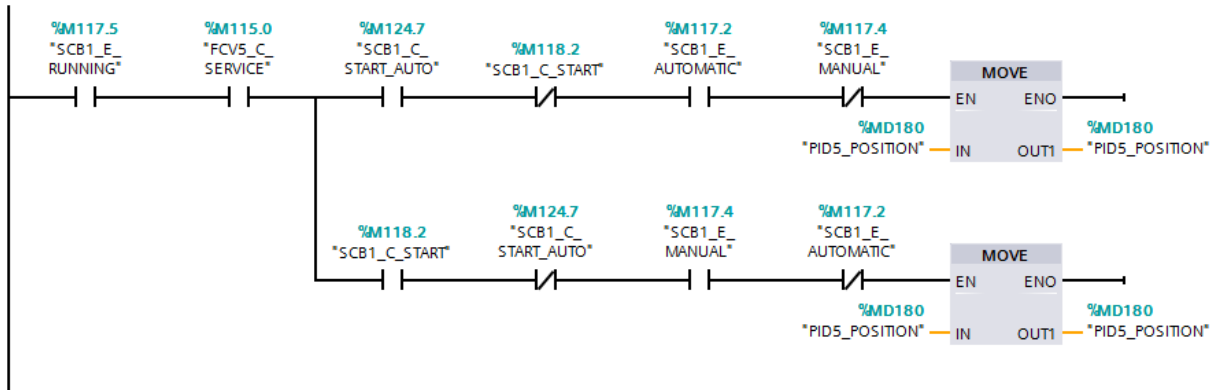
The input signal of "RUNNING" enables the "RUNNING" state, and now, the SCB1 is working.



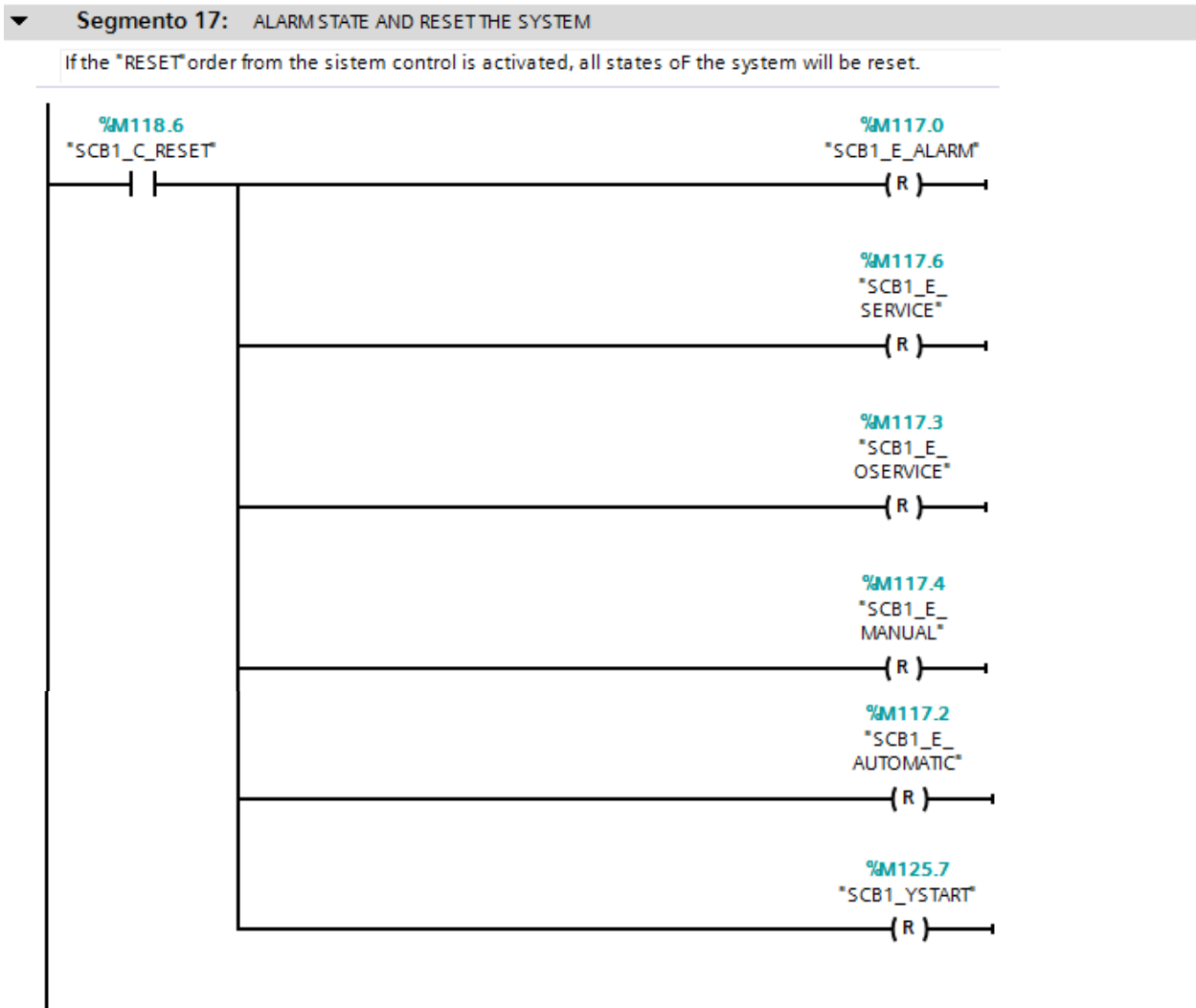
- When the SCB1 is in “RUNNING” state, it is necessary to let the FCV5 in “SERVICE” state to send the Set Point opening of the FCV5.

▼ **Segmento 16: CONTROL OF FCV5**

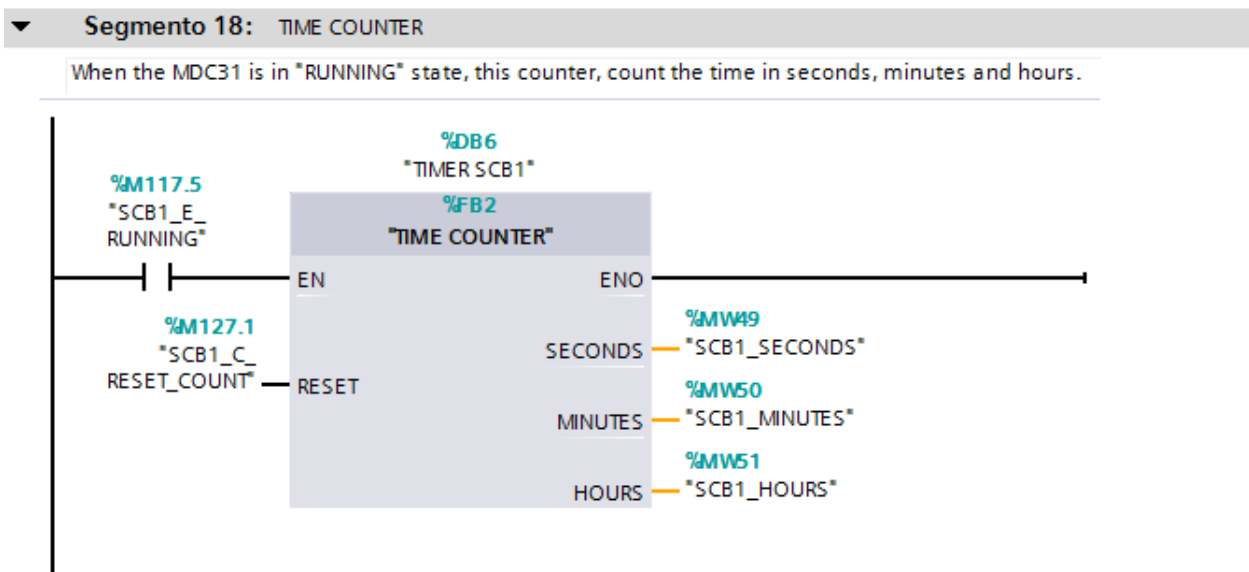
In this segment it is possible to control de value of opening percentage.



- The seventeenth segment is to let all the system in reset.



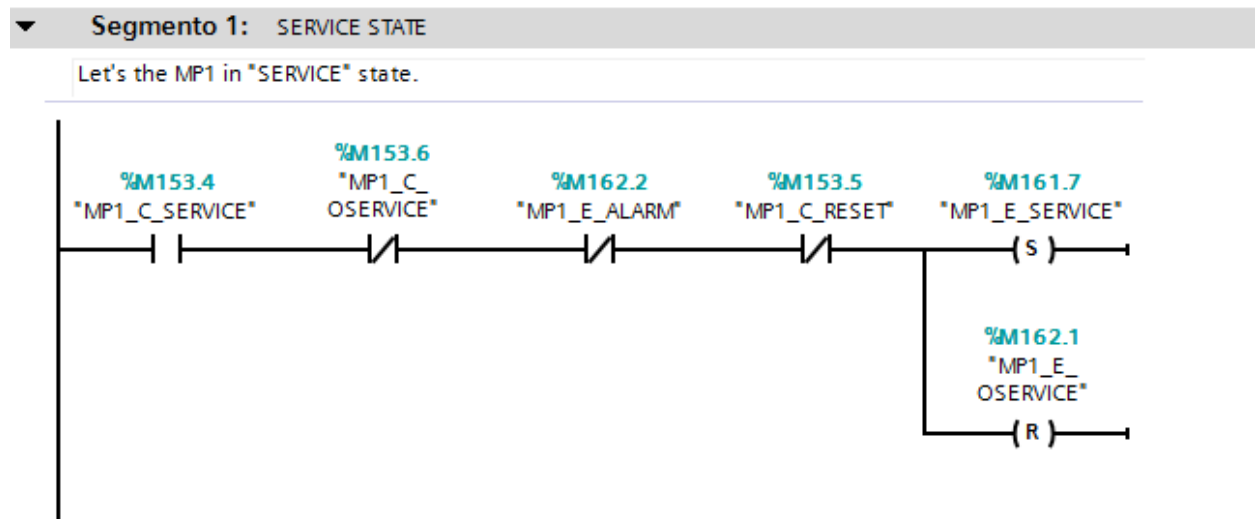
- The last segment is the time counter (TIME COUNTER [FB2]) used to count how many time the SCB1 is in "RUNNING" state.



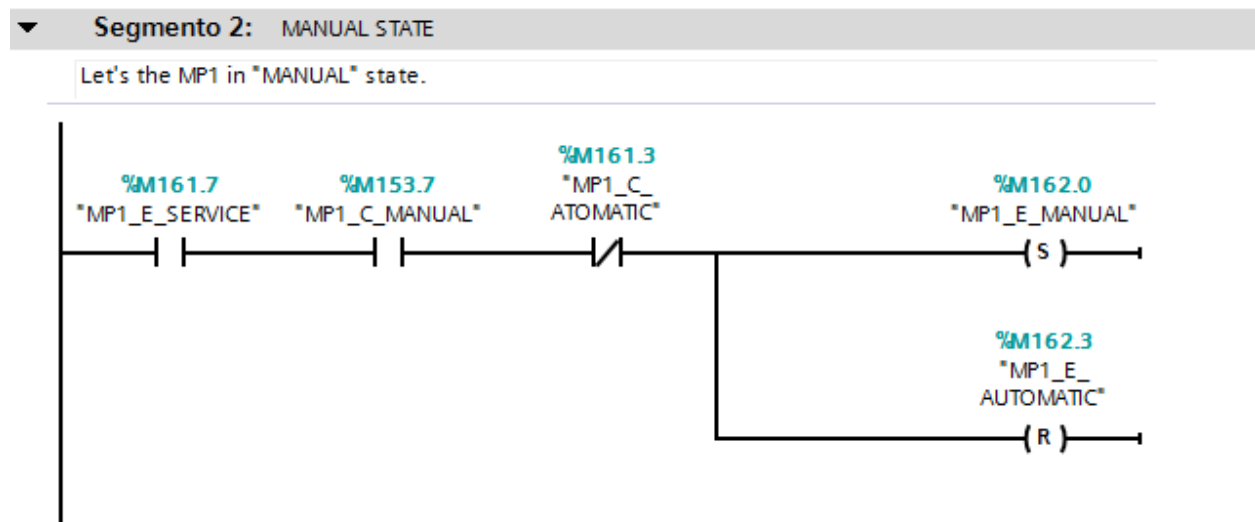
## i.vi Element (group) F

This element group is composed by the Mud Pumps (MP1, MP2 and MP3) with her respective Mud Sensors (MS1, MS2 and MS3) and by the Fat Pump (FP1) and her respective Fat Sensor (FS1). To explain the functioning of this type of element the program of MP1 will be used.

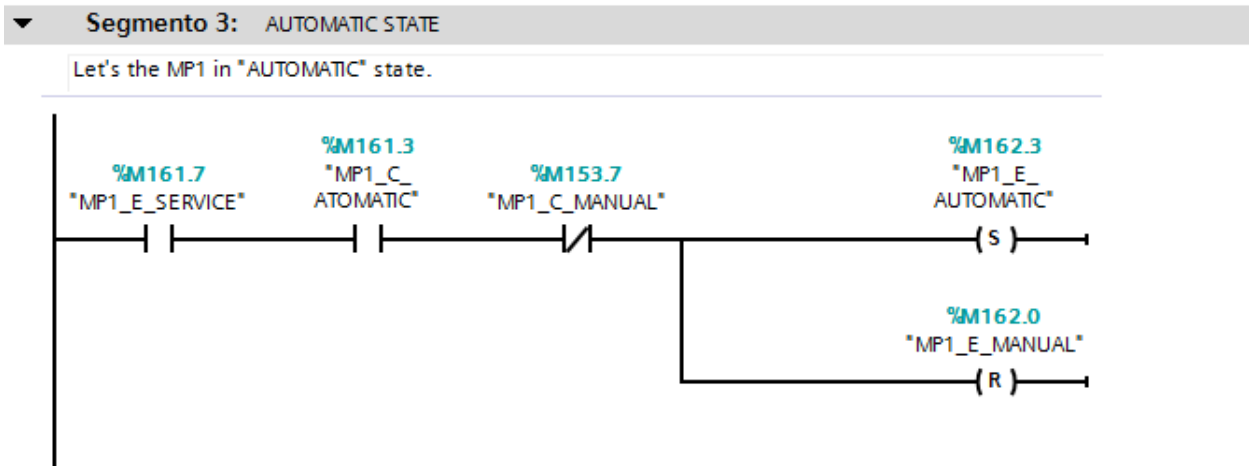
- The first segment is to let the system in “SERVICE” state.



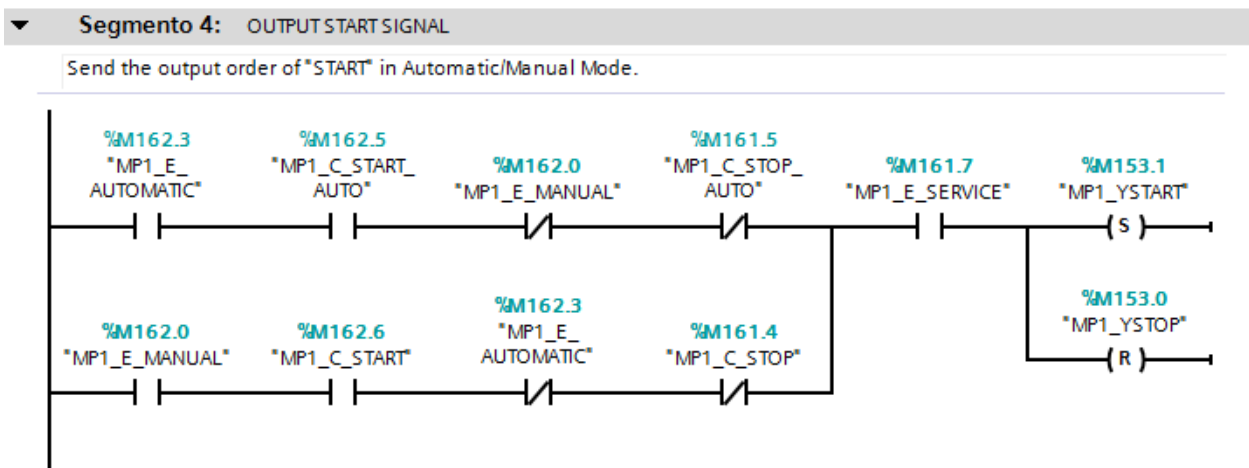
- The second segment is to let the system in “MANUAL” state.



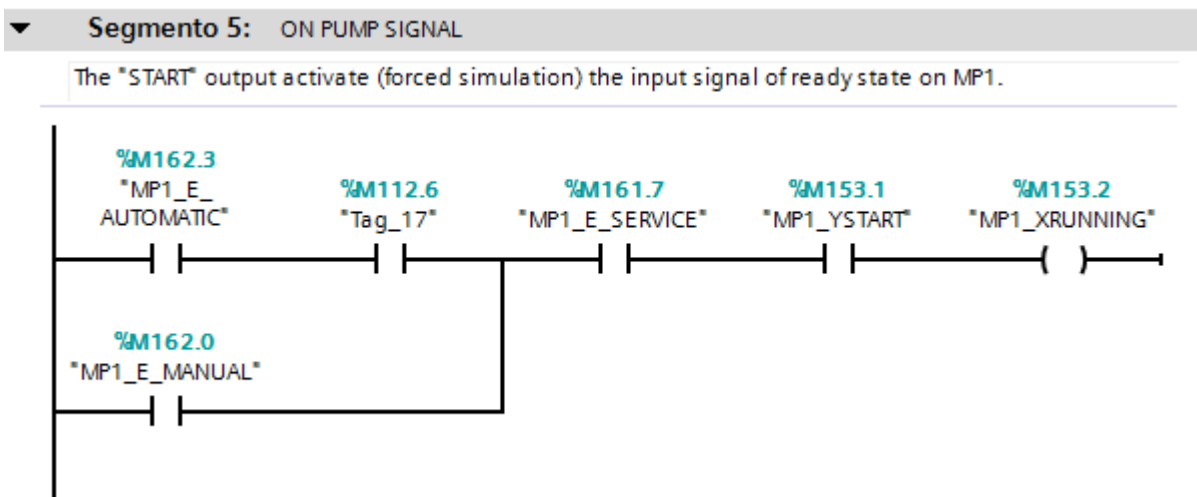
- The third segment is to let the system in “AUTOMATIC” state.



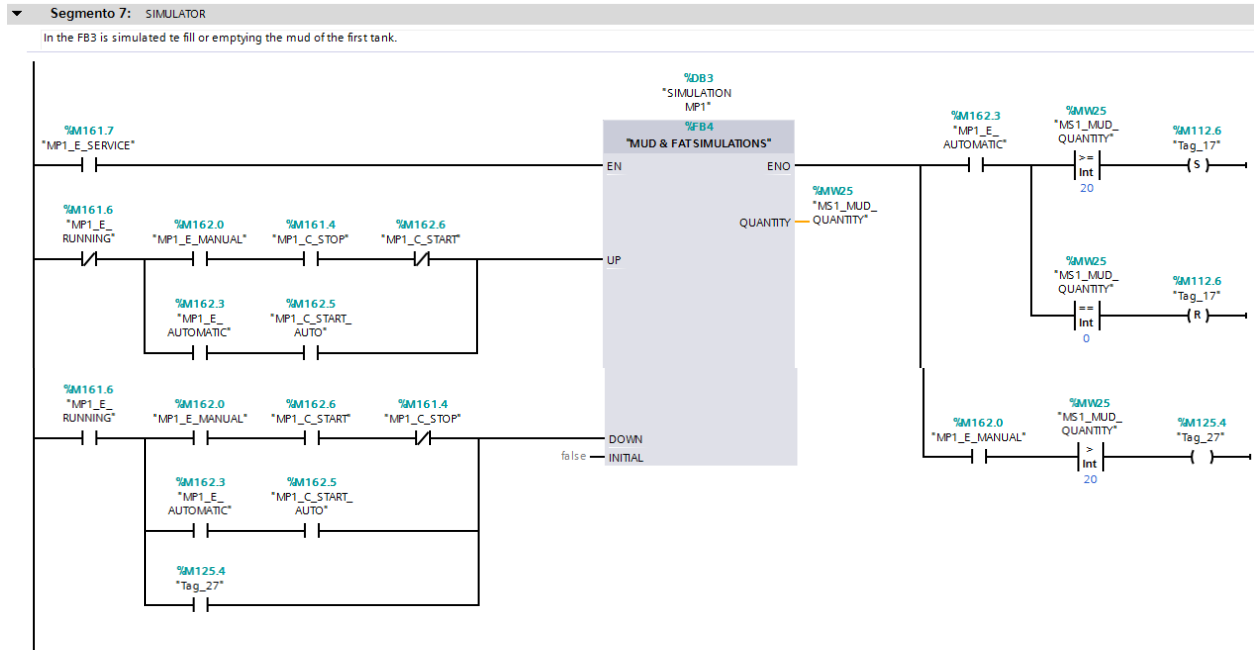
- The next segment is to send the “START” signal to the plant in “MANUAL” or “AUTOMATIC” mode.



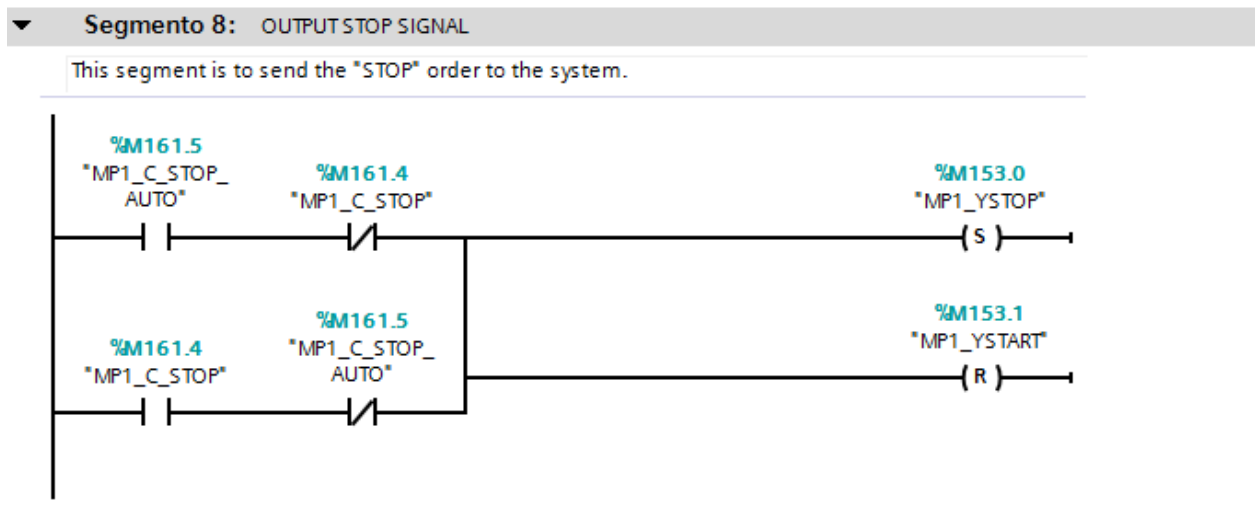
- The fifth segment is to send from the plant to the PLC the “RUNNING” signal in automatic or manual mode.



- The sixth segment is where the mud/fat level is generated (by simulation), depending of the work mode of the system (automatic or manual) the simulator (FB4) will work different. When the system is in “AUTOMATIC” state, the actuator will be active every 20 seconds to simulate the mud/fat removing. When the system is in "MANUAL" state, the mud/fat removing must be activated manually.



- The eighth segment is to send the “STOP” signal to the plant in “MANUAL” or “AUTOMATIC” mode.

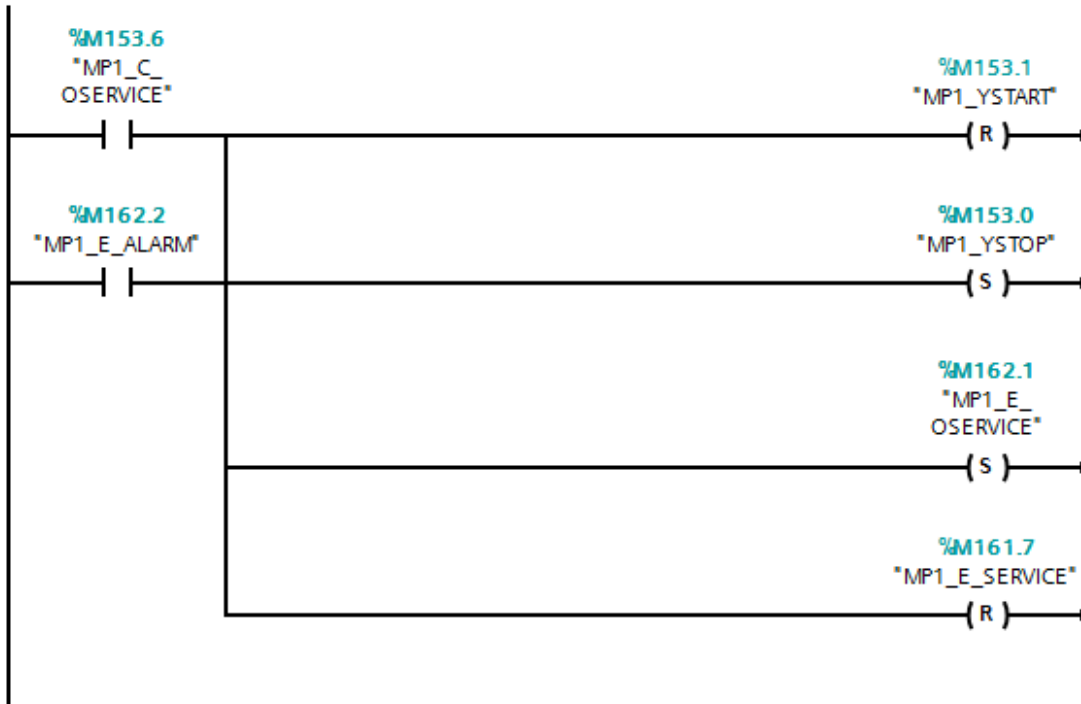




- The ninth segment is to let the system in “OSERVICE” state and without ”MANUAL” or ”AUTOMATIC” states, that happens when the command “OSERVICE” is sent or when the “ALARM” state is set.

▼ **Segmento 9: OUTPUT STOP SIGNAL**

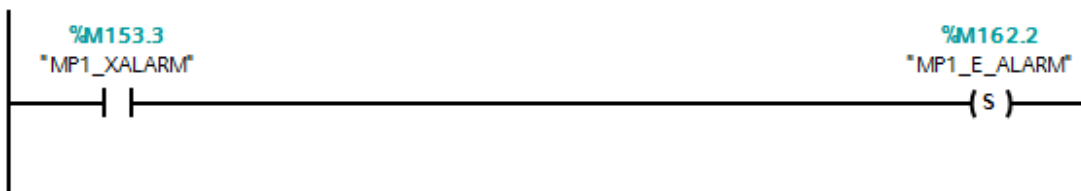
States, orders and signals whe let the MP1 in "OSERVICE" state. The "START" order is reset too.



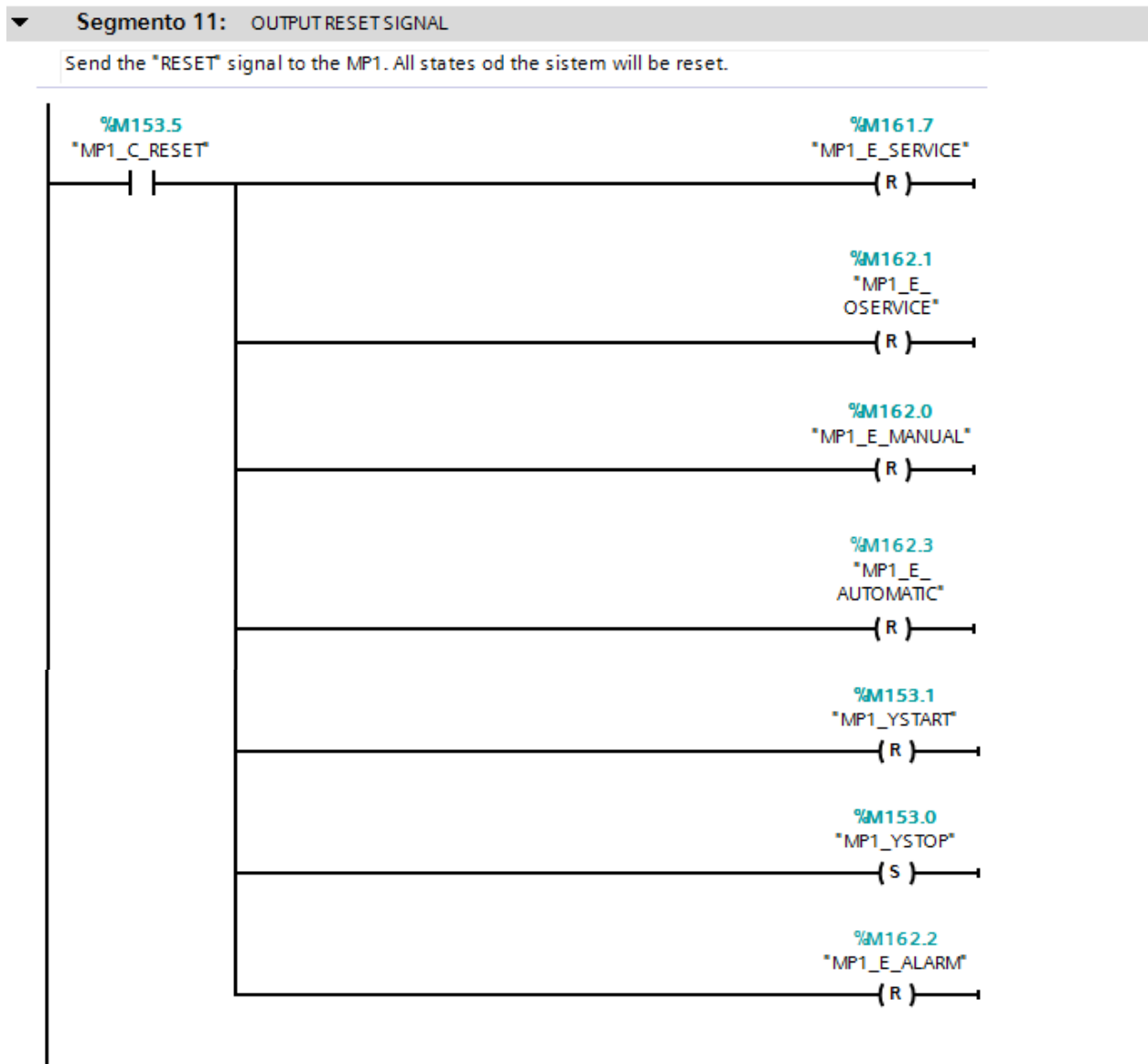
- The next segment is to simulate an alarm input from the plant.

▼ **Segmento 10: ALARM STATE**

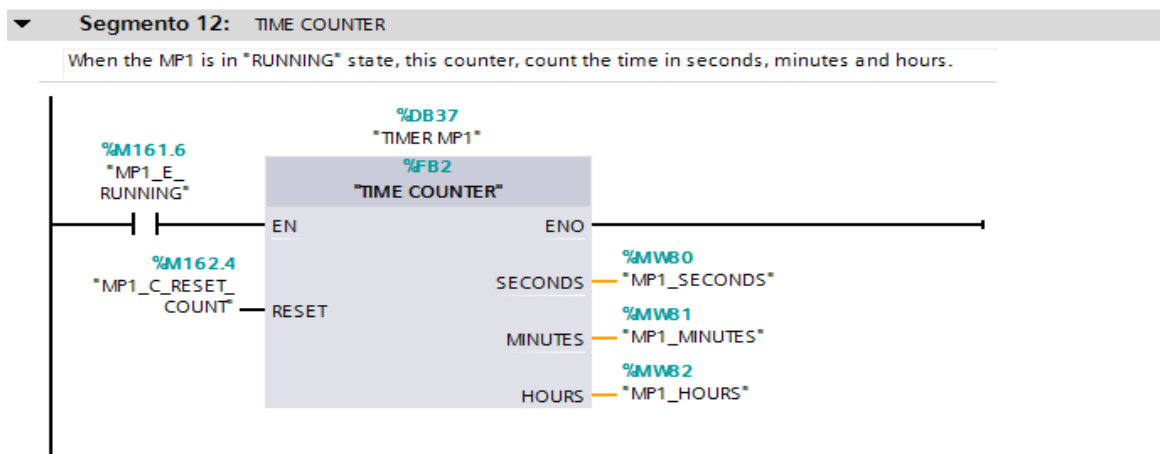
The input alarm signal puts the MP1 in "ALARM" state.



- The eleventh segment is to let all the system in reset.



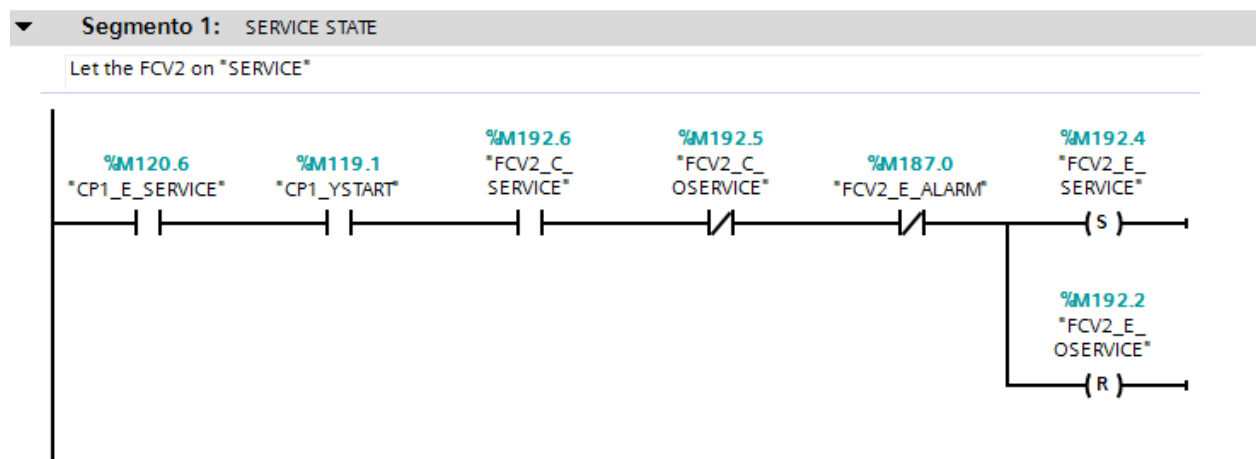
- The last segment is the time counter (TIME COUNTER [FB2]) used to count how many time the actuators (MP1, MP2, MP3 or FP1) are in "RUNNING" state.



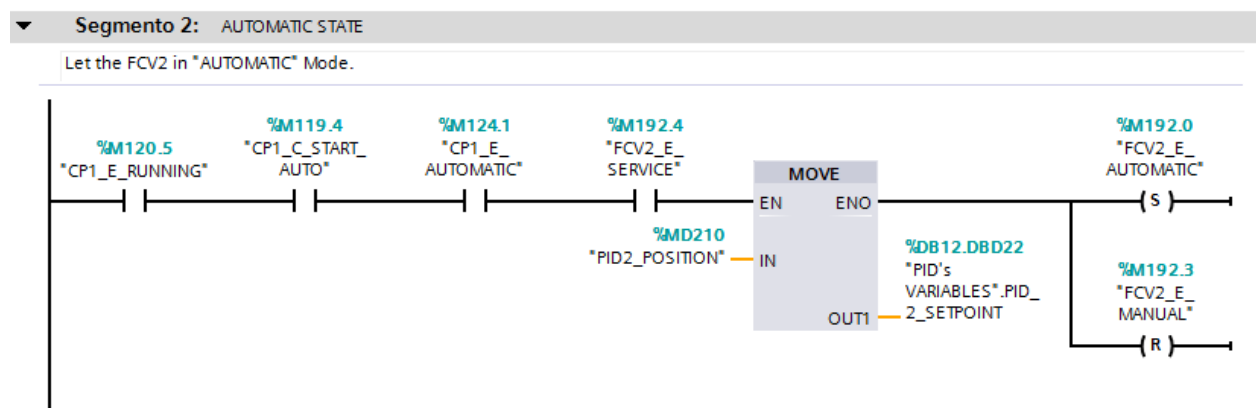
## i.vii Element (group) G

This element group is formed by the Flow Control Valve 2 (FCV2) and Flow Control Valve 8 (FCV8) and her respective Flow Meters (FM2 and FM8). To do the explanation of this element group, FCV2 controlled from the CP1 program will be used.

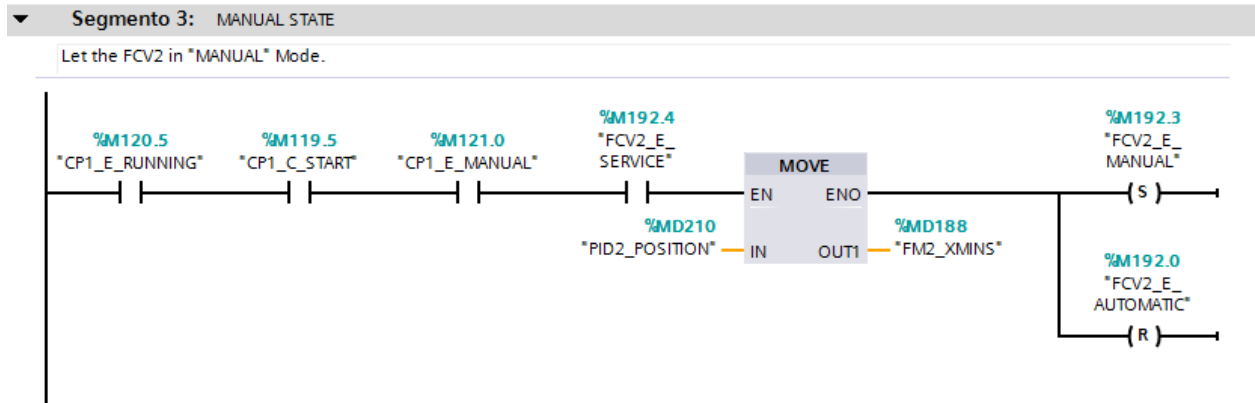
- The first segment is for let the FCV system in “SERVICE” state. To do that, it is necessary to set the SERVICE state in the control actuator (in the case of FCV2, the CP1 and in the case of FCV8 the CDP1).



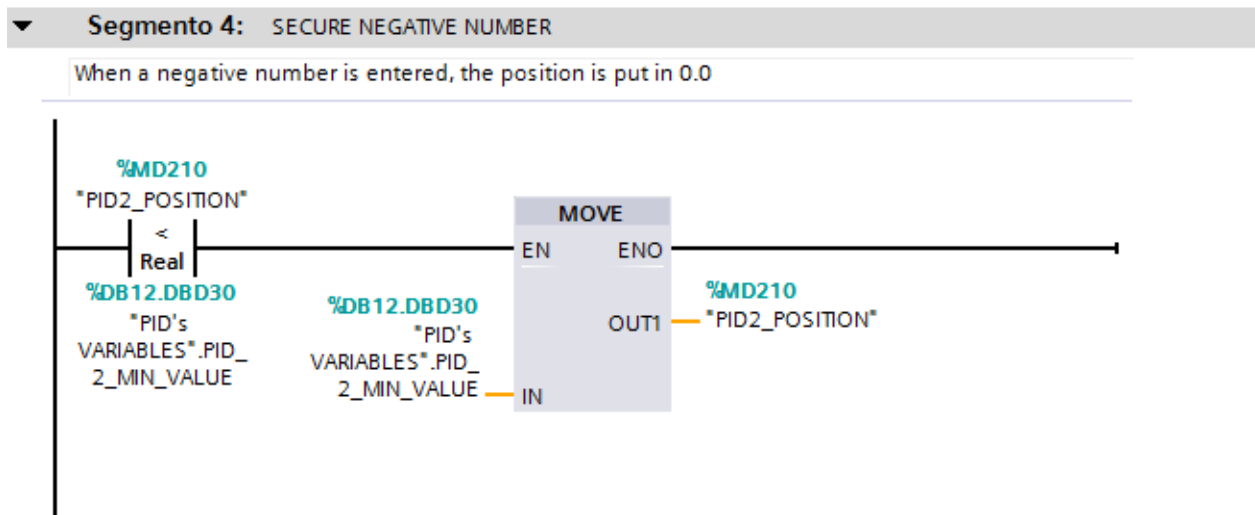
- The second segment is to choose the automatic mode of FCV and to enter the position of the valve (opening percentage). To do that, it is necessary the “RUNNING” state of the control actuator, the “START\_AUTO” command of the control actuator and the “AUTOMATIC” state of this actuator (in the case of FCV2, the CP1).



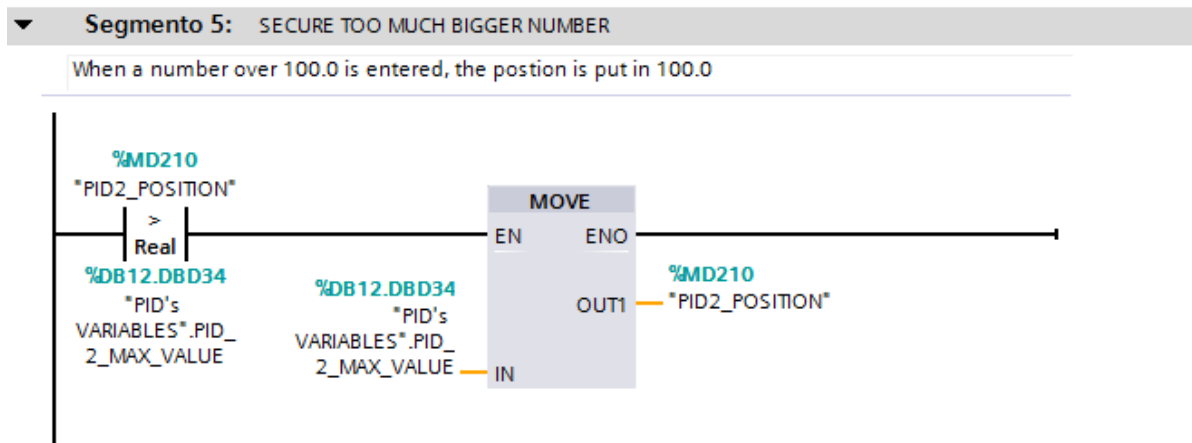
- The third segment is to choose the manual mode of FCV and to enter the position of the valve (opening percentage). To do that, it is necessary the “RUNNING” state of the control actuator, the “START” command of the control actuator and the “MANUAL” state of this actuator (in the case of FCV2, the CP1).



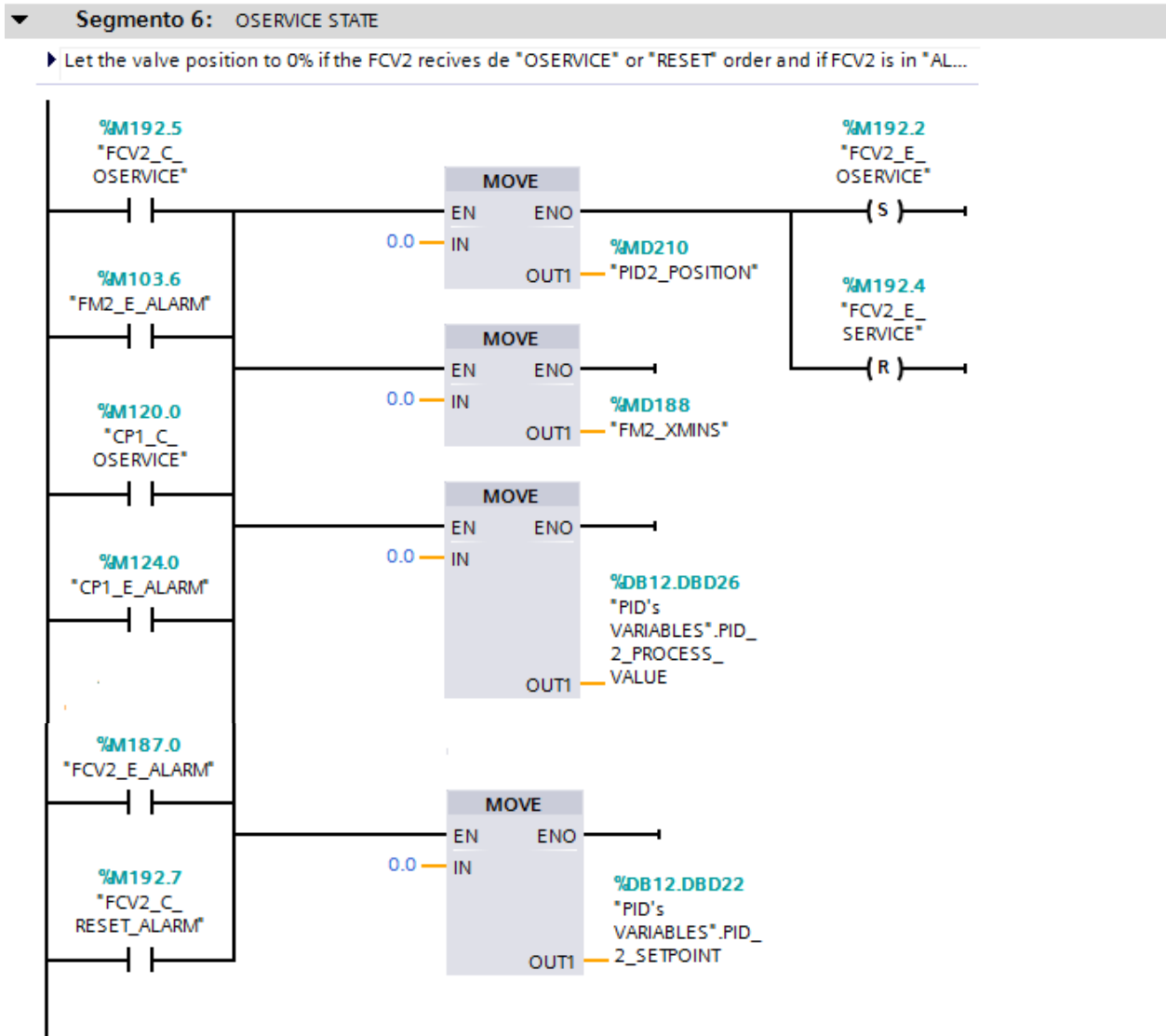
- This segment is to ensure that you will never enter a negative number to the valve.



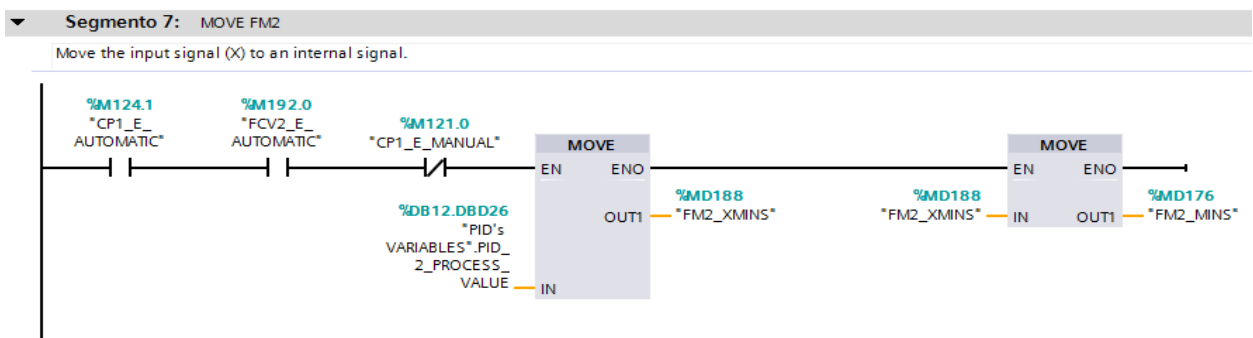
- This segment is to ensure that it will never enter a number bigger than 100 to the valve.



- The sixth segment is to let the system in "OSERVICE" state when the "OSERVICE" command is sent, when the alarm of FVC is active, when the "RESET" command is sent. when the alarm of FM is active, when the "OSERVICE" command of the controller (CP1 in the FCV2 case) is sent or when the controller is in "ALARM" state.



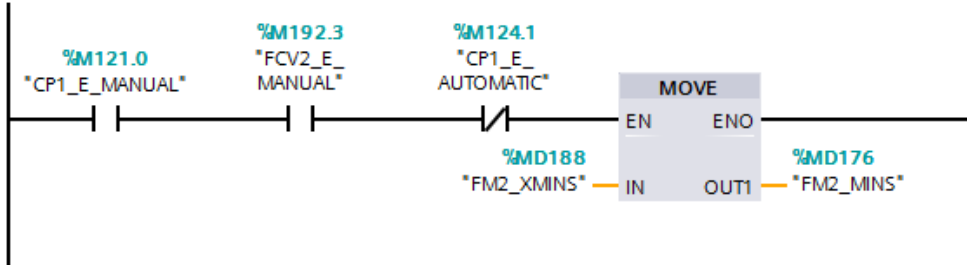
- This segment is to send the real number generated by PID to the Flow Meter when the system is in "AUTOMATIC" mode.



- This segment is to send the set point to the Flow Meter when the system is in “MANUAL” mode.

▼ **Segmento 8: MOVE FM2**

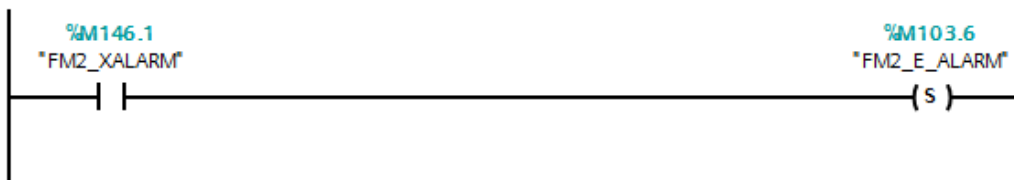
Move the input signal (X) to an internal signal.



- This segment is to simulate an alarm signal of Flow Meter that comes from the plant.

▼ **Segmento 9: FM1 ALARM STATE ON**

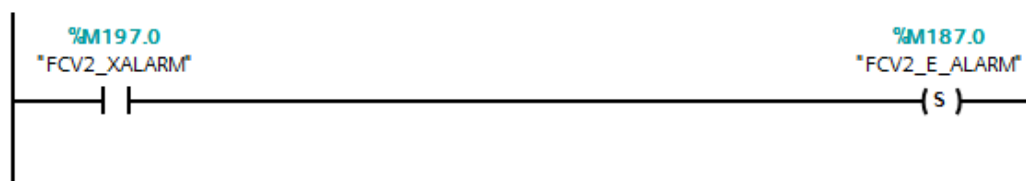
If the PLC receives an alarm signal (X), It puts the FM2 in "ALARM" state.



- This segment is to simulate an alarm signal of Flow Control Valve that comes from the plant.

▼ **Segmento 10: FCV1 ALARM STATE ON**

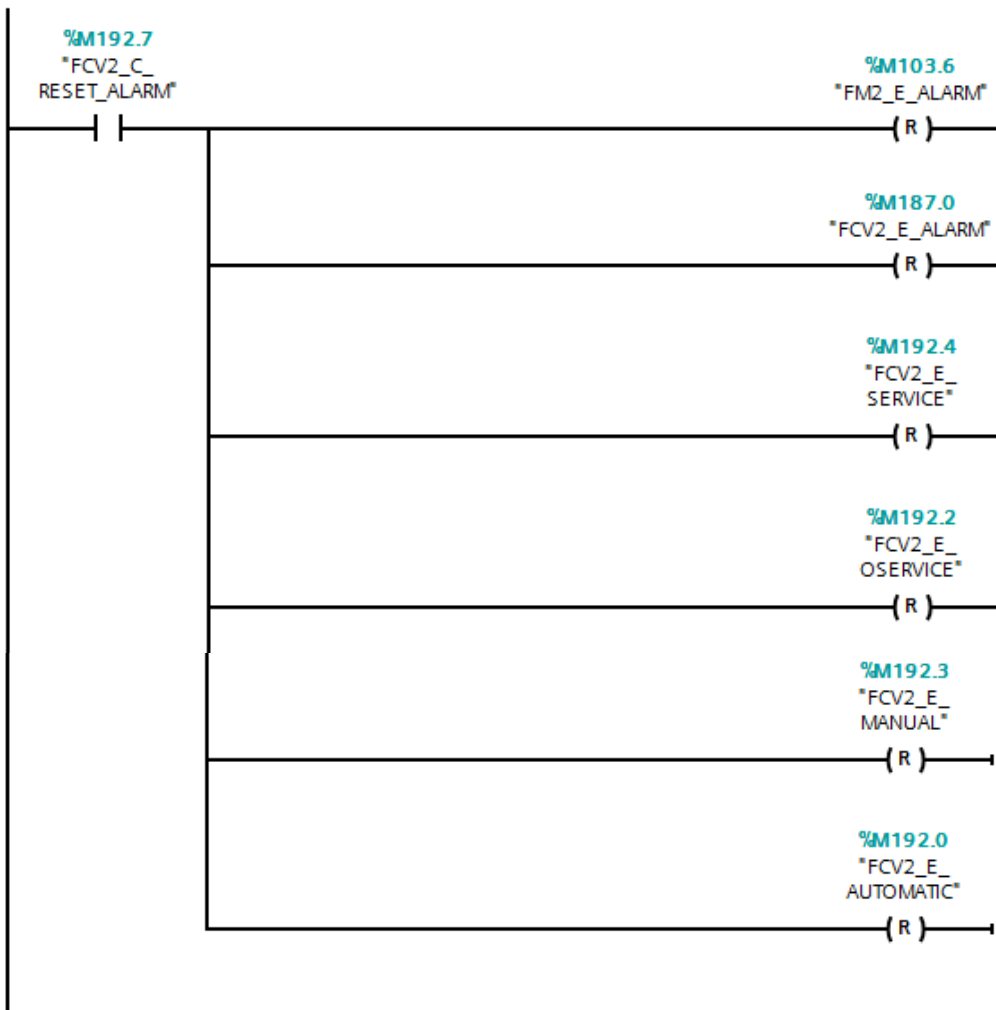
If the PLC receives an alarm signal (X), It puts the FCV2 in "ALARM" state.



- The last segment is to let all the system in reset.

▼ **Segmento 11: ALARM STATE AND RESET SYSTEM**

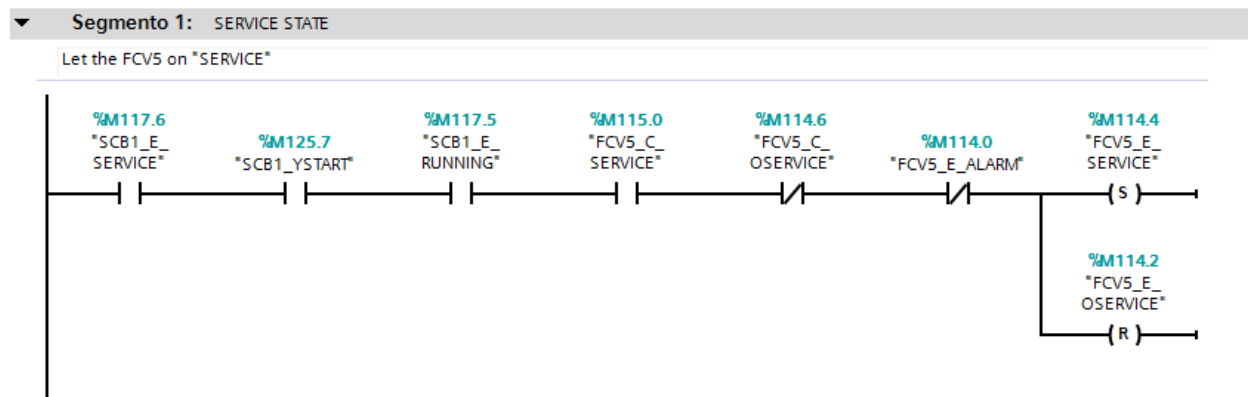
If the "RESET ALARM" order from the sistem control is activated, all states of the system will be reset.



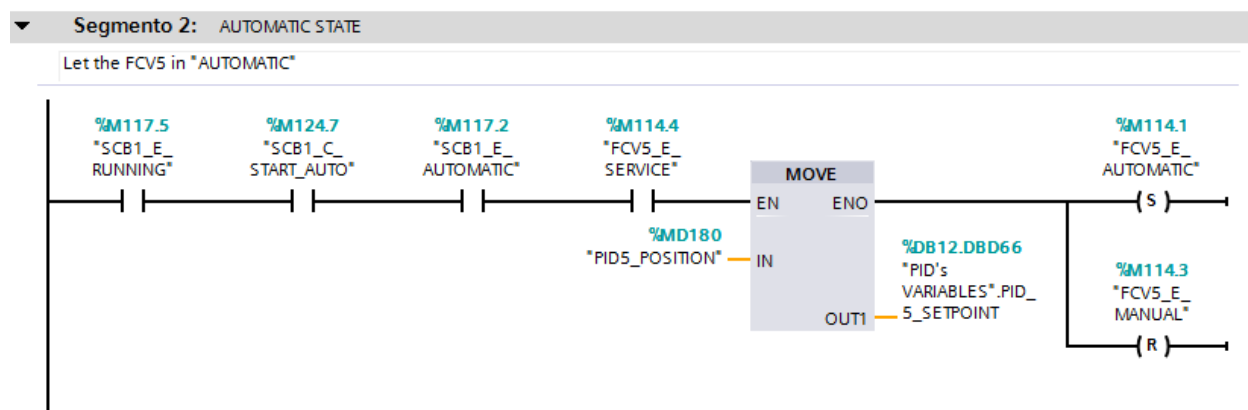
## i.viii Element (group) H

In this section, the functioning of Flow Control Valve 5 (FCV5) will be explained. This element is used in the project to control the Side Channel Blower 1 (SCB1). The Flow Meter 5 (FM5) will be explained in this section too.

- The first segment is for let the FCV5 system in “SERVICE” state. To do that, it is necessary the SERVICE state in the control actuator SCB1.

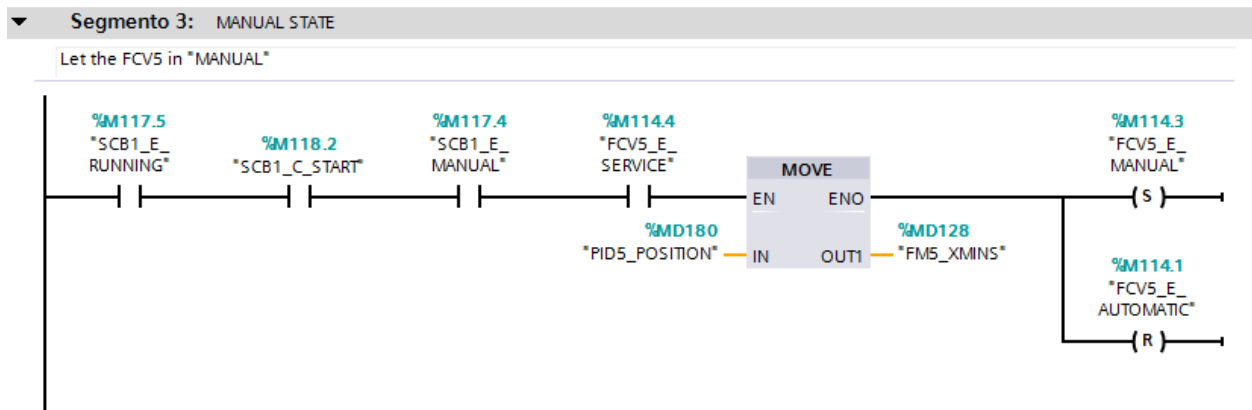


- The second segment is to choose the “AUTOMATIC” mode of FCV5 and to enter the position of the valve (opening percentage). To do that, it is necessary the “RUNNING” state, the “START\_AUTO” command the “AUTOMATIC” state of the control actuator SCB1.

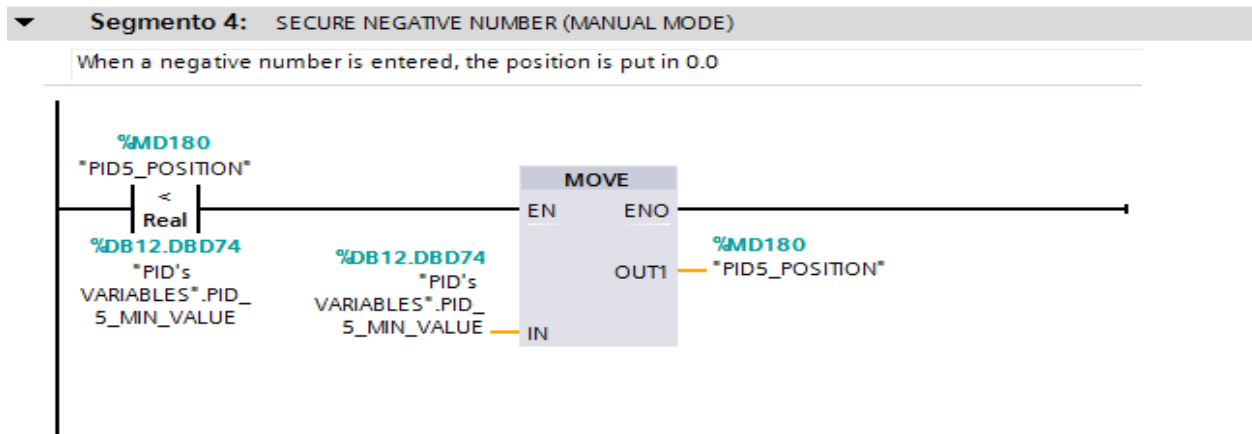




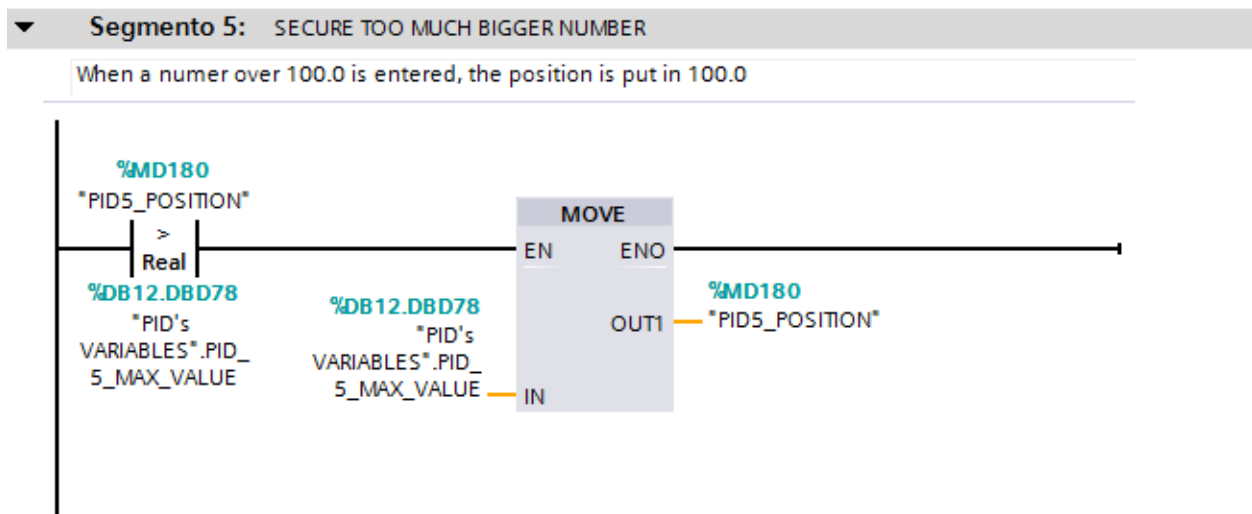
- The third segment is to choose the “MANUAL” mode of FCV5 and to enter the position of the valve (opening percentage). To do that, it is necessary the “RUNNING” state, the “START” command and the “MANUAL” state of the control actuator SCB1.



- This segment is to ensure that you will never enter a negative number to the valve.



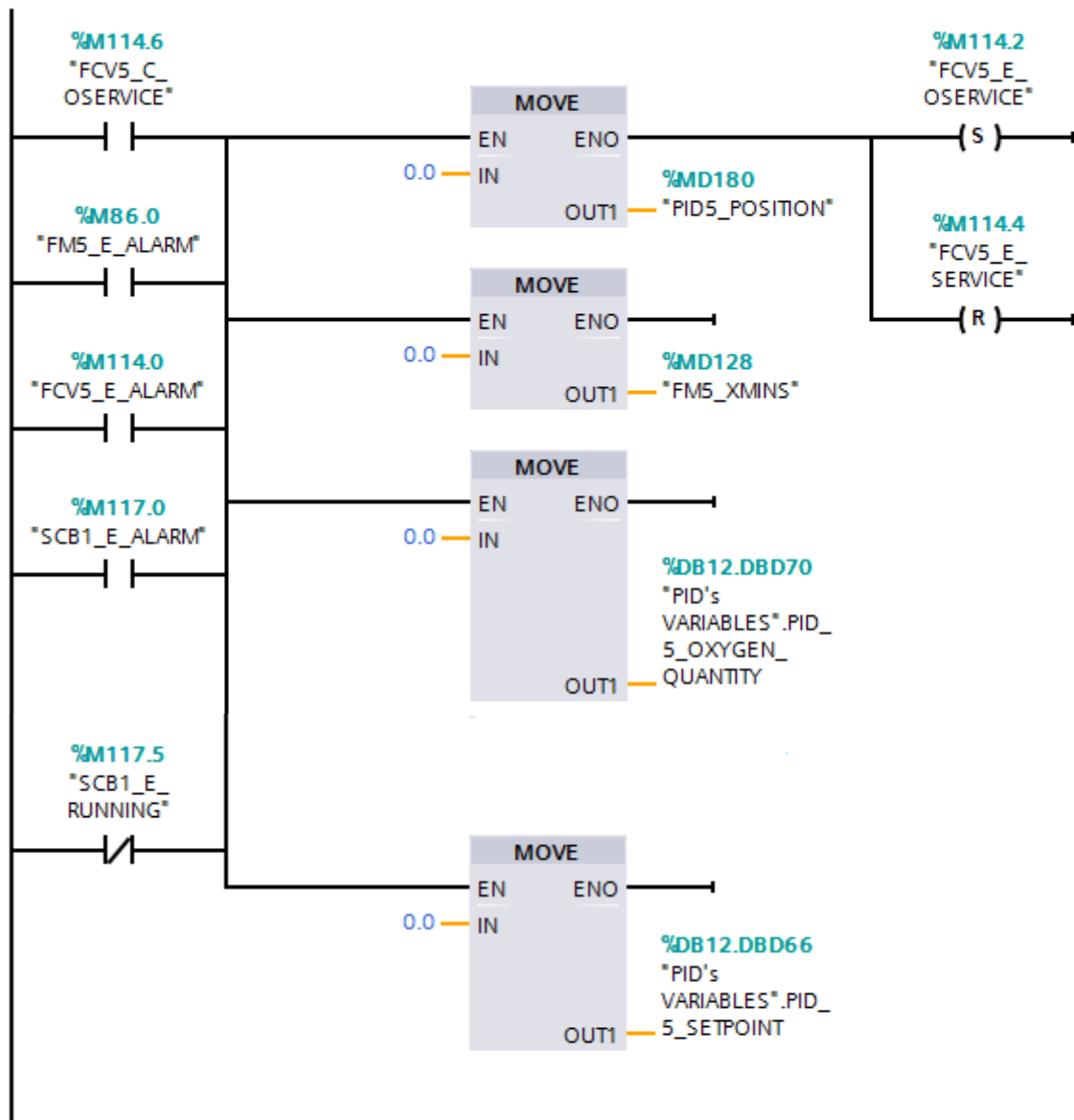
- This segment is to ensure that it will never enter a number bigger than 100 to the valve.



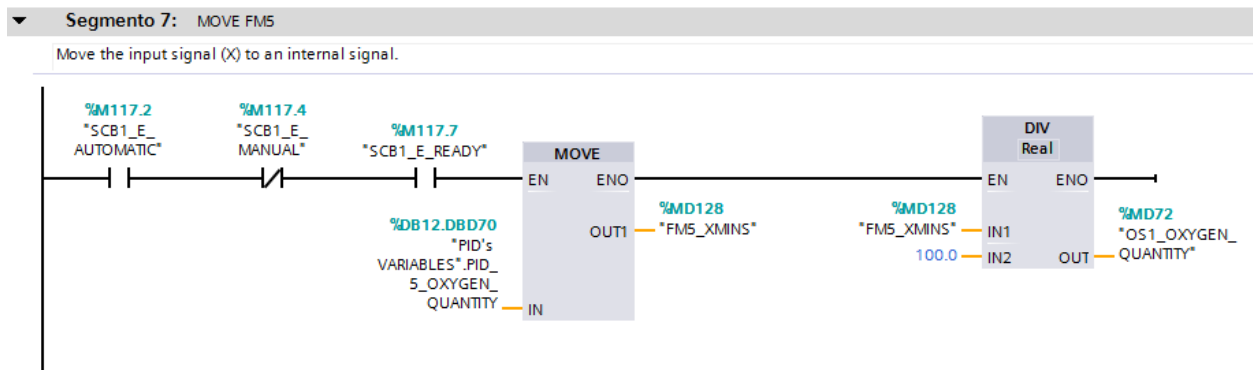
- The sixth segment is to let the system in “OSERVICE” state when the "OSERVICE" command is sent, when the alarm of FVC5 is active, when the “RESET” command is sent when the alarm of FM5 is active, when the “OSERVICE” command of the controller SCB1 is sent or when the controller is in “ALARM” state.

▼ **Segmento 6: OSERVICE STATE**

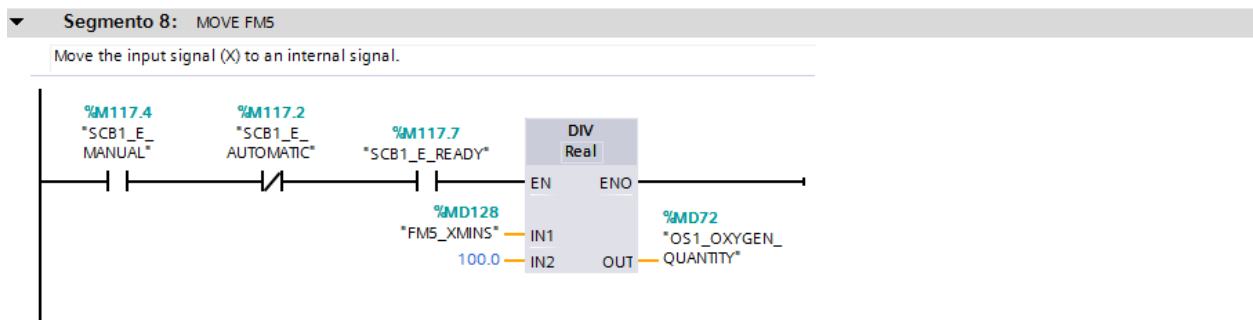
► Let the valve position to 0% if the FCV5 recives de "OSERVICE" or "RESET" order and if FCV5 is in "AL...



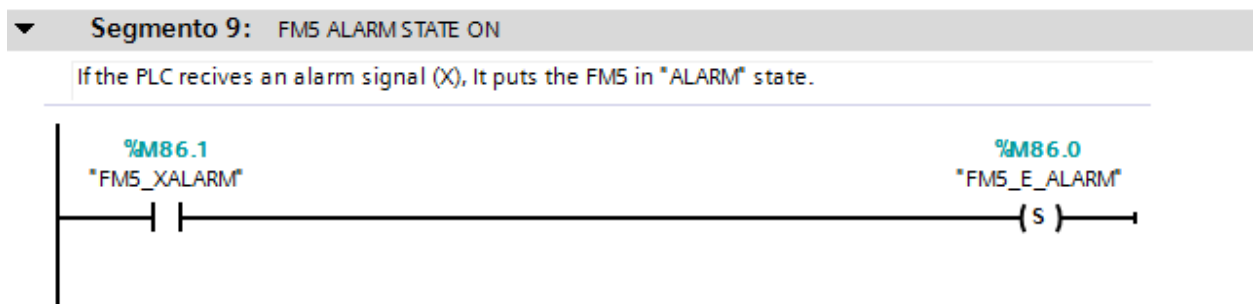
- This segment is to send the real number generated by PID to the Flow Meter when the system is in “AUTOMATIC” mode.



- This segment is to send the set point to the Flow Meter when the system is in “MANUAL” mode.



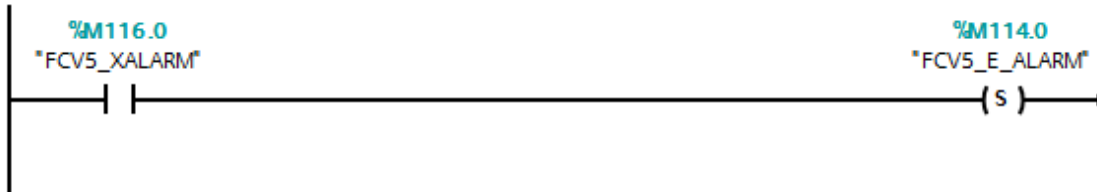
- This segment is to simulate an “ALARM” signal of Flow Meter that comes from the plant.



- This segment is to simulate an “ALARM” signal of Flow Control Valve that comes from the plant.

▼ **Segmento 10:** FCV5 ALARM STATE ON

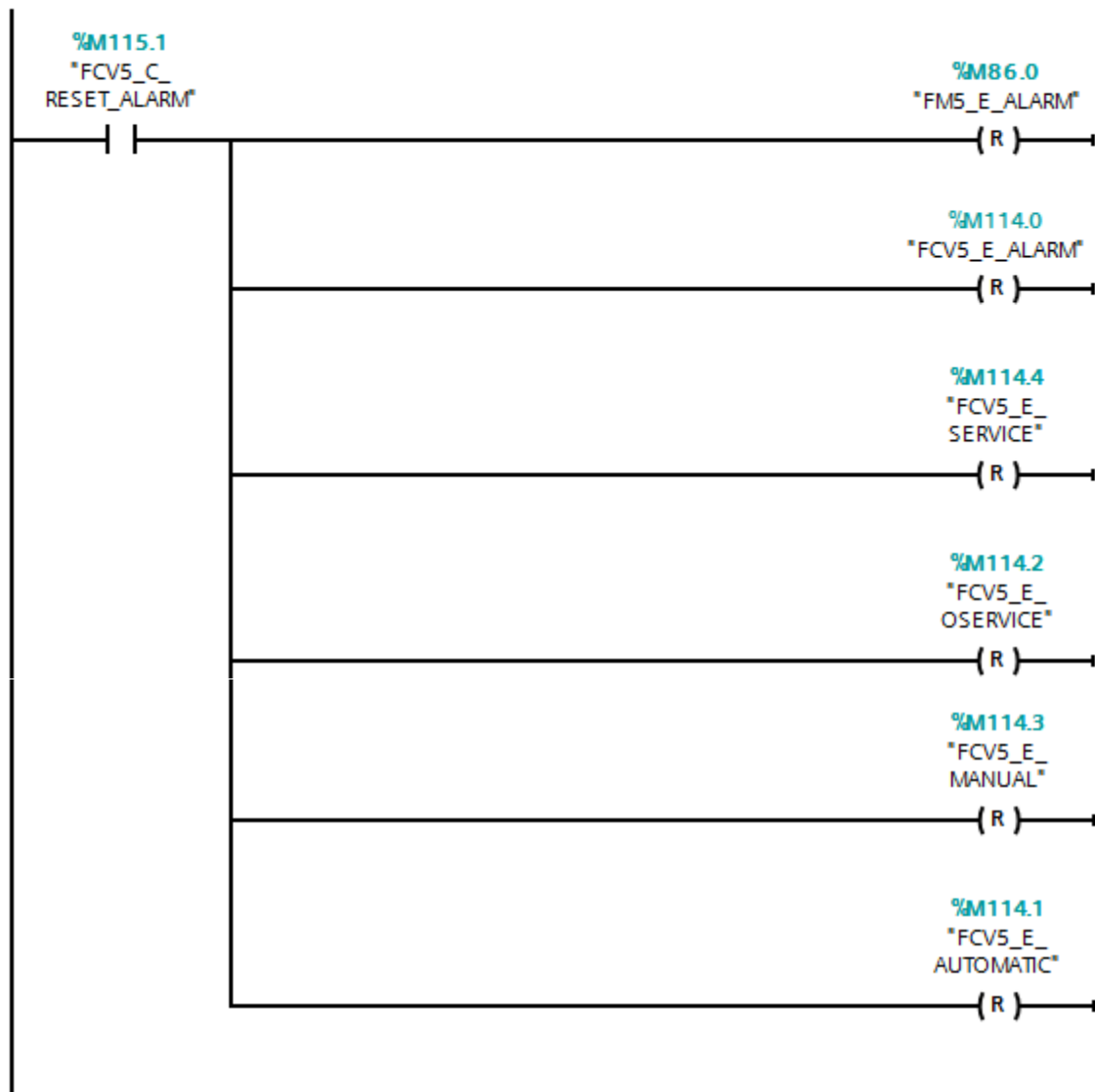
If the PLC receives an alarm signal (X), It puts the FCV5 in "ALARM" state.



- The last segment is to let all the system in reset.

▼ **Segmento 11:** ALARM STATE AND RESET SYSTEM

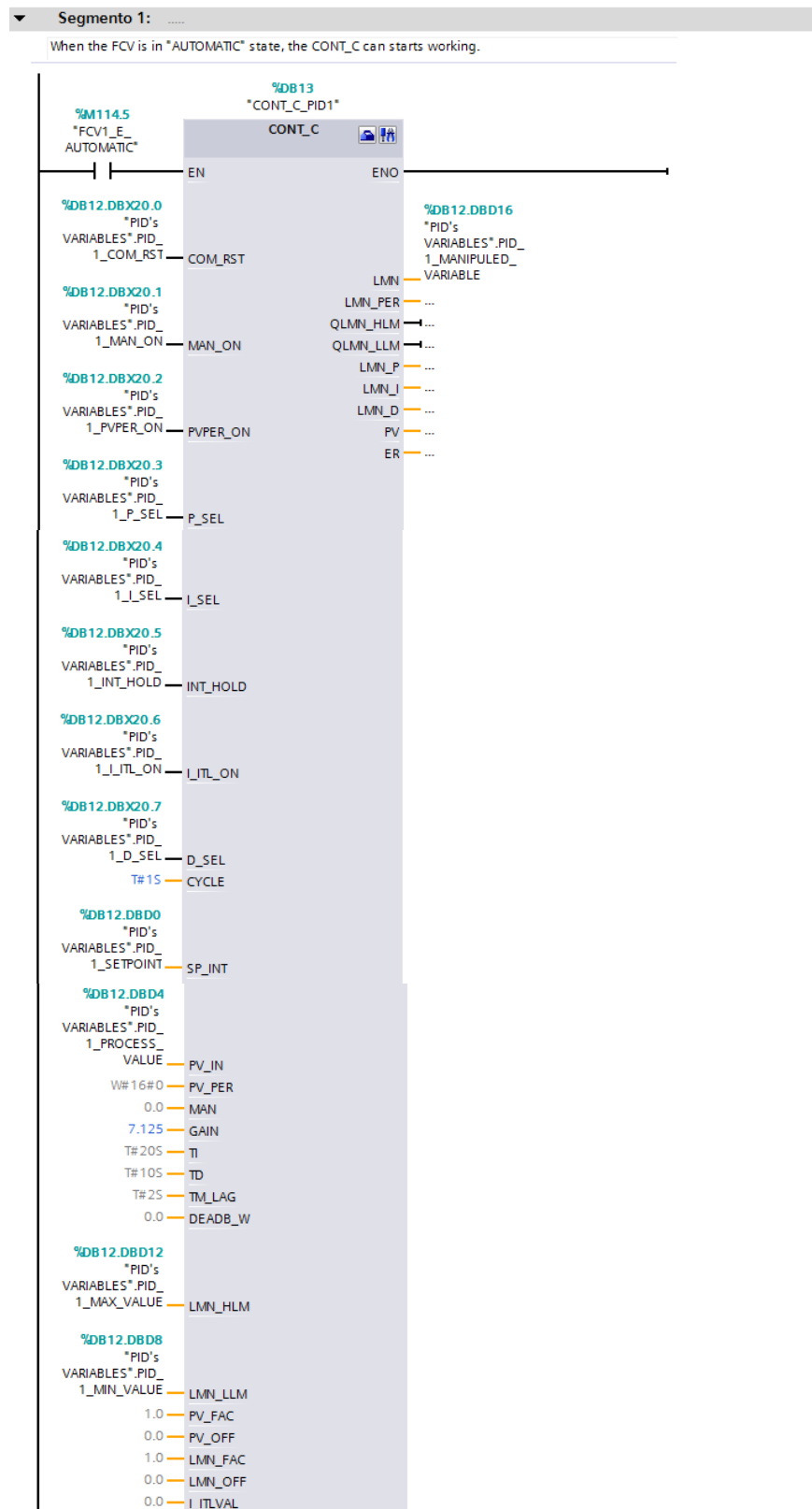
If the "RESET ALARM" order from the sistem control is activated, all states of the system will be reset.



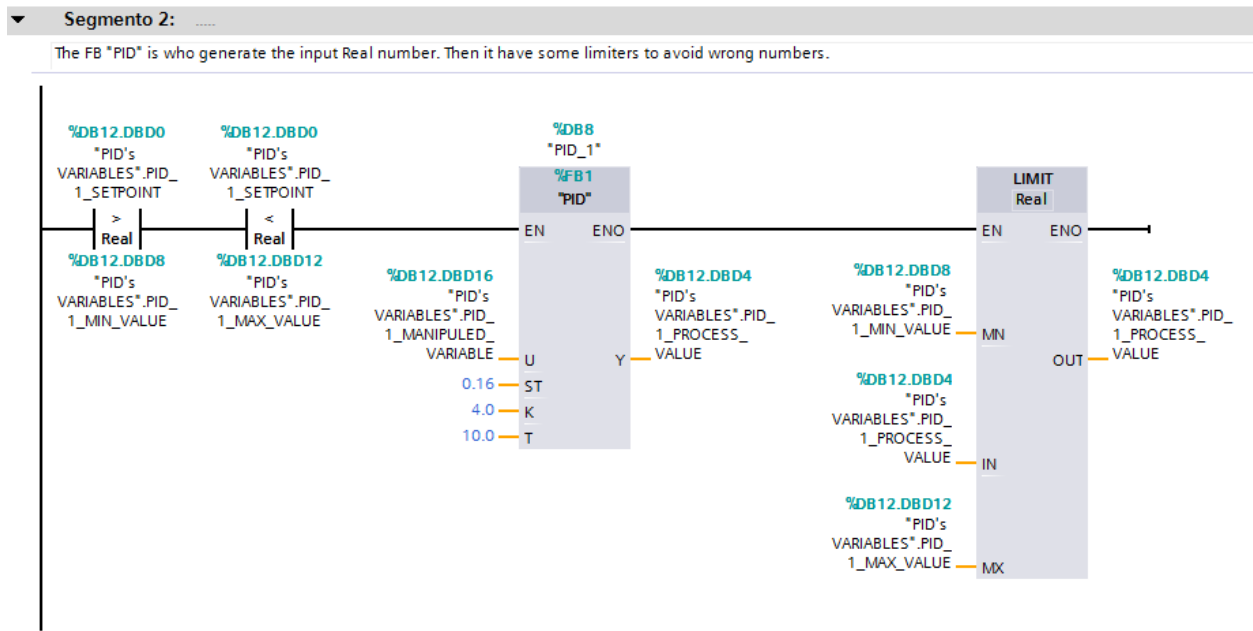
## i.ix Element (group) I

This section is for explain the PID element, this element will be used to control the simulated real input which is used to simulate a flow water. The PID 1 will be used to do the example.

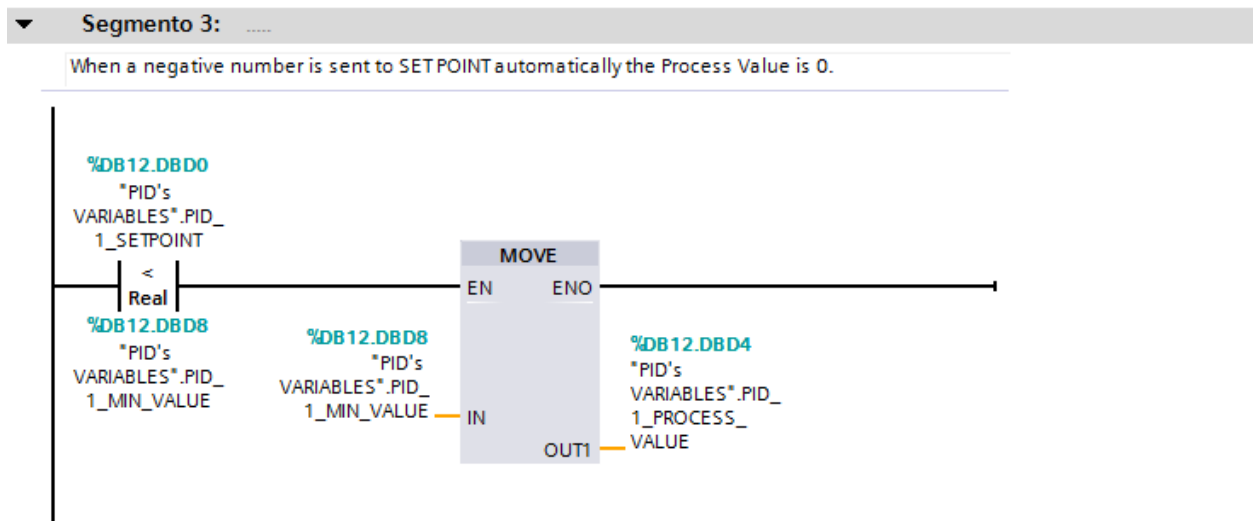
- When the FCV is in “AUTOMATIC” state, the PID starts working:



- The second segment is to generate the number that will be the flow water/oxygen/chlorine simulation. Two comparisons to assure that the input “SET POINT” will be always between 0 and 100 will be done.



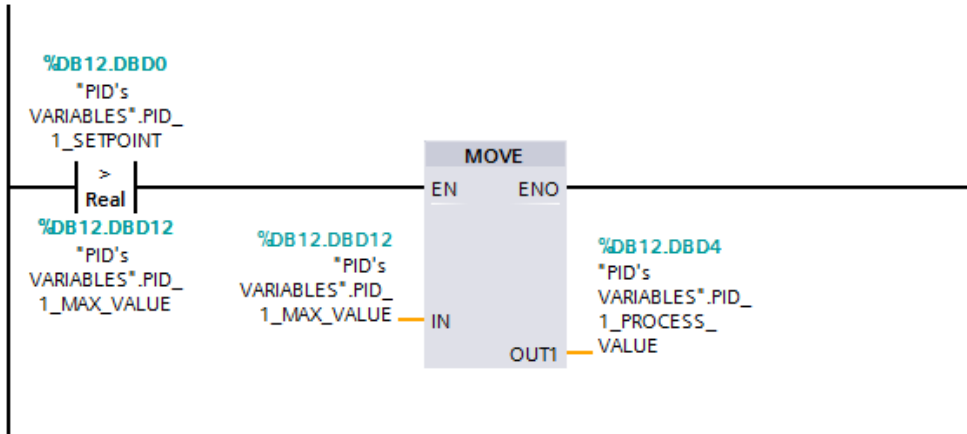
- When the SET POINT number sent is negative, automatically, the Process Value is 0. The Process Value is the number that simulates the flow water.



- When the SET POINT number sent is too much bigger (more than 100.0), automatically, the Process Value is 100.0.

▼ Segmento 4: .....

When to much bigger number is sent to SETPOINT automatically the Process Value is 100.



# ANNEX ii

In this annex, will be explained the functioning of the different function blocks done along the project. The Function Blocks (FB) has been really useful to do the different simulations that the project needs. It has four different Function Blocks that will be explained now.

## ii.i PID [FB1]

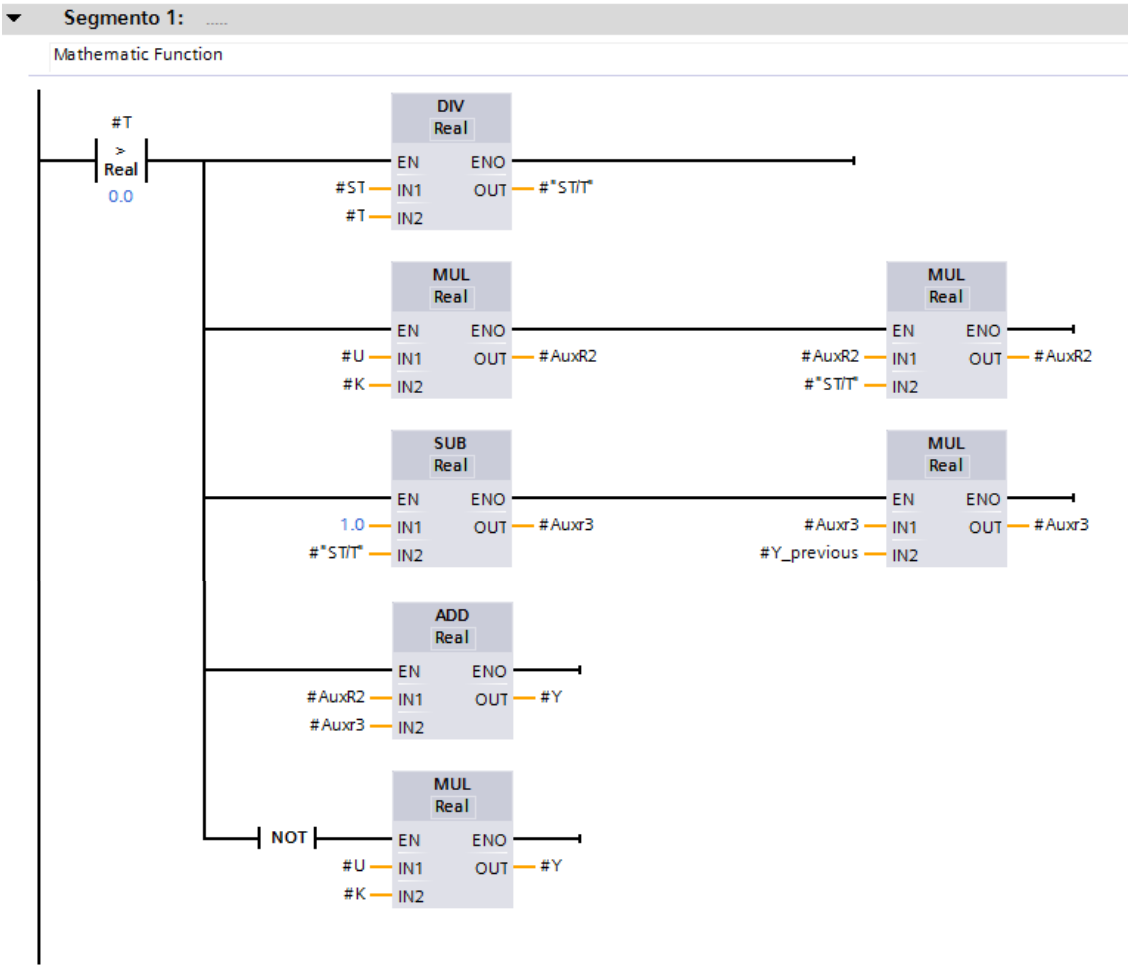
This FB it has done to simulate a Real number to simulate water/oxygen/chlorine flow. To do that has been generated a first order function. The generated function is:

$$y_n = \frac{ST}{T} \cdot U \cdot K + \left(1 - \frac{ST}{T}\right) y_{n-1}$$

**Equation 1.** *Mathematic function used along the project.*

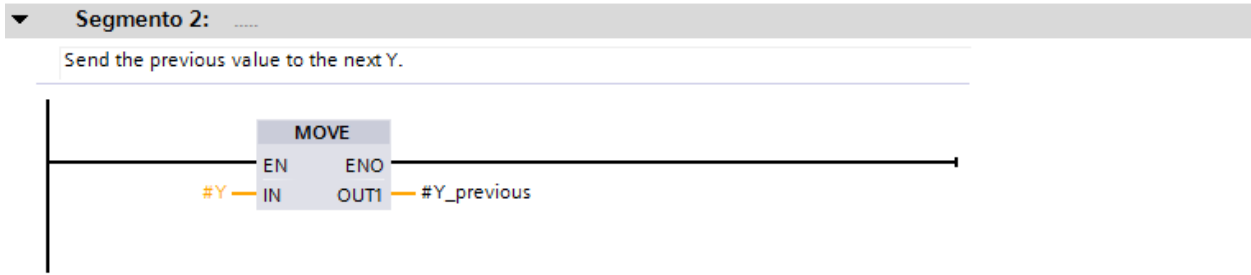
Where  $ST = 0.16, T = 10.0, K = 4.0$ ,  $U$  is the Manipulated Variable and  $y_n$  is the Process Value.

- The first segment is the Mathematic function explained above.





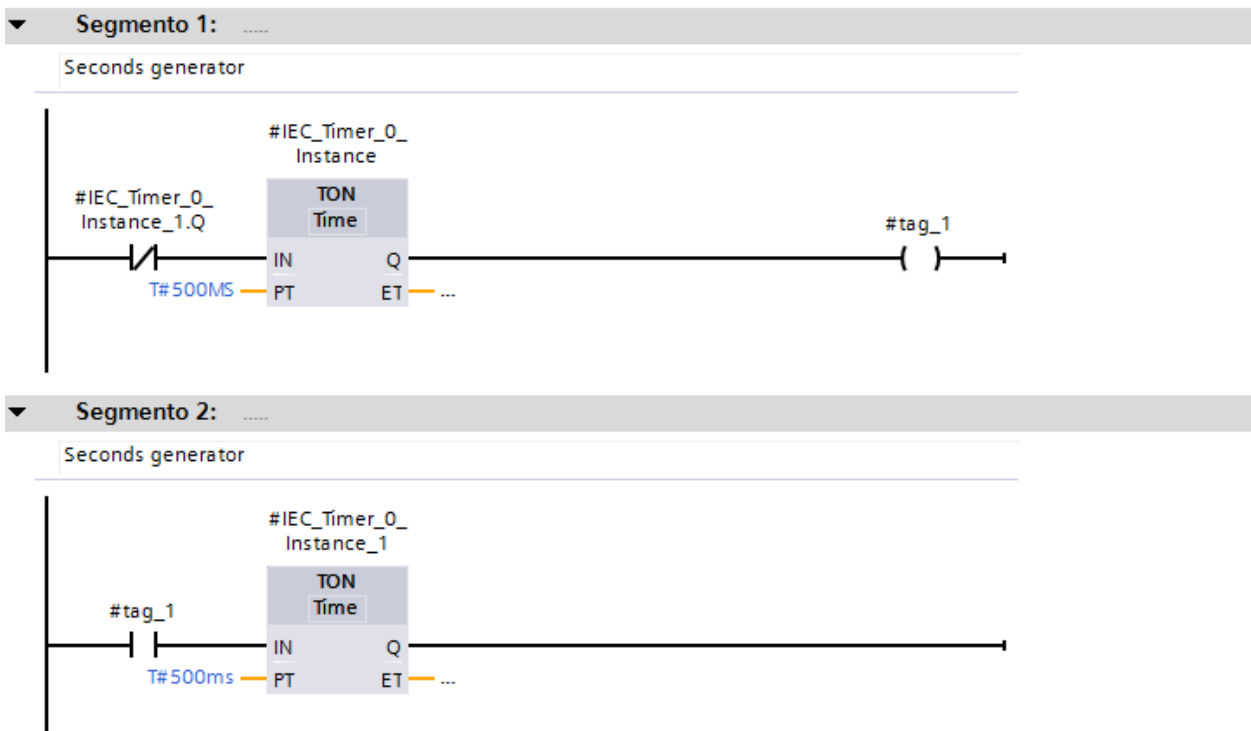
- The second segment is just to send  $y_n$  to  $y_{n-1}$ .



## ii.ii Time Counter [FB2]

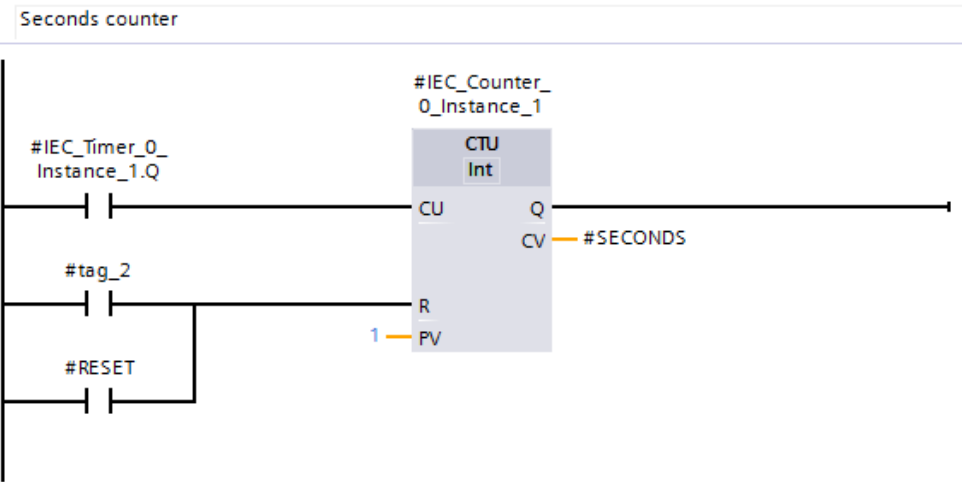
This Function Block is a counter to know how much time the different actuators are in “RUNNING” state. The program is a normal counter, and then the time is divided in seconds, minutes and hours with one Reset Counter.

- The first and second segments are used to generate a seconds pulse.



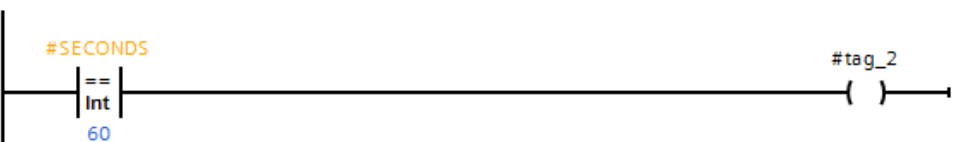
- The third and fourth segments are used to count the seconds and let the counter at 0 when the counter arrives at 60 seconds.

▼ Segmento 3: .....



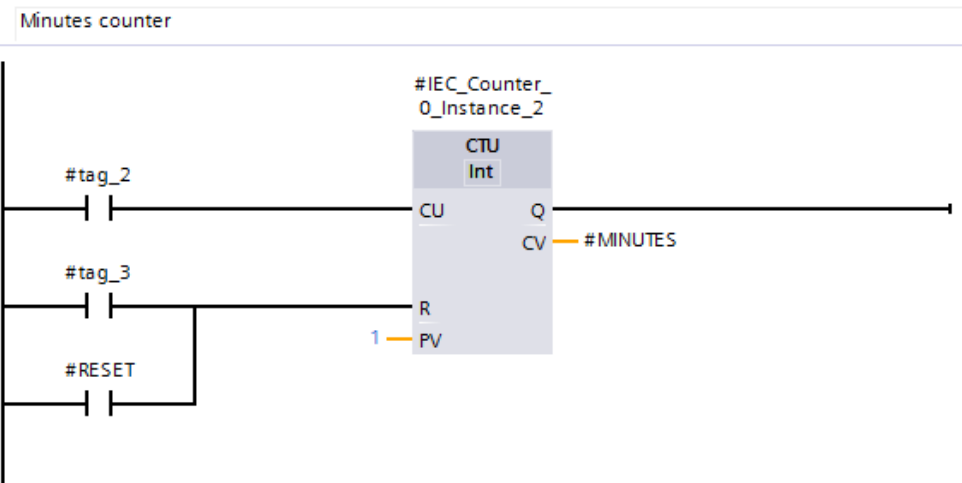
▼ Segmento 4: .....

When the seconds arrives to 60, the tag\_2 send a reset to the seconds counter.



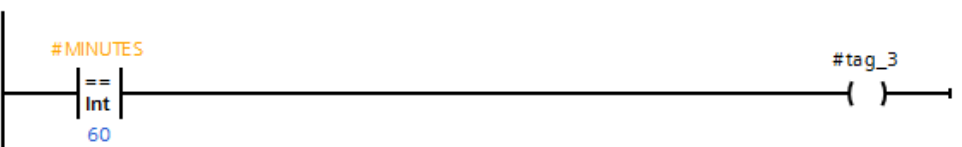
- The fifth and sixth segments are used to count the minutes and let the counter to 0 when the counter arrives at 60 minutes.

▼ Segmento 5: .....

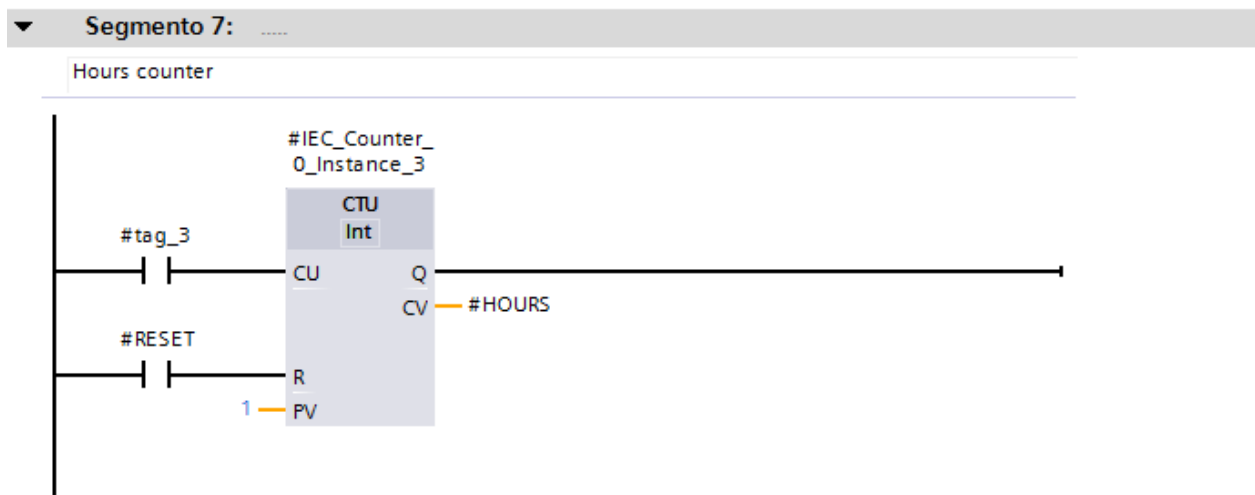


▼ Segmento 6: .....

When the minutes arrives to 60, the tag\_3 send a reset to the minutes counter.



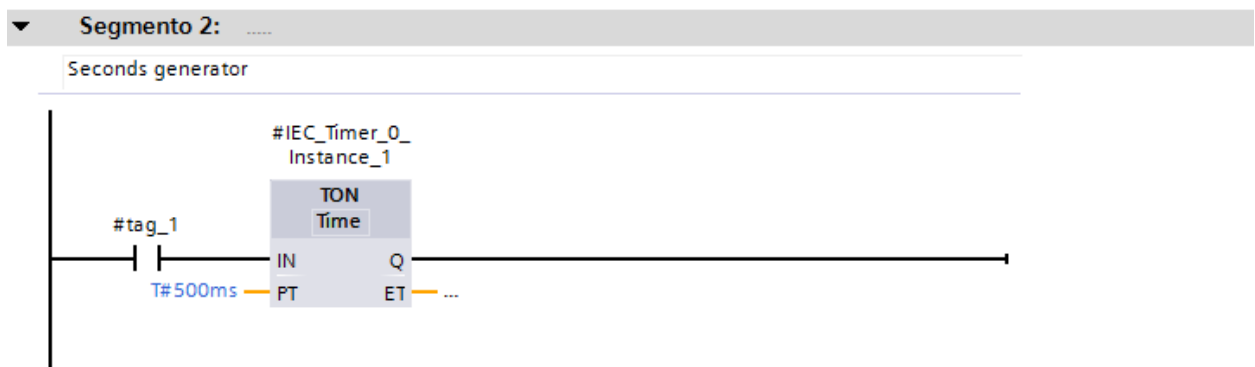
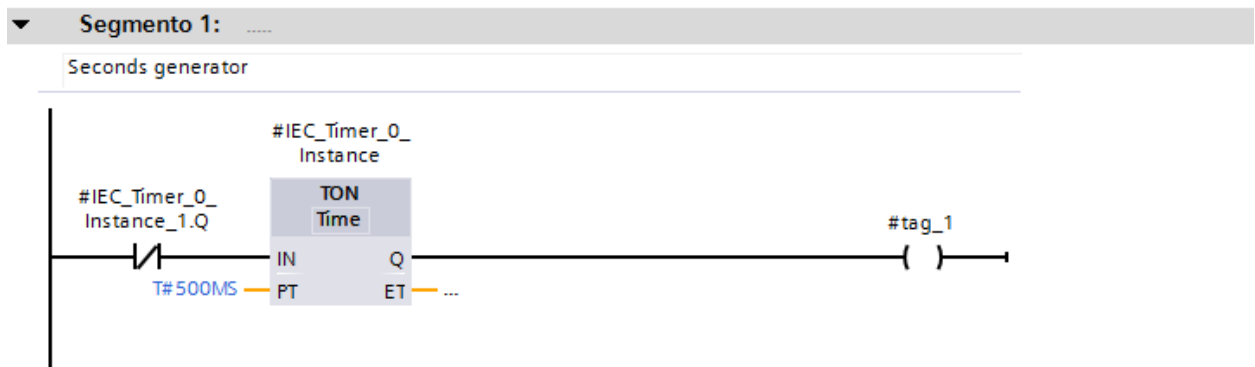
- The seventh segments are used to count the hours.



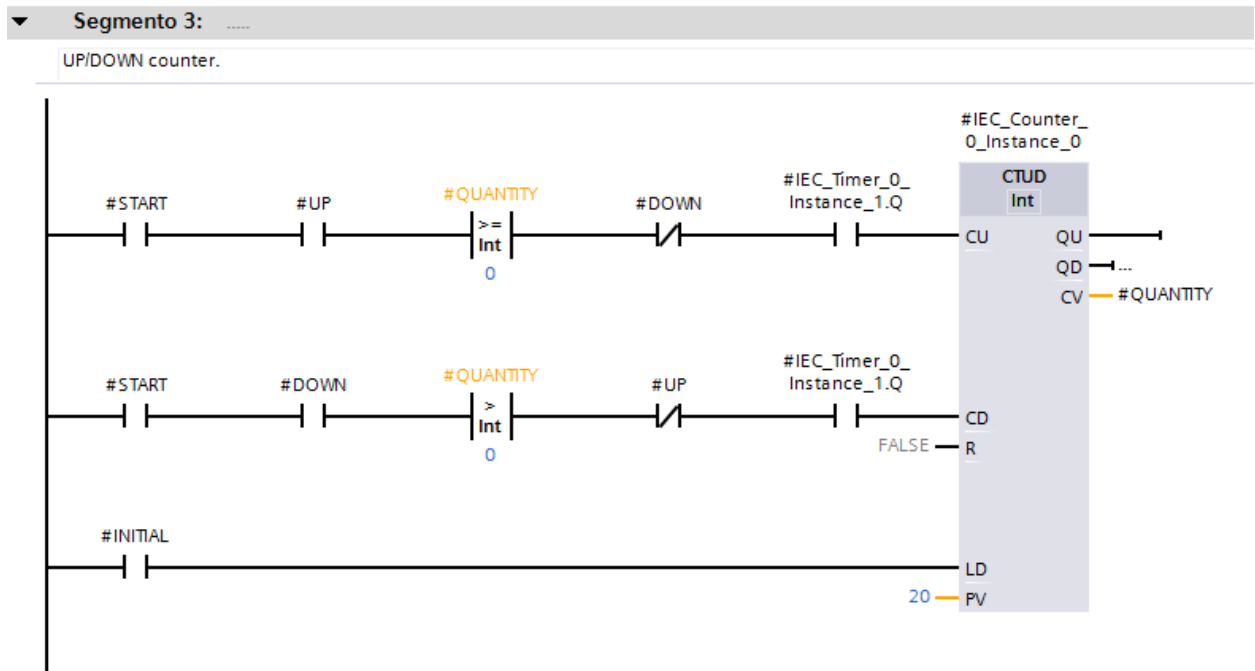
### ii.iii Height simulator [FB3]

This Function block is used to simulate the ascent and descend of the metal blades moved by MDC1, MDC2 and MDC3. It is a time counter until 20 seconds, it could be another cycle, but, to realize a simulation it is a good period.

- The first and second segments are used to generate a seconds pulse.



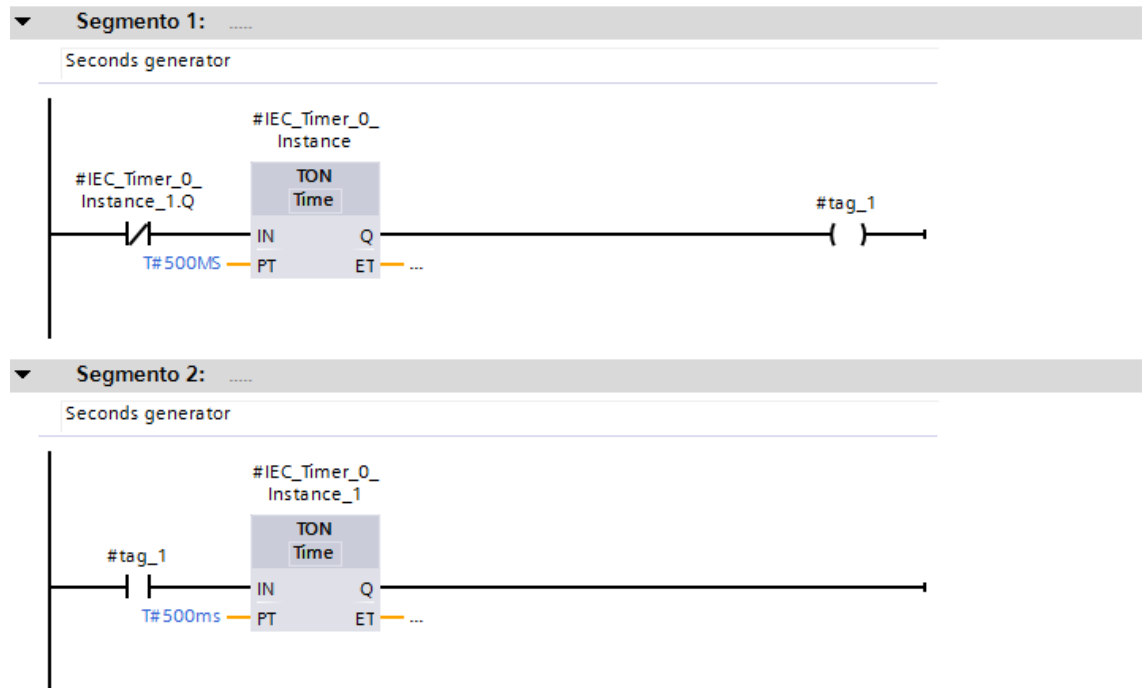
- The third segment is to control the UP/DOWN counter depending on the commands sent from the HMI control.



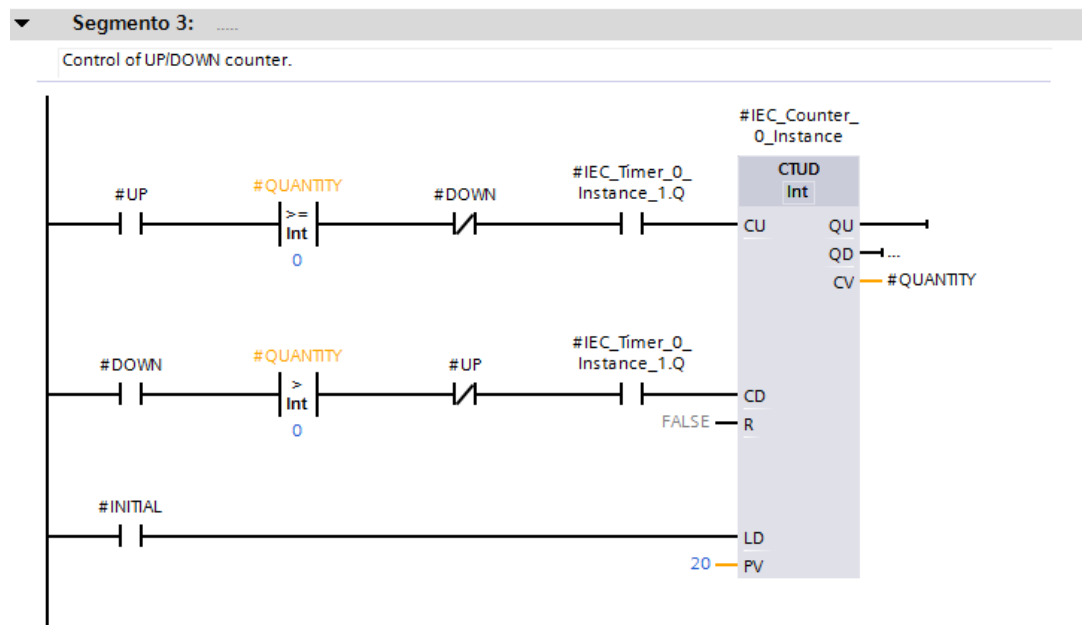
## i.iv Mud/Fat simulator [FB4]

This Function block is used to simulate the filling and emptying of the different tanks in the project through the MP1, MP2, MP3 and FP1. It is a time counter until 20 seconds, it could be another cycle, every second means a kg of mud, or a fat liter.

- The first and second segments are used to generate a seconds pulse.



- The third segment is to control the UP/DOWN counter depending on the commands sent from the HMI control.



## ANNEX iii

The third annex is to show the functioning of every part of HMI program. The program is made of thirteen images. The project has been divided in different sections, because when the control is made by HMI it is not possible to put all the process in the same screen (or image in our case), to do a good organization, it has done a principal screen where it is possible to go to the other images. All the images have a place in upper right corner where it can see the time and the day.

Next, every image and the utility of each one will be explained.

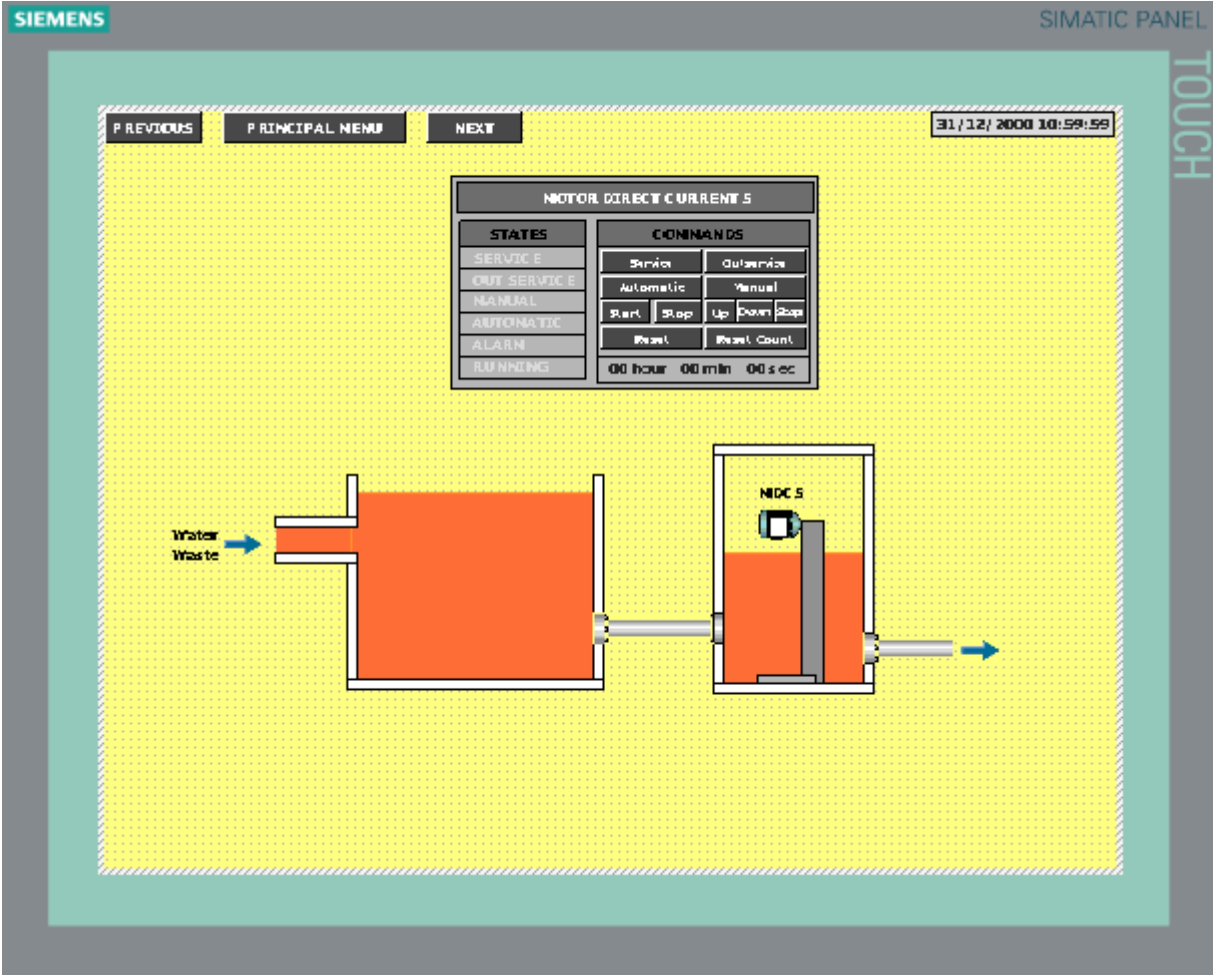
### iii.i Principal Menu

This image is the place where it is possible to go to the others images, that is the principal function of the Principal Menu. It has twelve buttons, where it can read the destination of the different processes. In the top right we find the date and time.

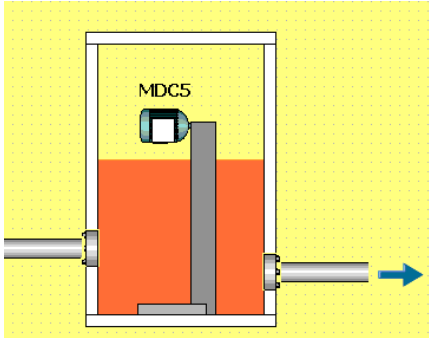
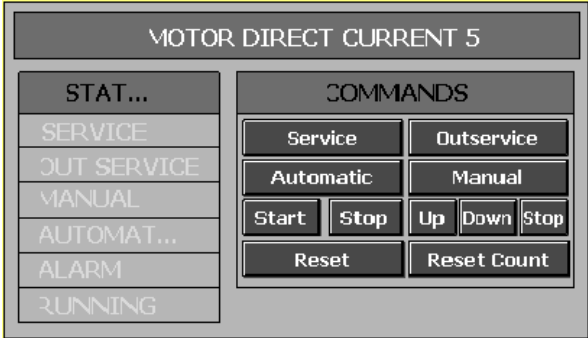


### iii.ii Pretreatment 1

In this image it can see the entrance of waste water to the plant; the first step is picking up the big solids through the metal blades of MDC5. In the upper left corner we find three buttons, one to go to the previous image, another to go to the principal menu and the other to go to the next image.

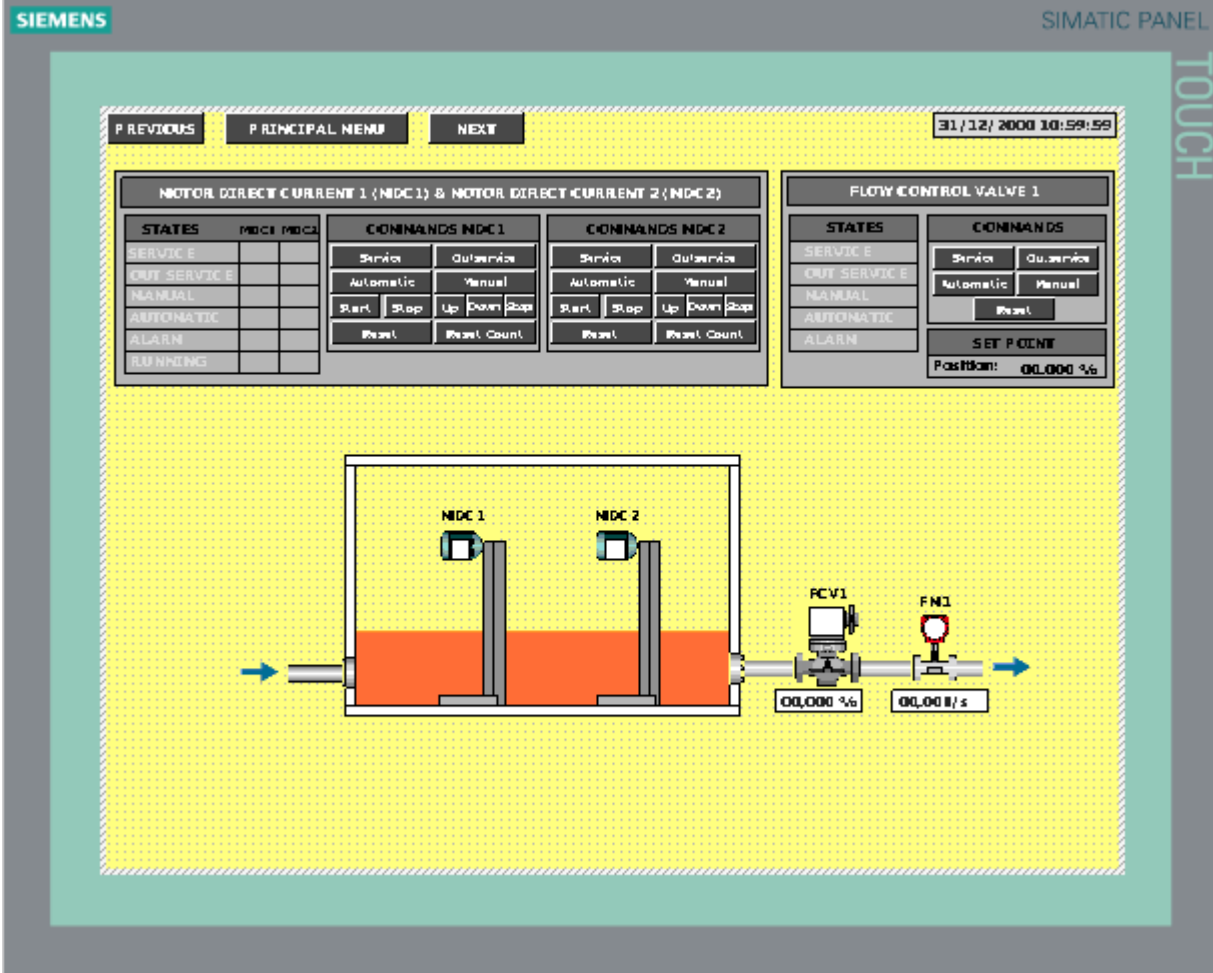


In the MDC5 controller, we can see the states in the left part, when one of these states is set in ON, the letters become black. In the right side we can see the command buttons, every button is connected with a command in the PLC program.



### iii.iii Pretreatment 2

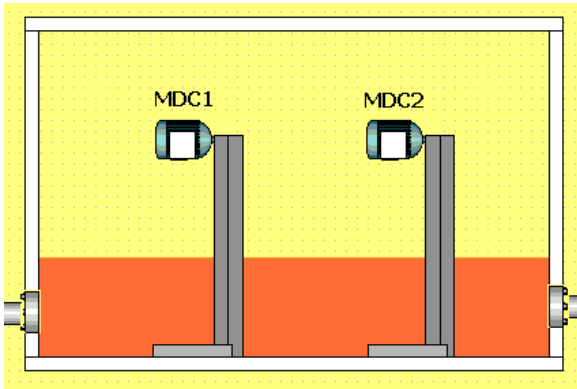
This image shows the second part of the pretreatment, there are two motors (MDC1 and MDC2) to ascend and descend the metal blades to remove the solid waste. Then it has a Flow Control Valve (FCV1) to control the flow before the first tank. In the upper left corner we find three buttons, one to go to the previous image, another to go to the principal menu and the other to go to the next image.





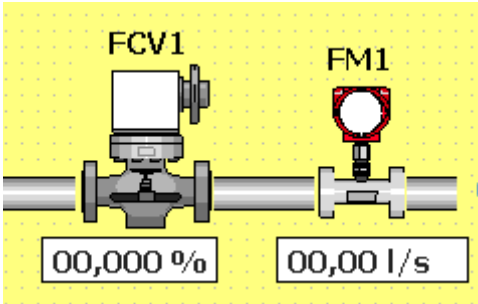
In this image, we can see two controllers; the first one is to control the motors MDC1 and MDC2, to optimize the space we have put together the states of both motors, when a states is sets in ON, the little square next to letters, becomes black.

MOTOR DIRECT CURRENT 1 (MDC1) & MOTOR DIRECT CURRENT 2 (MDC2)									
STATES	MDC1	MDC2	COMMANDS MDC1		COMMANDS MDC2				
SERVICE	<input type="checkbox"/>	<input type="checkbox"/>	Service	Outservice	Service	Outservice			
OUT SERVICE	<input type="checkbox"/>	<input type="checkbox"/>	Automatic	Manual	Automatic	Manual			
MANUAL	<input type="checkbox"/>	<input type="checkbox"/>	Start	Stop	Start	Stop	Up	Down	Stop
AUTOMATIC	<input type="checkbox"/>	<input type="checkbox"/>	Reset		Reset		Reset Count		
ALARM	<input type="checkbox"/>	<input type="checkbox"/>							
RUNNING	<input type="checkbox"/>	<input type="checkbox"/>							



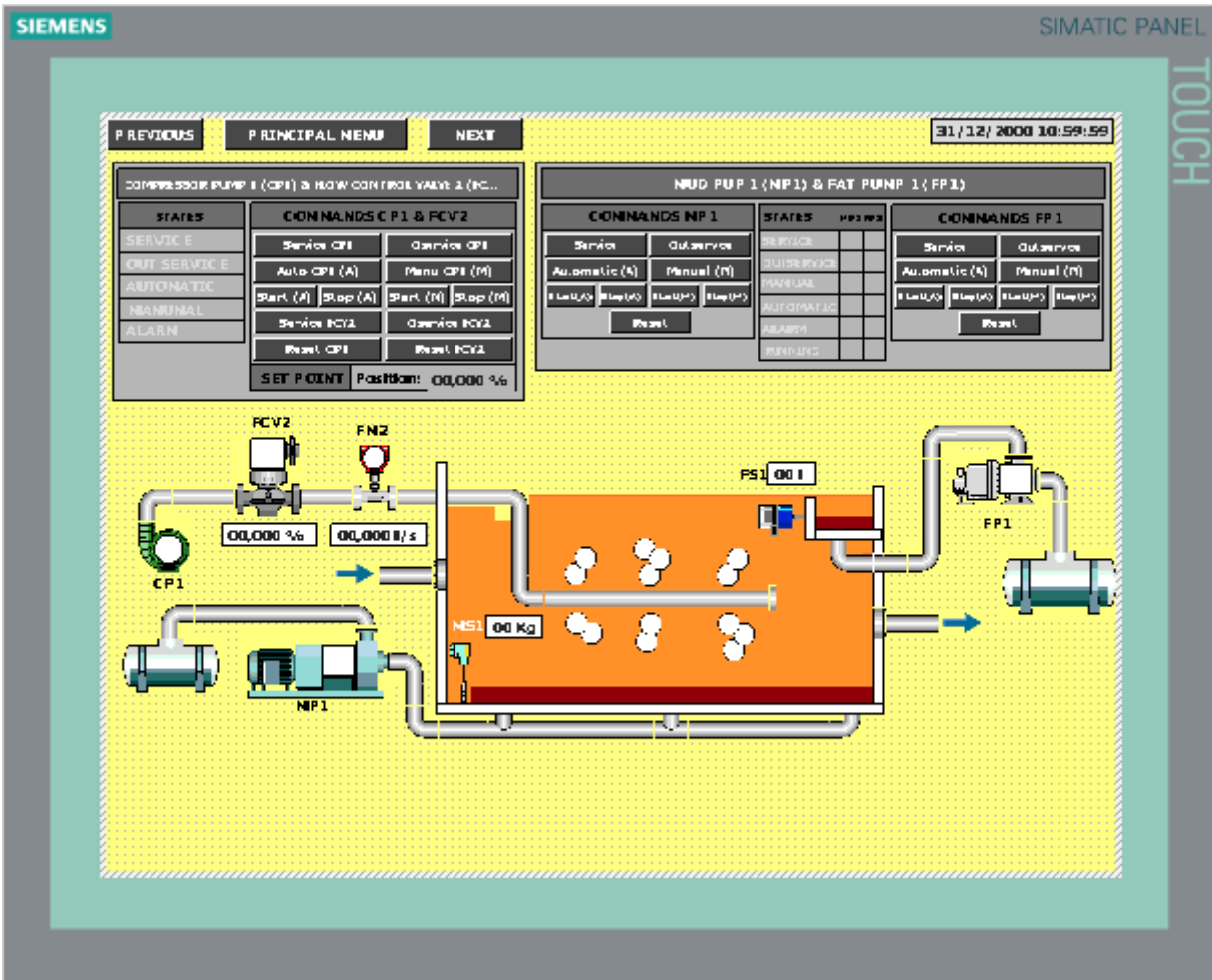
In the controller of FCV1 we can see the states in the left part. In the right side it has the buttons commands and below them, the set point percentage, that is an input to send to the PLC the opening percentage desired for the FCV1.

FLOW CONTROL VALVE 1	
STATES	COMMANDS
SERVICE	Service    Outservice
OUT SERVICE	Automatic    Manual
MANUAL	Reset
AUTOMATIC	SET POINT
ALARM	Position: 00,000 %



### iii.iv Pretreatment 3

This image shows the third part of the pretreatment. In this part, there is a tank where the fats and the muds are separated thanks of a Compressor Pump (CP1). In this image we can see four actuators and three sensors. The actuators are CP1, MP1, FCV1 and FP1; the sensors are FM1, MS1 and FS1. In the upper left corner we find three buttons, one to go to the previous image, another to go to the principal menu and the other to go to the next image.



To optimize the space, the controller of Compressor Pump (CP1) and Flow Control Valve 2 (FCV2) is the same, in the left part of this controller the states of CP1 can be seen and in the right side, button to send the commands to the PLC.

COMPRESSOR PUMP 1 (CP1) & FLOW CONTROL VALVE 2 (FCV2)	
STATES	COMMANDS CP1 & FCV2
SERVICE	Service CP1      Oservice CP1
OUT SERVICE	Auto CP1 (A)      Manu CP1 (M)
AUTOMATIC	Start (A)   Stop (A)   Start (M)   Stop (M)
MANUNAL	Service FCV2      Oservice FCV2
ALARM	Reset CP1      Reset FCV2
	SET POINT    Position: 00,000 %

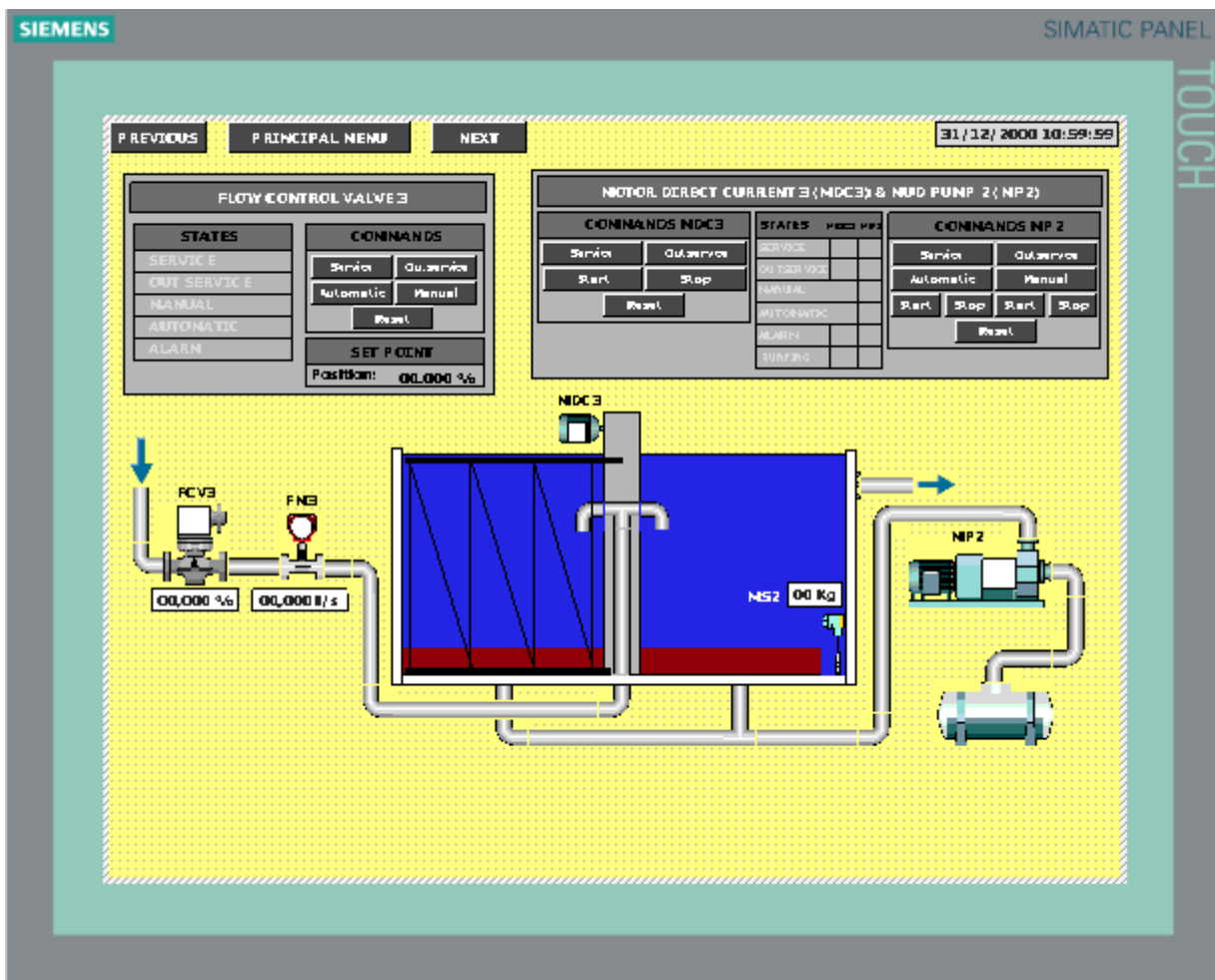
The other controller of this image is to control the Mud Pump 1 (MP1) and the Fat Pump 1 (FP1). In the left side the buttons to send the commands of MP1 to the PLC can be seen. In the middle we can see the different states of each actuator, when the state is ON, the little square next to letters become black. Finally, in the right side, the buttons to send the commands of FP1 to the PLC can be seen.

MUD PUP 1 (MP1) & FAT PUMP 1 (FP1)		
COMMANDS MP 1	STATES    MP1   FP1	COMMANDS FP 1
Service      Outservice	SERVICE <input type="checkbox"/> <input type="checkbox"/>	Service      Outservice
Automatic (A)    Manual (M)	OUTSERVICE <input type="checkbox"/> <input type="checkbox"/>	Automatic (A)    Manual (M)
Start(A)   Stop(A)   Start(M)   Stop(M)	MANUAL <input type="checkbox"/> <input type="checkbox"/>	Start(A)   Stop(A)   Start(M)   Stop(M)
Reset	AUTOMATIC <input type="checkbox"/> <input type="checkbox"/>	Reset
	ALARM <input type="checkbox"/> <input type="checkbox"/>	
	RUNNING <input type="checkbox"/> <input type="checkbox"/>	

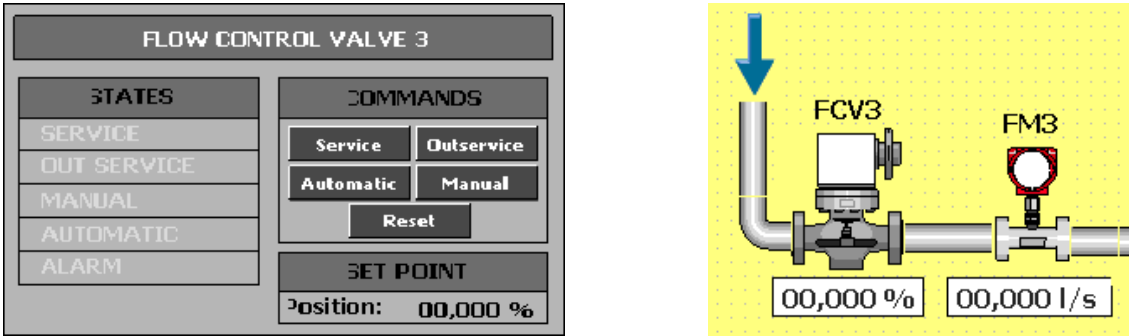
In the lasts images, we can see the MP1 with the Mud Sensor 1 (MS1) and the Fat Pump 1 (FP1) with his respective Fat Sensor 1 (FS1).

### iii.v Primary treatment 1

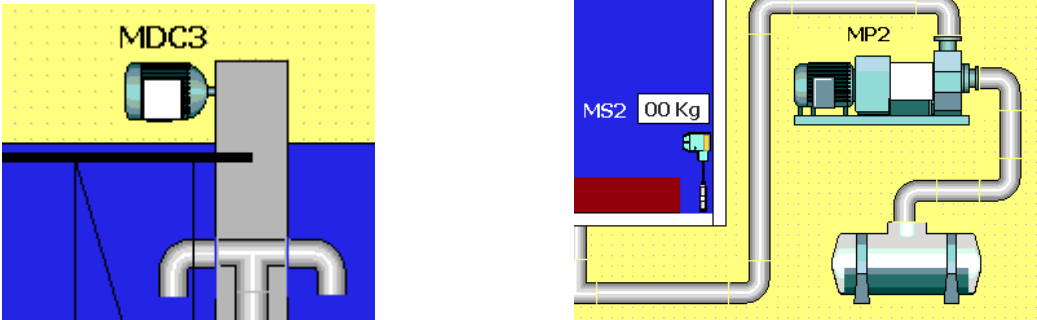
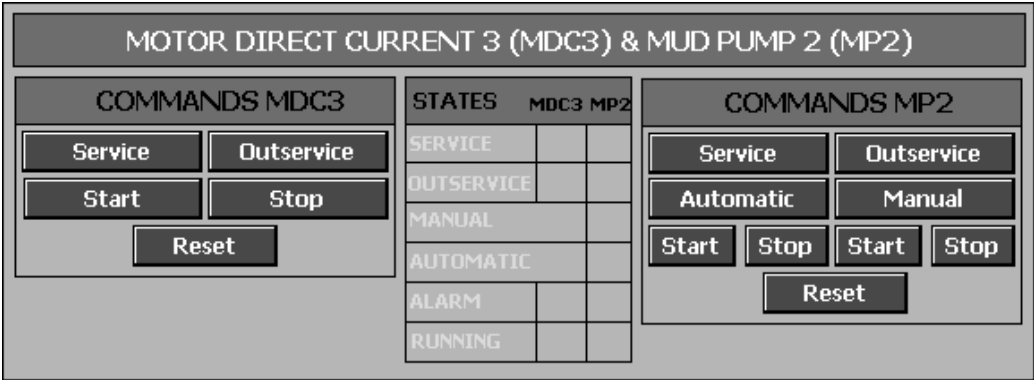
In this image we can see the first decantation of the process. To accomplish that, three actuators were needed, a Flow Control Valve (FCV3) to control the water flow that enter in the first decantation tank, one Motor Direct Current (MDC3) to turn the tool that help to settle the muds in the bottom of the tank, and a Mud Pump (MP2) to remove the mud settled. In the upper left corner we find three buttons, one to go to the previous image, another to go to the principal menu and the other to go to the next image.



In the controller of FCV3 we can see the states in the left part. The right side has the buttons commands and below them, the set point percentage, that is an input to send to the PLC the opening percentage desired for the FCV3.



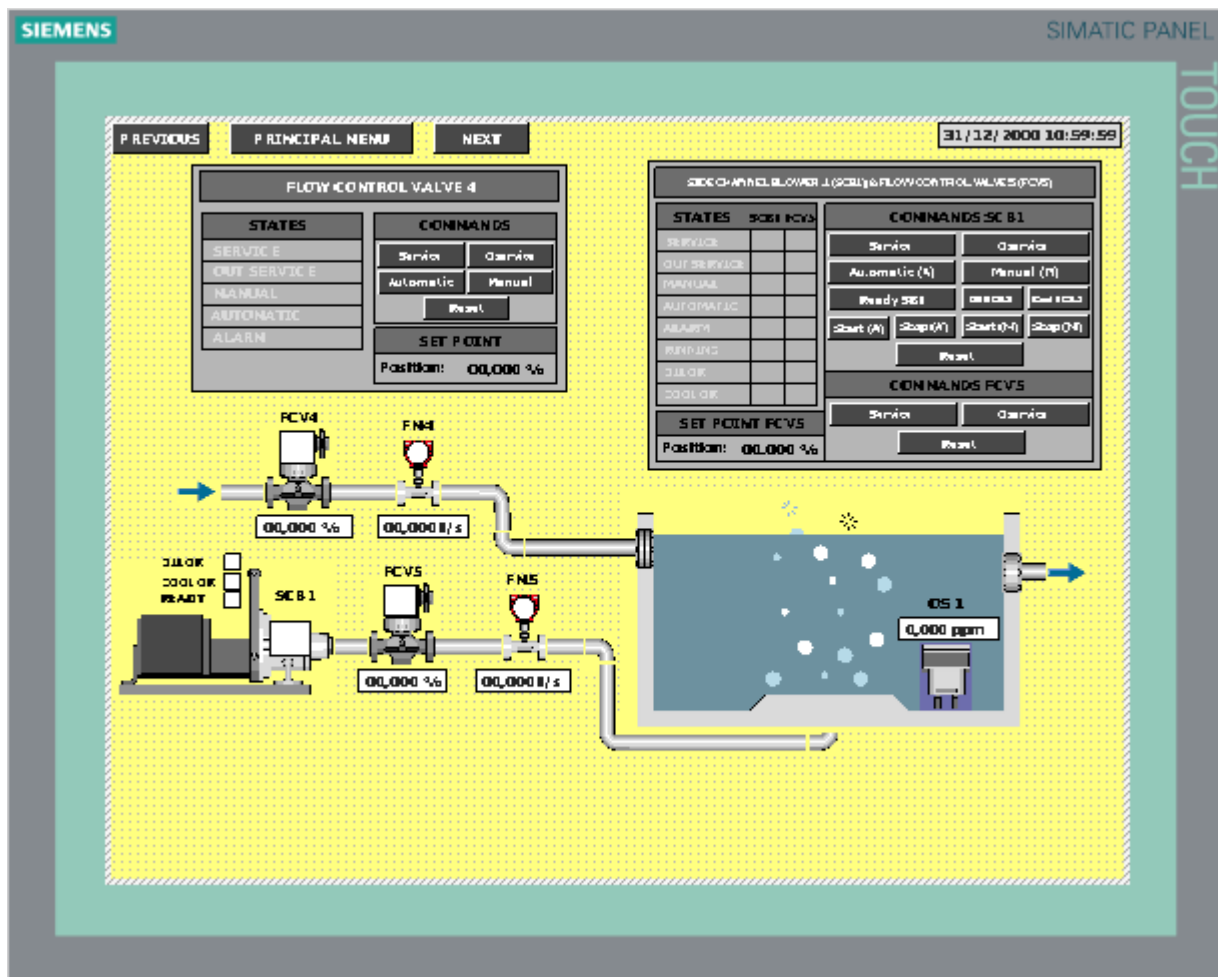
The other controller of this image is to control the Motor Direct Current 3 (MDC3) and the Mud Pump 2 (MP2). In the left side we can see the buttons to send the commands of MDC3 to the PLC. In the middle we can see the different states of each actuator, when the state is ON, the little square next to letters become black. Finally, in the right side are the buttons to send the commands of MP2 to the PLC.



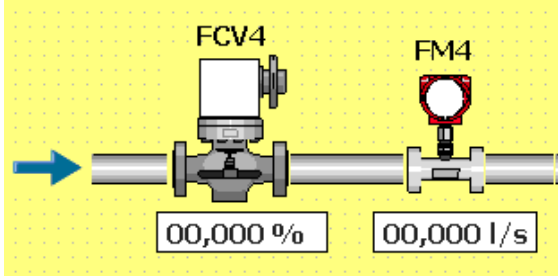
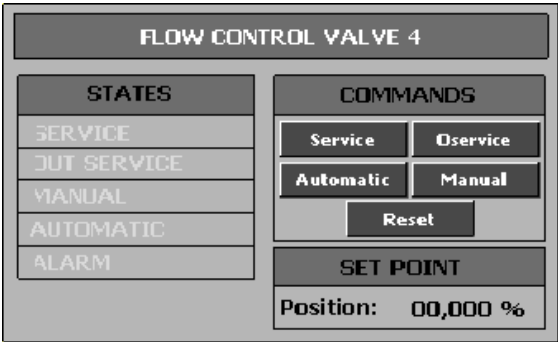
In the last images we can see the Motor Current Direct 3 (MDC3) and Mud Pump 2 (MP2) with his respective Mud Sensor 2 (MS2).

### iii.vi Secondary treatment 1

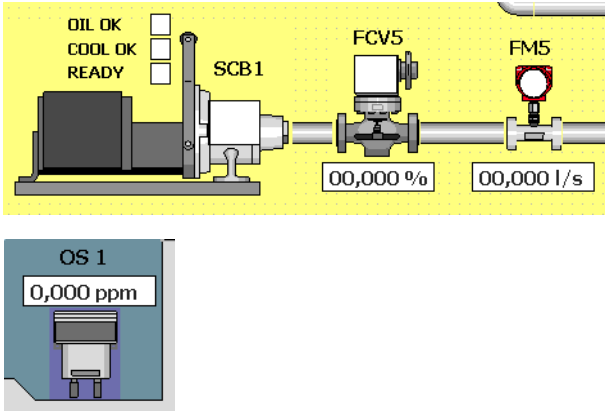
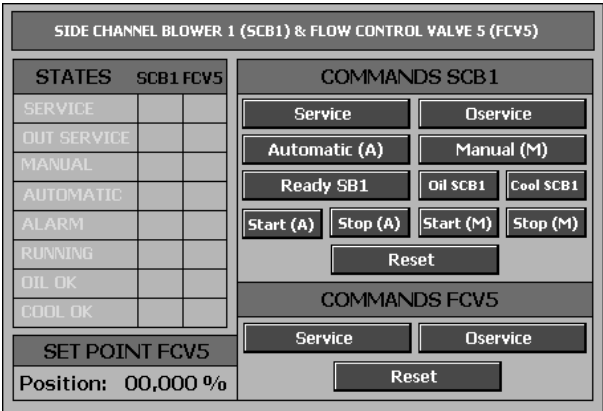
The next image shows the first part of secondary treatment. This treatment consists in the injection of oxygen into the water. To do that, it needs a Side Channel Blower (SCB1) controlled by Flow Control Valve (FCV5). The flow that enters in the oxygenation tank is regulated by another Flow Control Valve (FCV4). The SCB1 has a path to be ready to work, firstly it is necessary to switch on the oil circuit (“OILOK” state) and then the refrigerate circuit (“COOLOK” state). In the image, we can see that states (little squares) above SCB1. In the upper left corner we find three buttons, one to go to the previous image, another to go to the principal menu and the other to go to the next image.



In the controller of FCV4 we can see the states in the left part. In the right side it has the buttons commands and below them, the set point percentage, that is an input to send to the PLC the opening percentage desired for the FCV4.



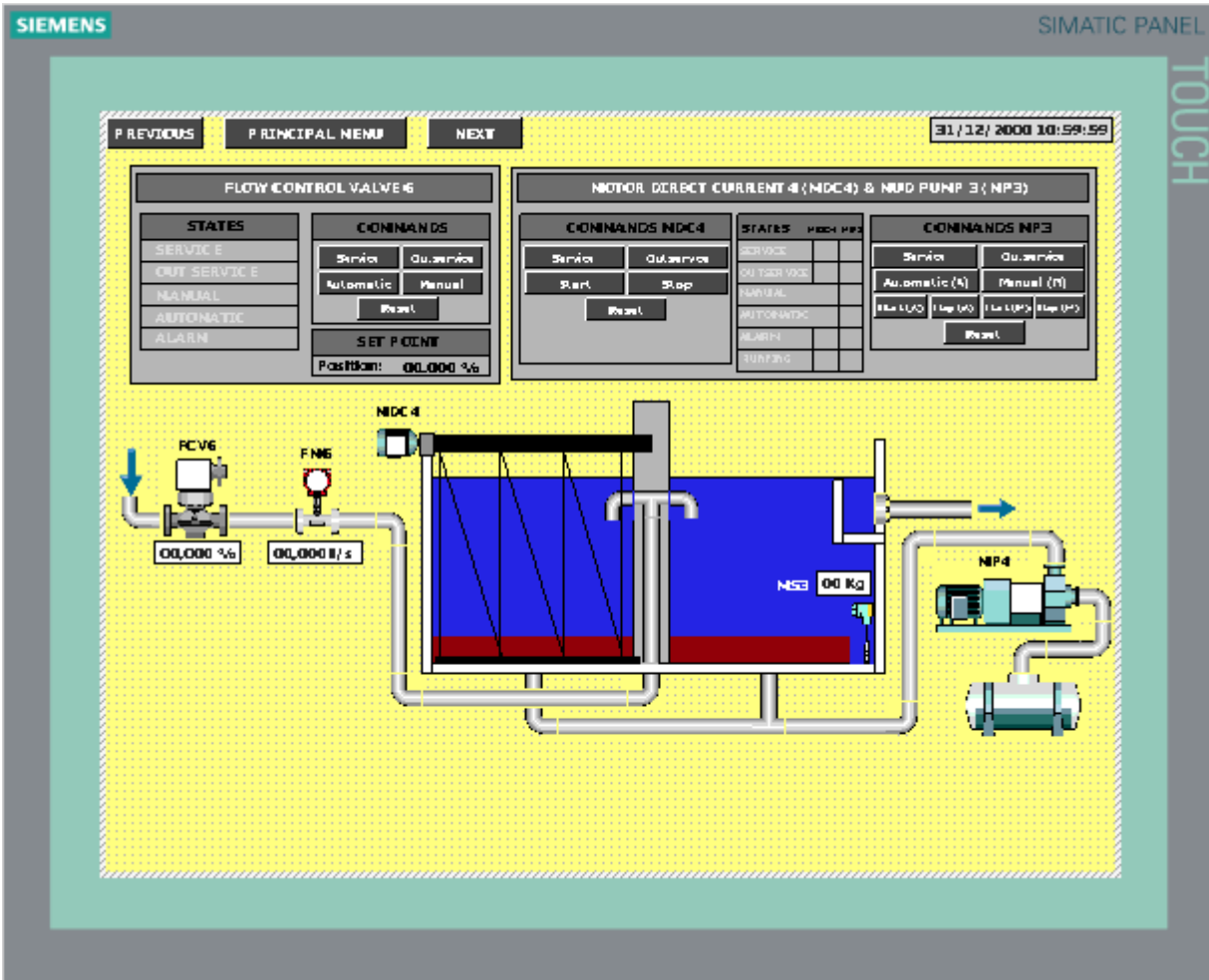
The other controller of this image is to control the Side Channel Blower 1 (SCB1) and the Flow Control Valve 5 (FCV5). In the left side we can see the different states of each actuator, when the state is ON, the little square next to letters lets black. Below the states indicators the Set Point of FCV5 can be seen. In the right side, the buttons to send the commands of SCB1 and below that, the buttons to send the commands of FCV5.



In the lasts images we can see the Side Channel Blower 1 (SCB1) with his respective Oxygen Sensor 1 (OS1) and Flow Control Valve 5 (FCV5).

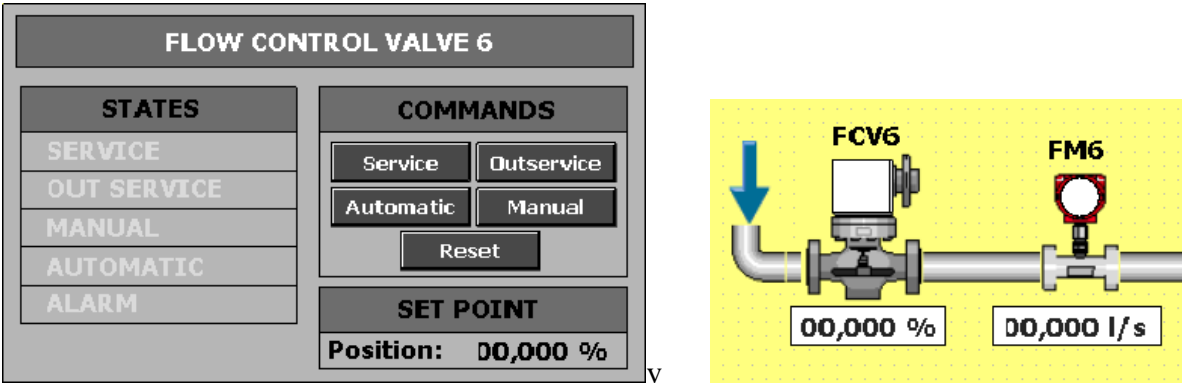
### iii.vii Secondary treatment 2

In this image we can see the second decantation of the process. To accomplish that, it needs three actuators, a Flow Control Valve (FCV6), to control the water flow that enter in the first decantation tank, one Motor Direct Current (MDC4) to turn the tool which helps the water to be deposited in the bottom of the tank and a Mud Pump (MP3) to remove the mud deposited in the bottom of the tank. In the upper left corner we find three buttons, one to go to the previous image, another to go to the principal menu and the other to go to the next image.

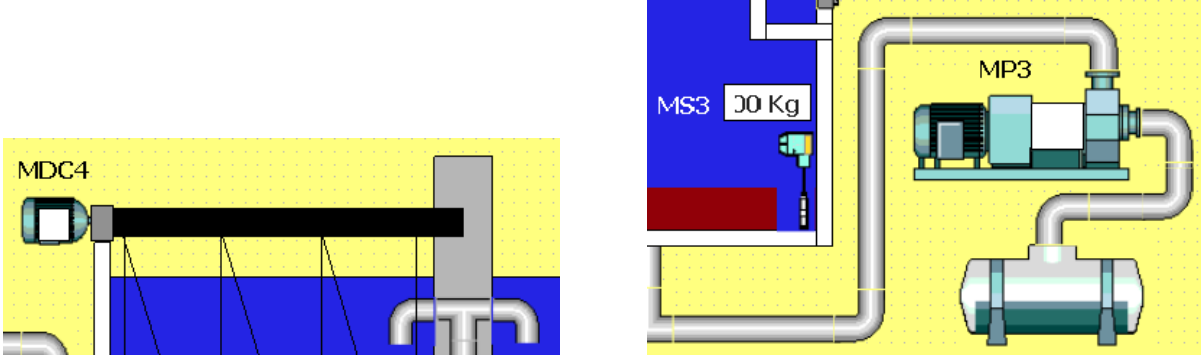
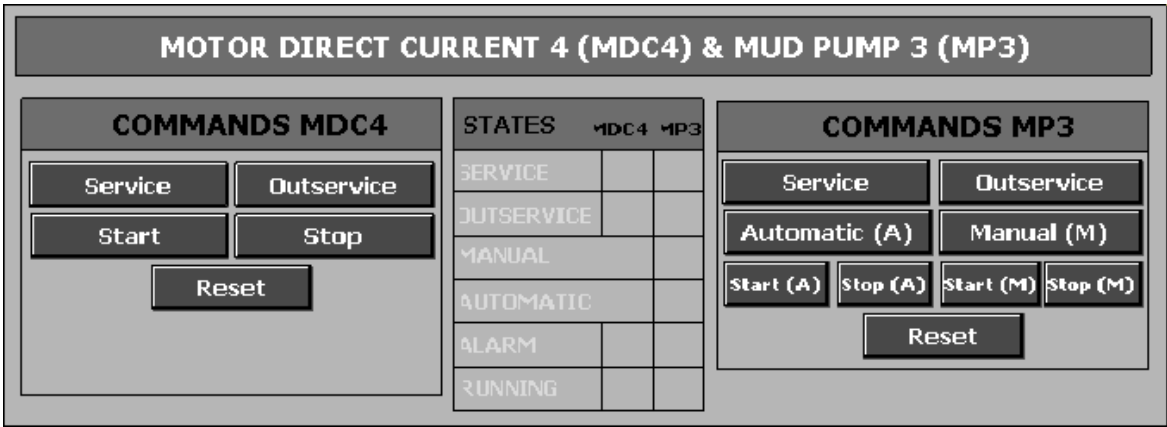




In the controller of FCV6 we can see the states in the left part. In the right side it has the buttons commands and below them, the set point percentage, that is an input to send to the PLC the opening percentage desired for the FCV6.



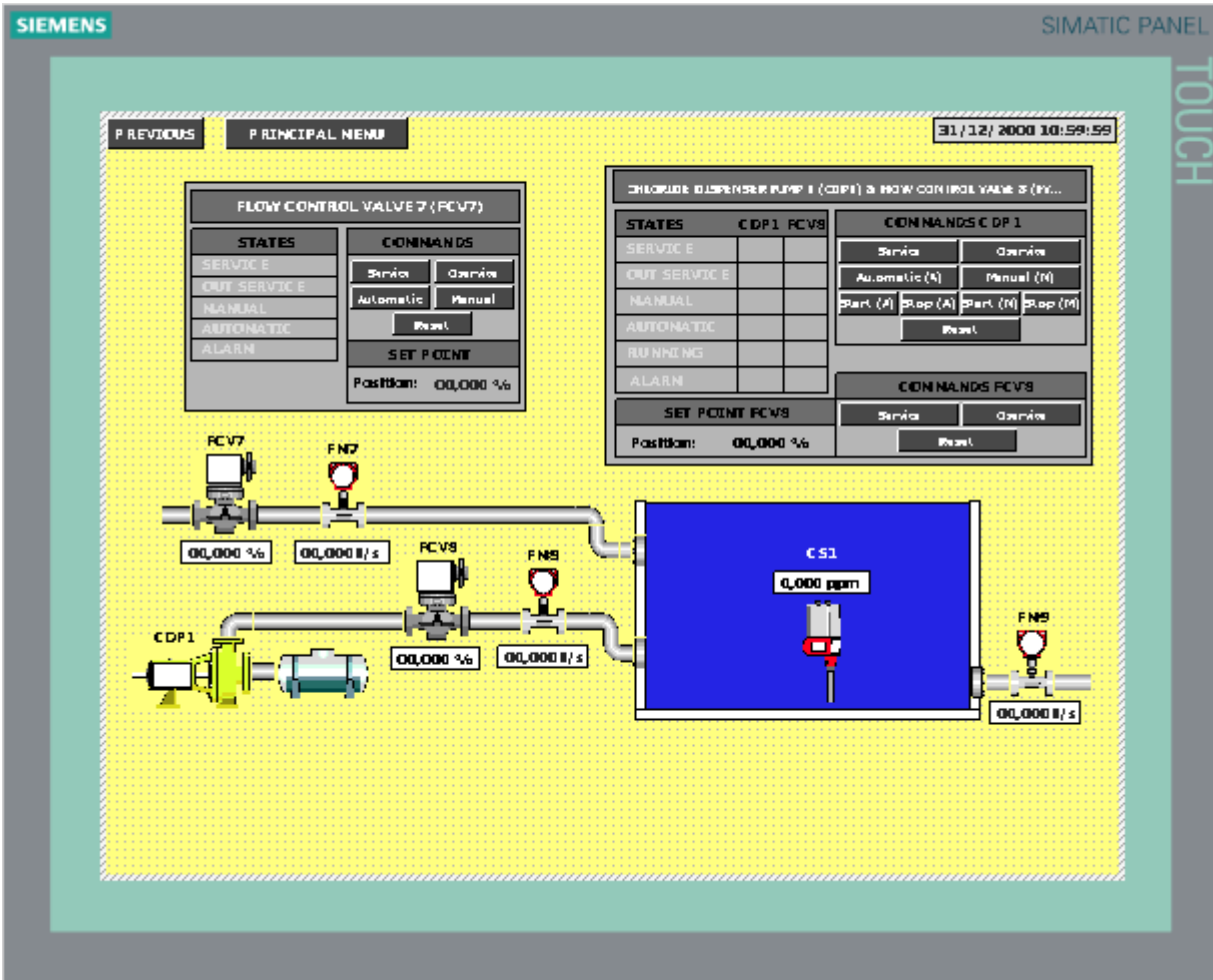
The other controller of this image is to control the Motor Direct Current 4 (MDC4) and the Mud Pump 3 (MP3). In the left side we can see the buttons to send the commands of MDC4 to the PLC. In the middle, the different states of each actuator, when the state is ON, the little square next to letters become black. Finally, in the right side, we can see the buttons to send the commands of MP3 to the PLC.



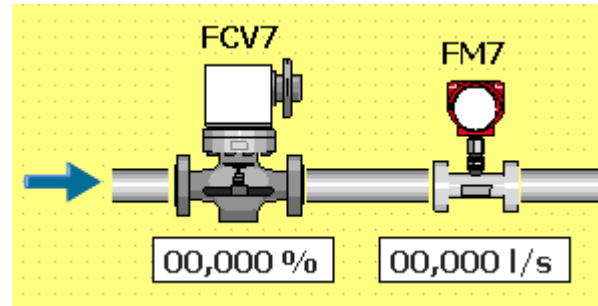
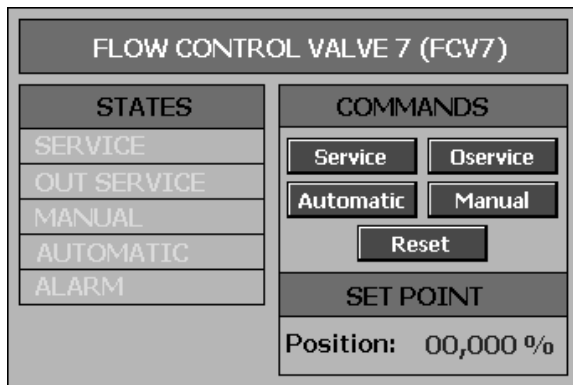
In the last images we can see the Motor Current Direct 4 (MDC4) and Mud Pump 3 (MP3) with his respective Mud Sensor 2 (MS2).

### iii.viii Tertiary treatment

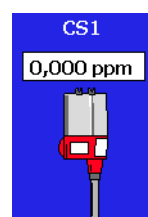
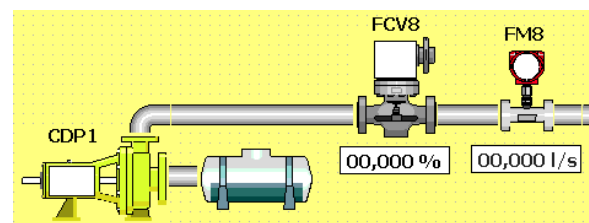
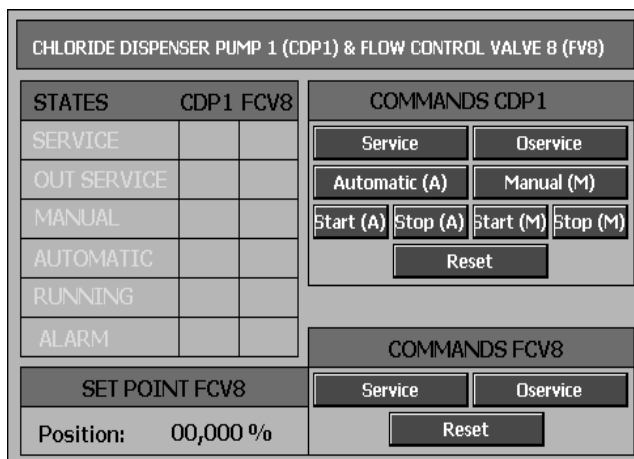
In this image we can see the last stage of this waste water treatment. To do that, it is necessary a Flow Control Valve (FCV7) to regulate the water flow that enter in the last tank, a Chlorine Dispenser Pump (CDP1) to inject chlorine to the water before come back to the river and his respective Chlorine Senor (CS1), and finally, another Flow Control Valve (FCV8) to control the quantity of chlorine injected in the water. In the upper left corner we find two buttons, one to go to the previous image and another to go to the principal menu.



In the controller of FCV7 we can see the states in the left part. In the right side it has the buttons commands and below them, the set point percentage, that is an input to send to the PLC the opening percentage desired for the FCV7.



The last controller of the process is to control the Chlorine Dispenser Pump 1 (CDP1) and his respective Flow Control Valve (FCV8). In the left side we can see the different states of each actuator, when the state is ON, the little square next to letters become black. Below the states indicators we found the Set Point of FCV8. In the right side, we can see the buttons to send the commands of SCB1 and below that, the buttons to send the commands of FCV8.

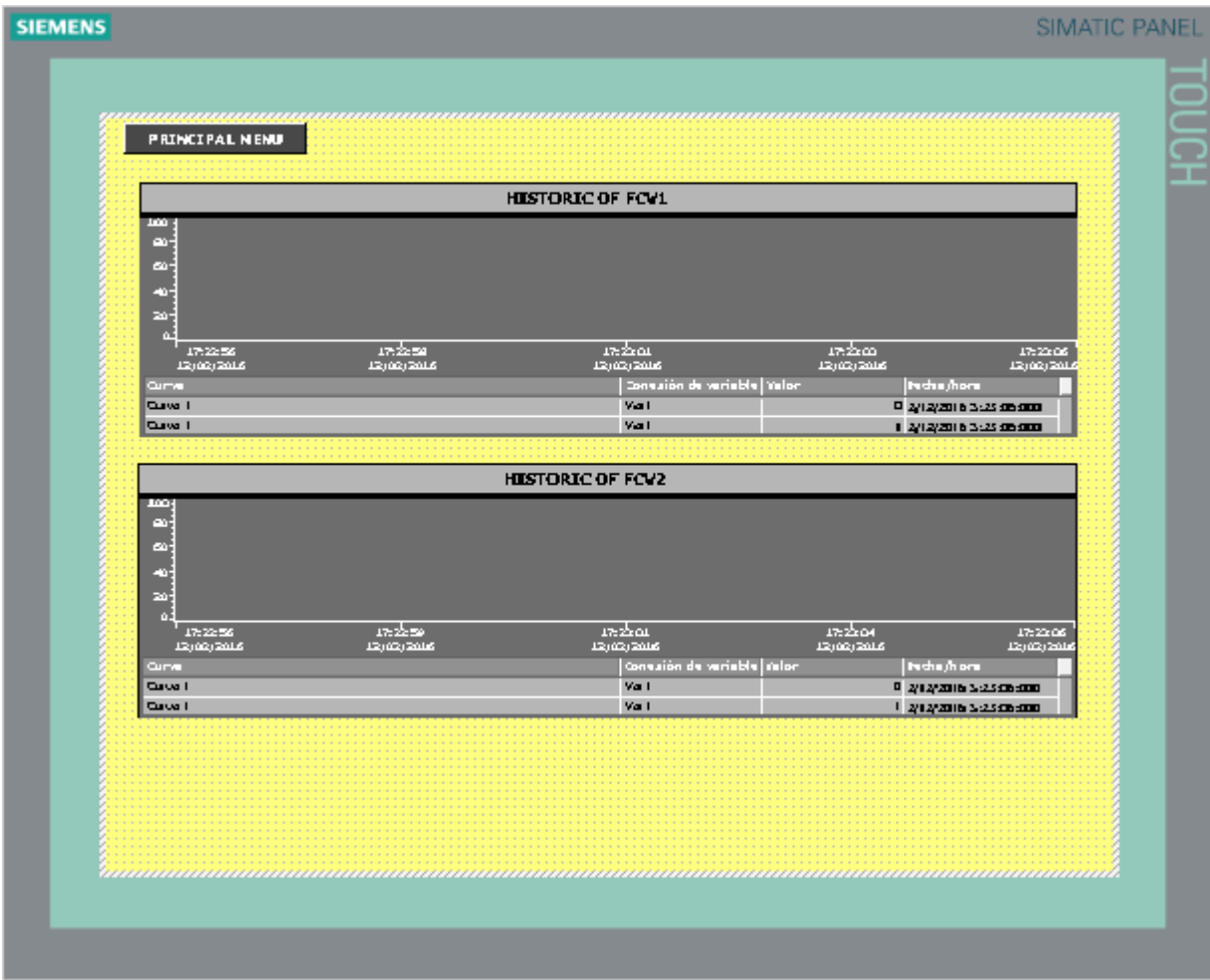


In the lasts images we can see the Chlorine Dispenser Pump 1 (CDP1) with its respective Chlorine Sensor 1 (CS1) and Flow Control Valve 8 (FCV8).

**iii.ix Visor of curves FCV1/FCV2**

This image is just to show the evolution of the variables of Flow Control Valve 1 (FCV1) and Flow Control Valve 2 (FCV2). In the upper left corner we find one button to go to the principal menu. In this graphics three variables can be seen:

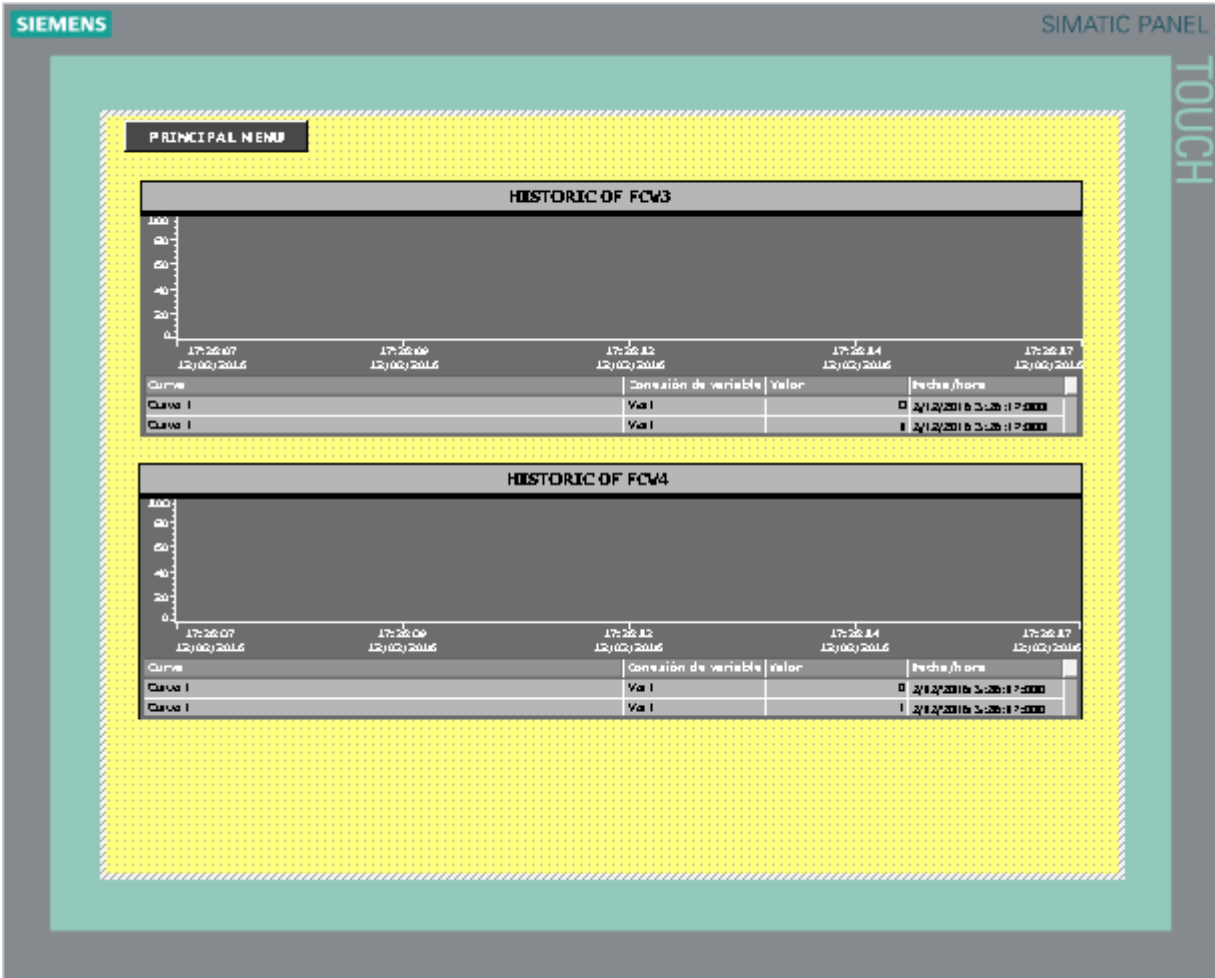
- SET POINT: the Set Points sent, is the white curve.
- PROCESS VALUE: The Real number simulated by the PID, is the red curve.
- MANIPULATED PROCESS: The pulse generated by CONT\_C to control the Process Value, is the blue curve.



**iii.x Visor of curves FCV3/FCV4**

This image is just to show the evolution of the variables of Flow Control Valve 3 (FCV3) and Flow Control Valve 4 (FCV4). In the upper left corner we find one button to go to the principal menu. In this graphics three variables can be seen:

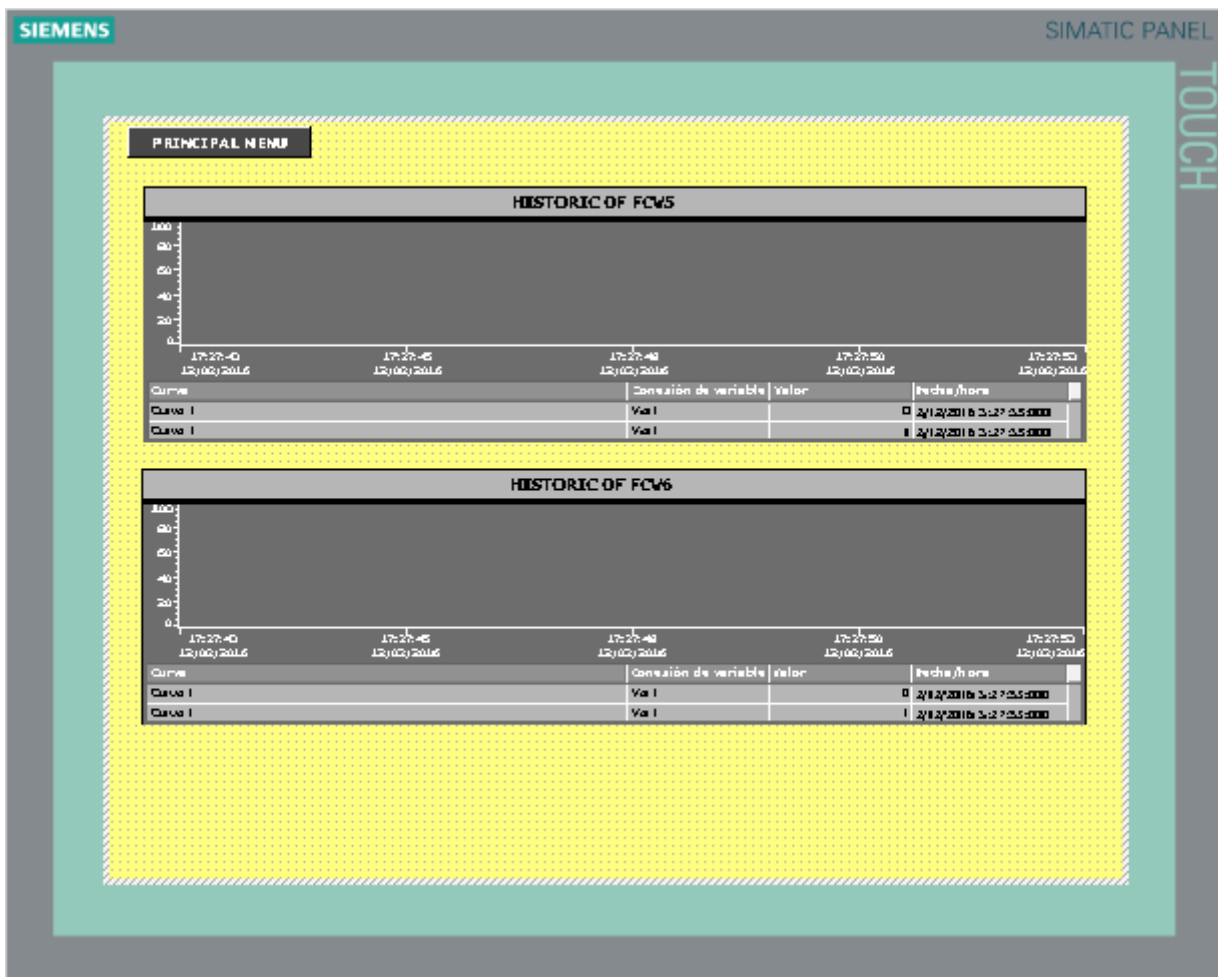
- SET POINT: the Set Points sent, is the white curve.
- PROCESS VALUE: The Real number simulated by the PID, is the red curve.
- MANIPULATED PROCESS: The pulse generated by CONT\_C to control the Process Value, is the blue curve.



### iii.xi Visor of curves FCV5/FCV6

This image is just to show the evolution of the variables of Flow Control Valve 5 (FCV5) and Flow Control Valve 6 (FCV6). In the upper left corner we find one button to go to the principal menu. In this graphics three variables can be seen:

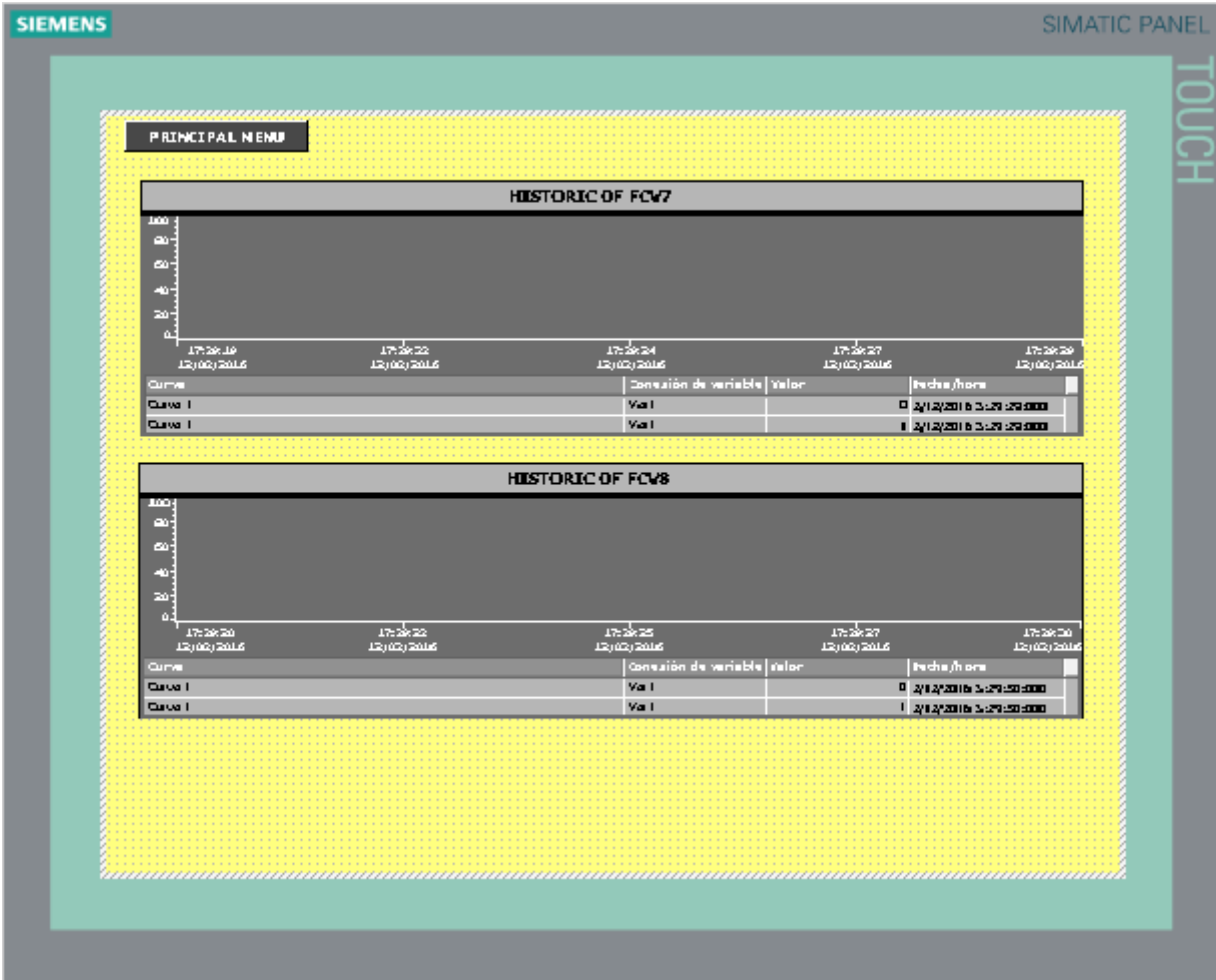
- SET POINT: the Set Points sent, is the white curve.
- PROCESS VALUE: The Real number simulated by the PID, is the red curve.
- MANIPULATED PROCESS: The pulse generated by CONT\_C to control the Process Value, is the blue curve.



**iii.xii Visor of curves FCV7/FCV8**

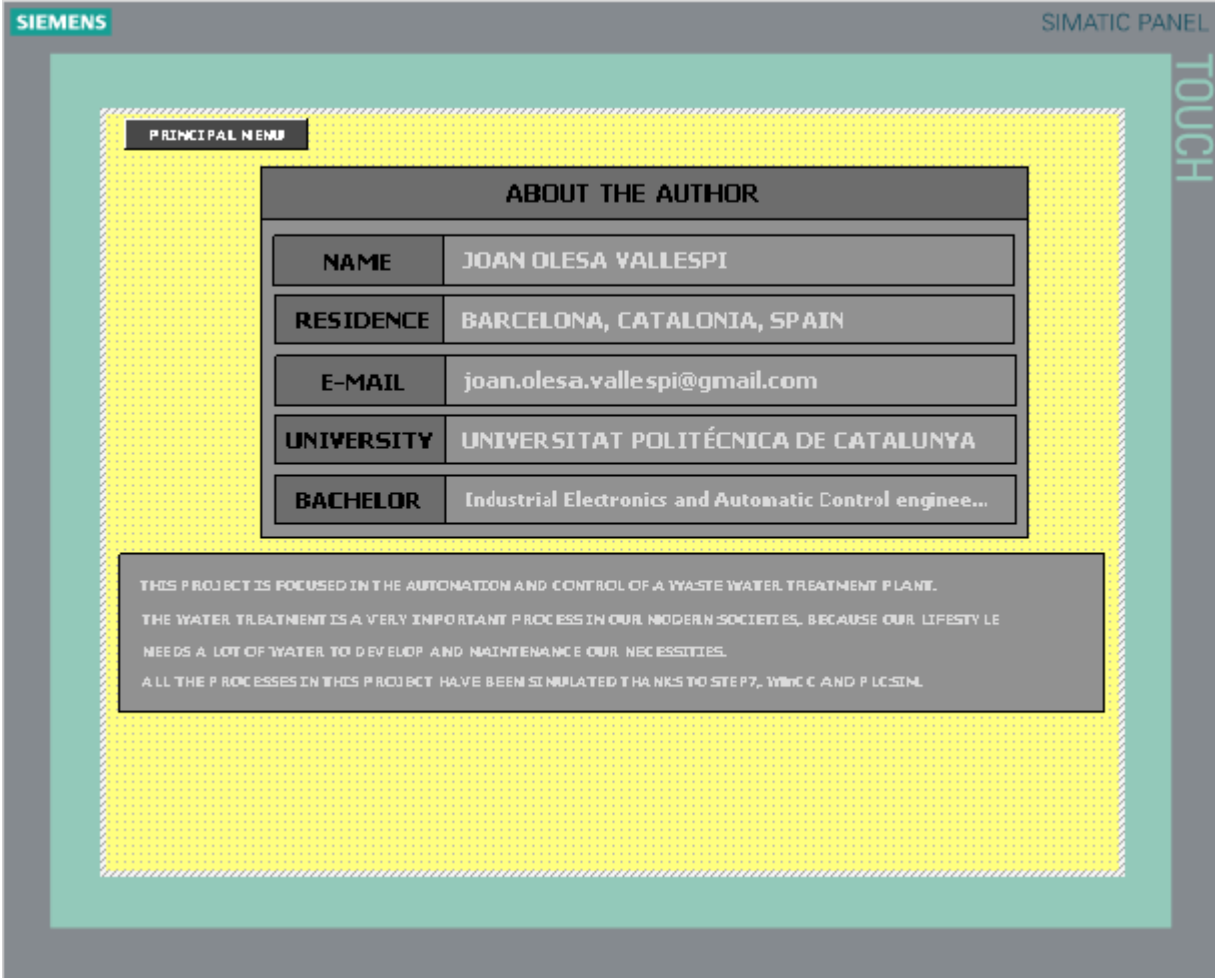
This image is just to show the evolution of the variables of Flow Control Valve 7 (FCV7) and Flow Control Valve 8 (FCV8). In the upper left corner we find one button, one to go to the principal menu. In this graphics three variables can be seen:

- SET POINT: the Set Points sent, is the white curve.
- PROCESS VALUE: The Real number simulated by the PID, is the red curve.
- MANIPULATED PROCESS: The pulse generated by CONT\_C to control the Process Value, is the blue curve.



iii.xiii Information

In the last HMI image, we can see some information about the author and the process. In the upper left corner we find one button, one to go to the principal menu.





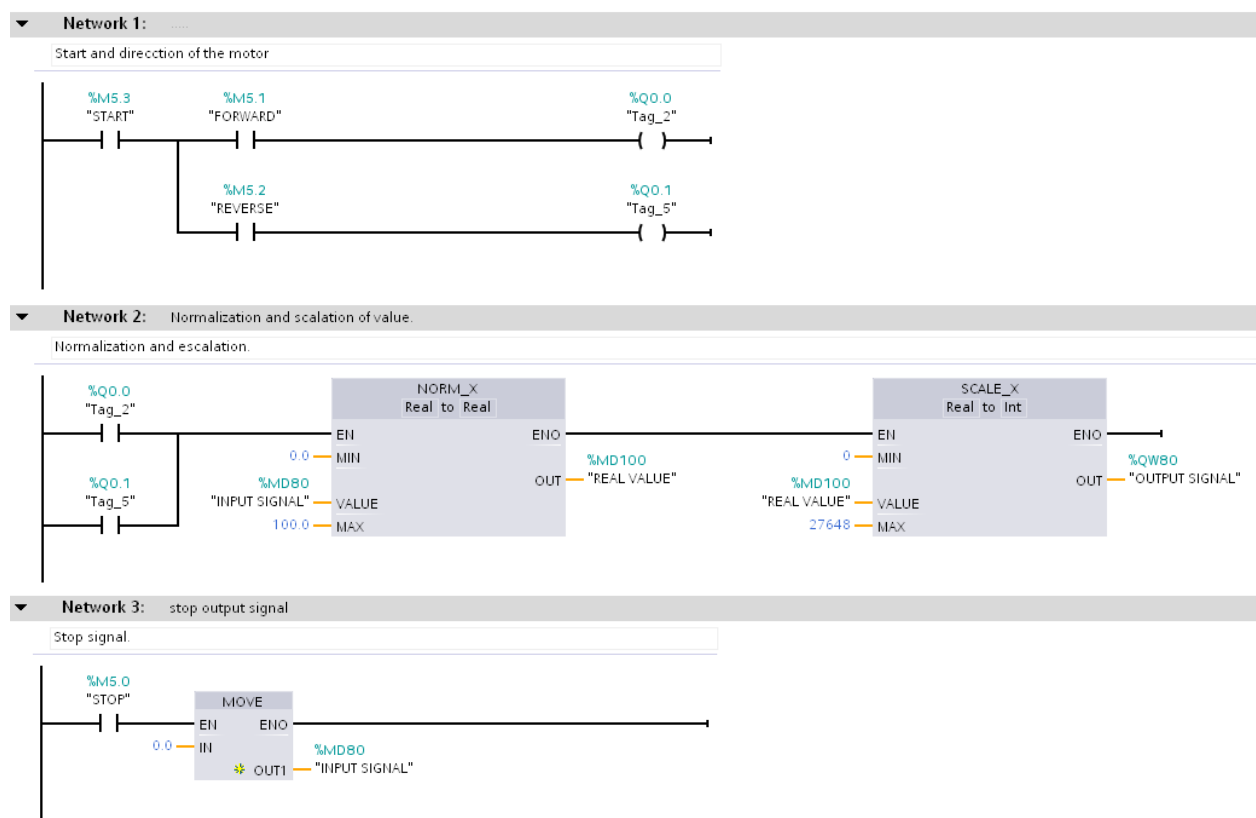
## ANNEX iv

In the fourth annex of the thesis, the program done for the real PLC will be explained. It is a program to control an analogic output (velocity percentage of the motor). The Analogic output is connected to a variable velocity drive who works between 0 and 10 V and that is connected to a motor.

Firstly, the START order is sent from the KTP600 Basic to the PLC, the next step is to choose the direction of the motor, FORWARD o REVERSE. When the direction is chosen, we can send the velocity percentage and the motor starts to work.

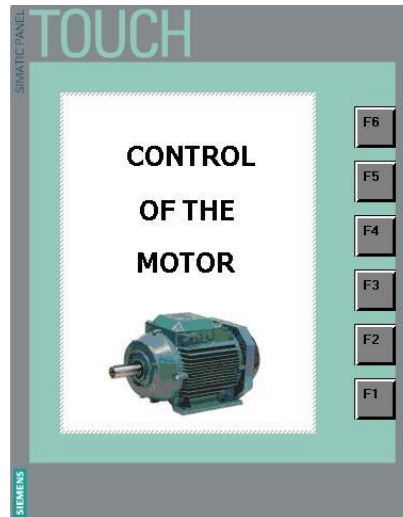
The value (velocity) is normalized from 0% to 100% and then, escalated from 0 (0V) to 27648 (10V). The analogic output could be in tension or in current, in this case we are working in voltage.

Finally, we have a STOP order that when is sent to the PLC, we send a 0% to the normalized value (INPUT SIGNAL).



## ANNEX v

This annex is to explain the functioning of the Real HMI KTP600. It is a simple program to control the Motor formed by two images. The first is a simple presentation image to go to the motor control; the buttons F1 to F6 have no function. The entire image is a button to go to the control motor image:



The CONTROL MOTOR image is to control the functioning of the motor, we can see five buttons.

- **Back:** this button is just to come to the principal menu.
- **START:** This button is to send the “START” command to the PLC.
- **FORWARD:** To choose the Forward direction of the motor.
- **REVERSE:** To choose the reverse direction of the motor.
- **STOP:** This button is to send the “STOP” command to the PLC.

Then, we can see an input to send the velocity percentage to the PLC.

