POLITECNICO DI TORINO

Master of Science
in
Mechatronic Engineering

Automatic Control, PLC Programming and Simulation of Pneumatic Cylinders' Lifetime Testing

Supervisor:

Professor Luigi Mazza

Candidate:

Alireza Qadiri

2020-2021

Contents

Abstract

In this thesis project, programming and simulating of the automatic testing process has been studied. The ambition is to develop an efficient test bench to determine the durability of pneumatic cylinders in working conditions.

Pneumatic cylinders have extensive application in different industries. For safety and financial reasons, just to name the most important, anticipation of their lifetime is crucial. To estimate the life service of a pneumatic cylinder, applying the progressive methods of testing and reliable mechanism is necessary.

The programmable logic controller (PLC) has a specific role in automation world. I have used Siemens' engineering framework, the TIA Portal, to control the test bench. Data acquisition is a key purpose in this project. Unlike previous studies related to the pneumatic cylinder lifetime testing, I have used a different way to record, save and analyze the data, namely by using the trace function in the TIA Portal. Previous solutions done by other students had some major limitations. One is in the ability to acquire and store the data. They needed to use an additional component like HMI or a PC that always had to be connected to the PLC CPU. Another limitation was the lack of analyzing tools. However, with the method used in this study there is no need to use any extra component to save the measured values. In addition, the trace function has the capability of creating the diagrams and analyzing the data.

There are some crucial measurements that can be achieved by two different tests; a wear test to measure the number of complete movements and a leak test to investigate any leakage inside the cylinder chambers. These two tests can be done either individually or successively one after another.

The outcome of this project is to learn how to assemble a test bench to perform different testing on pneumatic cylinders automatically and how to program and organize it in a proper manner for controlling the project. And finally, how to simulate the project, diagnose and develop the program virtually.

In this project I have used the TIA Portal V16, PLCSIM Advanced V3.0 and FluidSim Pneumatic for PLC programming and simulation of the test process. The program is developed in such a way that the major settings can be redefined by users according to the conditions and more cylinders can easily be added to the test bench.

And the results show that these programs and this setup are suited for this kind of testing and that a proper method can be found using it.

Introduction

The project consists of two major parts: hardware, including the test bench that is a combination of pneumatic and electrical systems, and software which operates and controls the different test processes. In the first section I will look at the hardware part, the different components of the test bench and how they are working.

In the second section the software part will be explained in detail, starting with the FluidSim Pneumatic software that is used to design the electro-pneumatic circuit and simulate the test bench. Then there is an overview of the PLCSIM Advanced software and why it is necessary for this project. Then the TIA Selection Tool and the advantages of applying this software in the project will be clarified. And finally, the most important part of the project, programming in the TIA Portal engineering framework, will be described in detail. The procedure of creating the project in the TIA Portal and programming is illustrated step by step. Then a study of the data acquisition system has been made using “Trace function”. The trace function is a dynamic capability of the TIA Portal to record, collect and analyze the data.

In the software section figures are used to help the reader better understand the explanations.

The test program has three modes. In the first mode the leak test is executed using high-pressure air. The next mode is the wear test which uses low pressure air. The third state is the full test mode in which both the wear and leak tests will be executed in succession. All these three test modes will be done automatically. It is crucial to predict possible faults that may happen during each test mode. For some errors it suffices to inform the operator, but in some cases the defective cylinder should be removed from test process. A function named Alarm Codes is defined for this purpose. In this function different errors and proper actions are considered. In this project the main language which is used in PLC programming is LAD. However, to simulate pressure dropping during the leak test SCL coding is used as well. For simulation of pressure decreasing, an exponential function is used. SCL coding is more suitable when using mathematical calculations.

In the third section the conclusions drawn from this project and its results are mentioned as well as their applicability in the industry. At the end the list of references which I used in my thesis is shown. The last section includes the complete program of the testing project in the TIA Portal.

A. Hardware

1. Electro-Pneumatic Systems

1.1 Pneumatics in industries

Automatic systems have a key role in today's world and one of the technologies which is commonly used in this area is pneumatics. Electro-pneumatic systems consist of two main parts: power and control. The power section is driven by a pneumatic circuit and the control section is driven by an electric circuit.

A pneumatic system consists of interconnected components using compressed air to transmit and control energy in mechanical works and automated processes. The flexibility and efficiency of this system makes it very useful for a large range of industries and can be found in various places, such as train doors, dentist offices etc.

1.2 Main and basic components

Some basic components are needed in pneumatic machines in order to make, control, store, move and use compressed air:

- A compressor which collects air.
- Receiver for storing air.
- Valves for controlling air.
- Some kind of circuit for moving air between the different parts of the system.
- An actuator which is activated by air to do requested job.

As the name indicates, pneumatic systems get their power from compressed air. In order for such a system to work two main parts are needed: something that compresses the air, called compressor, and something that can use the compressed air to hold, move or lift an object, called the actuator. Some kind of circuit, usually comprised of a network of pipes, is also needed in order to move the air from the compressor to the actuator. In order to control the system a valve is needed in order to switch the air on or off. In some cases, the valve should also be able to change directions, otherwise the machine is unable to lower something that it has lifted, or vice versa.

Another important part of a pneumatic system is the receiver where compressed air is stored under pressure, ready to be used instantly as soon as the operating valve is opened. Without this and because of compressibility of the air, it would take time for the compressor to achieve the pressure needed to make the actuator move.

Compressors

As mentioned above, the compressor is the starting point for any pneumatic circuit that compresses air, getting it to about seven to ten times the atmospheric pressure, 700-1,000 kPa, or 100-150 psi. This pressure is what generates the energy needed for the pneumatic system, and it is done by compressing air into a smaller space. However, energy does not come out of thin air, the compressor needs to be powered by a diesel or gasoline engine, thus converting one form of energy to another.

Actuators

An actuator is something that moves something else. It is the part of a pneumatic tool that does some kind of action by lifting or lowering an object, for instance a mechanical arm, a piston etc. They usually work in a straight line and they are often powered by pistons that slide in and out of cylinders as they are moved by compressed air, thus turning potential energy from the compressed air into kinetic energy and some kind of movement.
In order to get a rotating motion an air-powered motor can be used. This uses gas to make shafts rotate, in very much the same ways as a turbine which can be seen as a machine that uses an internal windmill. When the gas moves through this kind of air motor it pushes against vanes, thus making an axle spin in order to make a drilling motion, such as in a polishing machine.

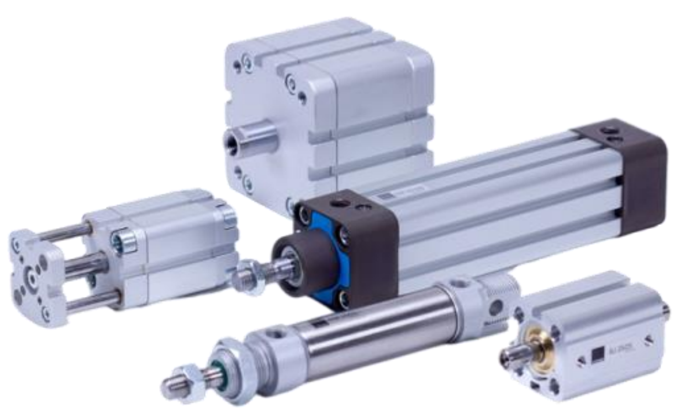


Figure 1

Circuits

Above we have described how a pneumatic system works, but in most industrial contexts many circuits coexist and need to be able to work together. They consist of different kinds of valves and actuators, and in a large factory one large compressor could be the energy source for a large range of machines, meaning that there are many complex circuits. There are dozens of pneumatic symbols to help drawing circuits clearly on the engineering plans, such as symbols for electronic and hydraulic components.
There is some engineering software to make very advanced logic circuits entirely from fluid-powered components in which fluidic components control complex process by changing the flow of air around analogous components. Like electrical circuits, there are fluid equivalents of AND/OR logic gates, timer circuits, latching units, switches, amplifiers etc.

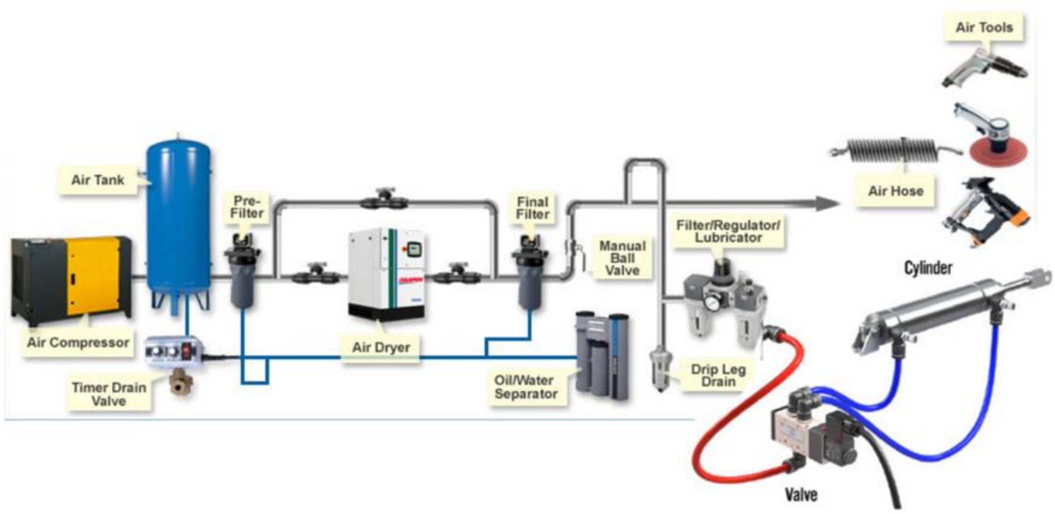


Figure 2

A pneumatic circuit can use many different components depending on the purpose. For instance:

- A timer drain valve drains the condensated water from the air tank when instructed to. But air alone cannot remove water and impurities from compressed air and therefore other components are also needed in order to increase the life span of the pneumatic components.

- A pre-filter is used to remove most water molecules as well as impurities and this filter is usually installed in the main line.

- An air dryer is installed in order to remove water vapor from compressed air. In order to dry compressed air, it is possible to lower the temperature and then increase it again, making condensation form at the lower temperature, allowing it to be discharged. Re-heating dries the air. This process is particularly efficient in refrigerated compressed air dryers.

- Air filter, regulator, lubricator:
  These can be used together or separately. They contribute to different cleaning: air filters clean the air from dust and vapor, a regulator decreases the pressure and air lubricators ensure that oil is added into the compressor air.

Pneumatic valves

Pneumatic valves govern the pressure, direction or flow of compressed air and they are usually divided into directional control valves (distributors), pressure regulating valves, and auxiliary valves.

Their function is to operate actuators and distribute compressed air in the circuits. Power valves directly distribute the fluid used to operate pneumatic actuators while control valves govern the fluid that controls other valves. Different kinds of ports or openings allow air to pass.

There are several types of valves depending on their manner of operation, such as diaphragm valves, poppet valves, rotary distributor valves and slide valves. These can control the pressure, direction, and flow of air. These can further be categorized based on their function such as pressure regulating valves, auxiliary valves, and directional control valves.

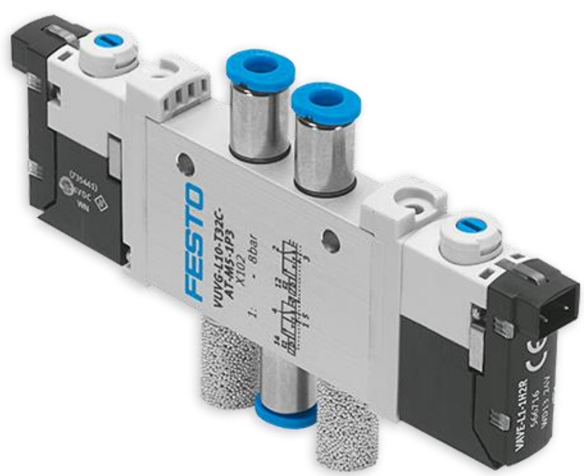


Figure 3

1. Directional control valves can direct fluid along different paths by switching the connections of an actuator, reversing a motion and stopping it. These valves are categorized based on the number of ports and positions. Since they are digital valves, usually have discrete number (two or three) of positions.

2. Pressure regulating valves are mainly used to control and reduce air pressure to ensure that it is constant and safe for the circuit at the output of the valve.

3. Auxiliary valves perform various functions associated with other valves and they can consist of selector valves, sequence valves, nonreturn valves, flow control valves and time delay valves.

Valves are either monostable or bistable. A monostable valve needs a signal for switching and without a signal it will return to its previous position. A bistable valve has two operating positions, without having a preference. When the valve moves after having received a signal it will retain this position until another signal comes (memory function).

Some common types of valves which are named according to their number of ports and switching positions:

- 2 way / 2 position valves
- 3 way / 2 position valves
- 4 way / 2 position valves
- 5 way / 2 position valves
- 4 way / 3 position valves

1.3 Electrical

In the electric parts of electro-pneumatic systems various components can be used according to the required functions. Solenoid valves, Relays, Switches, different types of Sensors (mechanical limit switches, magnetic, inductive, capacitive, optical, ultrasonic, etc.), Programmable controllers and so on.

2. PLC

PLC (Programmable Logic Controller) is the heart of the automation section of this project. PLCs are industrial digital computers, which are designed to perform any process automatically and with high reliability. A PLC is equipped with a programmable microprocessor that receives electric signals in inputs and according to the user program, sends commands to the outputs. Inputs are connected to the sensors or transducers and outputs are connected to the actuators in the system which it controls. In this project I used a CPU from Siemens S7-1500 family.

Figure 4

PLC Advantages

The PLCs are compact and reliable as they have no moving parts, and this make them particularly suitable for use in the industry as they operate safely. They are also modular and can therefore be configured as needed and maintained easily. This makes them flexible and changes to the process cycle can be made instantly. Despite of their simplicity they are able to handle very complicated logical operations. They have the capability to program counter, timer, comparator etc. and they possess powerful facilities when it comes to fault finding, diagnostics and documentation. In addition, they can handle analog signals and close loop control programming and can be connected to printers, video terminals, computers, other PLCs, displays etc. Finally, they can be reused and have a low power consumption making them compatible with an equivalent hard-wired logic system.

In this section we have looked at the different parts of an electro-pneumatic system and in the next segment the test bench for this project will be presented.

3. Test bench

In the hardware part of the test bench used in this project two double acting cylinders are used. For each cylinder two blocking valves (one for each chamber) and one distribution valve are used. The type of the blocking valves and the distribution valve are 2/2 and 5/2. In both cases the valves are monostable. The distribution solenoid valve will be actuated electrically while the blocking valve can only be activated pneumatically. There is one auxiliary solenoid valve to control all posterior blocking valves and one for all anterior blocking valves in the system. In this system we need two air supplies; high-pressure supply to perform the leak test and low-pressure supply for the wear test and in each part one 3/2 monostable valve at the input of the system will control the supplied air. In figure 5 we can see the pneumatic circuit of the test bench in this project.

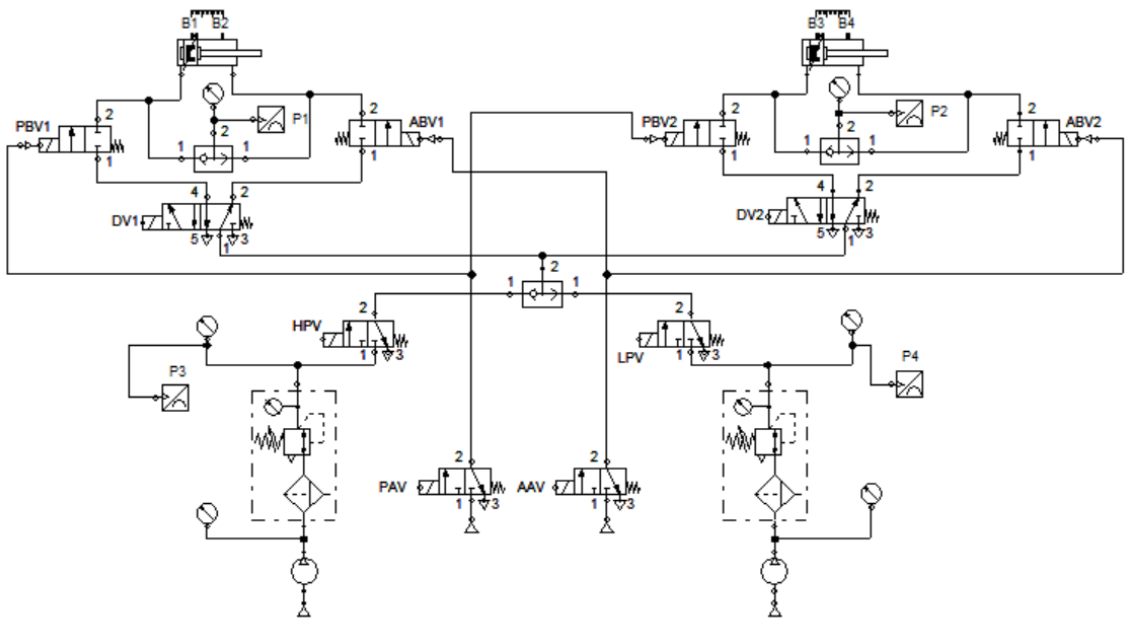


Figure 5

The power supply used in the electrical circuit is 24 volts. To send the commands and control the testing process, some detent switches, pushbuttons, and magnetic limit switches are used. The six modules as shown in figure 6 are related to the digital and analog input/output of the PLC.

Four pressure manometers show pressure inside the chambers and the supplies. Each limit switch is connected to a relay and a "normally opened" contact of the relay is an input signal to the PLC. All solenoid valves are connected to the output modules and receive commands from the PLC.

There are nine LEDs connected to the PLS' output and each of these are in their turn related to a different fault happens in the system.

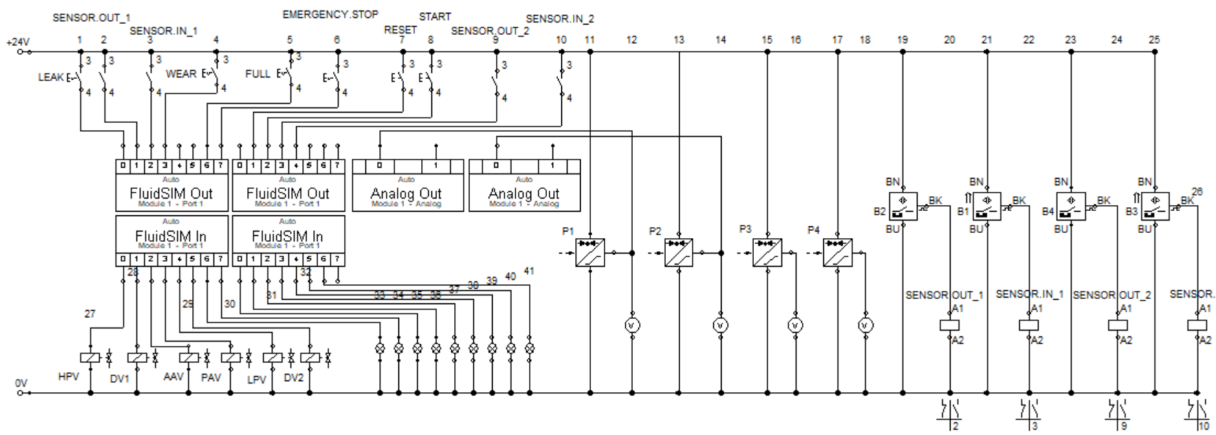


Figure 6

In the first part and until here we have discussed the main components required in the hardware section. The electro-pneumatic circuits are designed, and the hardware configuration is completed. We now have the base for our project and in the next part we will look at the different kinds of software that are used in this project, and in the subsequent part we will be looking at the programming and simulation of the test bench.

B. Software

4. FluidSim

Fluidsim is a robust software for simulating pneumatic, hydraulic, and electrical circuits. To simulate my project, I have used Fluidsim version 4. By means of the Fluidsim library, we can select our required components and create a project on the related window. There is also the possibility of simulating analog signals like pressure in this project. We can use the help section to find out the software possibility and its various functions.

My project in Fluidsim is a combination of pneumatics and electrical circuits. After designing the circuit, we can start the simulation. In fact, the Fluidsim plays the role of hardware in my project. When connecting my PLC program to Fluidsim I used OPC mode. In order to do so it was necessary to install an EzOPC file. I have used EzOPC V5.6 in my project.

There are different modules for digital and analog signals as inputs and outputs. In the following figure, we can see the project in Fluidsim. Analog out, Fluidsim In and Fluidsim out are three modules which I used to represent my PLC I/Os and its connection to the project hardware.

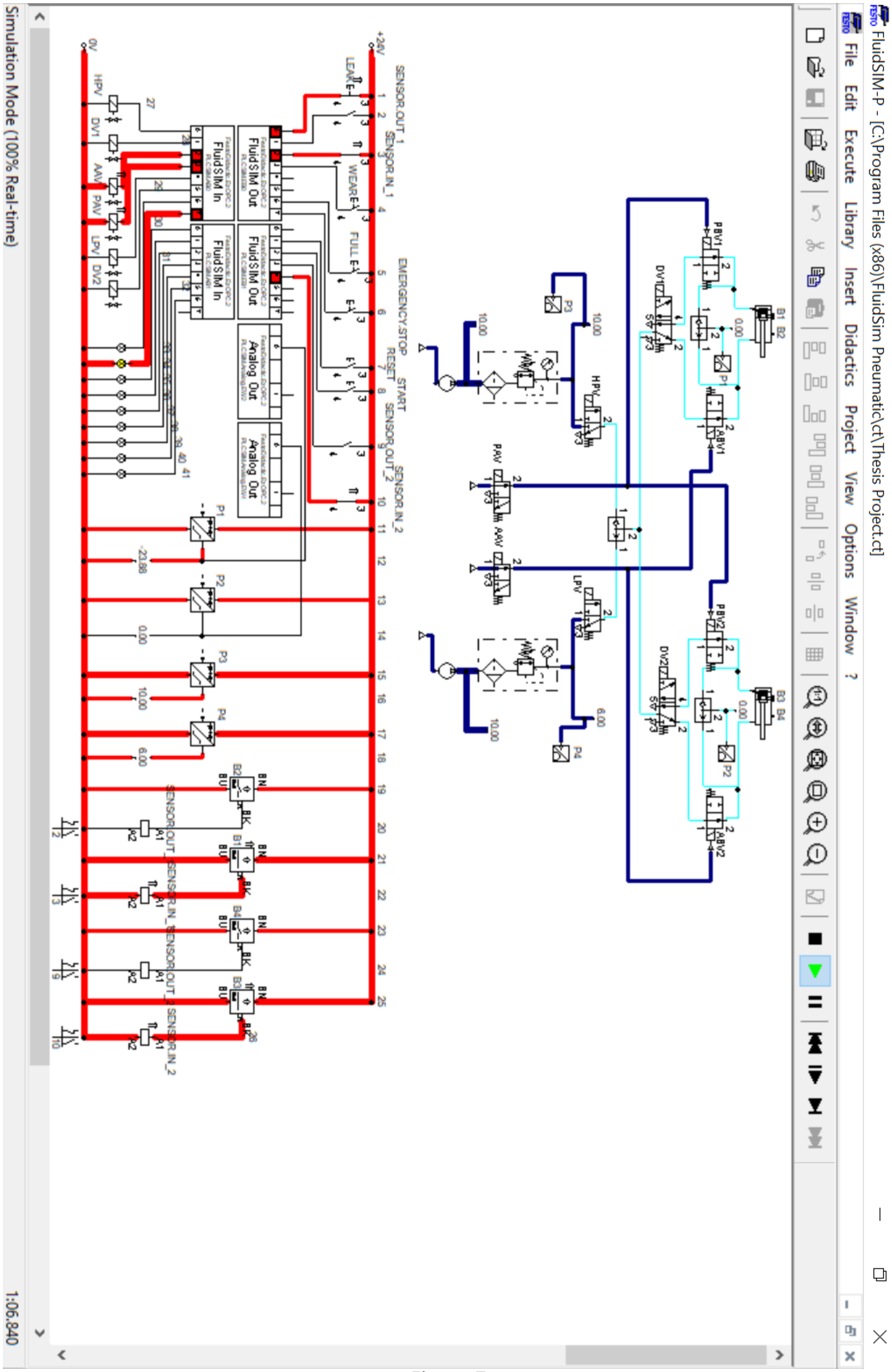

FluidSIM-P - [C:\Program Files (x86)\FluidSim Pneumatic\ct\Thesis Project.ct]
File Edit Execute Library Insert Didactics Project View Options Window ?
Simulation Mode (100% Real-time)
1:06.840

Figure 7

5. PLCSIM Advanced

In this project I have used PLCSIM Advanced V3.0 for simulation. In comparison with PLCSIM, PLCSIM Advanced has some important advantages. It tries to emulate CPU hardware behavior. In addition, it can communicate with other PCs and even with real PLC CPU.

After installing the PLCSIM Advanced, the first step is to check box the “Support simulation during block compilation” option in TIA Portal. It is placed on the “Protection” tab in the properties of the project. Otherwise the TIA Portal cannot connect to PLCSIM Advanced.

S7-PLCSIM Advanced Control Panel

At the top of the control panel, online access, we can switch to adopt the communication interface. If PLCSIM is selected, it can communicate with local via softbus. If PLCSIM virtual ethernet adapter is selected it communicates with the outside via TCP/IP and in the next section, TCP/IP communication will be activated. It is then possible to choose either local or ethernet network for distributed communication.

PLCSIM Advanced works only with CPU 1500 and ET 200SP CPU. To create a virtual controller, it is necessary to enter a unique name for the created “instance”, otherwise the start button will not be enabled.

The retentive data in addition to the hardware configuration, and the user program, will be saved in a file in the local PC. The path can be selected in the virtual simatic memory card section.

After creating an instance, we can load the program from the TIA Portal to the device and start the simulation.

Figure 8 shows the PLCSIM Advanced control panel. To connect FluidSim to the TIA Portal via PLCSIM Advance, it is necessary to install the Festo Didactic EzOPC software. Then in its S7-PLCSIM tab select PLCSIM Advanced and define I/O range. In FluidSim software and in options we have to select OPC mode as well. In Figure 9 the overview tab, PLCSIM tab and the setting options are shown.

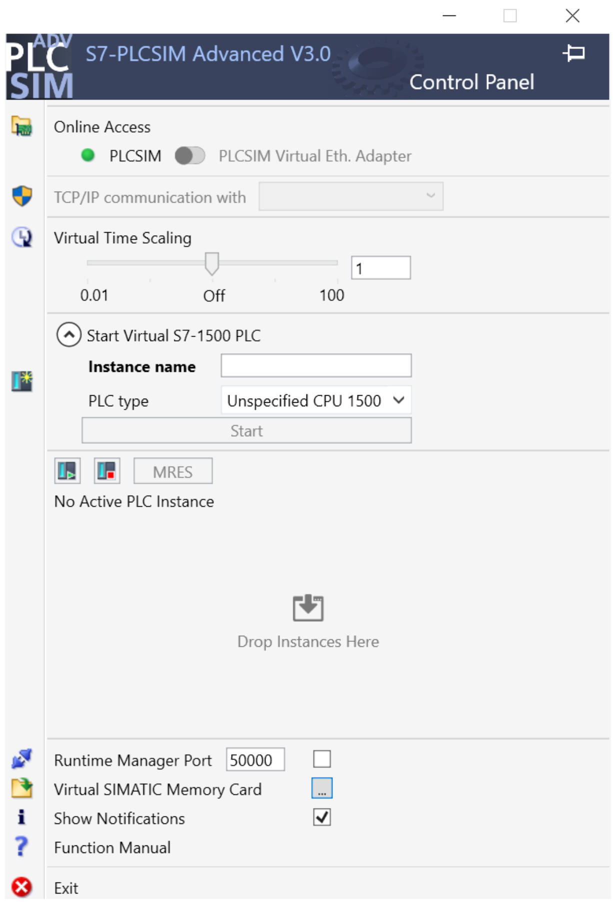


Figure 8

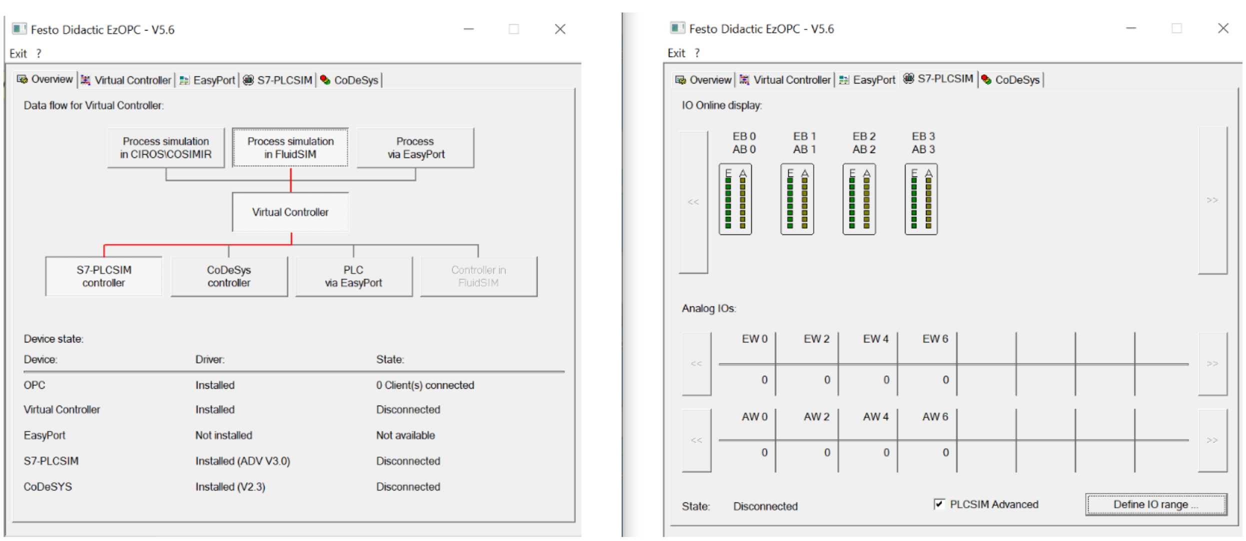


Figure 9

6. TIA Portal

Introduction

With continued technological development, the world of automation is changing as well. Processes need to be faster, tasks are becoming more complex and more and more of operations are becoming digital. To handle these demands, Siemens introduced a very powerful and comprehensive approach. The totally integrated automation portal, or TIA Portal for short, is a modern engineering framework for all digitalized automation tasks. The TIA Portal integrates STEP 7, WinCC, Startdrive, SIMOTION SCOUT TIA, SINAMICS and SIMCODE ES software. Furthermore, TIA Portal contains new functionalities like the TIA Portal Multiuser Engineering and SIMATIC Energy Suite that are designed for energy management. Another important advantage of TIA is digital workflow where it is possible to use a virtual model of machines and simulate, test, and analyze every possible scenario before building the real systems.

6.1 TIA Selection Tool

Starting a project with TIA Portal

The first step to design an automation project is to choose and configure the right hardware. In order to be able to correctly choose the devices required for our project, we can use TIA Selection Tool software, which is independent from the TIA Portal. This software helps users to select and configure components without error in a fast and easy way. As it is shown in figure 10, in the Tia Selection Tool main tab, we select plant configuration and wizards. By using this function and simply enter the number of input/output signals and requested module type, TIA Selection Tool will automatically generate the required modules and devices, including accessories.

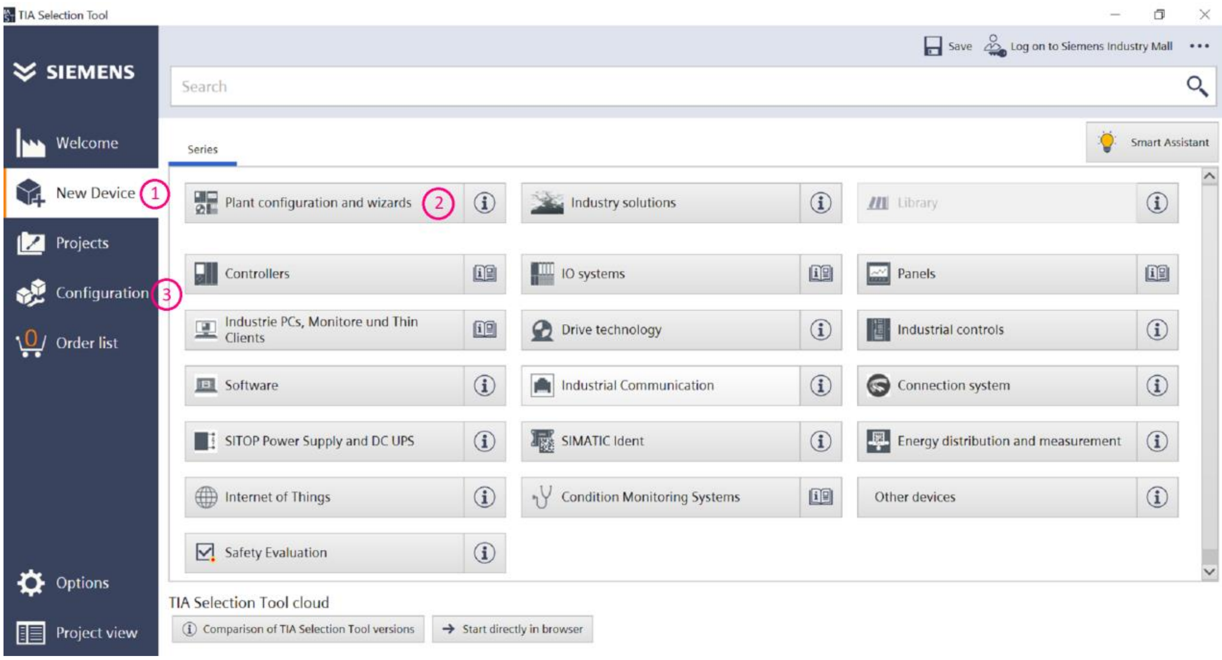


Figure 10

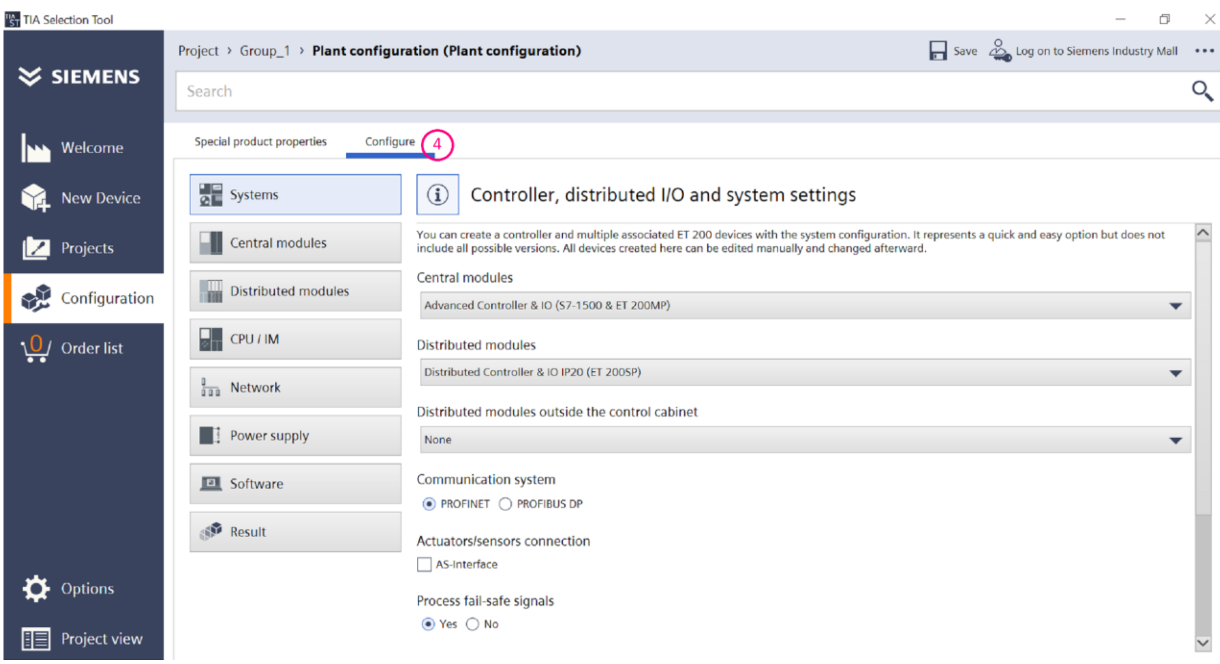


Figure 11

In the configuration section, we choose type of modules, communication system, software, and physical specifications of the required panel. According to our selection, this software gives us different options. In figure 11 we can see these options on the left side. It is possible to click on each of them one by one and set the required fields. For instance, by clicking on central and distributed module and then clicking on add module, a new tab will be opened. Here we can enter the type and number of signals which we want to use in our project, Figure 12. After completing the setting, we click on result. The

TIA Selection Tool computes and analyzes our requirement and shows the result, Figure 13.

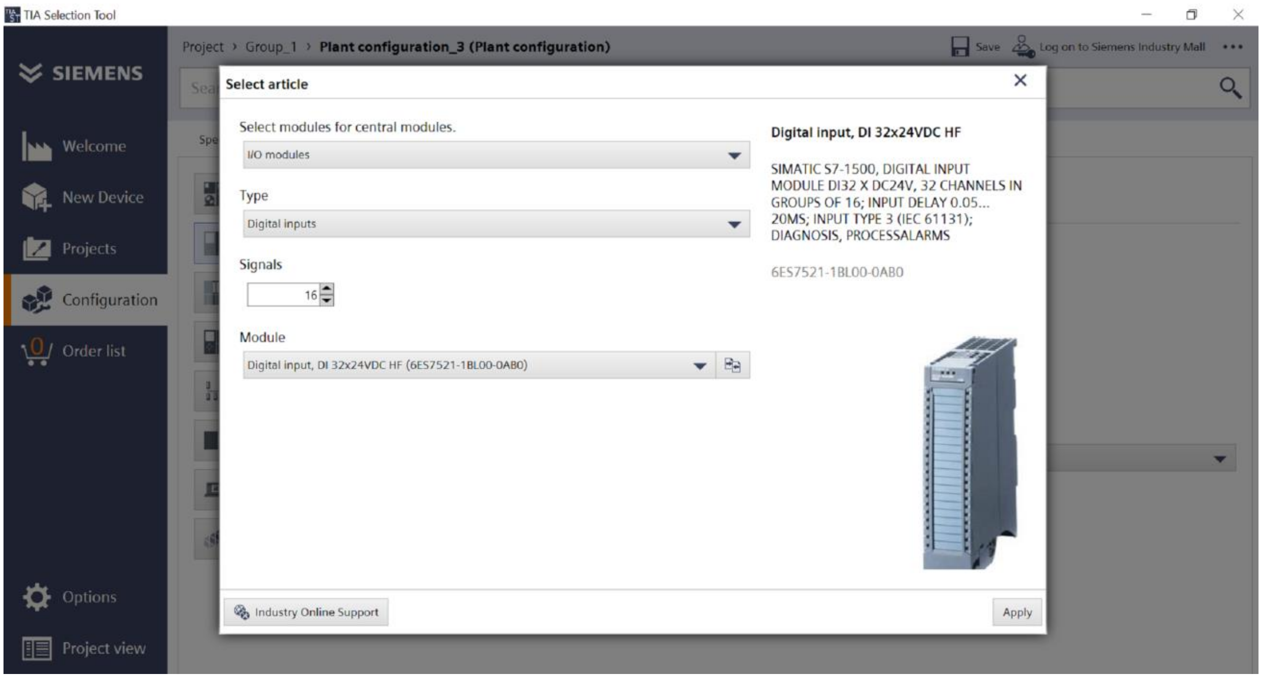


Figure 12

In the last step, by clicking on create devices, the software will generate a list of devices which can fulfill the project requirements, Figure 14. We could also click on controllers in the main tab (Figure 10) and instead make some selection manually.

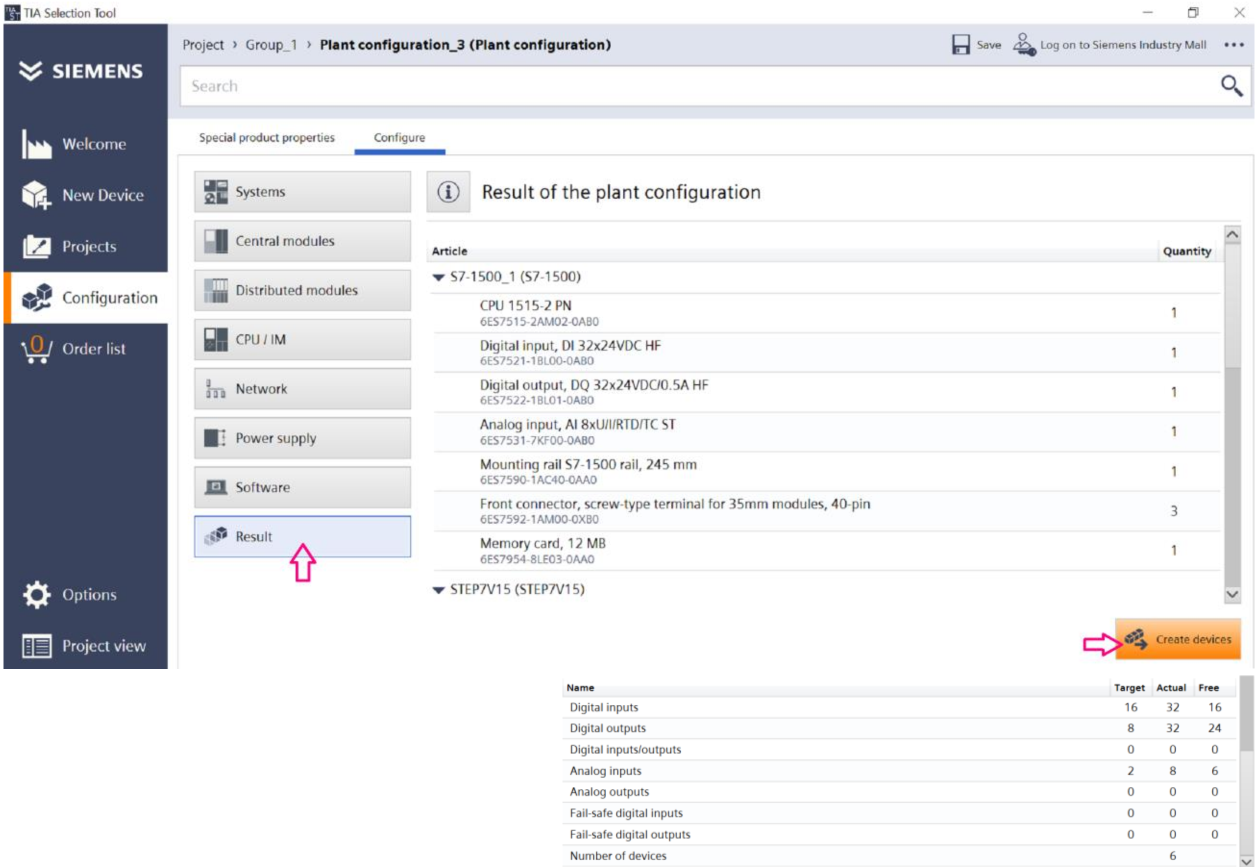


Figure 13

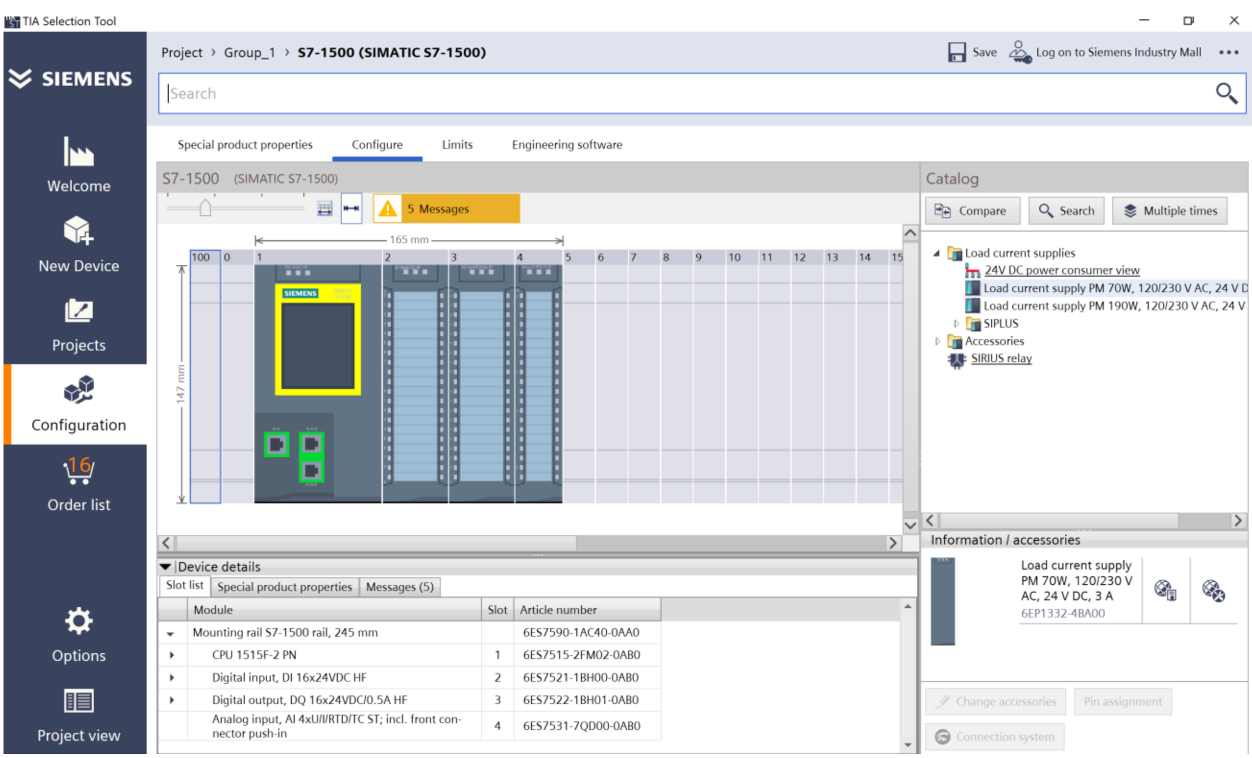


Figure 14

Now we can export the project to the TIA Portal. To do so in the project view we click on project / export / TIA Portal or ECAD system. It is also possible to do exporting in the portal view by clicking on “…” in the top right corner and then select export / TIA Portal or ECAD system.

There are many other functions in the TIA Selection Tool that we can use for different purposes. For instance, when our TIA Selection Tool project is completed, in the Project / Export / EPLAN section the TIA Selection Tool can automatically generate the circuit diagram of our project in the EPLAN software.

TIA Portal:

After exporting the project from the TIA Selection Tool, we switch over to the TIA Portal to import the generated export file. To do so, we click on Tools in the main bar and select import CAx data. After the import, we can see the imported devices with full details in the project tree on the device view.

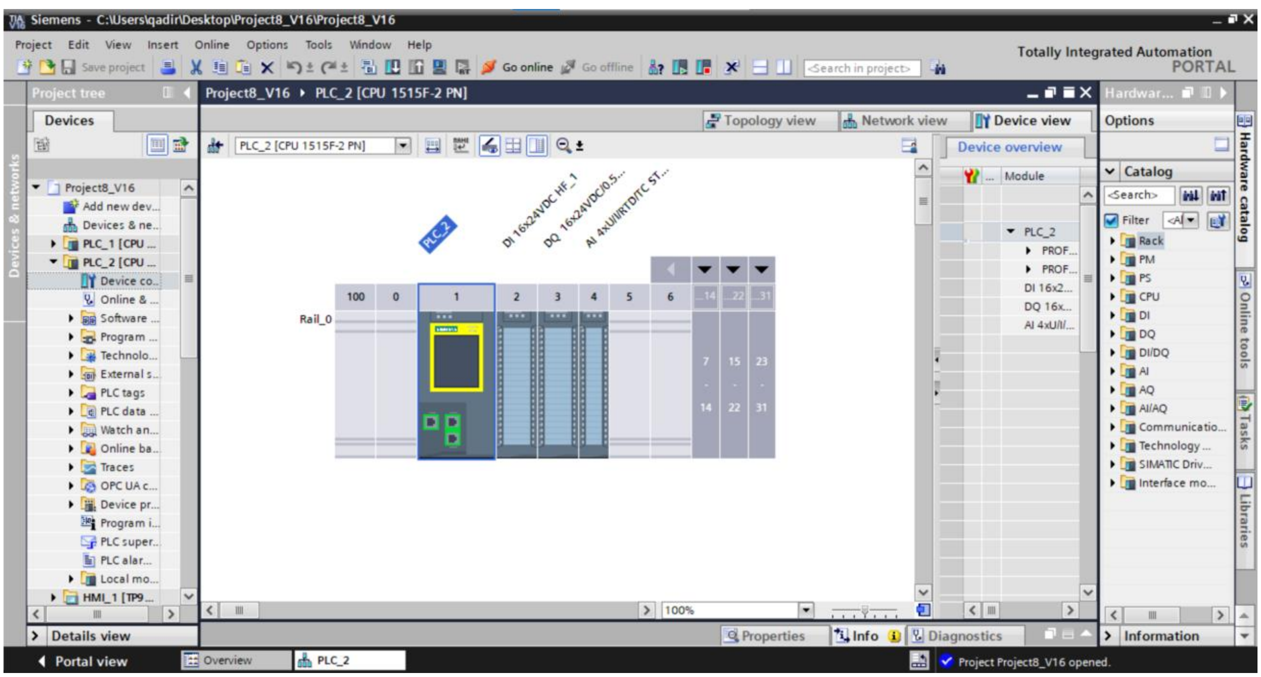


Figure 15

6.2 Data Types

After the device configuration, we are ready to start programming. But before that, I would like to add a short explanation on different data types in PLC programming.

It is crucial to have good knowledge about various data types and their usage in programming. In the S7-1500 series we can use 64 different data types. Every single data or variable is defined by two major specifications. First is data type and the second is address. In fact, data type specifies what we can do with this variable. For instance, Boolean data type can accept two values (0/1 or On/Off or True/False). Or Integer data type can accept any number without a fractional component. The second characteristic of a variable is the size and the place that variable is stored in memory. The address can show both. The size of a variable can be in range of bits, byte (8 bits), word (16 bits), double word (32 bits) and long word (64 bits). Long word is only supported by S7-1500. When we are defining a variable in the PLC tags table, we have to determine its data type as well. Figure 16

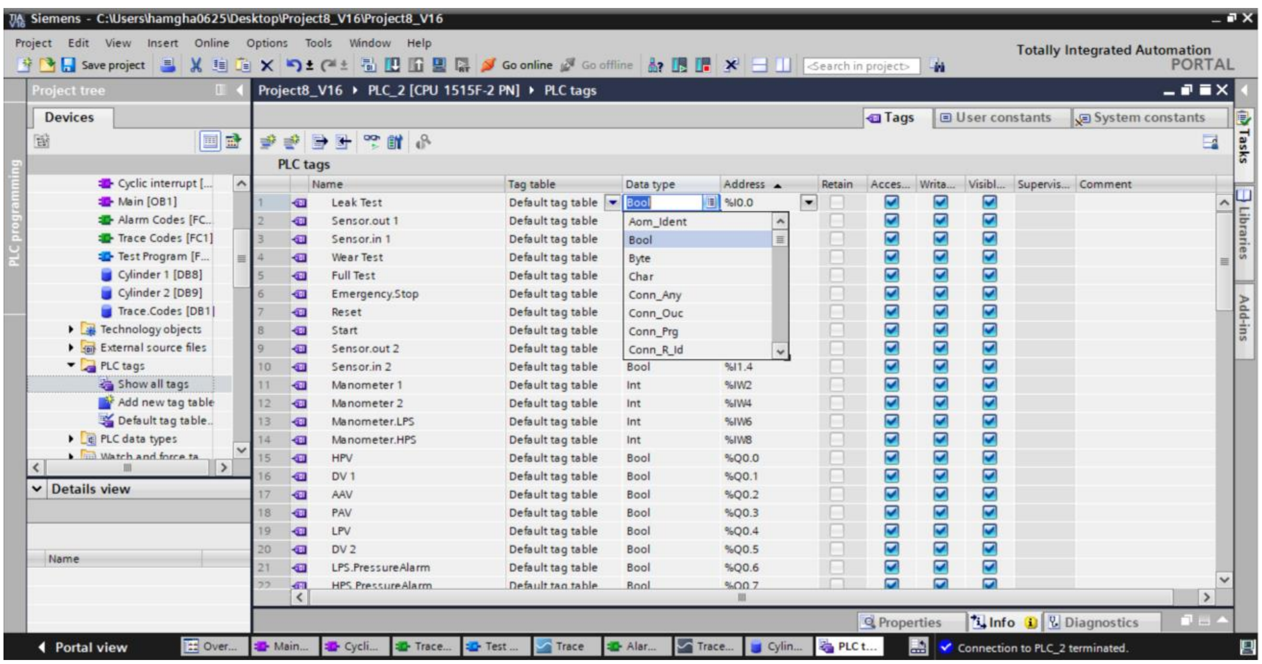


Figure 16

6.3 Programming blocks

I have structured the automation tasks in my program in order to increase the transparency of the project. This helps the maintainability of the plant and programming becomes easier. There are four different programming blocks in the TIA Portal. Organization block (OB), Function block (FB), Function (FC), and Data block (DB). The organization block has different types. I used "program cycle" as a main block of my program.

A "program cycle" OB is executed cyclically. In this block we place the instructions that control our application and call additional user blocks.

As I have two cylinders in my project, I wrote my code in a function block and then called it for each cylinder in the main block. We can add more cylinders to this project simply by calling the function block for each of them without needing to repeat the coding.

So, let’s start programming in the function block. Function blocks are code blocks that store their values permanently in the instance data blocks, so that they remain available after the block has been executed. In fact, a function block is a combination of a function (FC) with its own data block.

In the project view and from project tree, we click on “add new block”. A new tab will be opened where we can create a new function block. I renamed it “Test Program”, Figure 17.

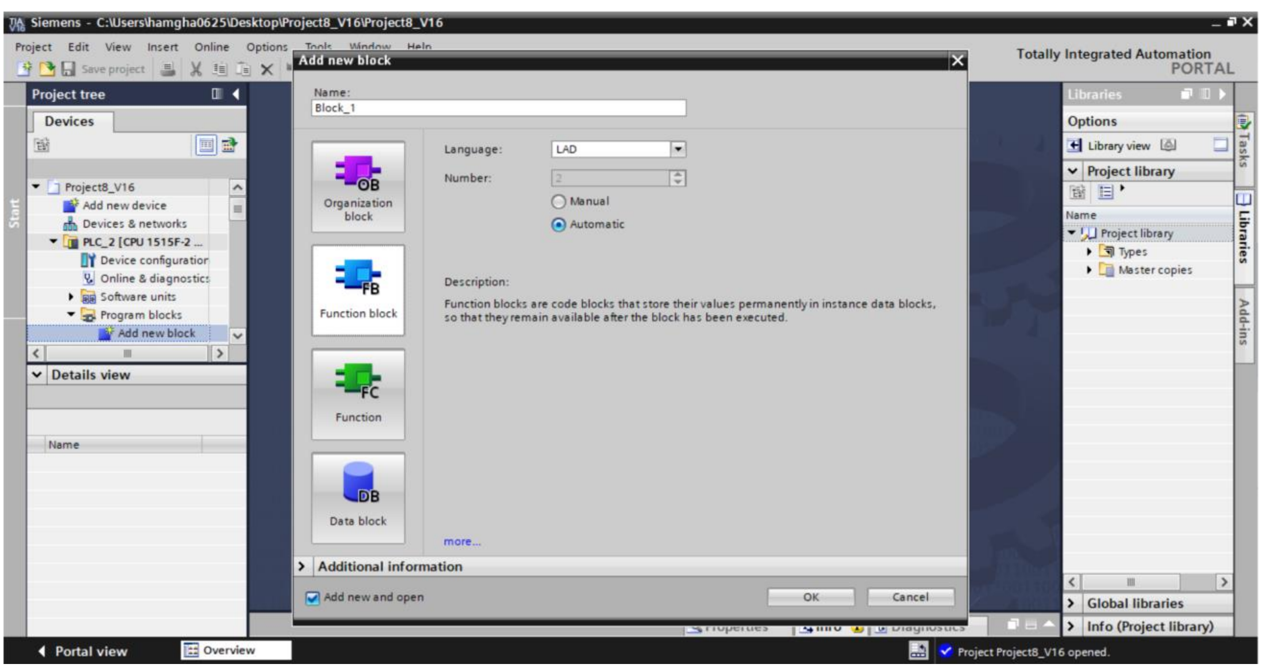


Figure 17

6.4 Test Program (Function Block)

6.4.1 Network 1

The code in this network prepares the leak test starting conditions, both in the leak test and full test modes. Full test mode is when both wear and leak tests are executed successively. As we can see in the Figure 18, after emergency a “normally closed” contact, there are two parallel lines. In the first line by selecting leak test mode with selector switch and then pushing start button, the leakage testing process will be started. By executing network 1, high pressure valve, distribution valve, anterior and posterior auxiliary valves, and a memory bit (M1) will be set. Then the piston rod moves to out. The second line is related to the leak test when the test bench is in full test mode. The low-pressure valve must not be energized during the leak test. In the leak test program, I have used a memory bit in each network which is set when the execution of the network’s code is finished. By setting this memory bit the next network can be executed. Using the “normally closed” contact of M1 in the start path guaranties that there be no disturbance in the testing process if before test completion the start button is activated again.

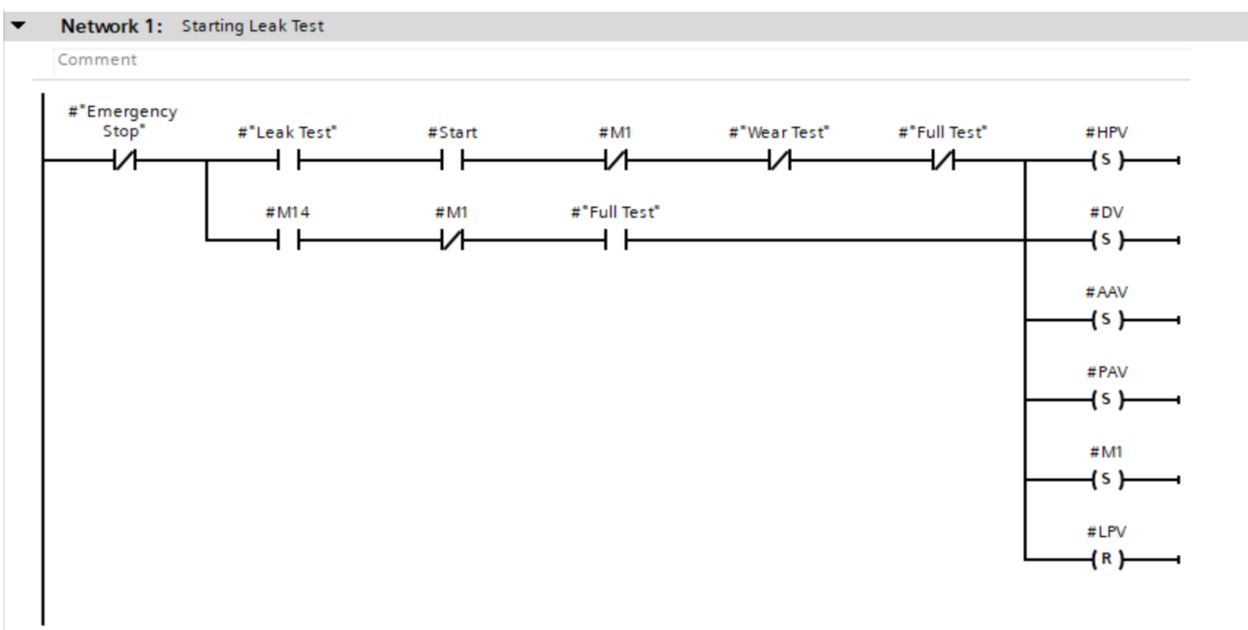


Figure 18

6.4.2 Network 2

When the cylinder reaches plus end-position, this network will be activated by a sensor out signal. When the timer is activated, the cylinder will be under high pressure for some seconds. Here the timer is set on 10 seconds, Figure 19. Then air paths from both back and front sides will be blocked for 5 seconds by the timer that is placed in the third network.

6.4.3 Network 3

By executing the code in this network, the high-pressure supply will be disconnected. The anterior auxiliary valve will be energized to open the front path. However, the posterior air path will remain in blocked position. The rear chamber memory bit will be set to activate measurements recording in trace function, Figure 20. This step lasts for 30 minutes. The timer that is in charge for doing so is placed in the next network, number 4.

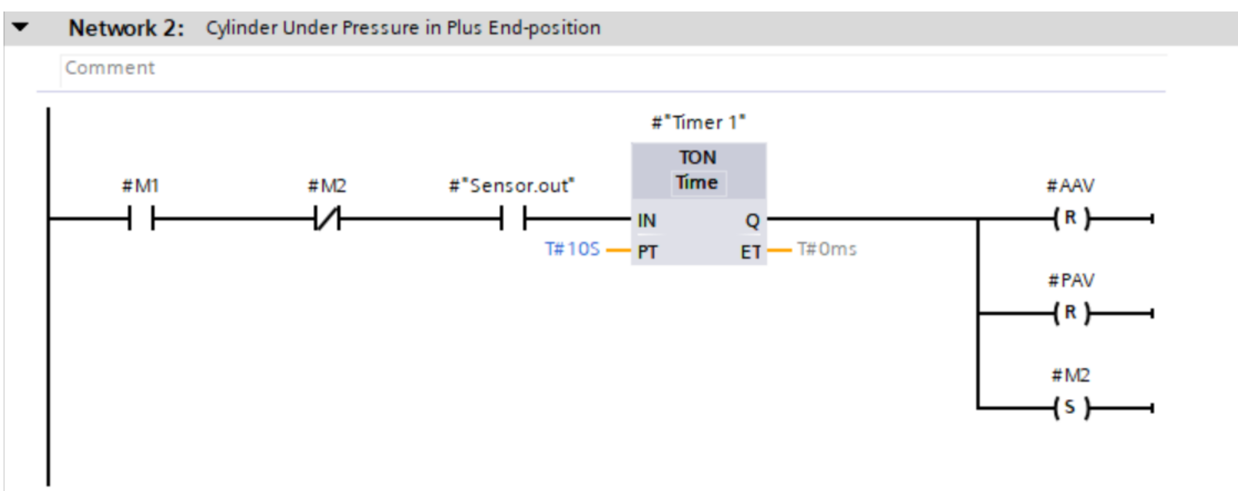


Figure 19

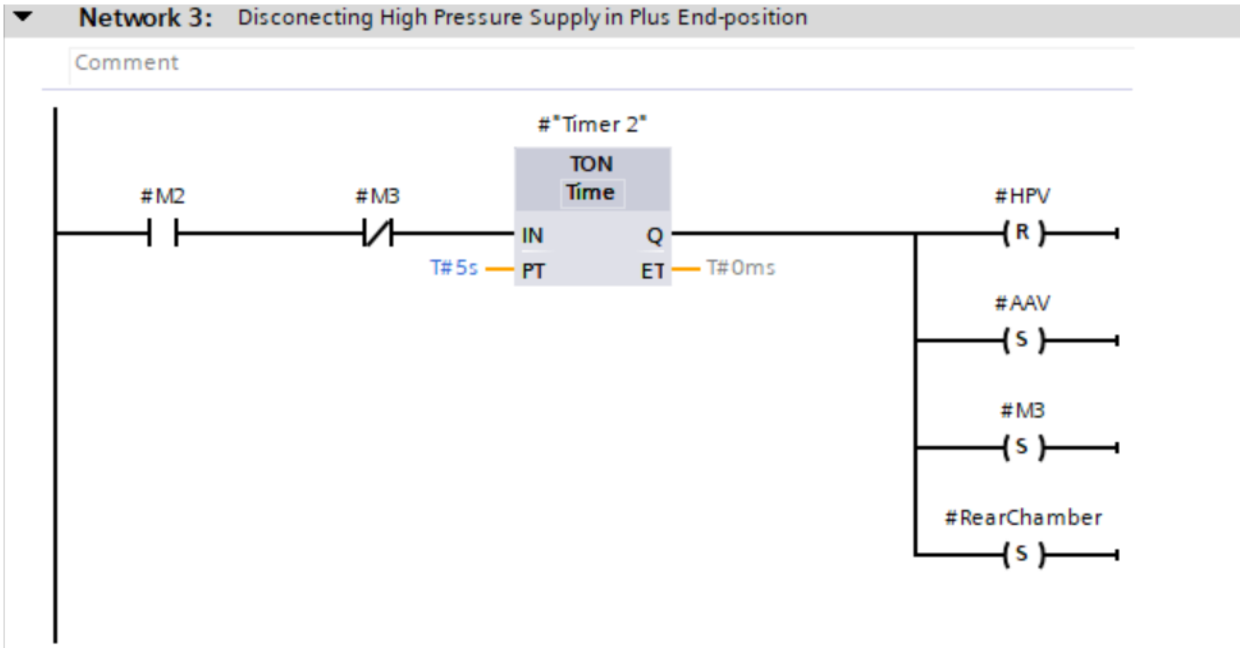


Figure 20

6.4.4 Network 4

The main step for the leakage test last for 30 minutes. When this step is finished, the high pressure and posterior auxiliary valves will be energized, and the distribution valve will be reset. Now the piston rod moves inside. The rear chamber memory bit will be reset to pause the recording, Figure 21.

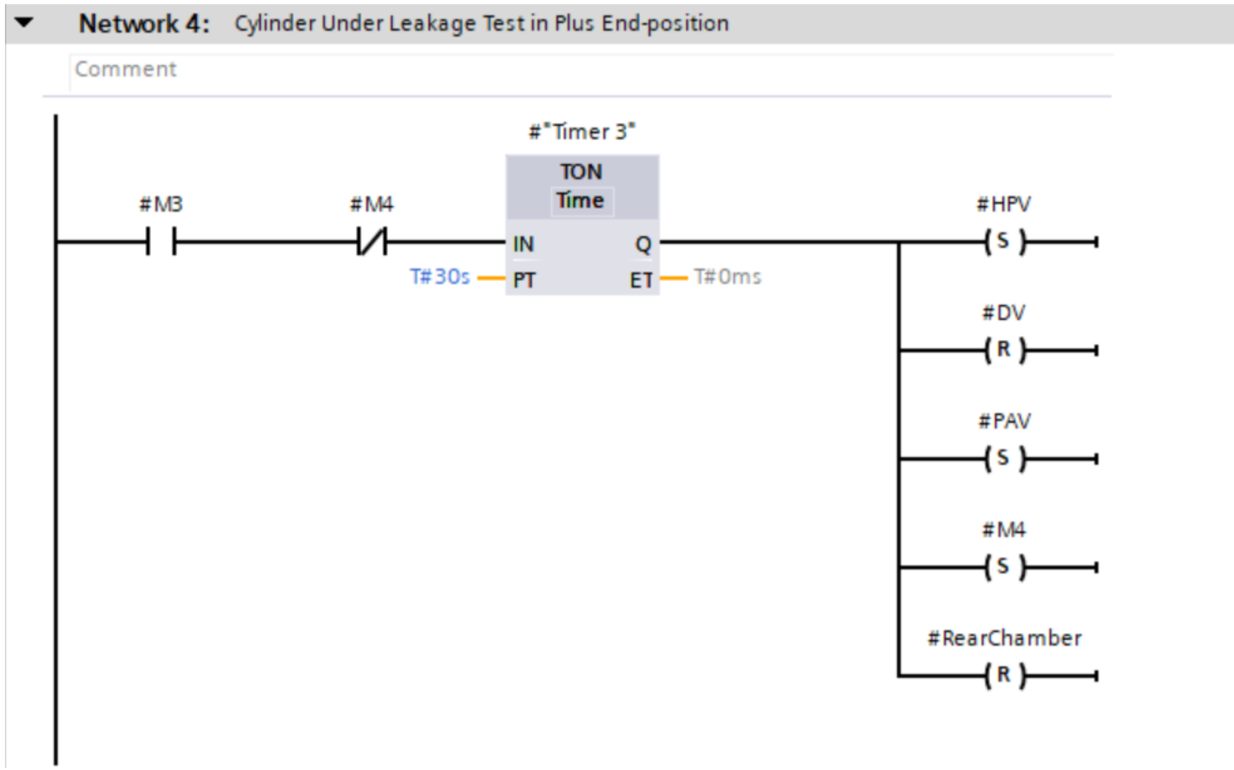


Figure 21

6.4.5 Network 5

When the piston rod moves inside the cylinder, the timer in network 5 will be activated by a sensor in signal. When the timer is activated, the cylinder will be under high pressure for some seconds. Here the timer is set to 10 seconds, Figure 22. Then the air paths from both back and front sides will be blocked for 5 seconds by the timer that is placed in the next network.

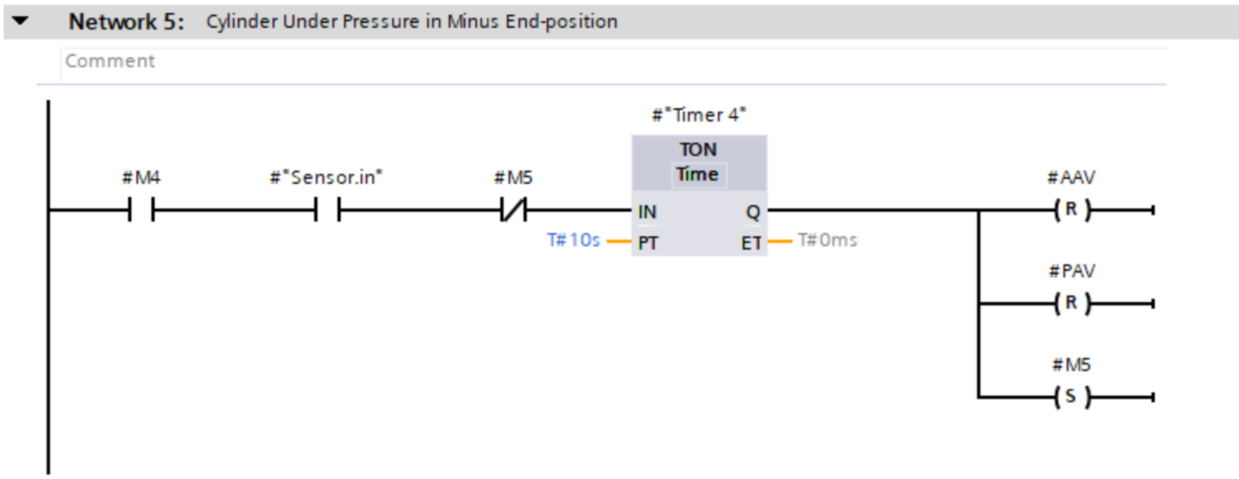


Figure 22

6.4.6 Network 6

By executing the code of this network, the high-pressure supply will be disconnected. The posterior auxiliary valve will be energized to open the backside path. However, the anterior air path will remain in a blocked position. The front chamber memory bit will be set to reactivate the measurements recording in trace function, Figure 23. These are the conditions required to test the leakage in the front chamber.

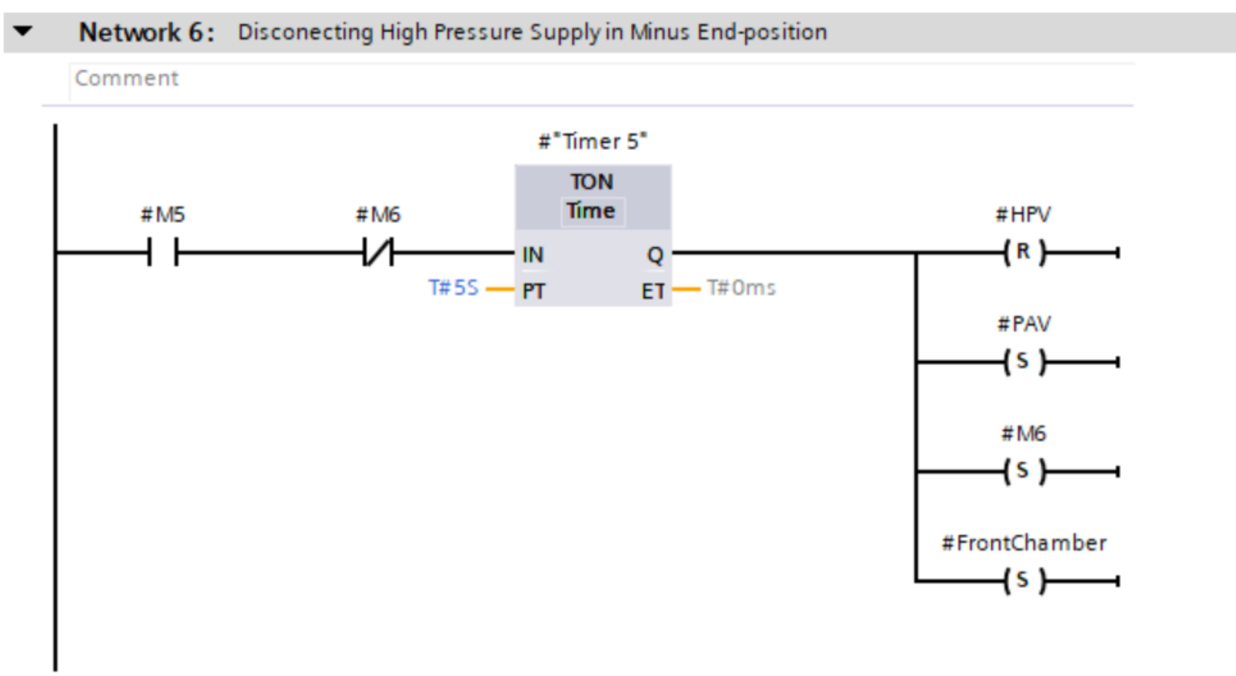


Figure 23

6.4.7 Network 7

By activating the timer on this network, the main leakage test will start and last for 30 minutes. During this period, the pressure changes will be monitored by the trace function. When the leakage monitoring is finished, all valves and memory bits will be reset at the output of the timer, and the data recording will be stopped by resetting the front chamber memory bit, Figure 24.

6.4.8 Network 8

Network 8 is in charge to execute the wear test and the first step of the full test. By using a selector switch we select wear or full test mode and then by switching the start button, the test process will be started. Using a memory bit M10 helps to make a supporting line to keep the circuit connected after releasing the start button. When the wear sequences reach to the setpoint of the counter, M13 will be set. Then by returning the piston inside the cylinder and setting M7, the wear test process will be terminated, Figure 25.

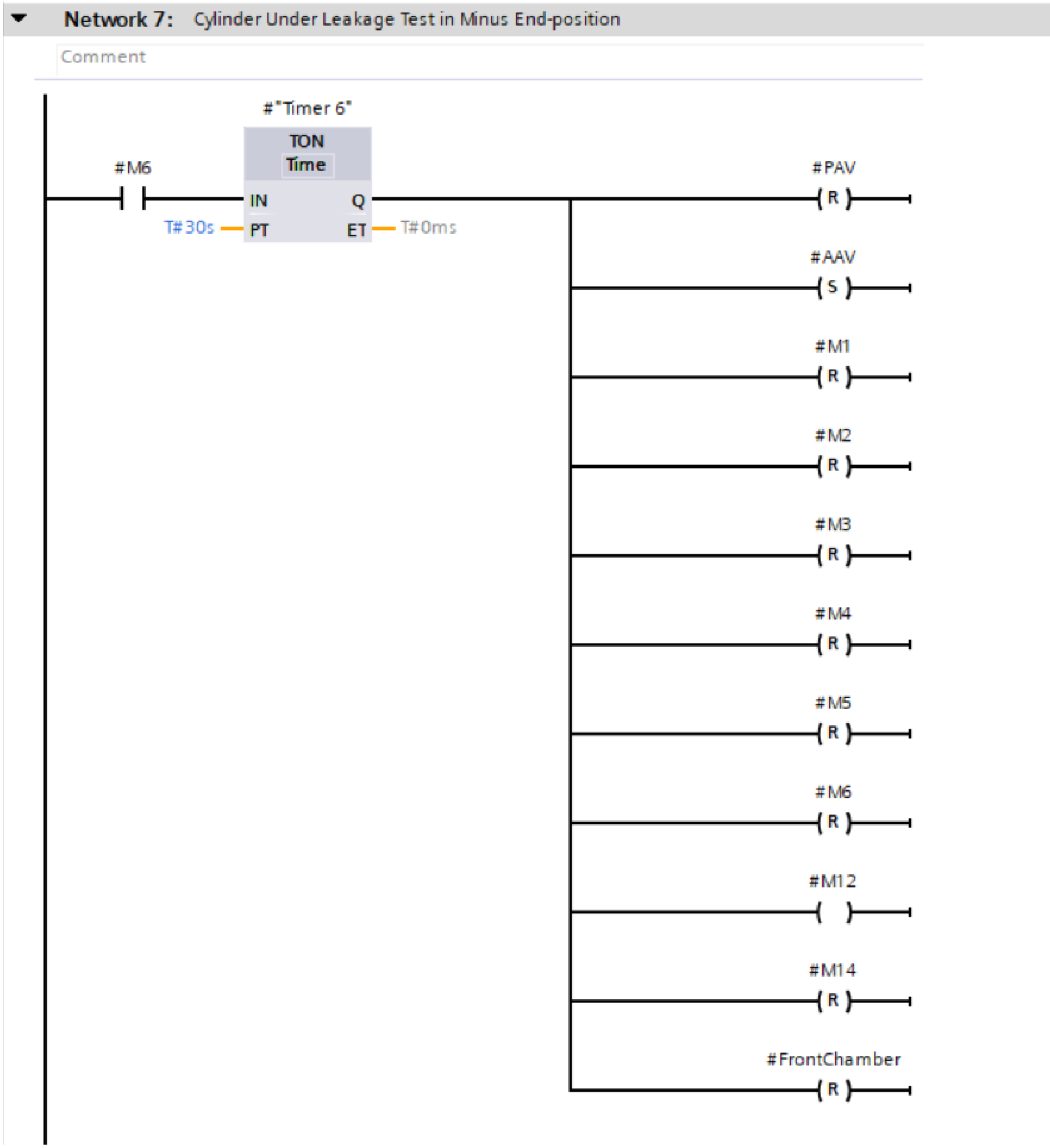

Network 7: Cylinder Under Leakage Test in Minus End-position
Comment
#"Timer 6"
TON
Time
#M6
IN
Q
T#30s
PT
ET
T#0ms
#PAV
R
#AAV
S
#M1
R
#M2
R
#M3
R
#M4
R
#M5
R
#M6
R
#M12
#M14
R
#FrontChamber
R

Figure 24

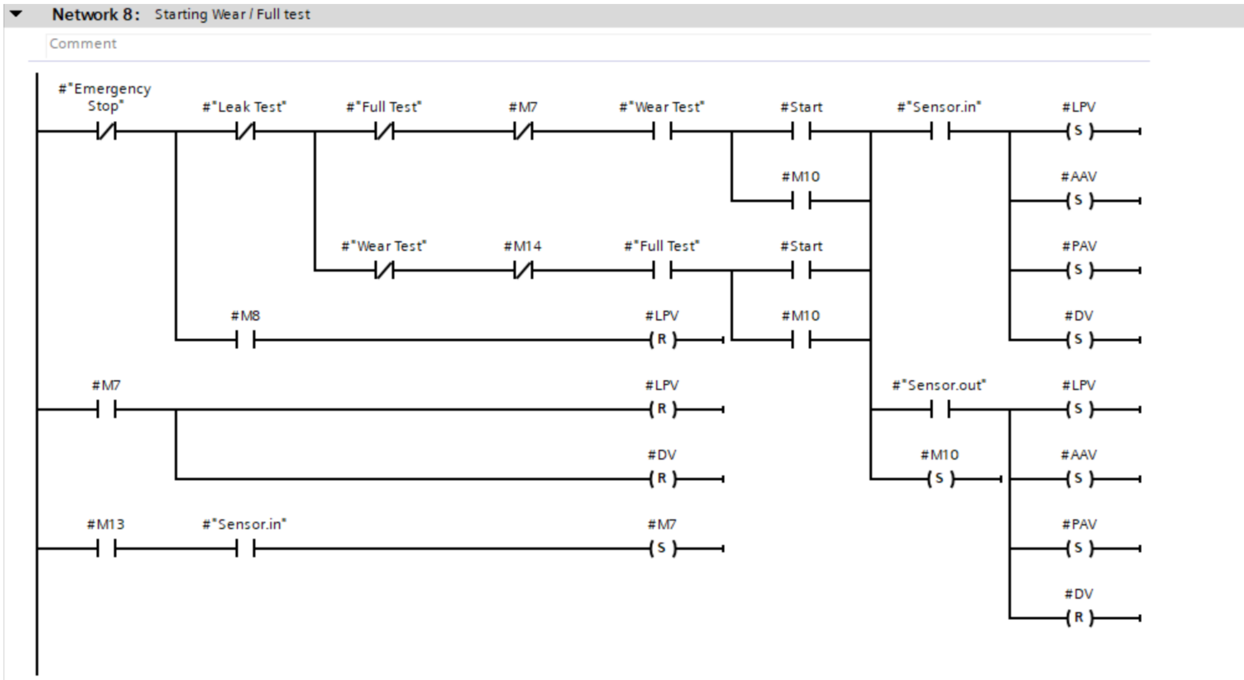


Figure 25

6.4.9 Network 9

When the test bench is in full test mode, the wear test is started by pressing the start button. The counter in network 9 counts the wear sequences in the full test mode. At the end of wear test, when the test bench switches to the leak test, air supply must be changed from low to high pressure. So, it is necessary that all cylinders finish their wear test first. Then the system is able to change the air supply and move to the leak test. As the movements of two cylinders are not synchronized, they may not finish their wear test process at the same time. Therefore, I defined the conditions in network 9 to guarantee this requirement. The lower parallel line will be activated in case of removing any defective cylinder from test process. This parallel line can bypass the condition in absence of the defective cylinder in the test process. By activating M8 and M14 and their contacts in networks 1 and 8, switching from wear to leak test process will be done automatically, Figure 26.

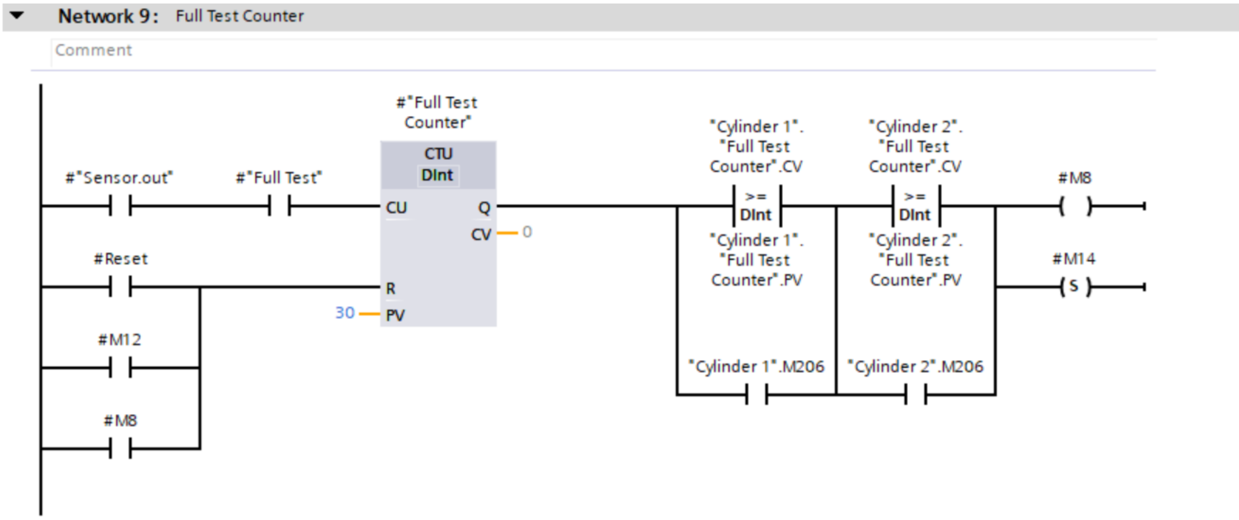


Figure 26

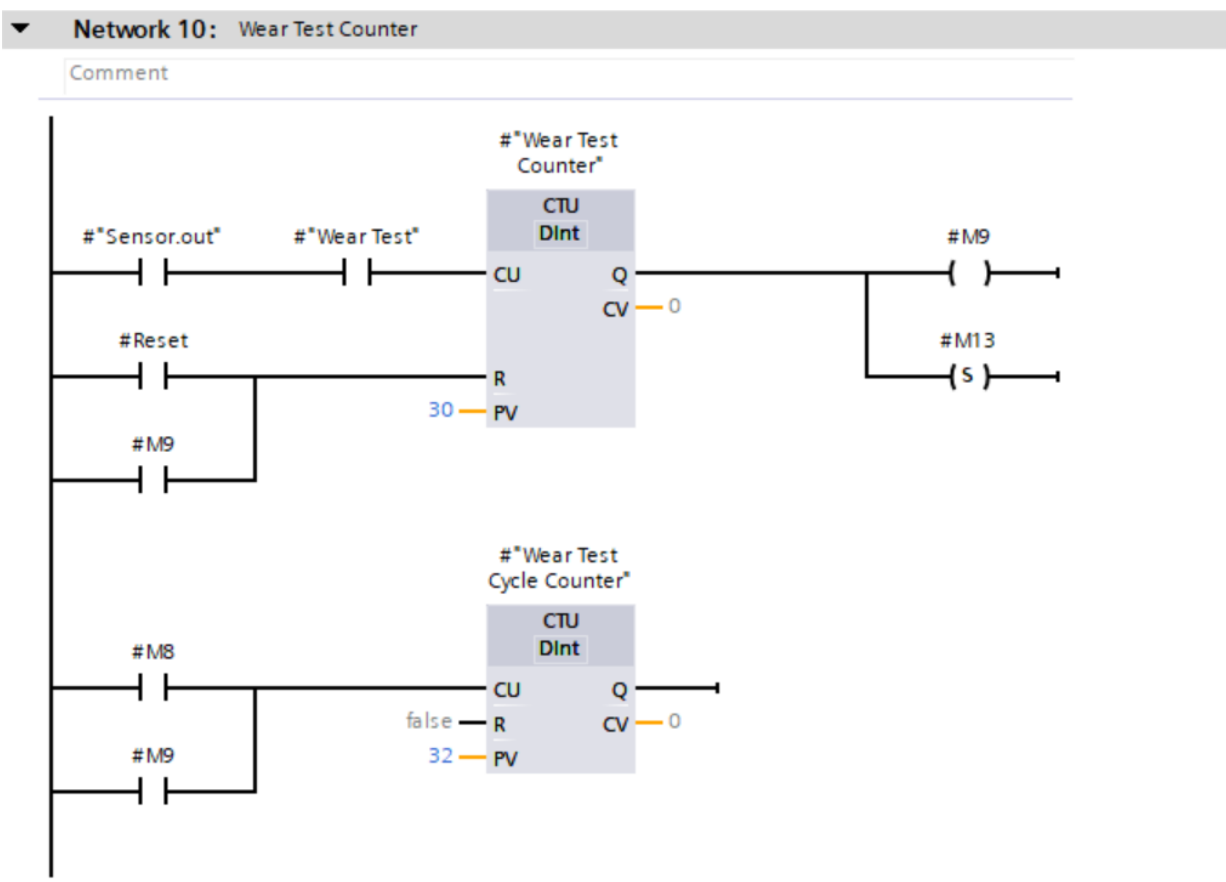


Figure 27

6.4.10 Network 10

In this network there are two counters. The first one counting the wear sequences during wear test and the second one counting the number of complete wear cycles. Each wear cycle is equal to the setpoint of the first counter. In this project each wear cycle contains 30,000 complete wear sequences. The wear cycle counter is set on 32,000, Figure 27.

6.4.11 Network 11

In this network the total number of wear sequences either in wear or full test mode will be shown through the use of the "calculate" function and a mathematical formula, Figure 28.

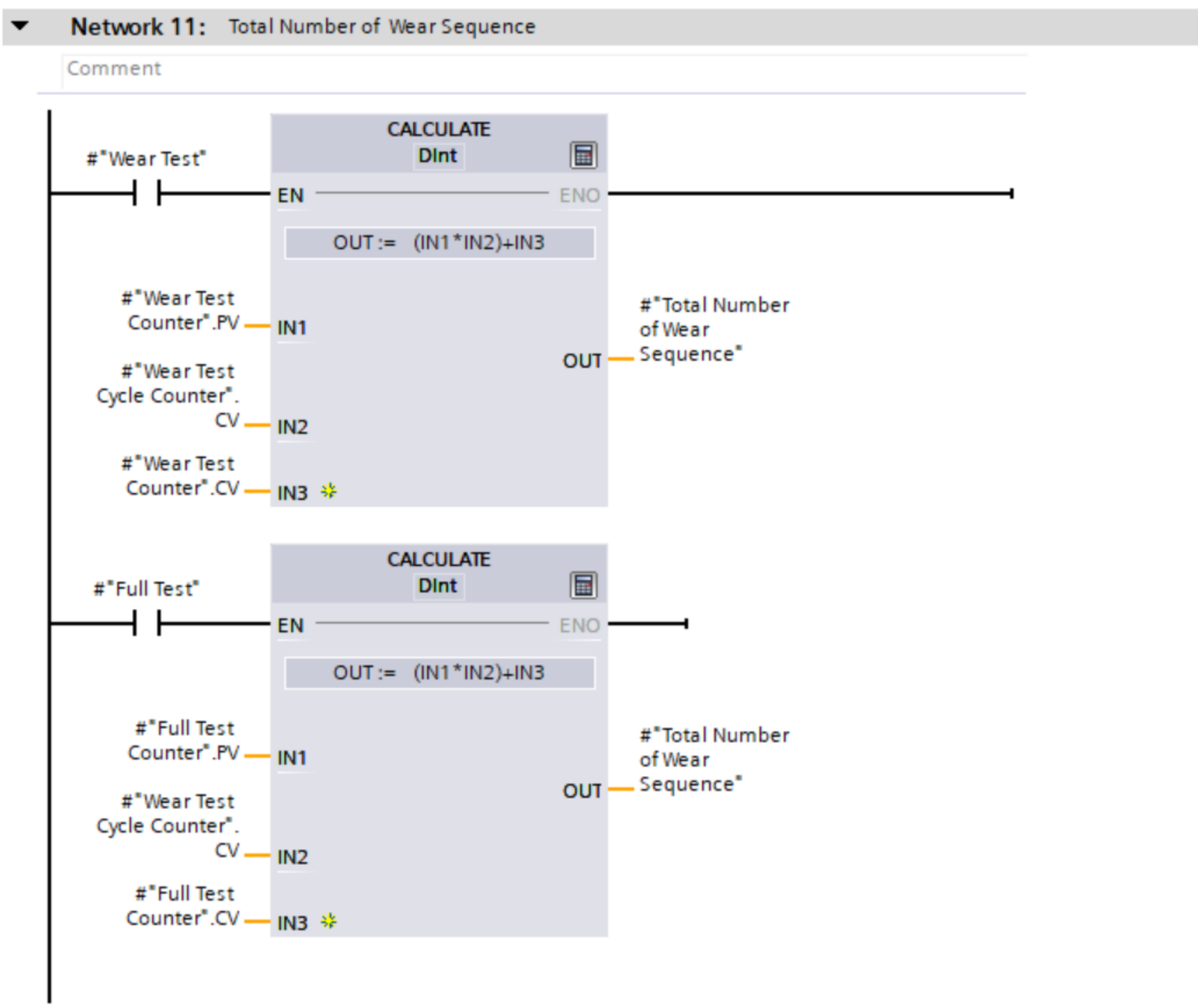


Figure 28

6.4.12 Network 12

The analog signal which is receiving from the digital manometer is in the form of voltage. We create this network to normalize and scale this analog signal. The minimum and maximum value is depending on the system air pressure, Figure 29.

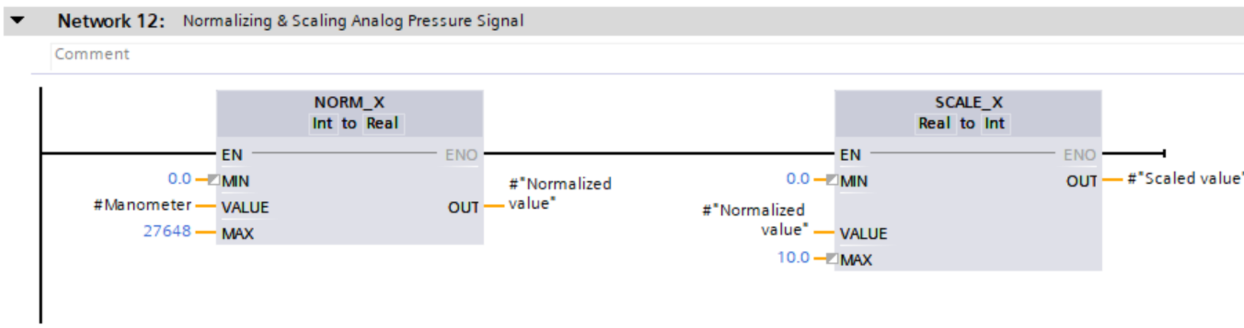


Figure 29

6.4.13 Network 13

If a piston stuck inside a cylinder or each of sensors is not working properly or if there is any other defect that causes the piston movement to last more than the predefined time, it can be detected in this network. By activating M206 the defective cylinder will be removed from the testing process. Figure 30

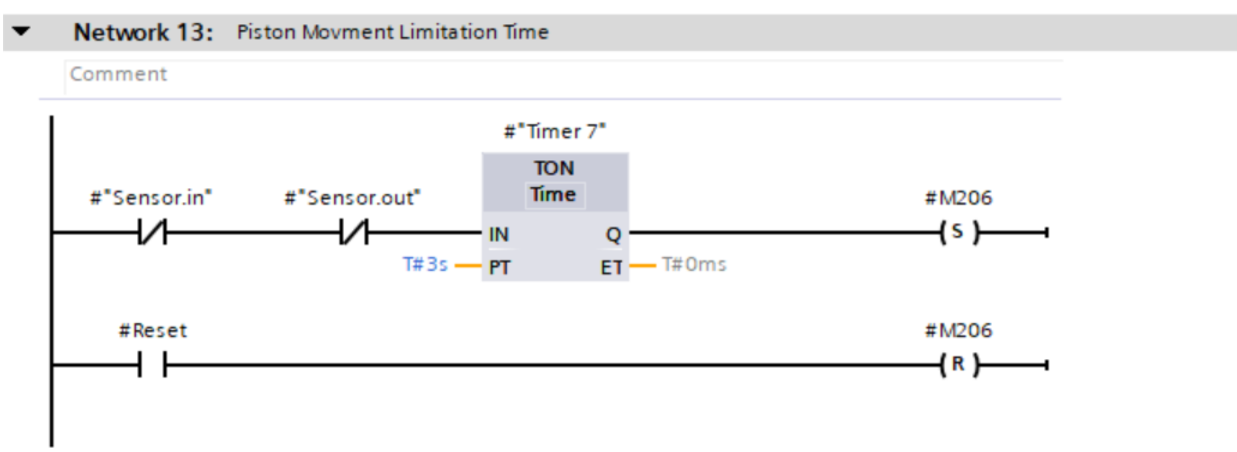


Figure 30

6.4.14 Network 14

Network 14 is created to provide safety during all testing modes. If in each step of any testing modes the emergency stop switch is pushed, the working process will be stopped immediately, Figure 31. To resume the test process the emergency stop switch should be released and the reset push button should be pressed. If the piston is stopped in the middle of its path, you should first select the wear test mode switch and then press the reset button.

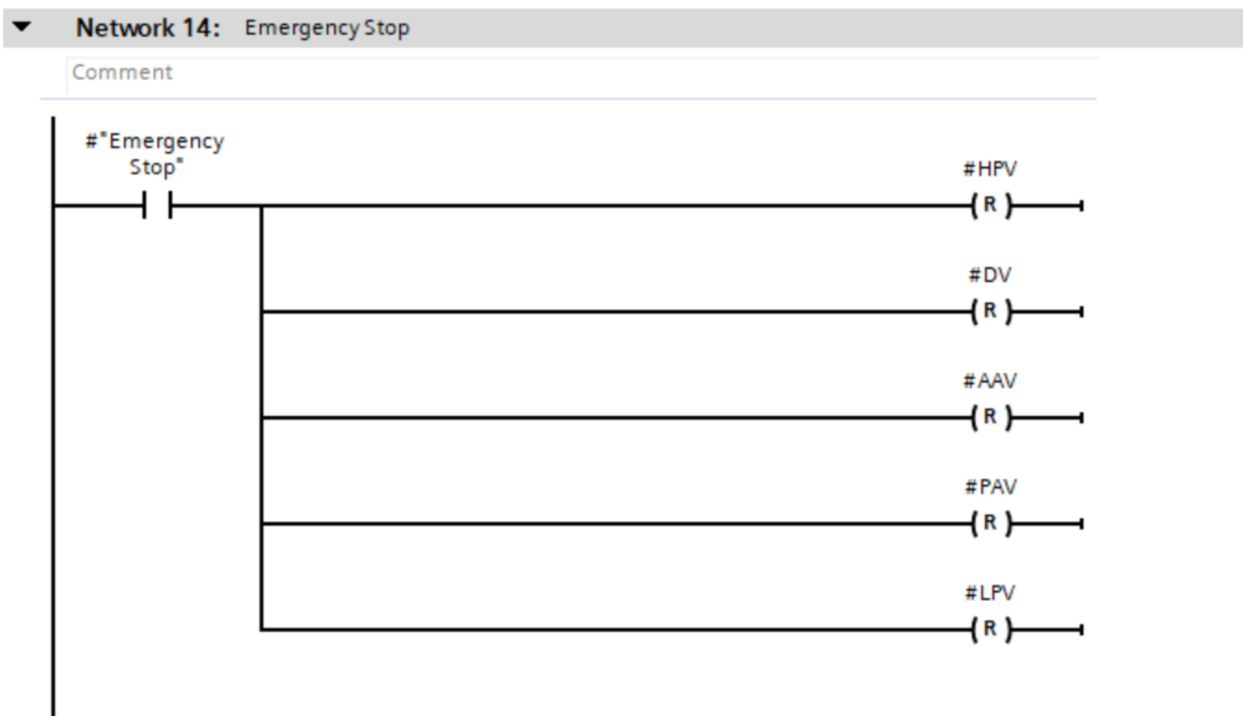


Figure 31

6.4.15 Network 15

Whenever it is required to reset the testing system, which can be done by pressing the reset push button, the code in network 15 will be executed, Figure 32.

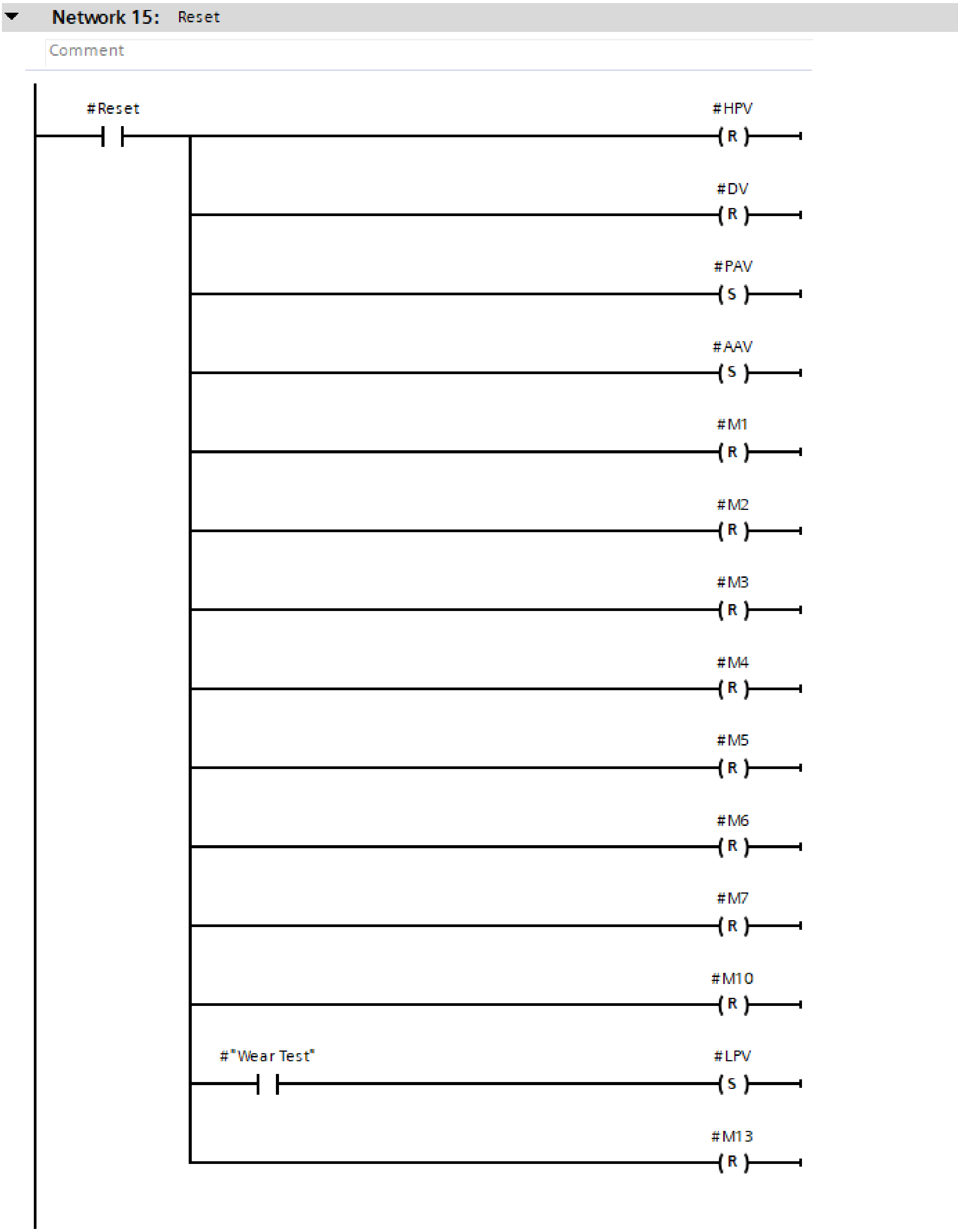

Network 15: Reset
Comment
#Reset
#HPV
R
#DV
R
#PAV
S
#AAV
S
#M1
R
#M2
R
#M3
R
#M4
R
#M5
R
#M6
R
#M7
R
#M10
R
#"Wear Test"
#LPV
S
#M13
R

Figure 32

6.5 Main Block (Organization Block)

The main block in the PLC programming is an organization block which, as I mentioned before, the selected subsection is ‘’program cycle’’. The first and second networks are in charge to call cylinders function blocks. Network 1 is related to cylinder 1 and the network 2 to the second cylinder.

6.5.1 Network 1 & 2

As a way of not having to repeat the programming for each cylinder, we have an option in TIA Portal. We write the main program in a function block and we call this FB for each cylinder in the main block. As seen in Figure 33, there are two types of inputs and outputs. Some of them are common between the two cylinders and some must be used independently. The uncommon I/O comes from different physical addresses. For example, the start switch is common for all cylinders and we use only one start switch that occupies just one physical address. However, we use one manometer for each cylinder individually, so they are uncommon I/O and occupy different physical addresses. For each uncommon signal, we use an independent global tag from the tag table. There is no need to use a physical address tag for common I/O in the first and second networks. However, we create a specific network to define them. In the following pictures, Figures 33 and 34, the first and second networks with all common and uncommon I/O are shown.

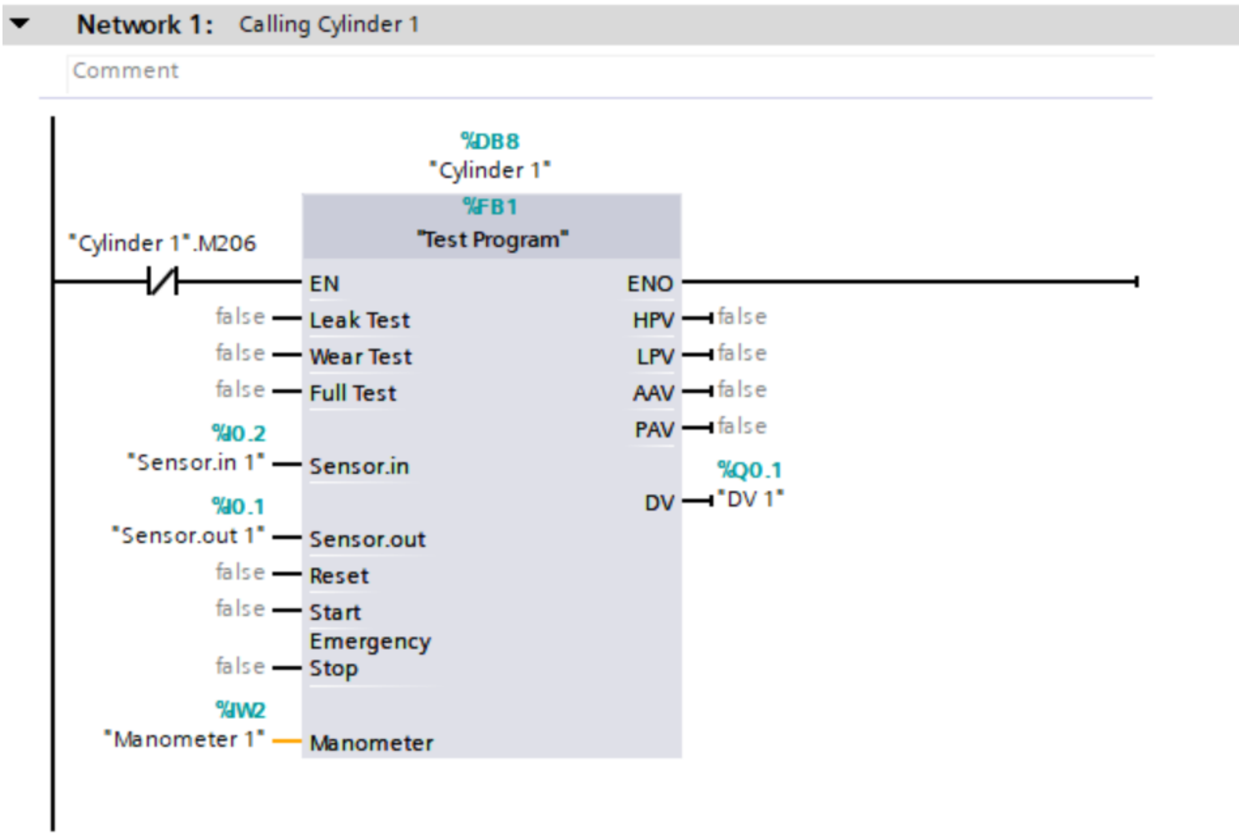

Network 1: Calling Cylinder 1
Comment
%DB8
"Cylinder 1"
%FB1
"Test Program"
"Cylinder 1".M206
EN
ENO
false
Leak Test
HPV
false
false
Wear Test
LPV
false
false
Full Test
AAV
false
PAV
false
%I0.2
"Sensor.in 1"
Sensor.in
%Q0.1
DV
"DV 1"
%I0.1
"Sensor.out 1"
Sensor.out
false
Reset
false
Start
Emergency Stop
false
%IW2
"Manometer 1"
Manometer

Figure 33

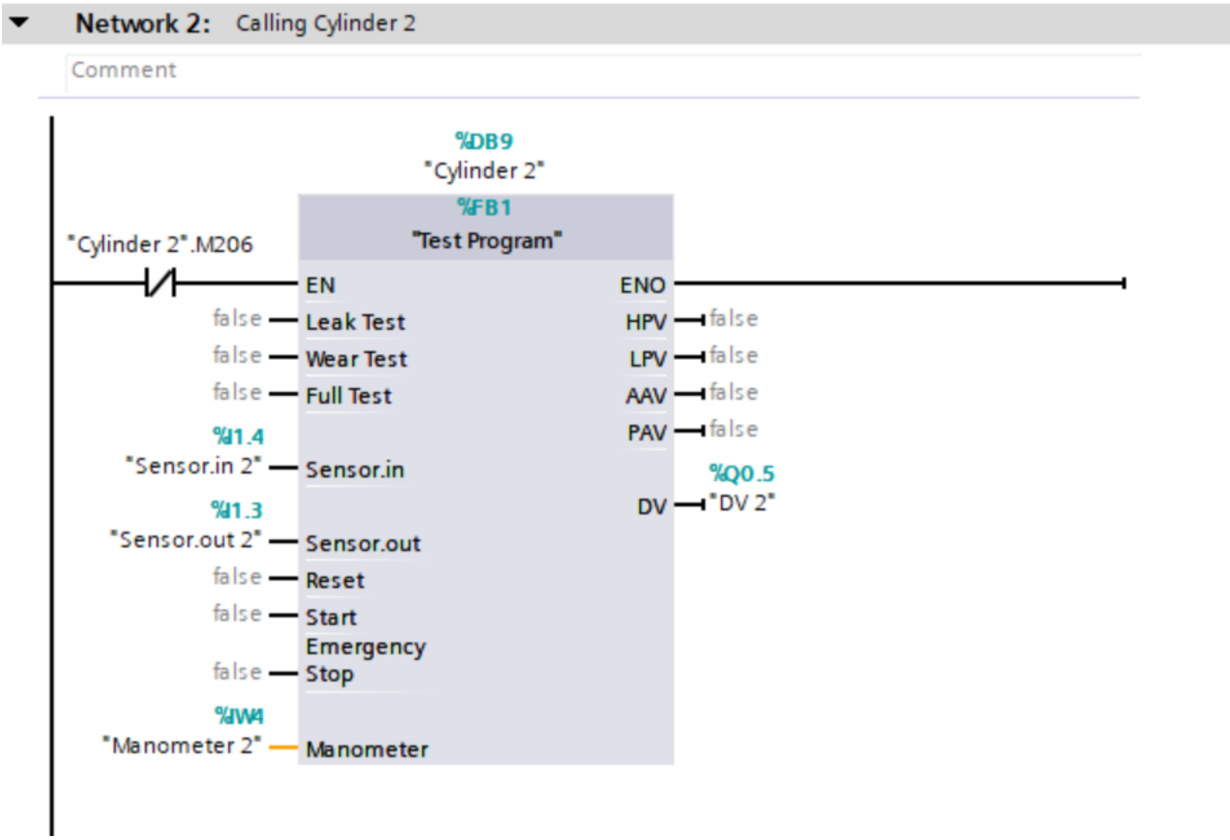


Figure 34

6.5.2 Network 3 to 8

In these networks, I have defined the common inputs for all cylinders. Physical selector switch and pushbuttons: Leak test, wear test, full test, start, reset and emergency stop. (Figure 35 and Figure 36)

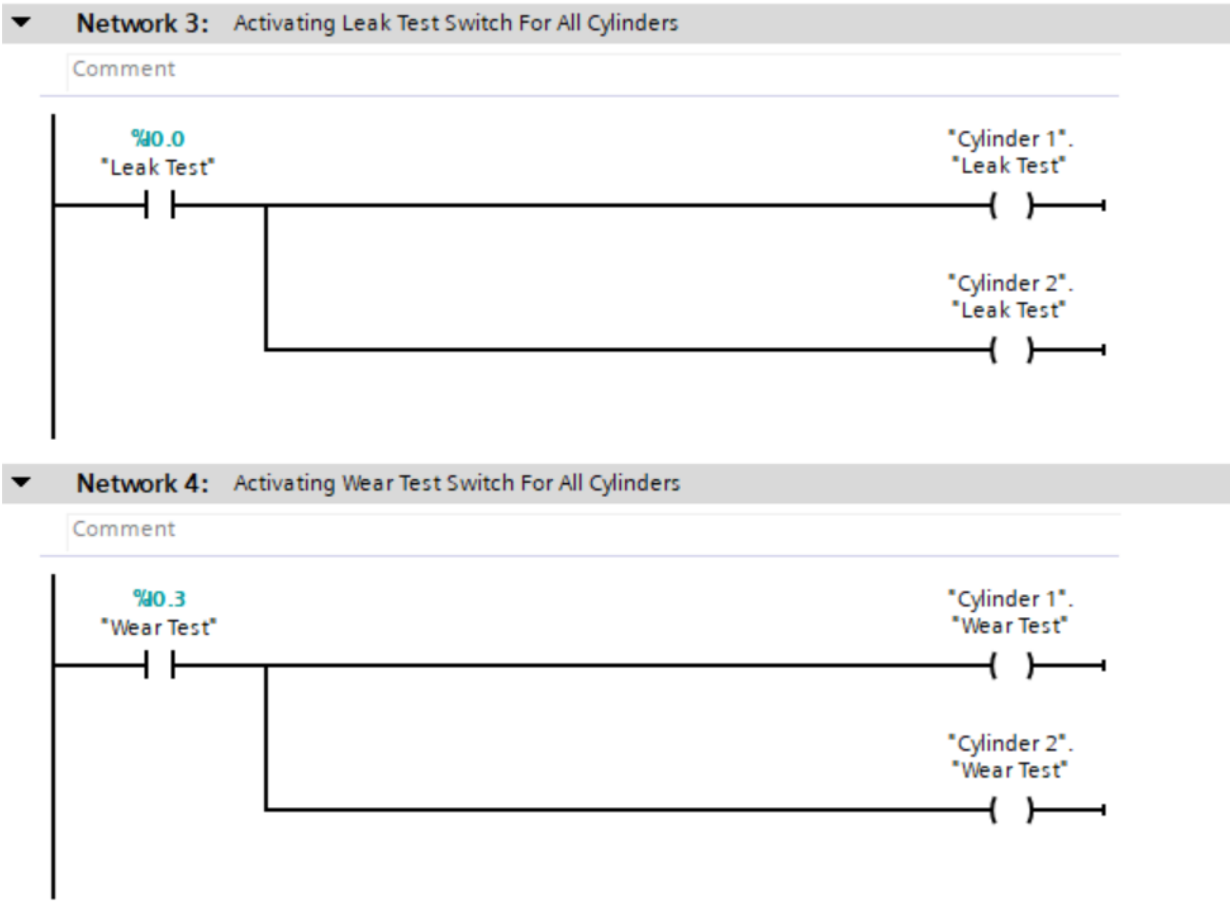

Network 3: Activating Leak Test Switch For All Cylinders
Comment
%I0.0
"Leak Test"
"Cylinder 1".
"Leak Test"
"Cylinder 2".
"Leak Test"
Network 4: Activating Wear Test Switch For All Cylinders
Comment
%I0.3
"Wear Test"
"Cylinder 1".
"Wear Test"
"Cylinder 2".
"Wear Test"

Figure 35

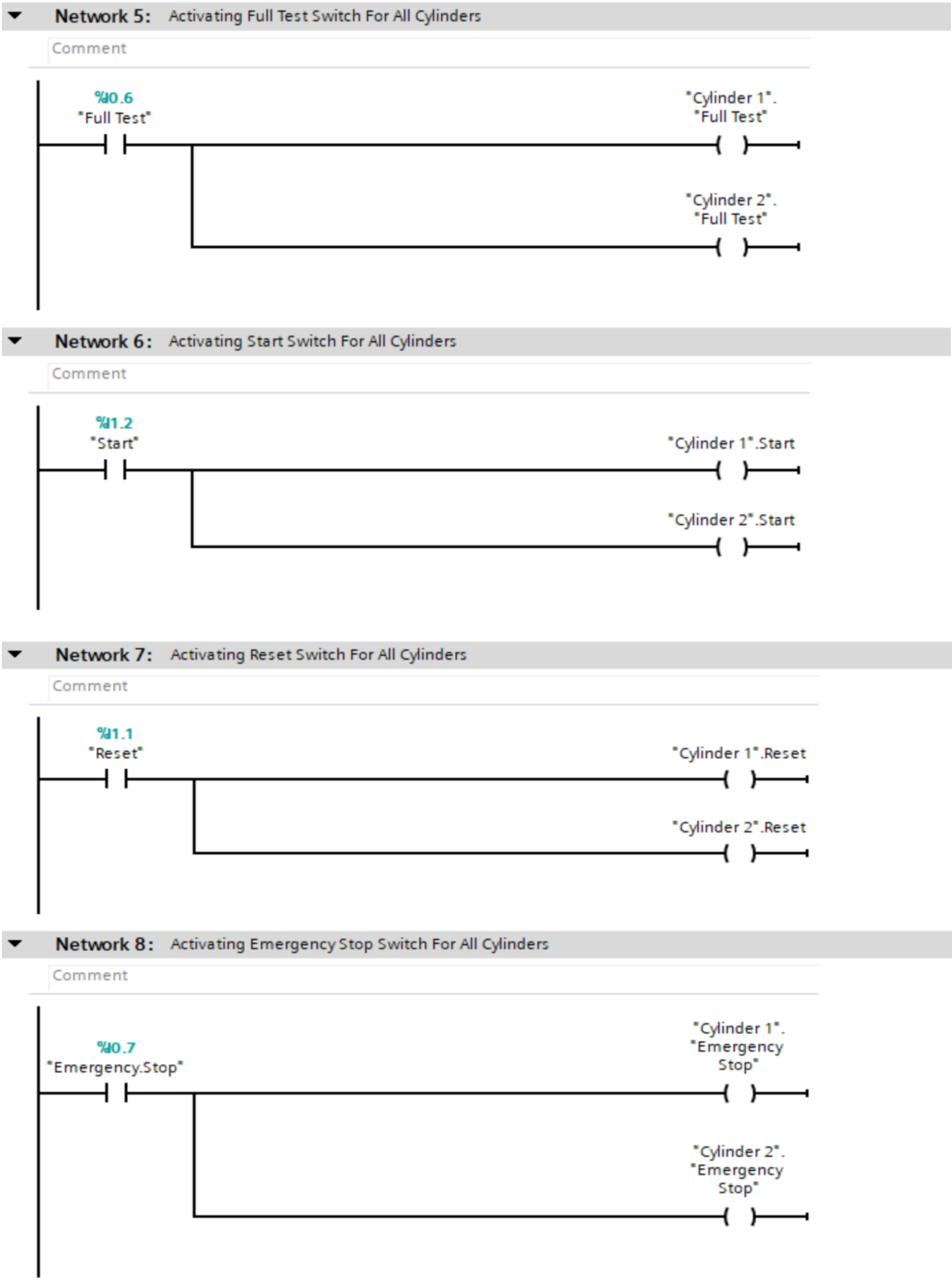

Network 5: Activating Full Test Switch For All Cylinders
Comment
%I0.6
"Full Test"
"Cylinder 1".
"Full Test"
"Cylinder 2".
"Full Test"
Network 6: Activating Start Switch For All Cylinders
Comment
%I1.2
"Start"
"Cylinder 1".Start
"Cylinder 2".Start
Network 7: Activating Reset Switch For All Cylinders
Comment
%I1.1
"Reset"
"Cylinder 1".Reset
"Cylinder 2".Reset
Network 8: Activating Emergency Stop Switch For All Cylinders
Comment
%I0.7
"Emergency.Stop"
"Cylinder 1".
"Emergency
Stop"
"Cylinder 2".
"Emergency
Stop"

Figure 36

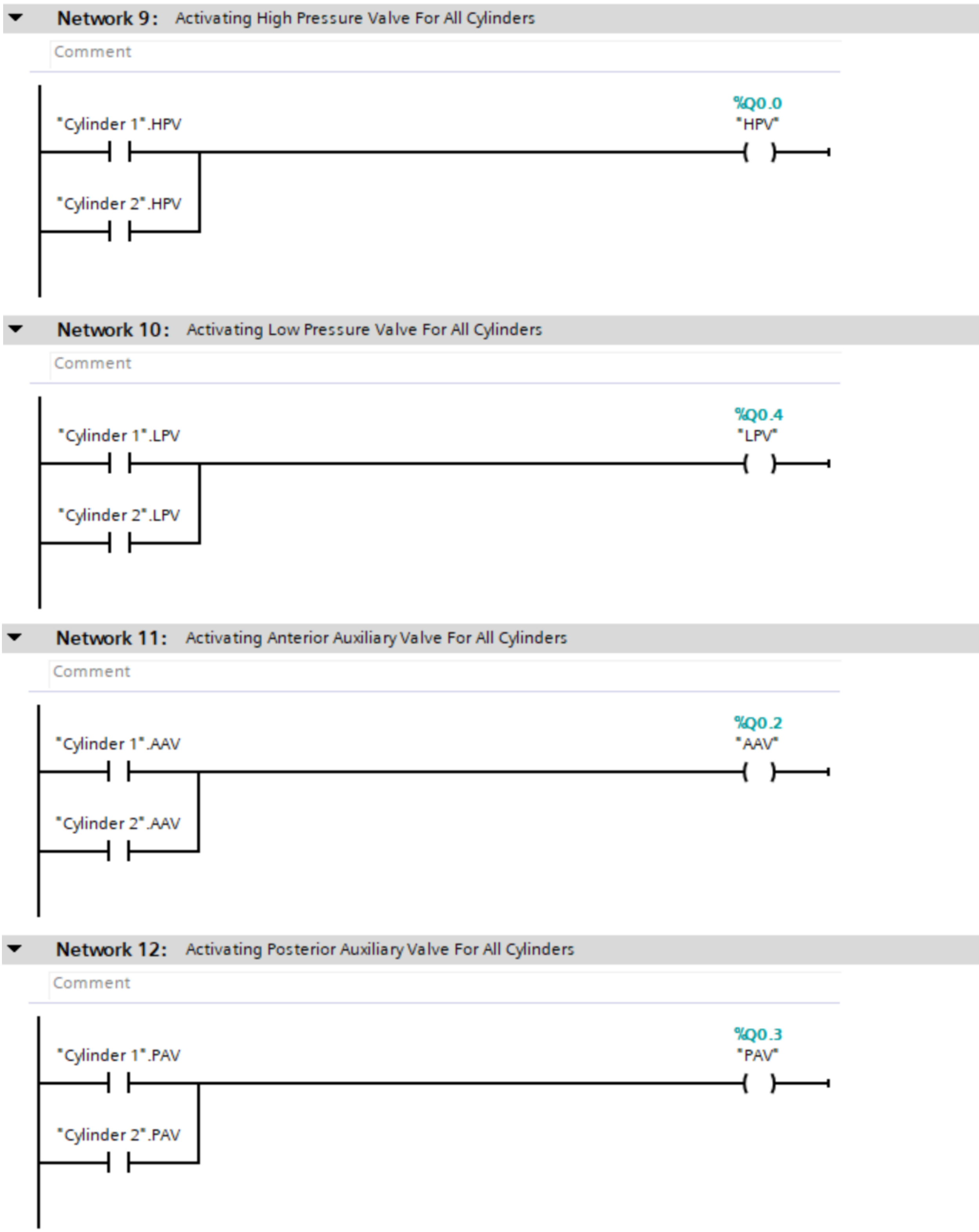

Network 9: Activating High Pressure Valve For All Cylinders
Comment
"Cylinder 1".HPV
"Cylinder 2".HPV
%Q0.0
"HPV"
Network 10: Activating Low Pressure Valve For All Cylinders
Comment
"Cylinder 1".LPV
"Cylinder 2".LPV
%Q0.4
"LPV"
Network 11: Activating Anterior Auxiliary Valve For All Cylinders
Comment
"Cylinder 1".AAV
"Cylinder 2".AAV
%Q0.2
"AAV"
Network 12: Activating Posterior Auxiliary Valve For All Cylinders
Comment
"Cylinder 1".PAV
"Cylinder 2".PAV
%Q0.3
"PAV"

Figure 37

6.5.3 Network 9 to 12

In these networks, I have defined all common outputs to be valid for all cylinders. High- and low-pressure valves, anterior and posterior auxiliary valves, Figure 37.

6.5.4 Network 13

I have created a FC to write codes in order to simulate pressure dropping during the leak test. In network 13 this FC is called. To enable FC1 the test bench has to be in leak or full test modes, Figure 38.

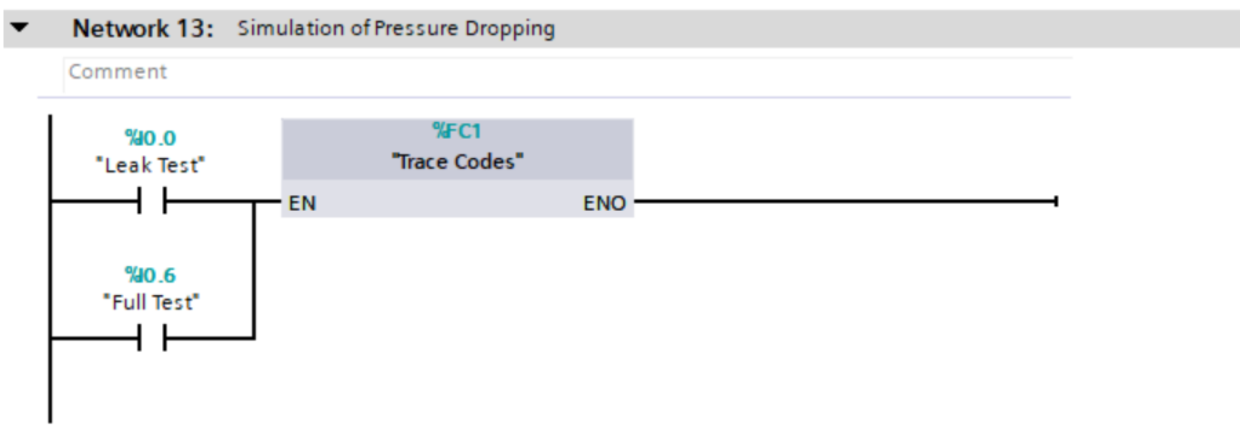


Figure 38

6.5.5 Network 14

During the execution of tests, various errors may occur. There is a specific function in main block which is in charge to detect any possible errors. FC2 is called in this network, Figure 39.

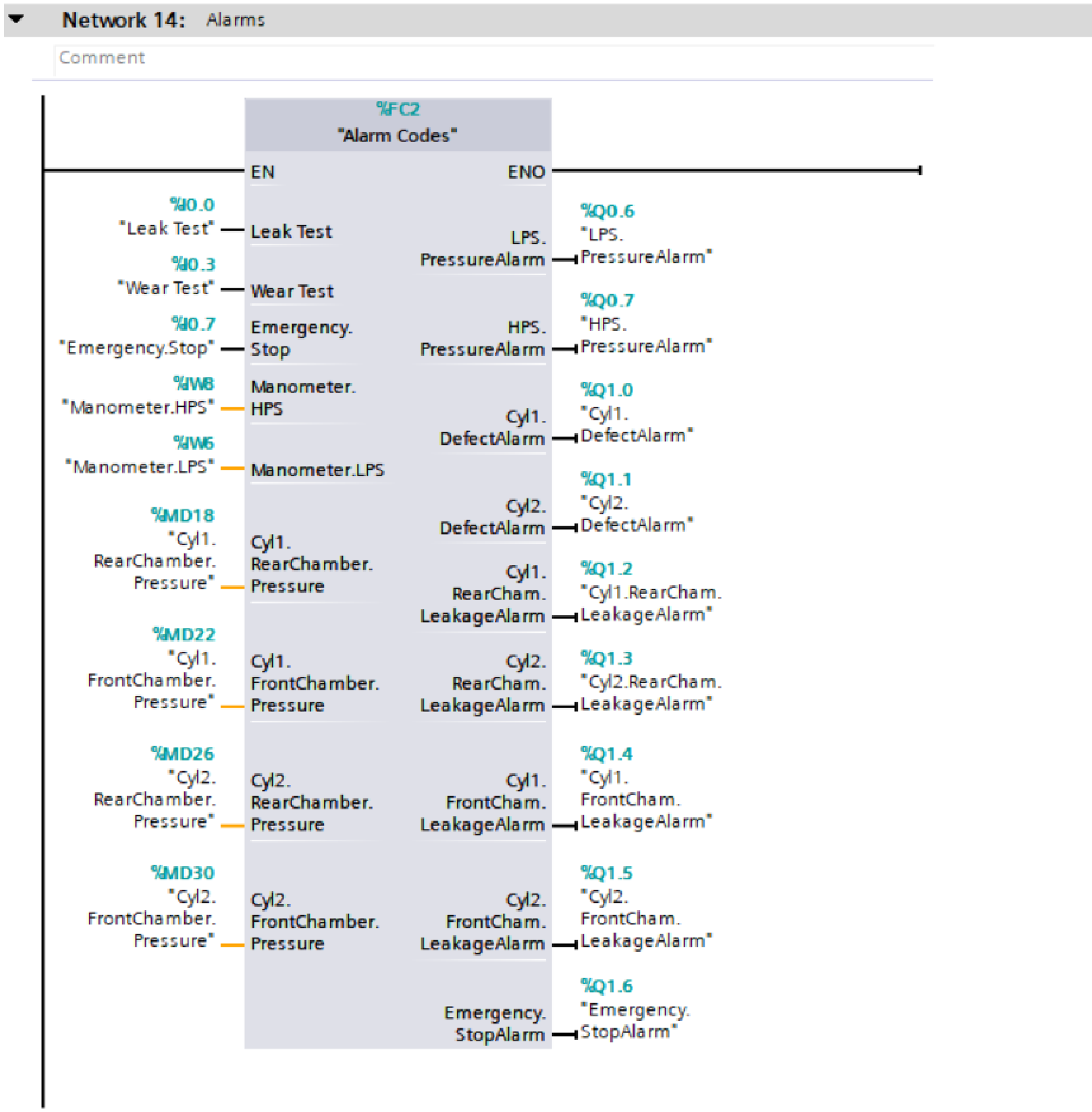


Figure 39

6.5.6 Network 15

When the piston movement limitation time error occurs, the defective cylinder will be removed from test process by disabling its function block in main OB. To reset this fault, it is essential to execute the related code in the main block because there is no access to the defective cylinder FB as it is disabled, Figure 40.

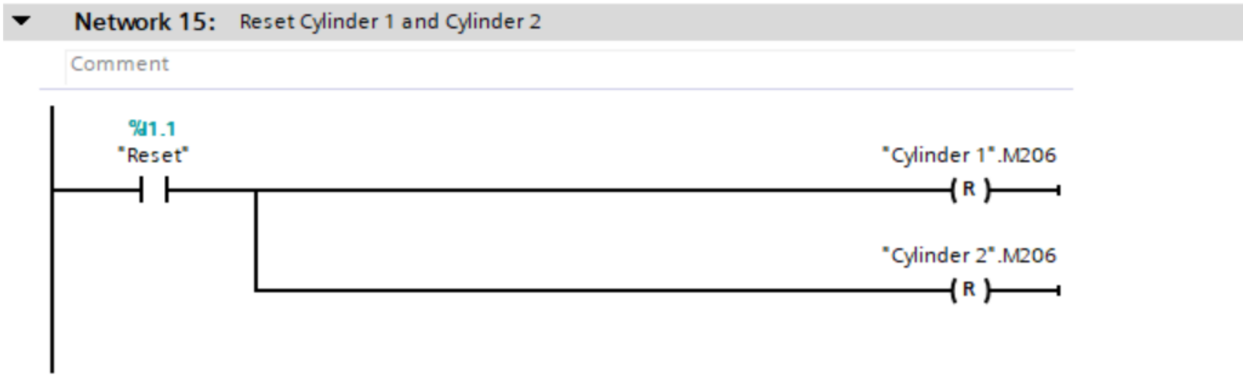


Figure 40

6.5.7 Network 16

This network provides the required conditions to trigger trace function recording during the leak test. Whenever the individual chambers of the cylinders are under leakage test, its related memory bit is activated. This changes the state of “M0”, and data recording starts, Figure 41.

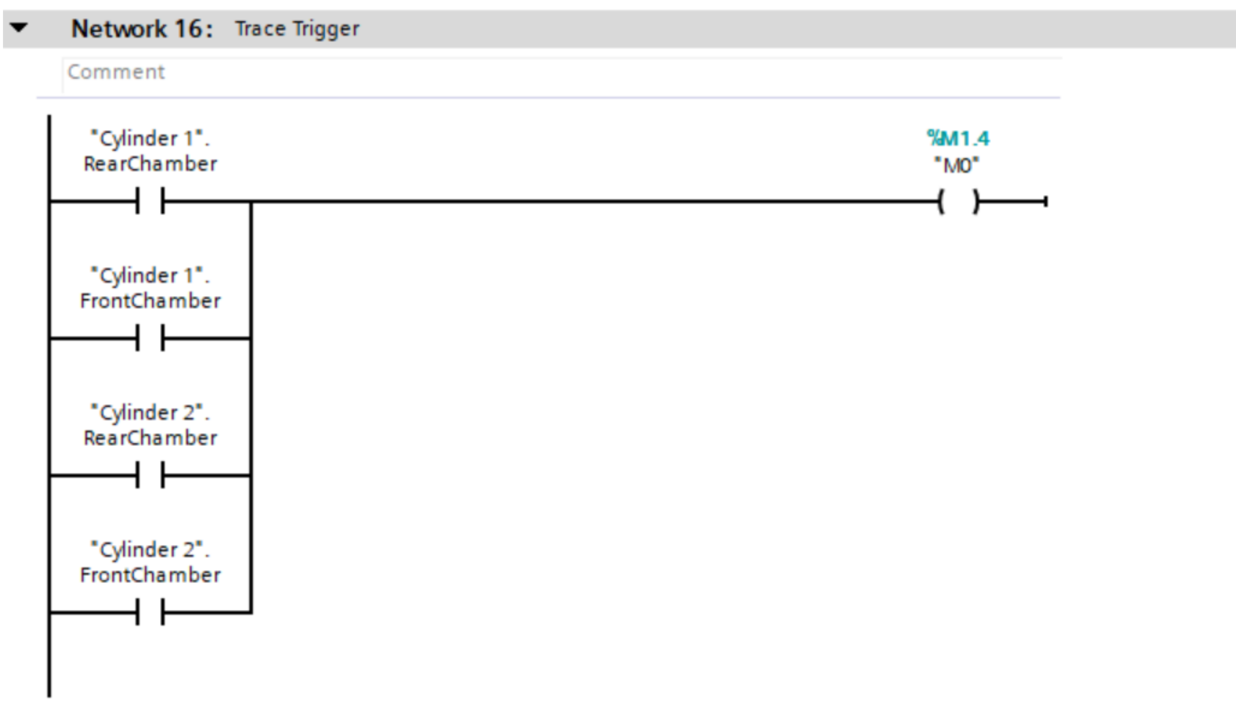


Figure 41

6.5.8 Network 17

This network provides the required conditions to trigger the trace function recording during the wear test in both “wear and full” modes. As shown in figure 42, when we are either in wear or full mode, as soon as pushing the start button, the recording will be started.

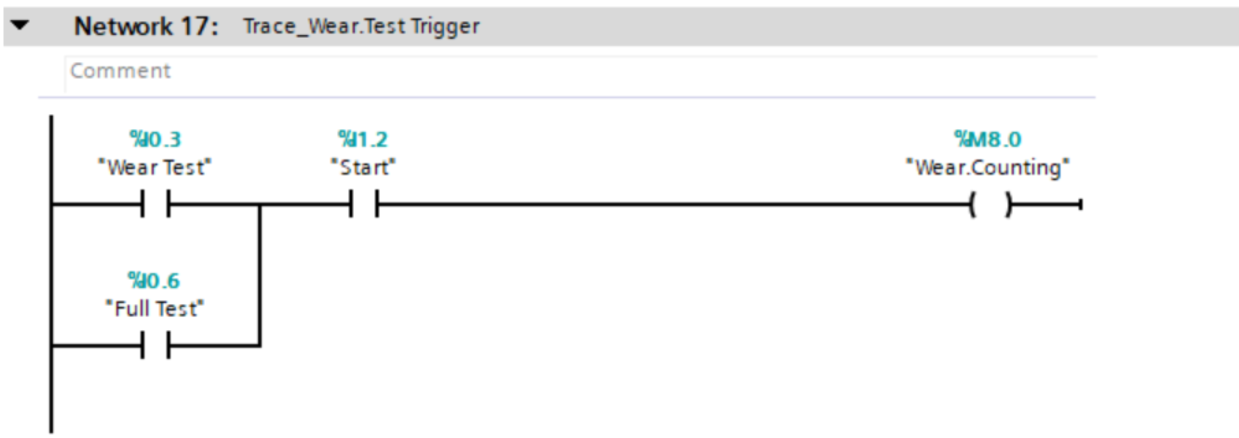


Figure 42

6.6 Alarm Codes (Function 2)

As I mentioned in introduction It is essential to predict possible faults that may occur during each test mode. In this section I used a function named Alarm Codes for this purpose where different errors and proper actions are considered. FCs in comparison with FBs do not occupy the memory for cyclic data storages. So, when it is possible it is better to use the FC instead of a FB to increase the performance speed. As the alarm system is common between all cylinders and have to be provided with actual parameters when called, it can be defined in a FC. To create a new function we use project tree to add a new block, Figure 17. Then in the new function header, we should define the local tags.

6.6.1 Network 1

The first network of this function is related to the air supply pressure. The practical air pressure in this project is 6 bar for low and 10 bar for high pressure. Two digital manometers sense the pressure of both supplies. When the low-pressure air used in the wear test drops from 4 bar and the high-pressure air used in the leak test drops from 8 bar an alarm will be activated in each case, Figure 43.

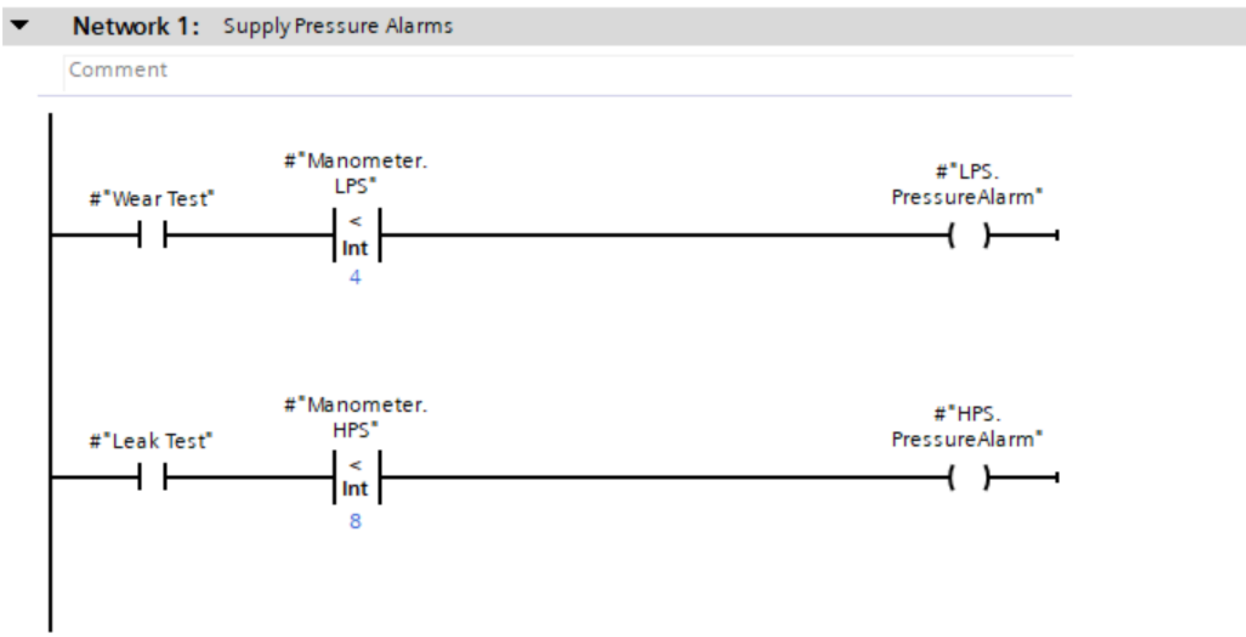


Figure 43

6.6.2 Network 2

As I explained in previous sections, when a cylinder is removed from the test process due to a defect an alarm will inform operator as well, Figure 44.

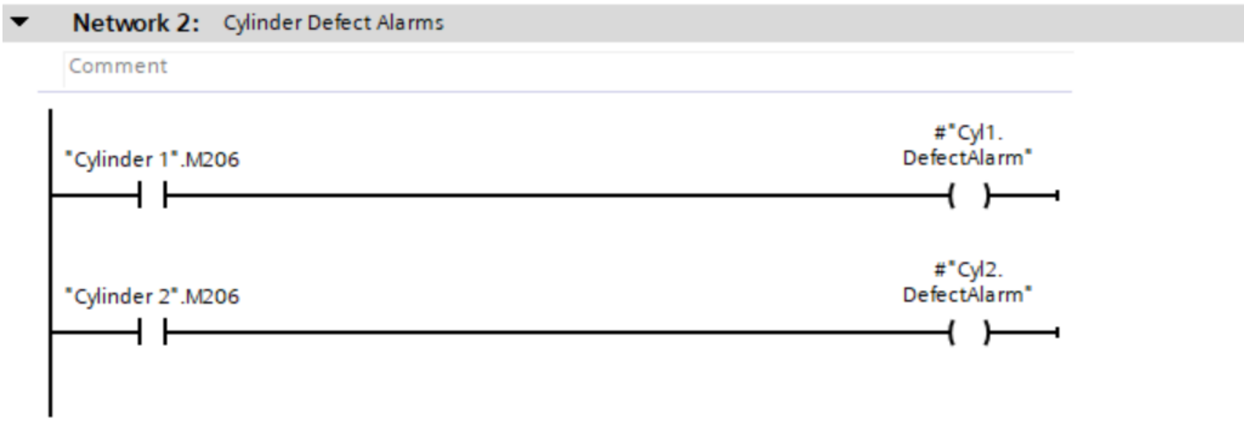


Figure 44

6.6.3 Network 3

Monitoring the pressure during leak test is very important because we consider any possible leakage in the cylinders' chambers. If the pressure drops below 8 bar during the leakage test for each chamber, an alarm will be activated for each individually, Figure 45.

6.6.4 Network 4

The output in this network will be activated if the emergency stop switch is pressed and an alarm will inform the operator, Figure 46.

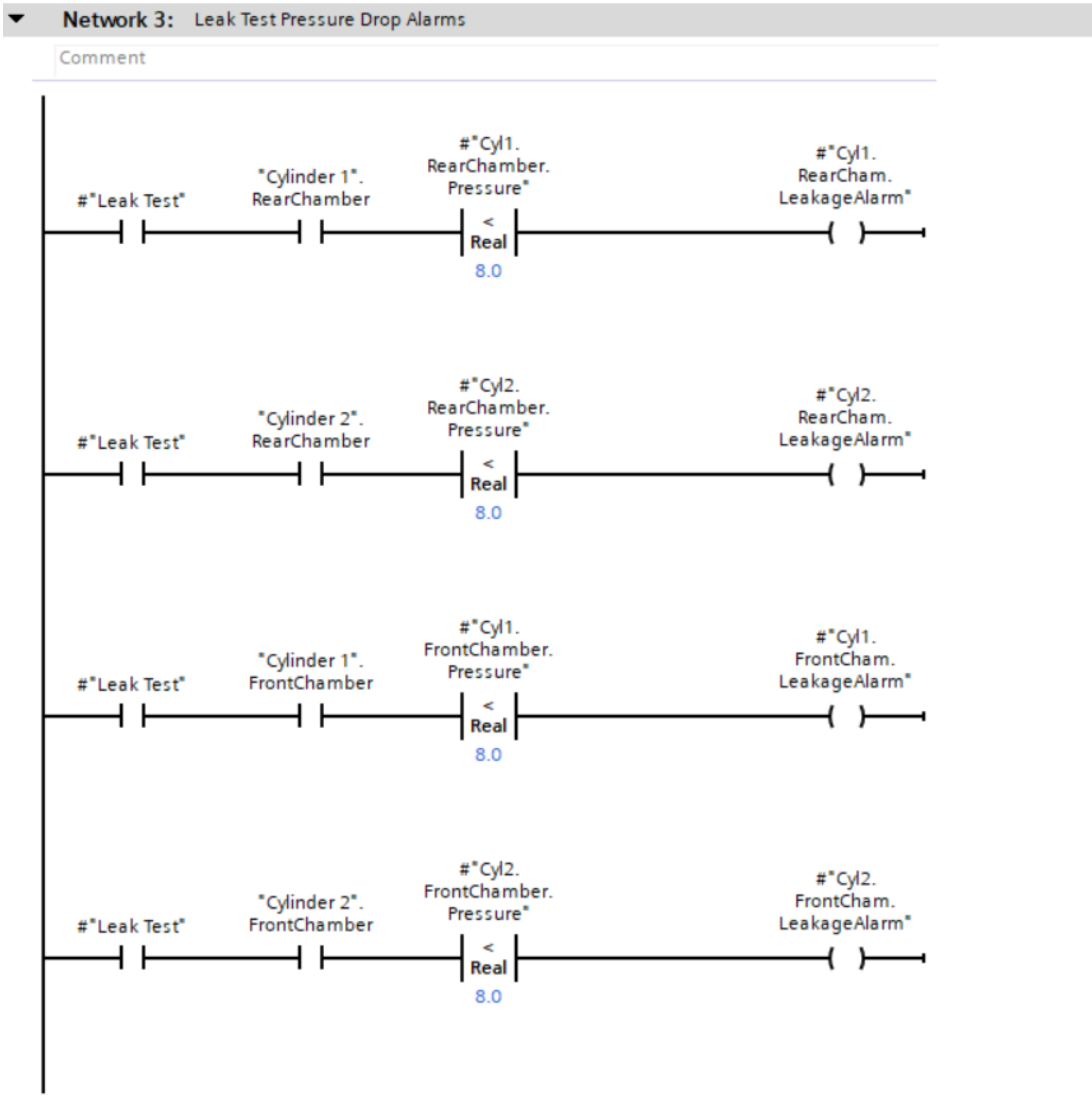


Figure 45

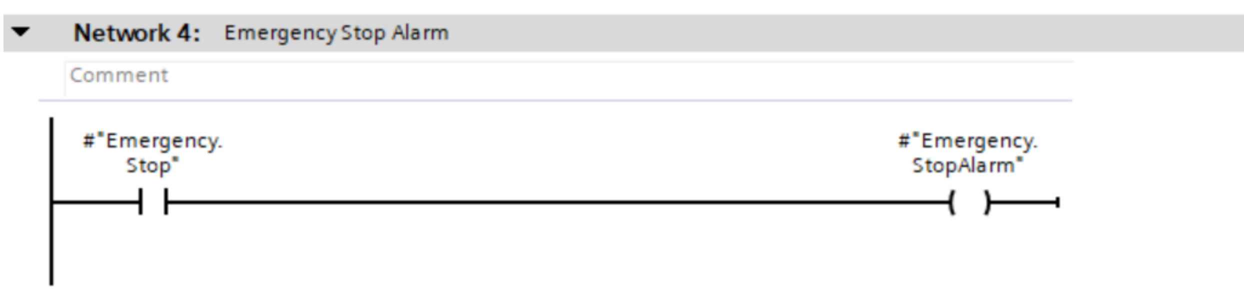


Figure 46

6.7 Trace Codes (Function 1)

6.7.1 Global data blocks

In order to simulate pressure dropping during the leak test and applying it in the trace function, I have used a function to create the required codes. I could use a function block as well. However, I used a function just to show how we can use it with a global data block. To create a global data block like other programming blocks, double click on "add new block" in the project tree, Figure 17. Variables which are located in global data blocks can be used by all other programming blocks (OB, FB, FC), Figure 47. As I need to use static variables in FC1, first I defined them in the global data block and then I used them in the function codes.

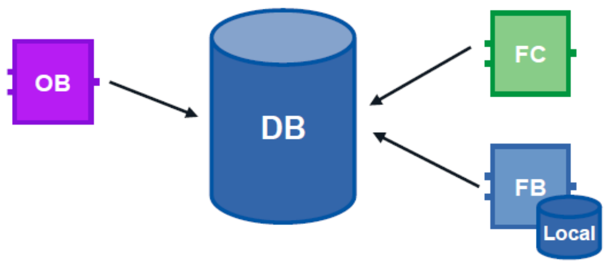


Figure 47

Some advantages of global data blocks are that they allow high-speed access and have a well-structured memory area. In addition, global DBs can be accessed by all other blocks in the user program and the structure of the global DBs can be composed of all data types.

6.7.2 Network 1 to 5

In the first network's code, (U) and (V) are two static variables which are called from Trace.Codes [DB1]. In this code, (V) is a constant that adjusts the rate of pressure dropping. I selected V=0.00001 which can be defined by the user.

I used in the calculation block in the second network and exponential function formula y=e^x, as I wanted to generate 10 bar as starting pressure in the chamber. To have 10 in output (y), y=10=e^x so x=ln(y)=ln(10)=2.3025... I used x=2.308 as a start value to have y=10.05 bar inside the chamber.

Network 1: Codes to Generate Pressure Drop Over Time

Comment

```
IF "M0" THEN
    // Statement section IF
    "Trace.Codes".U  += "Trace.Codes".V ;
ELSE
    "Cyl1.RearChamber.Pressure" := 0 ;
    "Cyl2.RearChamber.Pressure" := 0 ;
    "Cyl1.FrontChamber.Pressure" := 0 ;
    "Cyl2.FrontChamber.Pressure" := 0 ;
    "Trace.Codes".U := -2.308 ;
END_IF ;
```

Figure 48

When the code: "Trace.Codes".U += "Trace.Codes".V is executed, (U) will be incremented by rate of (V). As a result, the output of the calculation block which simulates the pressure dropping in the chamber, will be decreased by an exponential rate due to the formula output=exp(u)=e^(u). I need to have a negative value of x=U=-2.308 as a start value and execute y=output=e^(-x) to have the decreasing form of an exponential function.

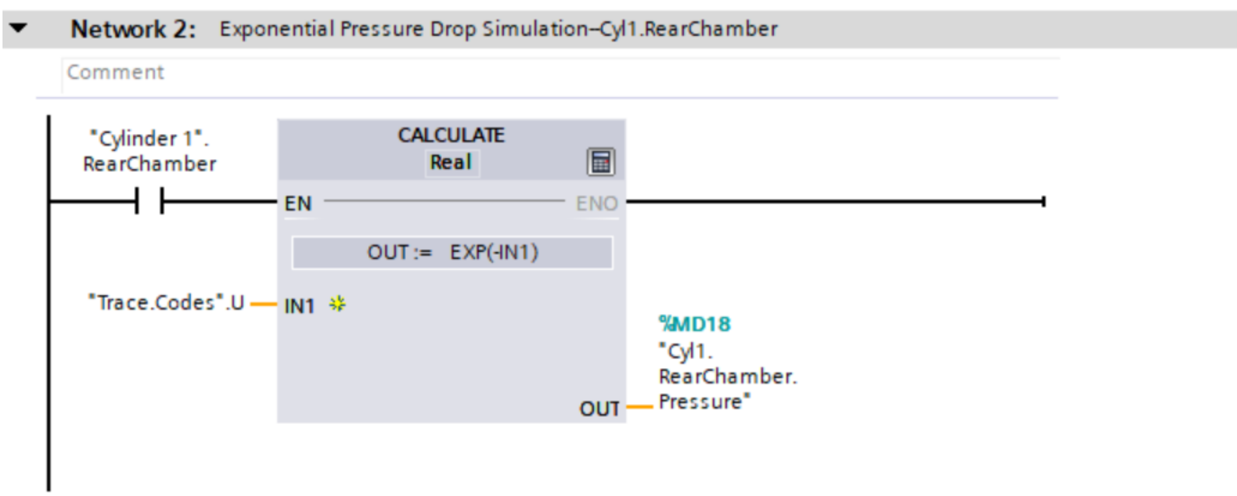


Figure 49

Networks 2 to 5 generate exponential pressure dropping simulation in the rear chamber of cylinder 1, the front chamber of cylinder 1, the rear chamber of cylinder 2 and the front chamber of cylinder 2.

Using SCL for mathematical calculations is very useful, but another possibility is to use the math functions “calculate”. I have used both in the trace codes function.

6.7.3 Trace Function and Data Recording

In the pneumatic cylinder’s lifetime testing, it is crucial to record the data and analyze them. And since the duration of this test may be weeks and months, the importance of storing information becomes much greater. The TIA Portal has a powerful capability for recording and analyzing measurements. In my project I have used trace function for this purpose. These measurements are saved to the device and can then be read and saved permanently. This ability to trace and make logical analysis makes this system well-adapted to monitoring highly dynamic processes. It is also possible to combine signal waves from several measurements and make an overlay measurement. These can then be compared and synchronized. In a last step the measurements on the memory card can also be used in the diagnostic interface on the web server.

In this project the total number of wear sequence during the wear test and the pressure value inside the cylinder’s both chambers during the leak test are recorded. For this purpose, I created two separate “traces”. “Trace_Wear.Test” and “Trace_Leak.Test”. Here I will explain how to create and configure the trace function to record and save the measurements for the leakage testing. Tracing process in the wear test including the steps and settings are the same.

To create this part of the program it is necessary to call the trace and logic analyzer function from project tree. By double clicking on the project tree, the section ‘’add new trace’’, the trace configuration tab will be opened. These settings are important and determine how data is stored. In the trace project tree and under configuration, the signals which need to be recorded are selected. To do so we can call signals from the tag table or the cylinders’ data block(s), Figure 50. It is possible to record tags from the different operand areas including process image input, process image output, bit memory, data blocks and I/O devices.

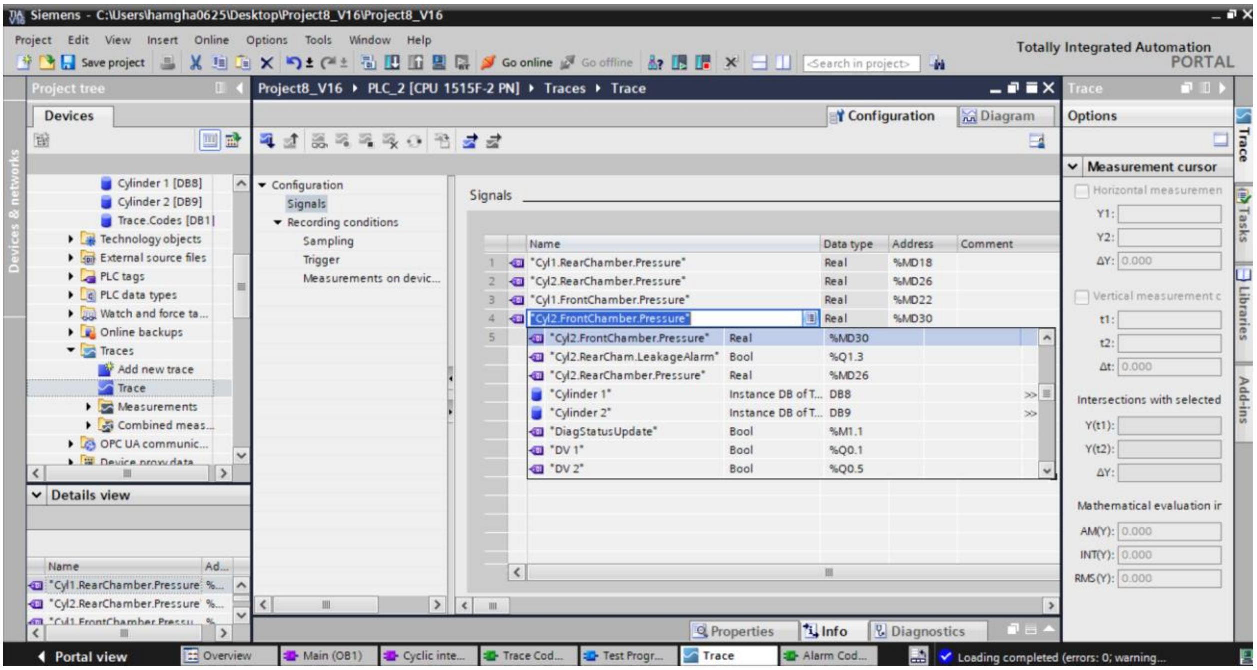


Figure 50

If the main [OB1] is selected under recording conditions for sampling, the program will record the data whenever the main [OB1] is executed. Here the problem is that the duration of the main [OB1] execution is not always the same and will vary in length. To solve this issue, I created another organization block with subsection “cyclic interrupt”. By sampling the data with cyclic interrupt, it is possible to record in a time frame. Now we have two options to select time instead of cycle. I set to record every 1 second. This can be further defined by the user according to the specific requirements. By selecting “max recording duration”, the program will record continuously to use all memory space. I set recording duration to cover only the leakage testing duration and when each chamber is under high pressure.

The next section is related to the trigger setting. In the trigger mode I select “trigger on tag” and choose a memory bit “M0” from the tag table. This memory bit is placed on network 16 in the main block. “M0” will be activated whenever in leak test mode and when one of the cylinder chambers is under pressure. As I defined the Event equals to True, recording will be started by activating “M0”. If it is required to start recording earlier than trigger, it can be set in pre-trigger section by entering the duration time before the trigger as well.

One of the advantages of the trace function in the TIA Portal is its capability of repeating measurements automatically and storing the recorded value in the device retentively. This can be the internal memory of CPU or a memory card. In this case even when there is no connected PC or HMI to the CPU, the trace function is able to record and save the data on the memory card. In the configuration tab, the last part is related to the setting of “Measurements on device (memory card)”. There is one important point here. Only when “trigger on tag” is selected in the previous section, is there the possibility to check the box “save measurements on device”, otherwise it is not possible to activate this function. Then we choose the number of measurements and the variety of response when the number of measurements is reached. It can be selected to deactivate recording or to overwrite starting from the oldest recording.

After setting is finished, it is necessary to transfer the trace configuration to the device by clicking on the related icon in the trace toolbar.

The diagram tab displays the selected signals of a recording. Under the curve diagram there is signal table with its setting options.

During the leak test and when the trigger tag becomes active, recording is started. It lasts until the leakage test of first chamber is finished. Then the recording is paused until the next chamber is tested for leakage. By reactivating “M0” for the next chamber, recording resumes immediately. Figure 51

Recordings can be added to the measurements folder on the project tree. This folder can be accessible in both offline and online modes.

When we set the trace configuration to record and save the data on the device memory, this will be done when CPU is in run mode and the program is executing. The big advantage is that the recording function is active in offline mode as well and measurements will be saved on device (memory card). These recorded measurements are accessible on the "Measurements on device (memory card)" folder. This folder is displayed when there is an online connection to the CPU.

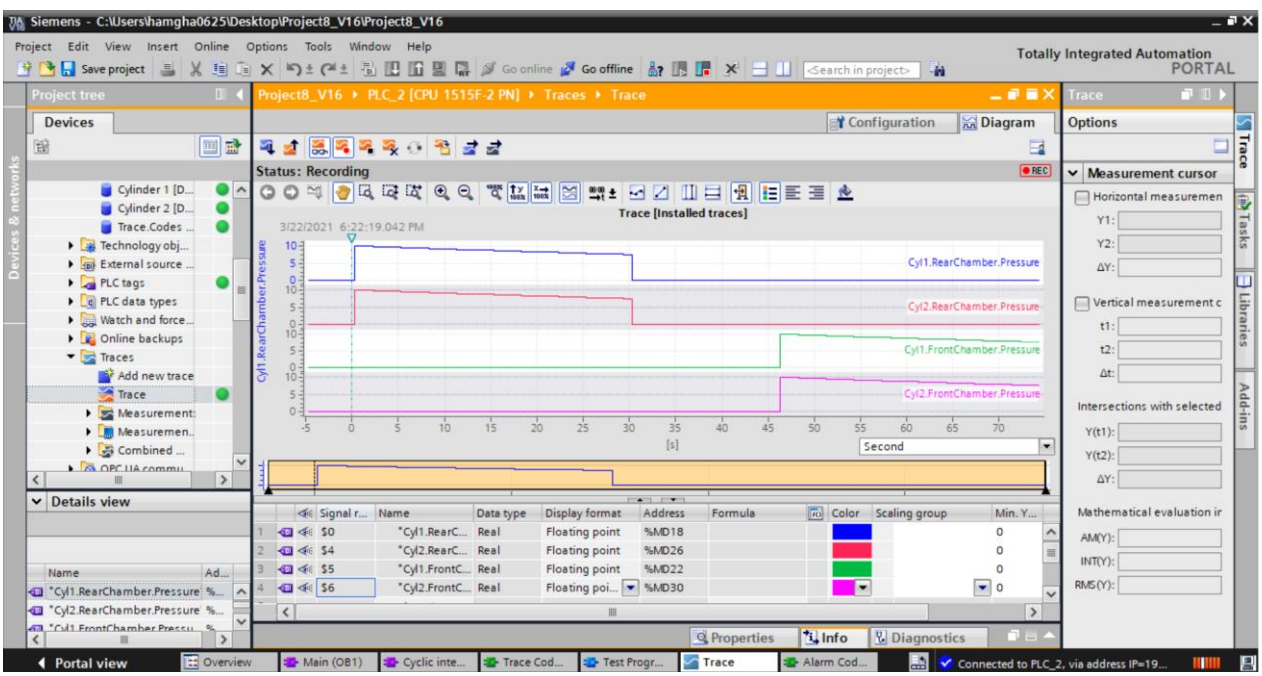


Figure 51

In figure 52 the configuration and recording tabs for the wear test is shown. One of the advantages of using the trace function is that even if there is no physical counter in the system, it is possible to count wear cycle and store it in the memory. By switching from online to offline mode, disconnecting the PC from CPU or switching the CPU to the stop mode, these stored measurements will be retained.

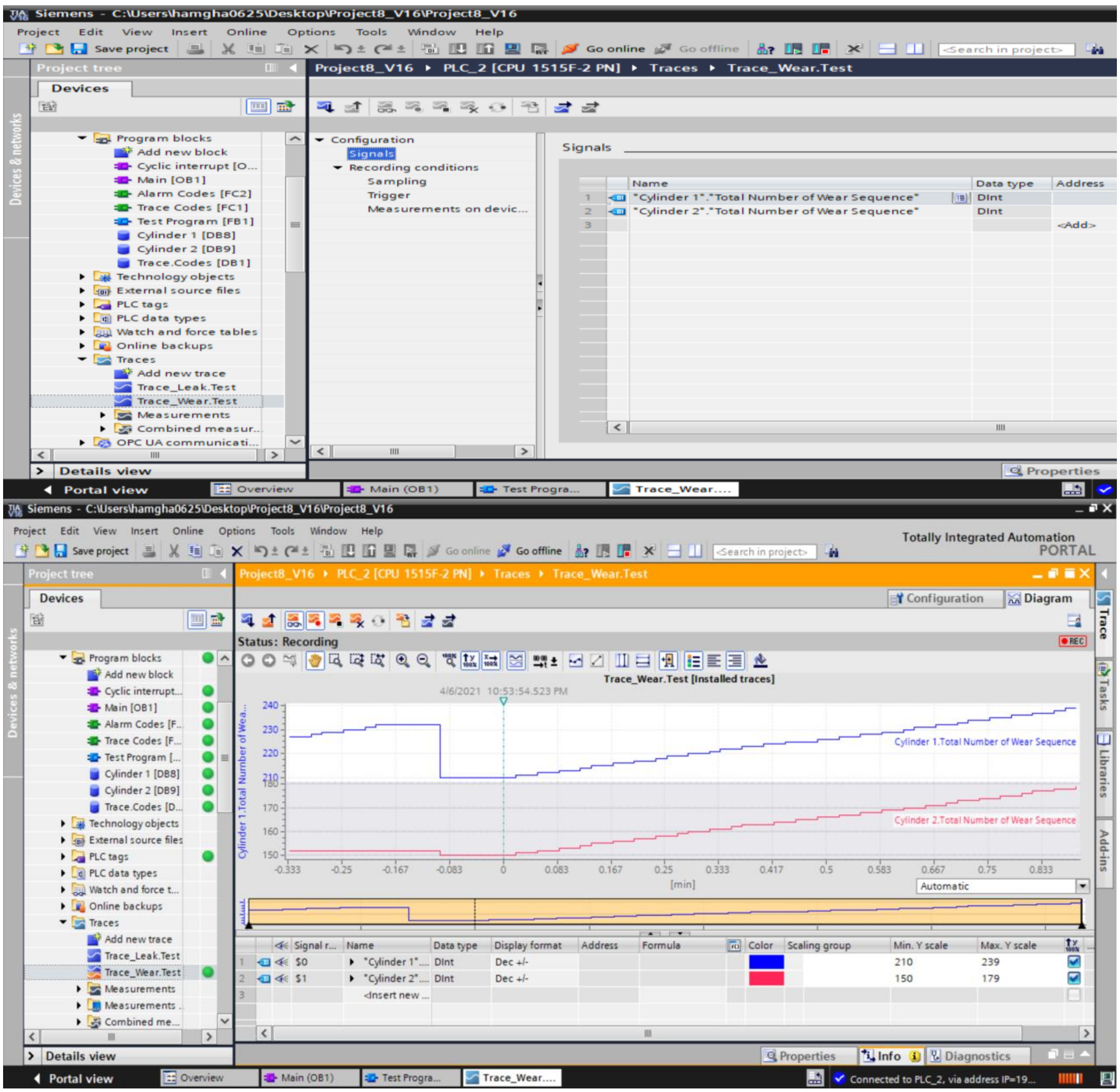


Figure 52

6.7.4 Exporting the measurements

When it is required to export the recorded data from the measurements folder in the project or from the device (memory card), it is possible to transfer the measurements as a file with the file extension "*.ttrecx" or "*.csv". If we need to export a file which is compatible with Excel, we have to select the "csv" format, Figure 53.

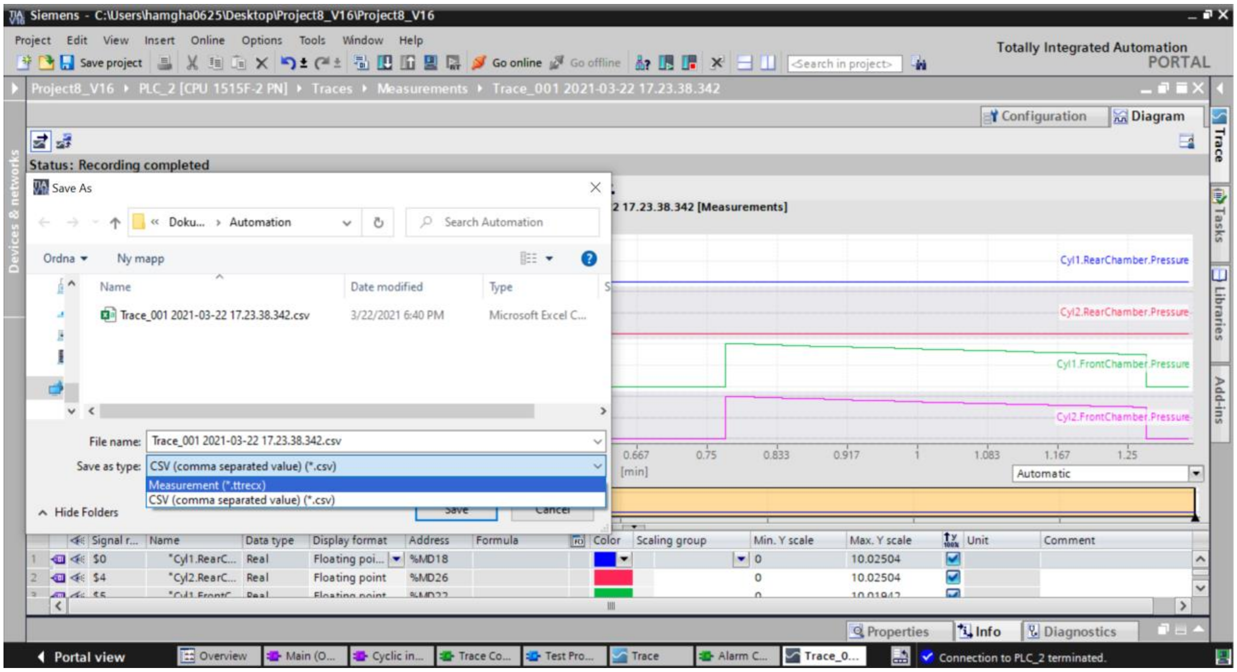


Figure 53

Conclusion

In this thesis project a test bench is designed in order to measure the pneumatic cylinders' lifetime. The main objective of the project is programming, data acquisition and automatic control of the entire process.

Siemens' PLC and the TIA Portal are the major cores of the project. I used the TIA Selection Tool to select the required hardware precisely. The main programming language used in the project is LAD, although SCL language is also used for simulating pressure dropping.

During the project I studied how to organize the program, the application of various programming blocks, how to use different data types, diagnostic and troubleshooting.

A new approach is applied in this project for data acquisition, storing and analyzing the test measurement. By using the trace function, it is possible to save the data without needing an additional device. In this way the data is stored in the CPU itself.

With simulation of the program and whole testing process, it is possible to develop the program at low cost and with little effort. In this project FluidSim and PLCSIM Advanced are used to simulate the test bench and control the program. Festo didactic EzOPC software is also used to help to connect FluidSim and TIA Portal. A good program is one that meets all necessary safety conditions and provides easy way for diagnostic and maintenance. It should be as well constructed and understandable as possible. The software used in this project met all these requirements and will become useful for developing and improving the project in the future.

References

- Siemens Programming Guideline for S7-1200/1500
- Siemens function manual, Simatic/Sinamics using the trace and logic analyzer function Edition 12/2019
- Siemens function manual, Simatic S7-1500 and PLCSIM Advanced Edition 11/2019.
- L. Mazza, “Mechatronic Engineering: Fluid Automation-unit4-Pneumatic Actuators Valves”.
- L. Mazza, “Mechatronic Engineering: Fluid Automation-unit12
- www.xpneumatic.com/what-are-pneumatic-components
- www.explainthatstuff.com/pneumatics.html

Appendix and Full PLC Codes

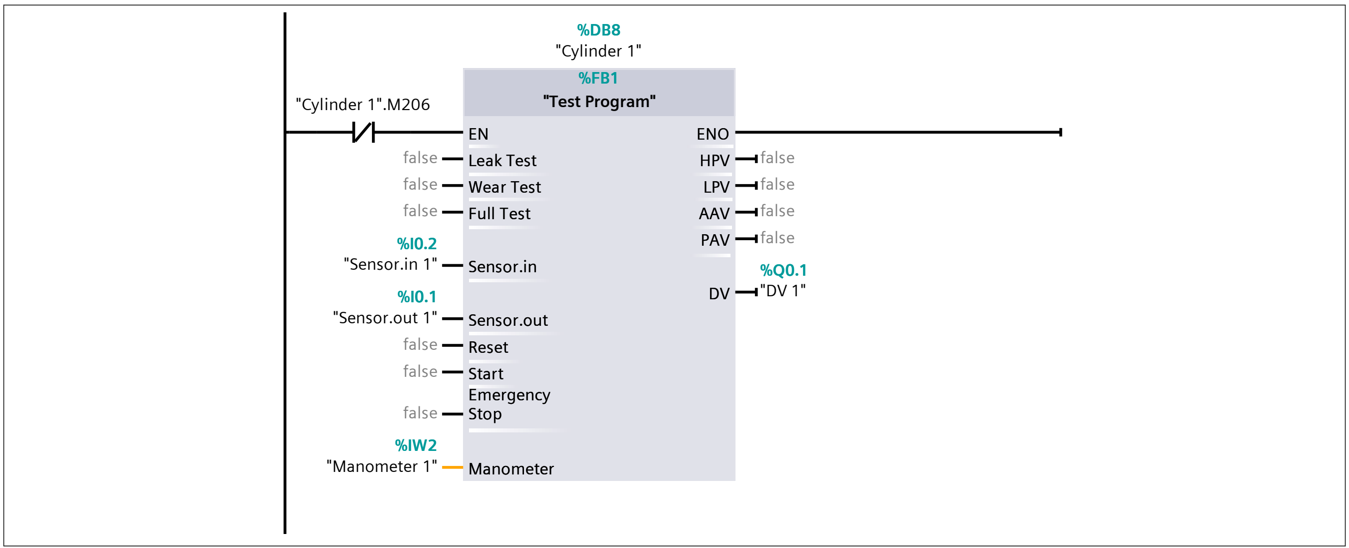
In the following section the complete codes and the project data in the TIA Portal are attached.

Main [OB1]

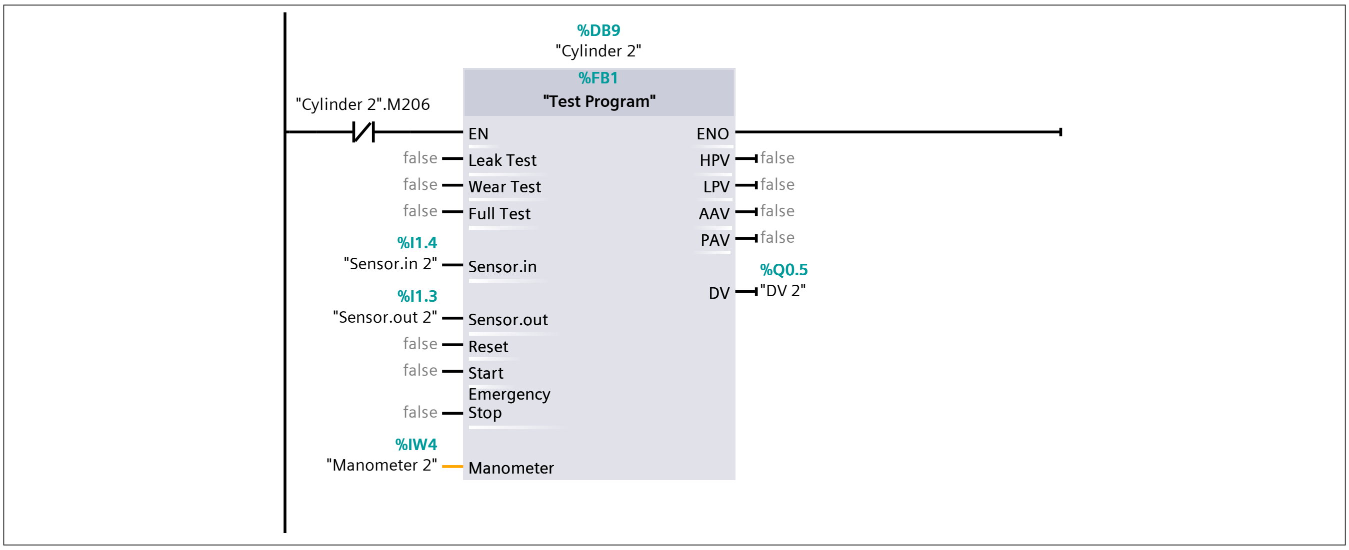
Main Properties					
General					
Name	Main	**Number**	1	**Type**	OB
Language	LAD	**Numbering**	Automatic		
Information					
Title	"Main Program Sweep (Cycle)"	**Author**	Alireza.Qadiri	**Comment**	The main OB is executed cyclically. In this block we place the instructions that control our application and call additional user blocks.
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

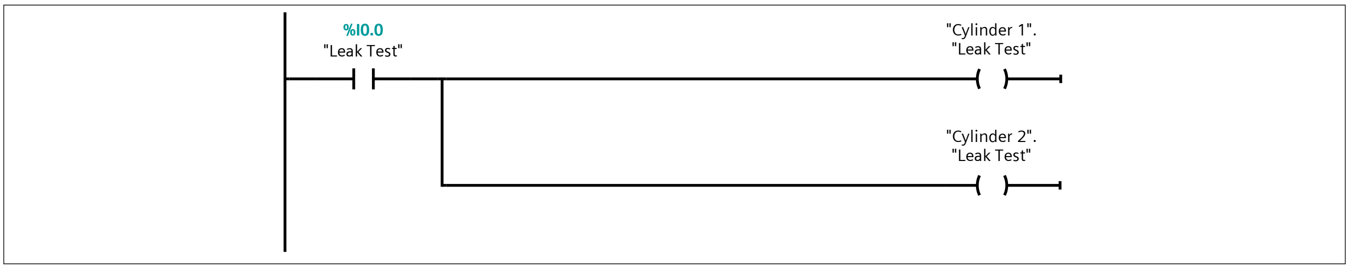
Network 1: Calling Cylinder 1



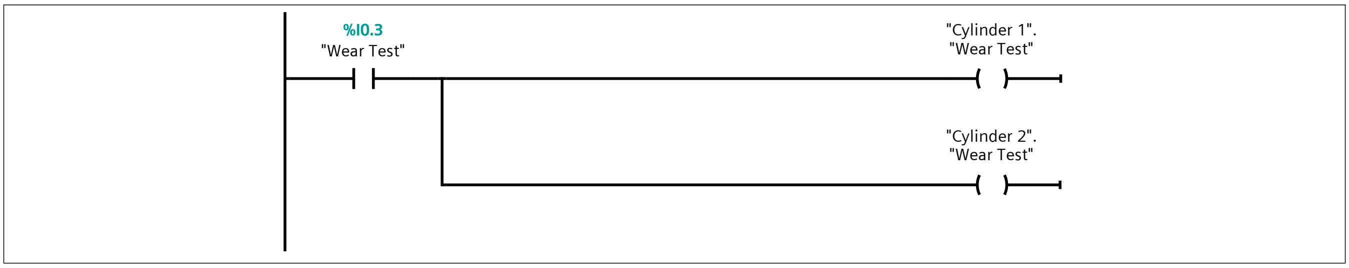
Network 2: Calling Cylinder 2



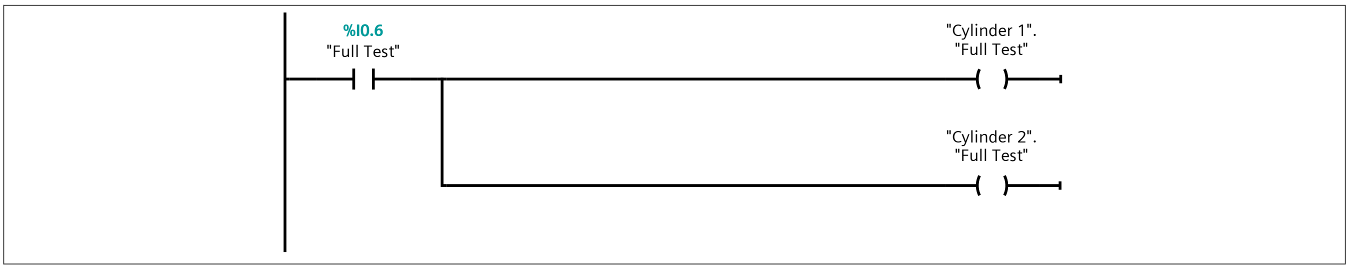
Network 3: Activating Leak Test Switch For All Cylinders



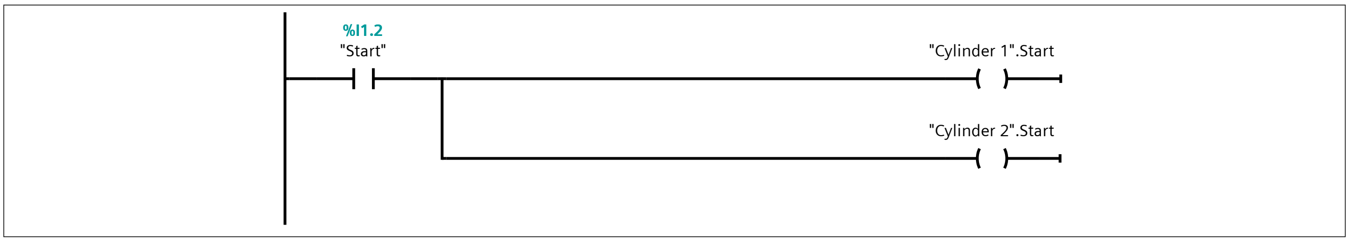
Network 4: Activating Wear Test Switch For All Cylinders



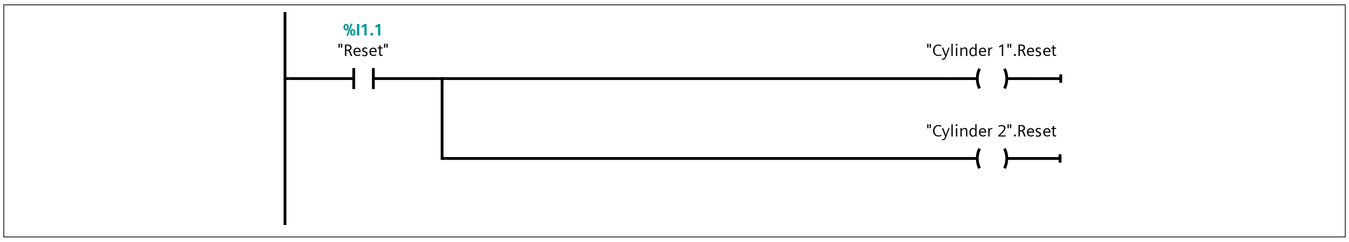
Network 5: Activating Full Test Switch For All Cylinders



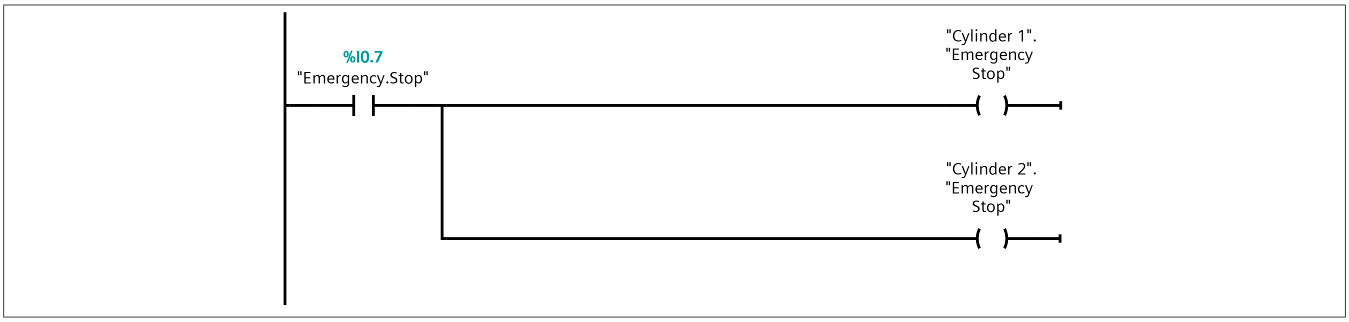
Network 6: Activating Start Switch For All Cylinders



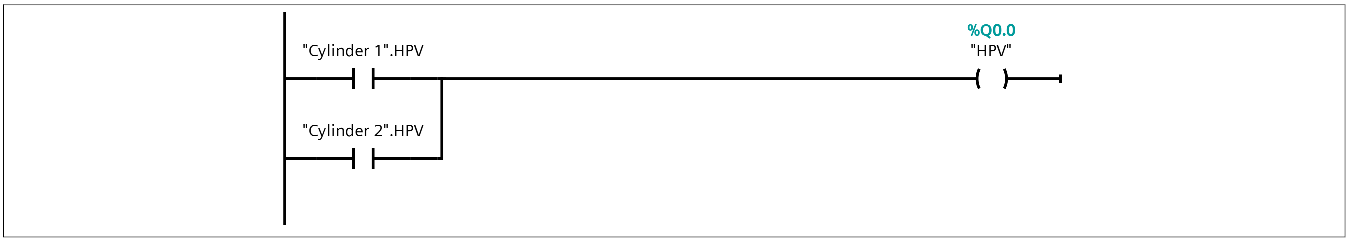
Network 7: Activating Reset Switch For All Cylinders



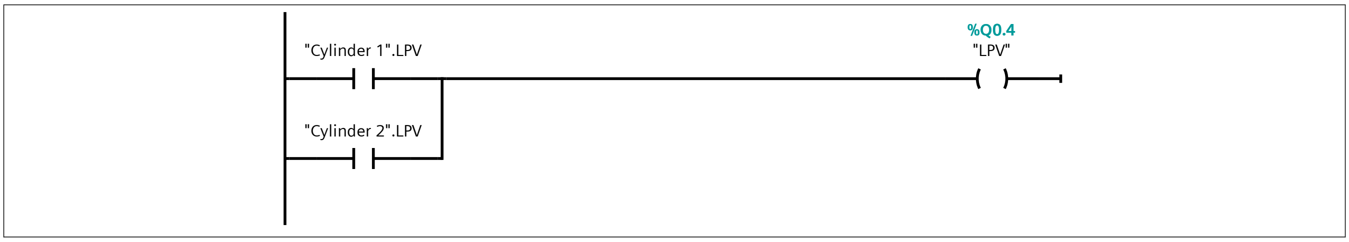
Network 8: Activating Emergency Stop Switch For All Cylinders



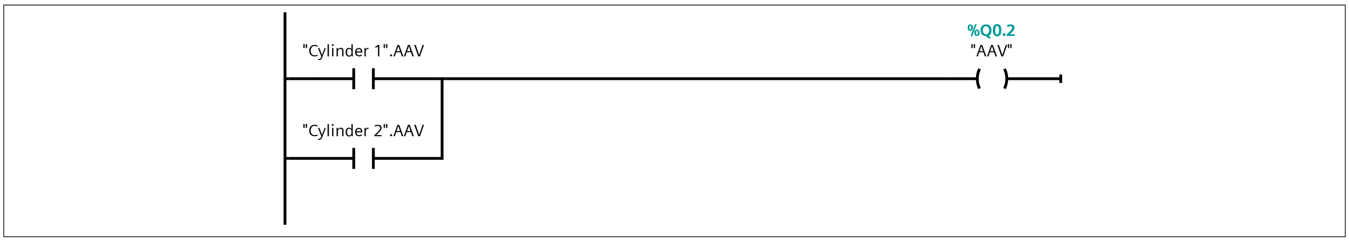
Network 9: Activating High Pressure Valve For All Cylinders



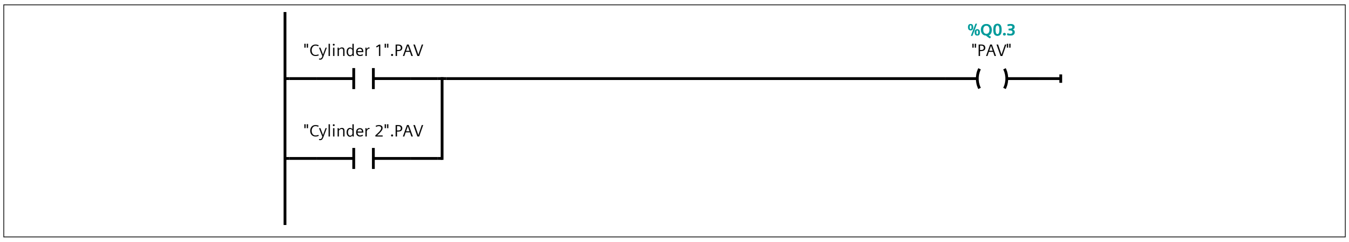
Network 10: Activating Low Pressure Valve For All Cylinders



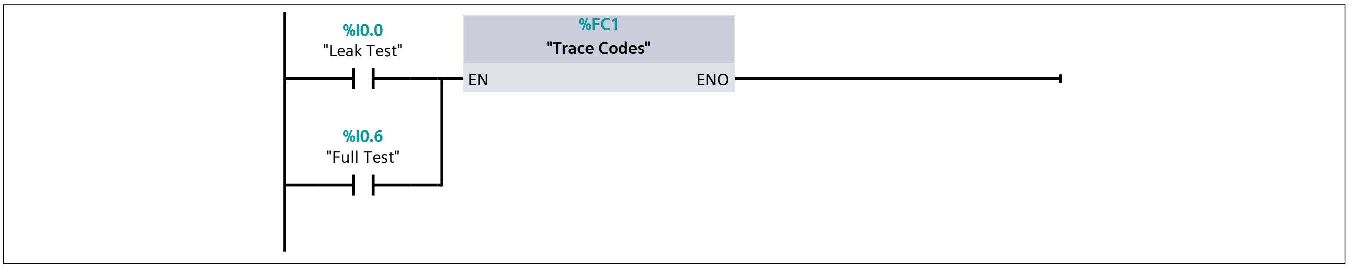
Network 11: Activating Anterior Auxiliary Valve For All Cylinders



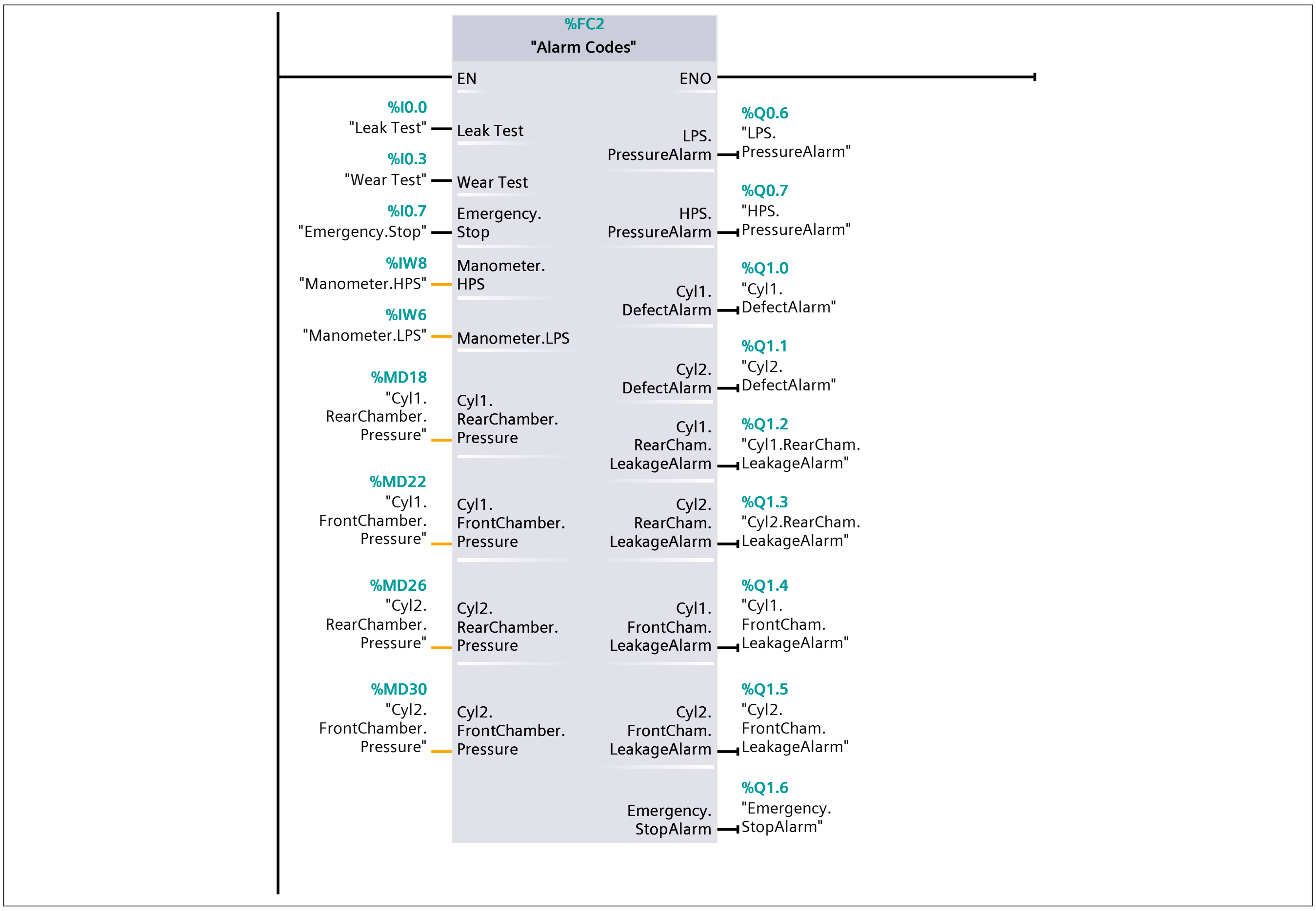
Network 12: Activating Posterior Auxiliary Valve For All Cylinders



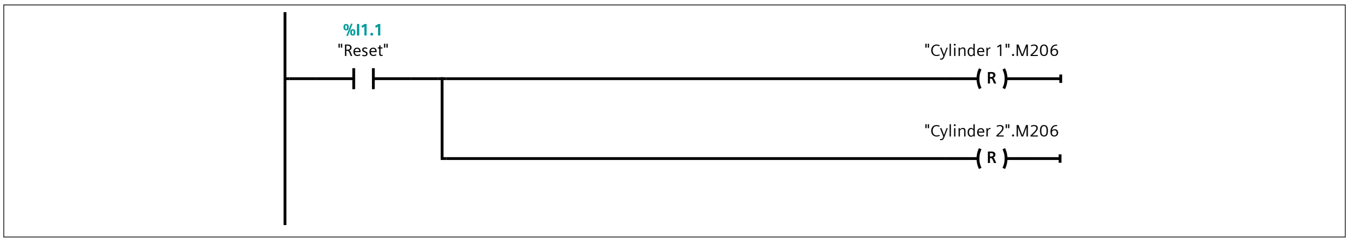
Network 13: Simulation of Pressure Dropping



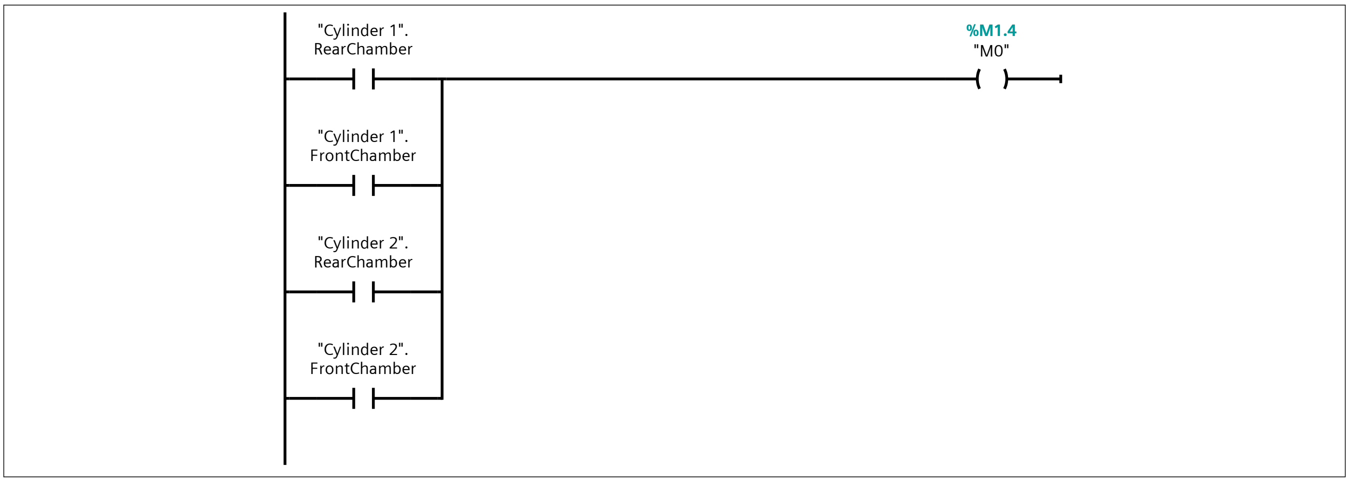
Network 14: Alarms



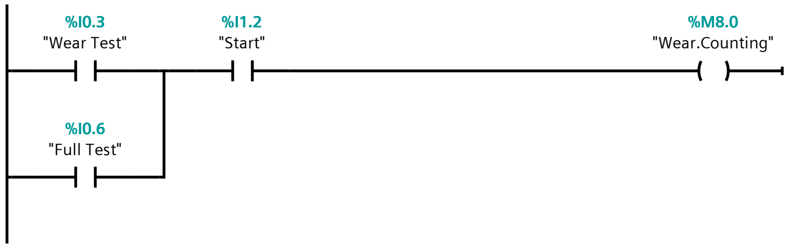
Network 15: Reset Cylinder 1 and Cylinder 2



Network 16: Trace_Leak.Test Trigger



Network 17: Trace_Wear.Test Trigger



Cyclic interrupt [OB30]

Cyclic interrupt Properties					
General					
Name	Cyclic interrupt	**Number**	30	**Type**	OB
Language	LAD	**Numbering**	Automatic		
Information					
Title	Cyclic interrupt	**Author**	Alireza.Qadiri	**Comment**	This Cyclic interrupt OB is created to be used in trace function. No need to write any codes here.
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
Temp			
Constant			

Test Program [FB1]

Test Program Properties					
General					
Name	Test Program	**Number**	1	**Type**	FB
Language	LAD	**Numbering**	Automatic		
Information					
Title	Test Bench Program	**Author**	Alireza.Qadiri	**Comment**	In this Function block the main program of the test bench is written.
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
Leak Test	Bool	false	Non-retain	True	True	True	False		
Wear Test	Bool	false	Non-retain	True	True	True	False		
Full Test	Bool	false	Non-retain	True	True	True	False		
Sensor.in	Bool	false	Non-retain	True	True	True	False		
Sensor.out	Bool	false	Non-retain	True	True	True	False		
Reset	Bool	false	Non-retain	True	True	True	False		
Start	Bool	false	Non-retain	True	True	True	False		
Emergency Stop	Bool	false	Non-retain	True	True	True	False		
Manometer	Int	0	Non-retain	True	True	True	False		
▼ Output									
HPV	Bool	false	Non-retain	True	True	True	False		
LPV	Bool	false	Non-retain	True	True	True	False		
AAV	Bool	false	Non-retain	True	True	True	False		
PAV	Bool	false	Non-retain	True	True	True	False		
DV	Bool	false	Non-retain	True	True	True	False		
InOut									
▼ Static									

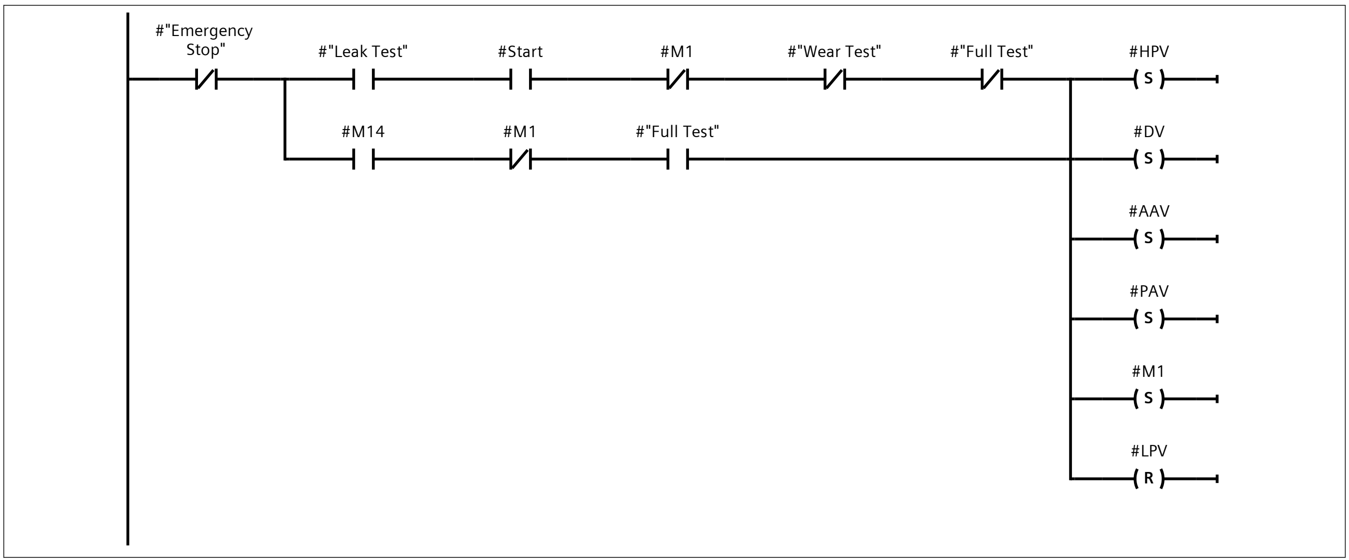
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
M1	Bool	false	Non-retain	True	True	True	False		
M2	Bool	false	Non-retain	True	True	True	False		
M3	Bool	false	Non-retain	True	True	True	False		
M4	Bool	false	Non-retain	True	True	True	False		
M5	Bool	false	Non-retain	True	True	True	False		
M6	Bool	false	Non-retain	True	True	True	False		
M7	Bool	false	Non-retain	True	True	True	False		
M8	Bool	false	Non-retain	True	True	True	False		
M9	Bool	false	Non-retain	True	True	True	False		
M10	Bool	false	Non-retain	True	True	True	False		
M11	Bool	false	Non-retain	True	True	True	False		
M12	Bool	false	Non-retain	True	True	True	False		
M13	Bool	false	Non-retain	True	True	True	False		
M14	Bool	false	Non-retain	True	True	True	False		
RearChamber	Bool	false	Non-retain	True	True	True	False		
FrontChamber	Bool	false	Non-retain	True	True	True	False		
M206	Bool	false	Non-retain	True	True	True	False		
Normalized value	Real	0.0	Non-retain	True	True	True	False		
Scaled value	Int	0	Non-retain	True	True	True	False		
▼ Full Test Counter	CTU_DINT		Non-retain	True	True	True	True		
CU	Bool	false	Non-retain	True	True	True	False		
CD	Bool	false	Non-retain	True	True	True	False		
R	Bool	false	Non-retain	True	True	True	False		
LD	Bool	false	Non-retain	True	True	True	False		
QU	Bool	false	Non-retain	True	True	True	False		

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
QD	Bool	false	Non-retain	True	True	True	False		
PV	DInt	0	Non-retain	True	True	True	False		
CV	DInt	0	Non-retain	True	True	True	False		
▼ Wear Test Counter	CTU_DINT		Non-retain	True	True	True	True		
CU	Bool	false	Non-retain	True	True	True	False		
CD	Bool	false	Non-retain	True	True	True	False		
R	Bool	false	Non-retain	True	True	True	False		
LD	Bool	false	Non-retain	True	True	True	False		
QU	Bool	false	Non-retain	True	True	True	False		
QD	Bool	false	Non-retain	True	True	True	False		
PV	DInt	0	Non-retain	True	True	True	False		
CV	DInt	0	Non-retain	True	True	True	False		
▼ Wear Test Cycle Counter	CTU_DINT		Retain	True	True	True	True		
CU	Bool	false	Retain	True	True	True	False		
CD	Bool	false	Retain	True	True	True	False		
R	Bool	false	Retain	True	True	True	False		
LD	Bool	false	Retain	True	True	True	False		
QU	Bool	false	Retain	True	True	True	False		
QD	Bool	false	Retain	True	True	True	False		
PV	DInt	0	Retain	True	True	True	False		
CV	DInt	0	Retain	True	True	True	False		
Total Number of Wear Sequence	DInt	0	Retain	True	True	True	False		
▼ Timer 1	TON_TIME		Non-retain	True	True	True	True		
PT	Time	T#0ms	Non-retain	True	True	True	False		
ET	Time	T#0ms	Non-retain	True	False	True	False		

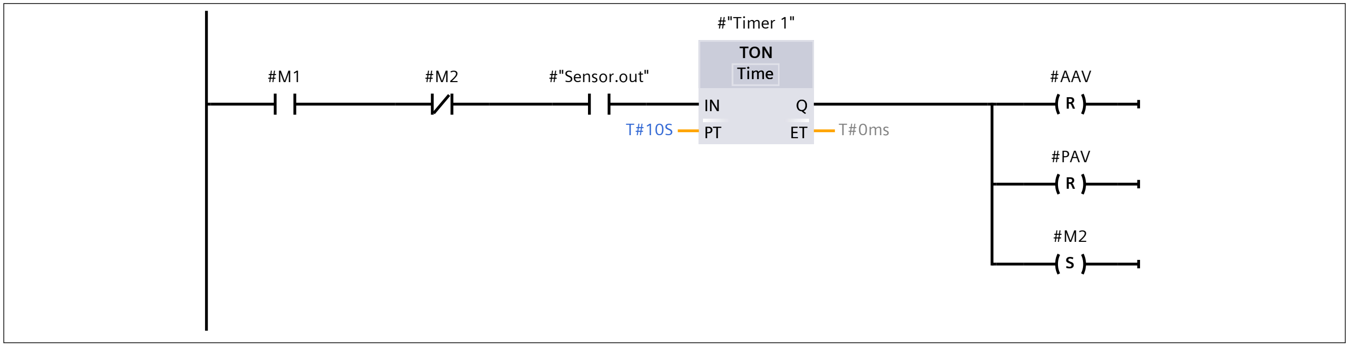
Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
IN	Bool	false	Non-retain	True	True	True	False		
Q	Bool	false	Non-retain	True	False	True	False		
▼ Timer 2	TON_TIME		Non-retain	True	True	True	True		
PT	Time	T#0ms	Non-retain	True	True	True	False		
ET	Time	T#0ms	Non-retain	True	False	True	False		
IN	Bool	false	Non-retain	True	True	True	False		
Q	Bool	false	Non-retain	True	False	True	False		
▼ Timer 3	TON_TIME		Non-retain	True	True	True	True		
PT	Time	T#0ms	Non-retain	True	True	True	False		
ET	Time	T#0ms	Non-retain	True	False	True	False		
IN	Bool	false	Non-retain	True	True	True	False		
Q	Bool	false	Non-retain	True	False	True	False		
▼ Timer 4	TON_TIME		Non-retain	True	True	True	True		
PT	Time	T#0ms	Non-retain	True	True	True	False		
ET	Time	T#0ms	Non-retain	True	False	True	False		
IN	Bool	false	Non-retain	True	True	True	False		
Q	Bool	false	Non-retain	True	False	True	False		
▼ Timer 5	TON_TIME		Non-retain	True	True	True	True		
PT	Time	T#0ms	Non-retain	True	True	True	False		
ET	Time	T#0ms	Non-retain	True	False	True	False		
IN	Bool	false	Non-retain	True	True	True	False		
Q	Bool	false	Non-retain	True	False	True	False		
▼ Timer 6	TON_TIME		Non-retain	True	True	True	True		
PT	Time	T#0ms	Non-retain	True	True	True	False		
ET	Time	T#0ms	Non-retain	True	False	True	False		

Name	Data type	Default value	Retain	Acces-sible from HMI/OPC UA/Web API	Wri ta-ble from HMI/OPC UA/Web API	Visible in HMI engi-neer-ing	Set-point	Super-vision	Comment
IN	Bool	false	Non-retain	True	True	True	False		
Q	Bool	false	Non-retain	True	False	True	False		
▼ Timer 7	TON_TIME		Non-retain	True	True	True	True		
PT	Time	T#0ms	Non-retain	True	True	True	False		
ET	Time	T#0ms	Non-retain	True	False	True	False		
IN	Bool	false	Non-retain	True	True	True	False		
Q	Bool	false	Non-retain	True	False	True	False		
Temp									
Constant									

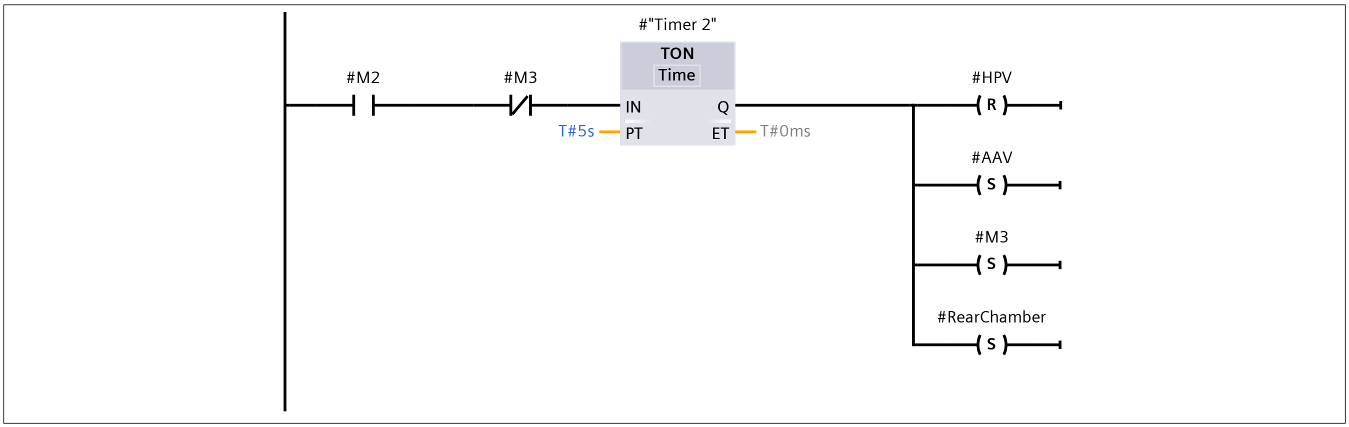
Network 1: Starting Leak Test



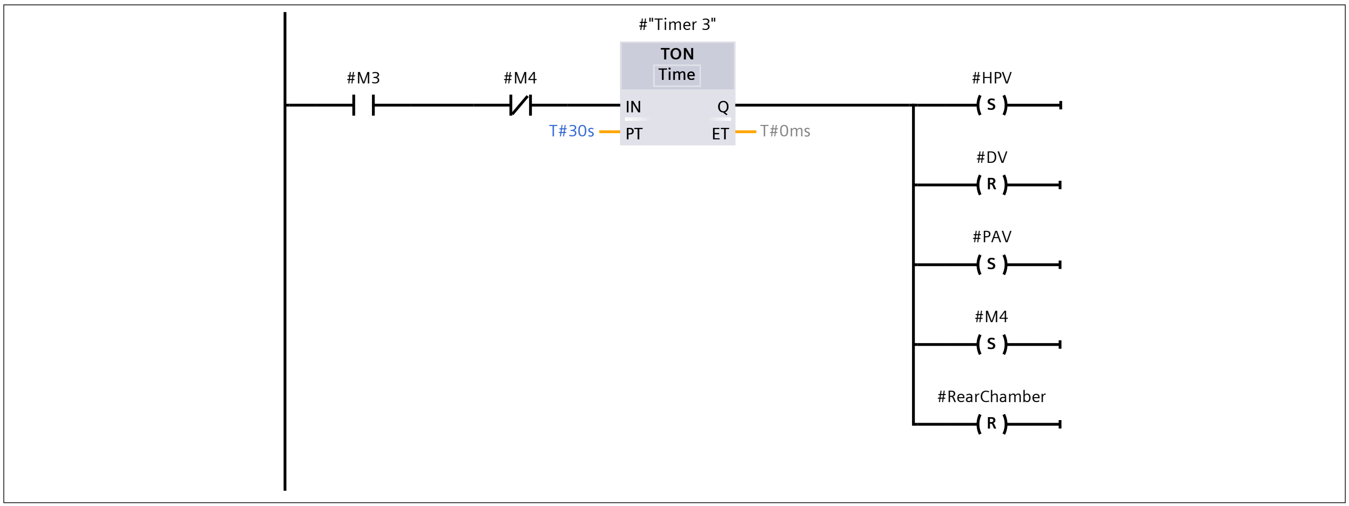
Network 2: Cylinder Under Pressure in Plus End-position



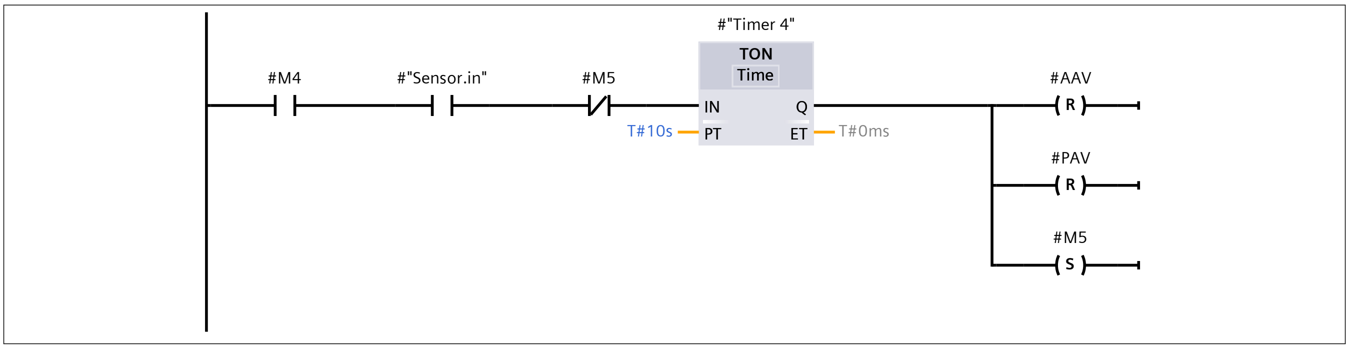
Network 3: Disconecting High Pressure Supply in Plus End-position



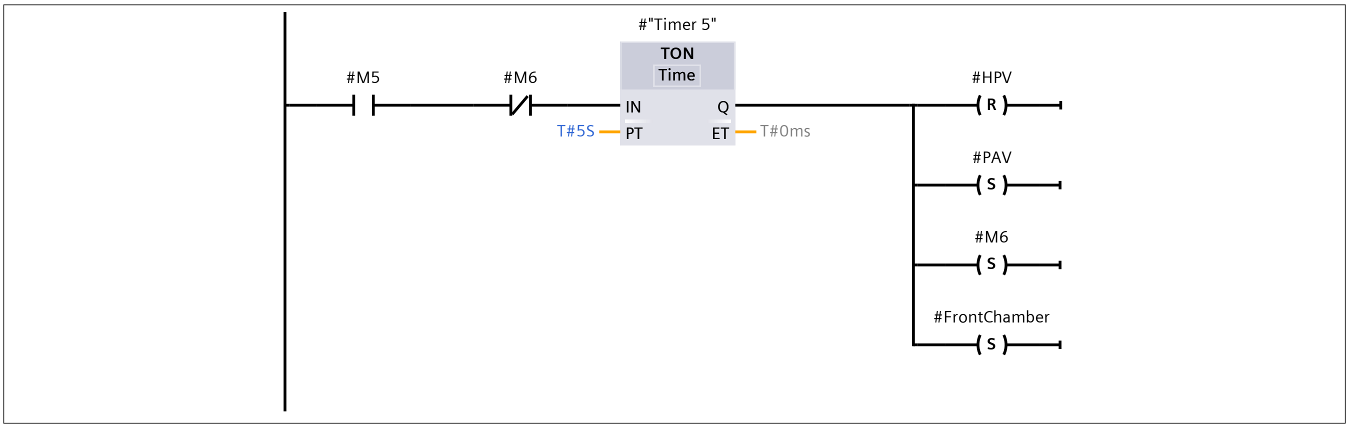
Network 4: Cylinder Under Leakage Test in Plus End-position



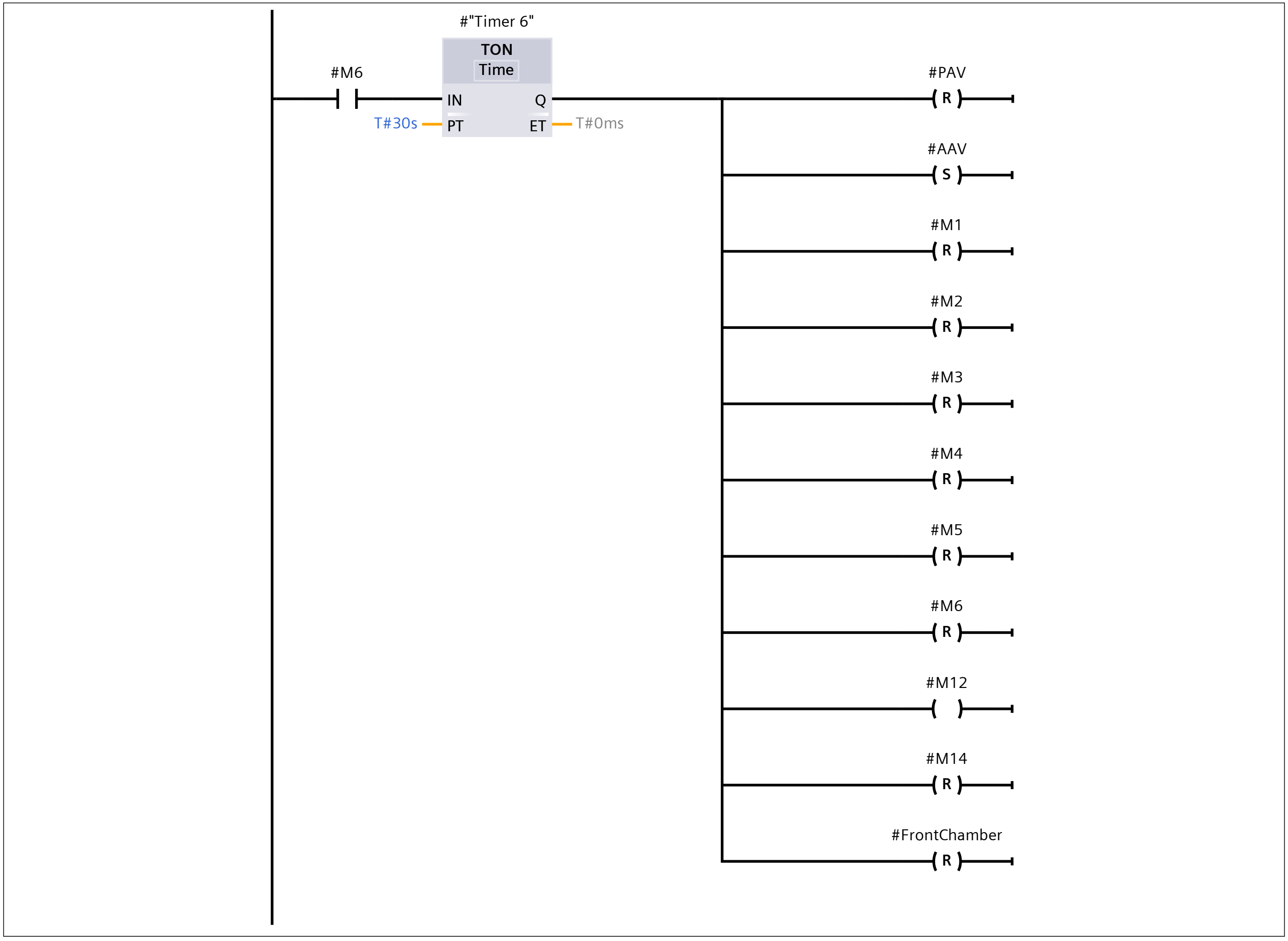
Network 5: Cylinder Under Pressure in Minus End-position



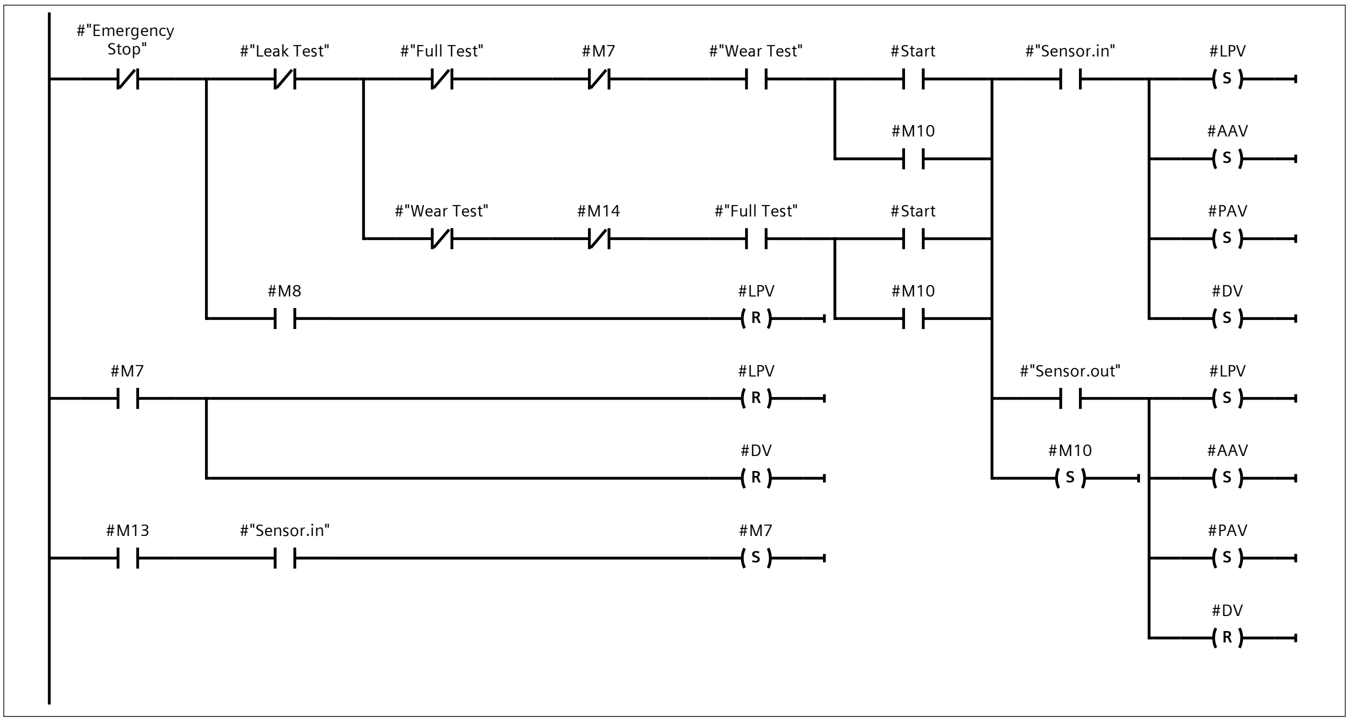
Network 6: Disconecting High Pressure Supply in Minus End-position



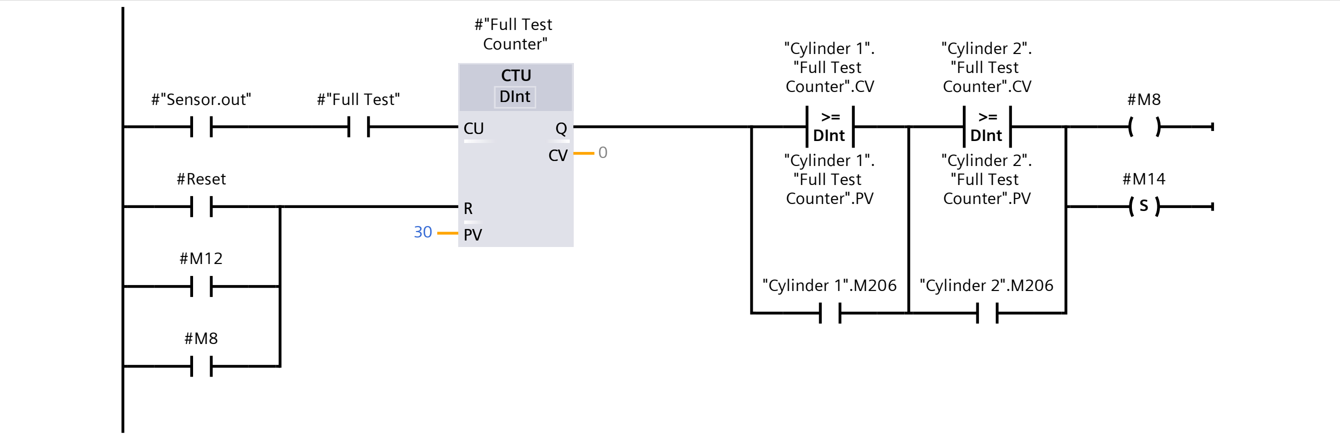
Network 7: Cylinder Under Leakage Test in Minus End-position



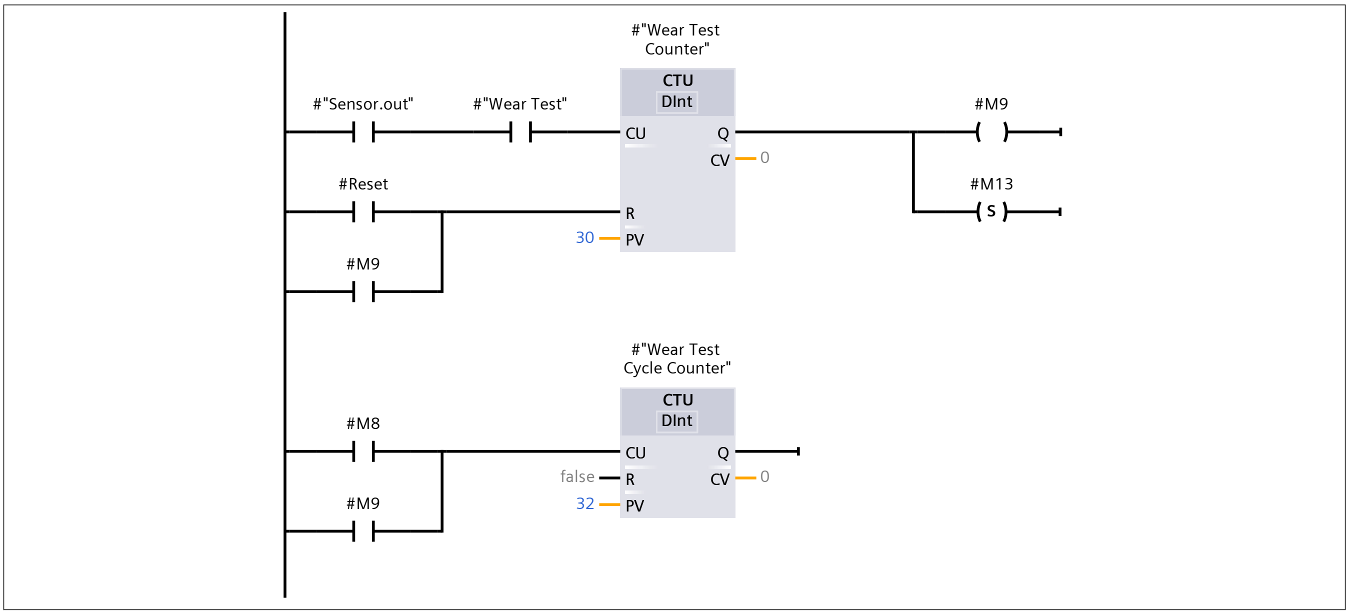
Network 8: Starting Wear / Full test



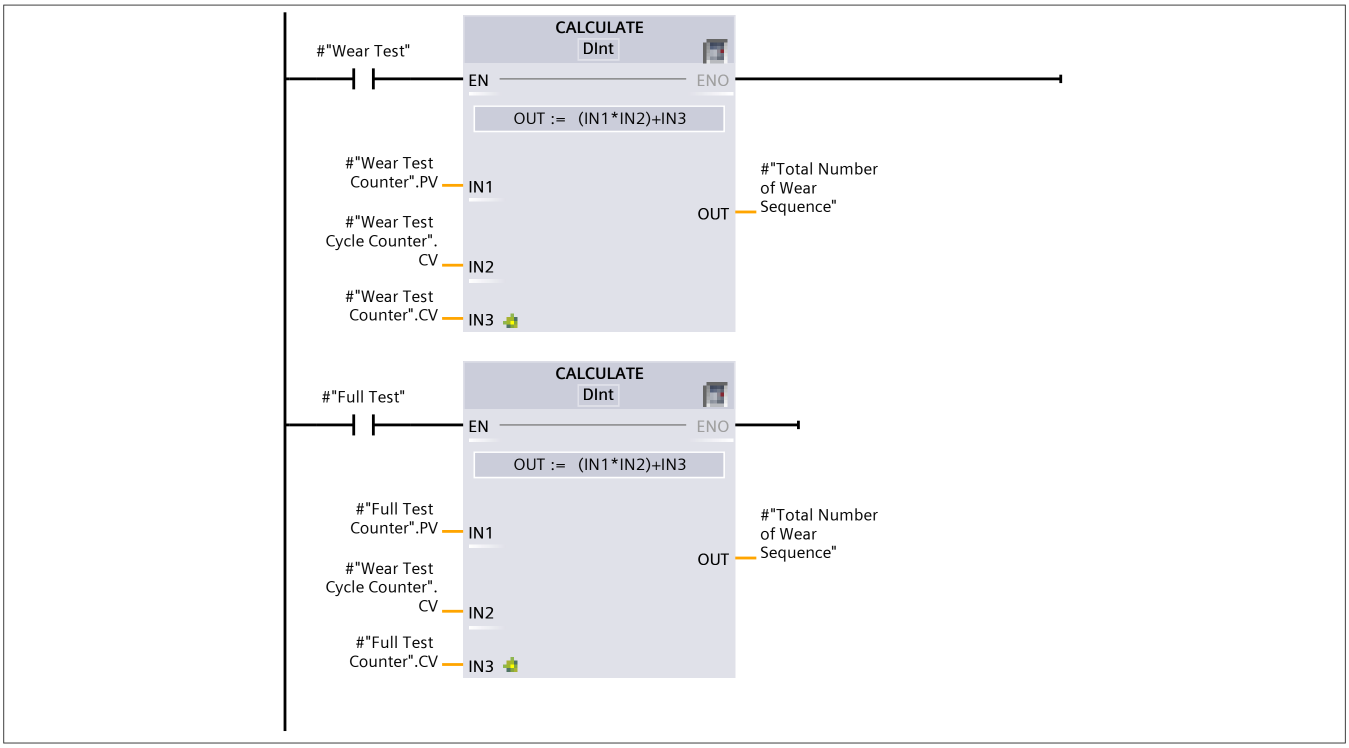
Network 9: Full Test Counter



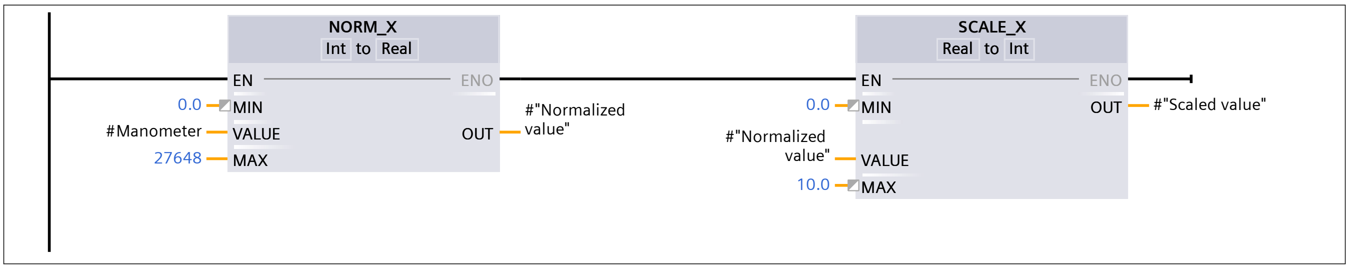
Network 10: Wear Test Counter



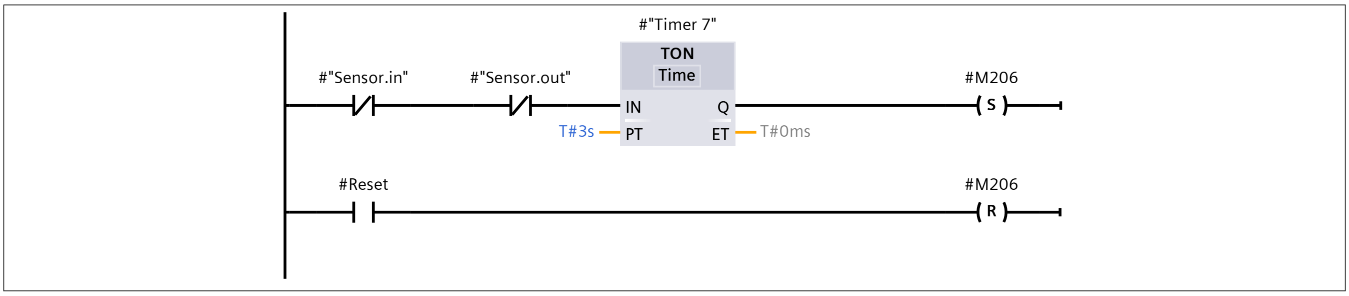
Network 11: Total Number of Wear Sequence



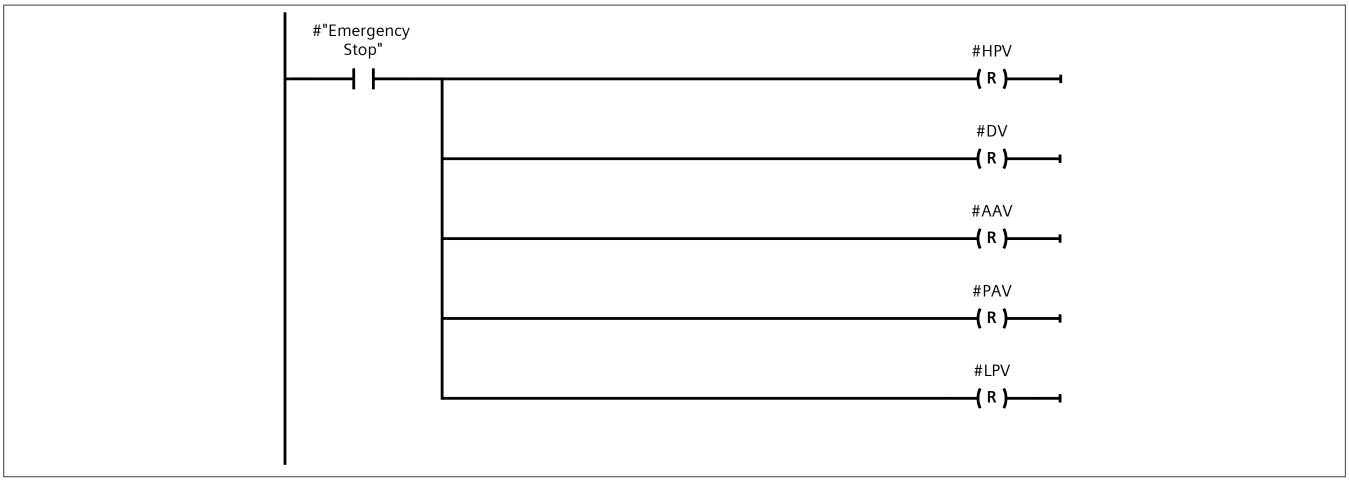
Network 12: Normalizing & Scaling Analog Pressure Signal



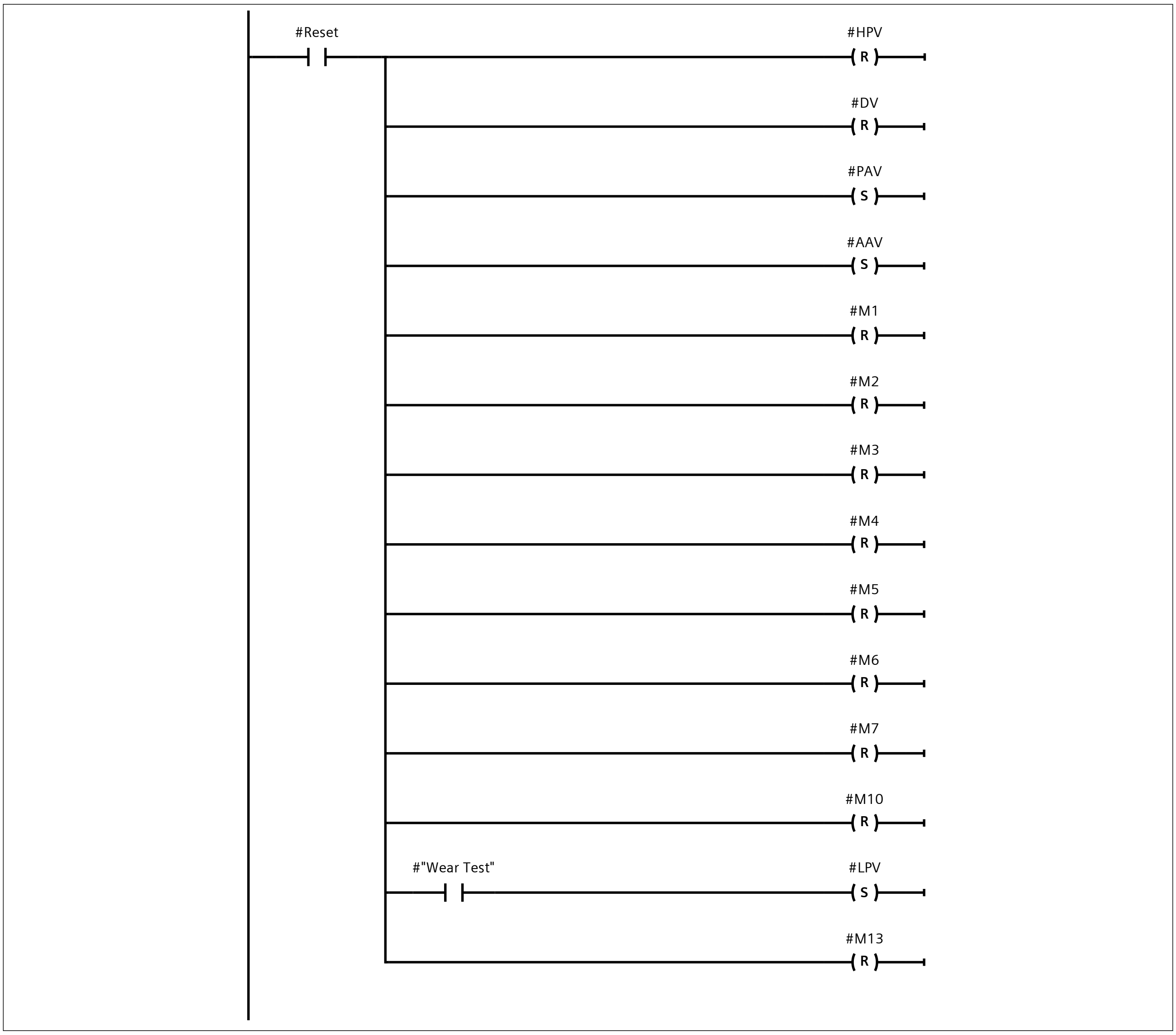
Network 13: Piston Movment Limitation Time



Network 14: Emergency Stop



Network 15: Reset



Trace Codes [FC1]

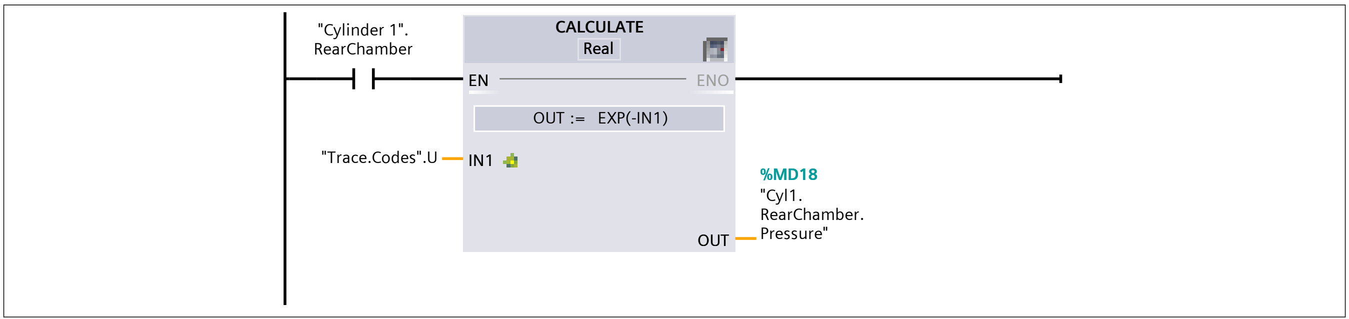
Trace Codes Properties					
General					
Name	Trace Codes	**Number**	1	**Type**	FC
Language	LAD	**Numbering**	Automatic		
Information					
Title	Simulation of pressure dropping	**Author**	Alireza.Qadiri	**Comment**	In this function the codes needed to simulate pressure dropping during the leak test is written. The value of the pressure will be recorded by "Trace Function".
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Default value	Comment
Input			
Output			
InOut			
Temp			
Constant			
▼ Return			
Trace Codes	Void		

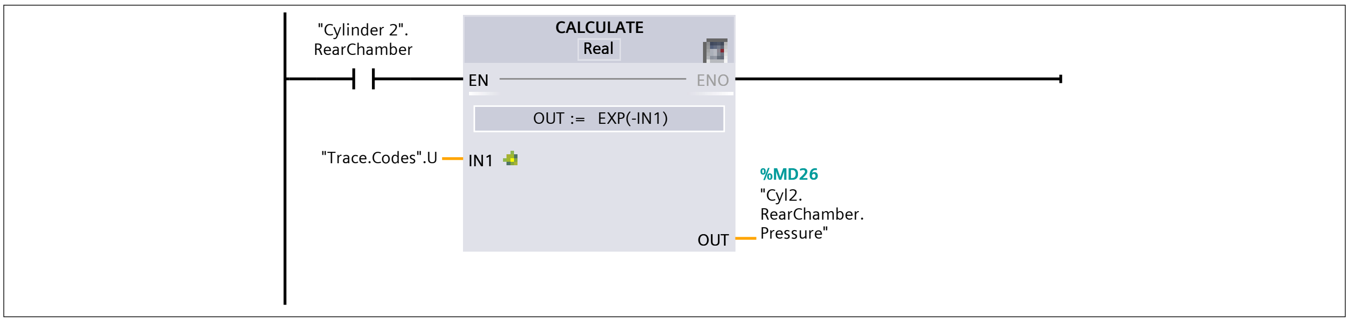
Network 1: Codes to Generate Pressure Drop Over Time

```
IF "M0" THEN
    // Statement section IF
    "Trace.Codes".U  += "Trace.Codes".V ;
ELSE
    "Cyl1.RearChamber.Pressure" := 0 ;
    "Cyl2.RearChamber.Pressure" := 0 ;
    "Cyl1.FrontChamber.Pressure" := 0 ;
    "Cyl2.FrontChamber.Pressure" := 0 ;
    "Trace.Codes".U := -2.308 ;
END_IF ;
```

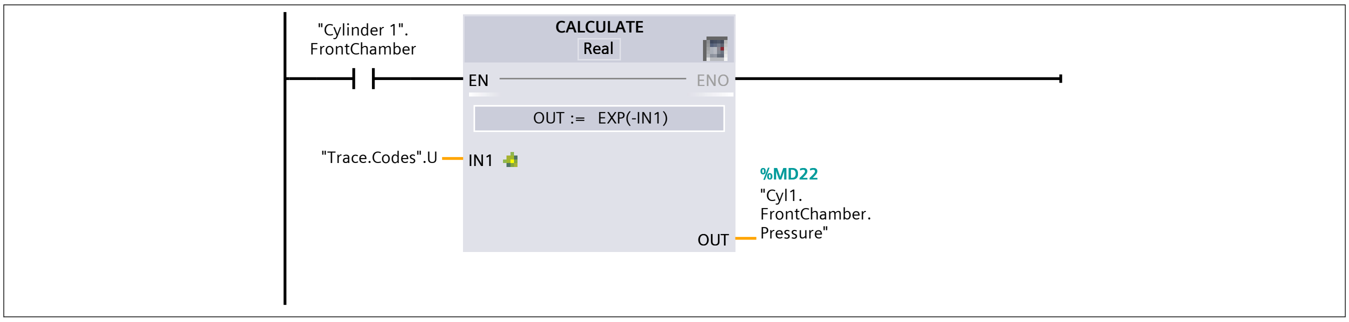
Network 2: Exponential Pressure Drop Simulation--Cyl1.RearChamber



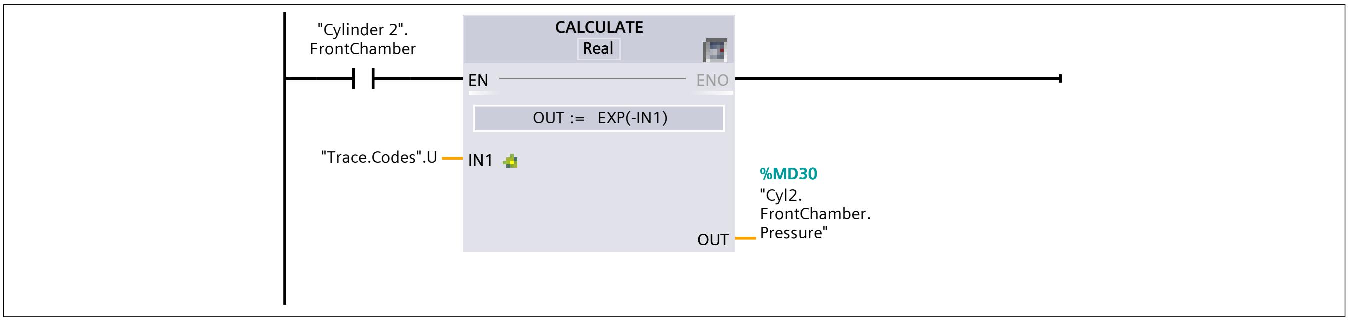
Network 3: Exponential Pressure Drop Simulation--Cyl2.RearChamber



Network 4: Exponential Pressure Drop Simulation--Cyl1.FrontChamber



Network 5: Exponential Pressure Drop Simulation--Cyl2.FrontChamber

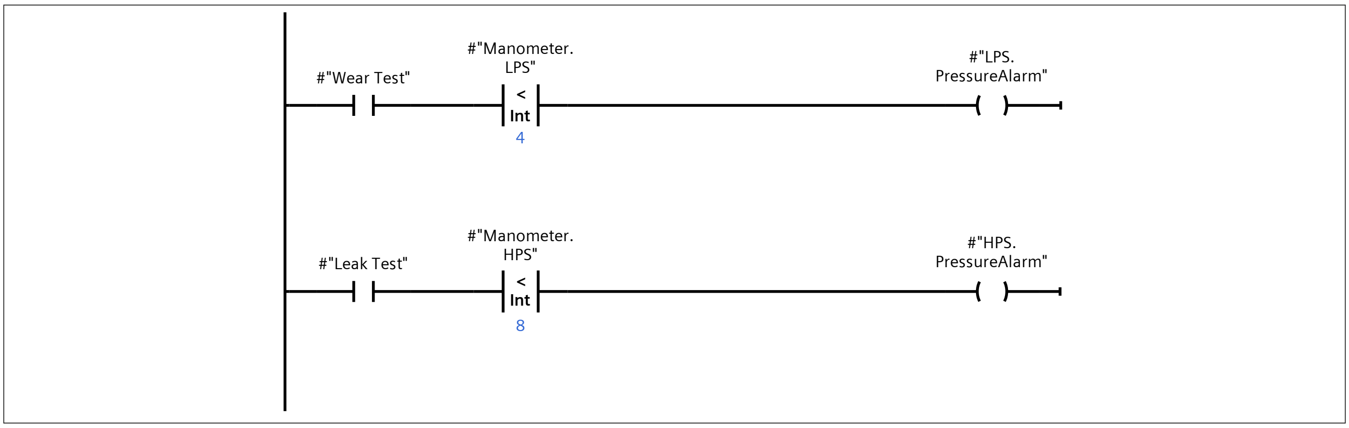


Alarm Codes [FC2]

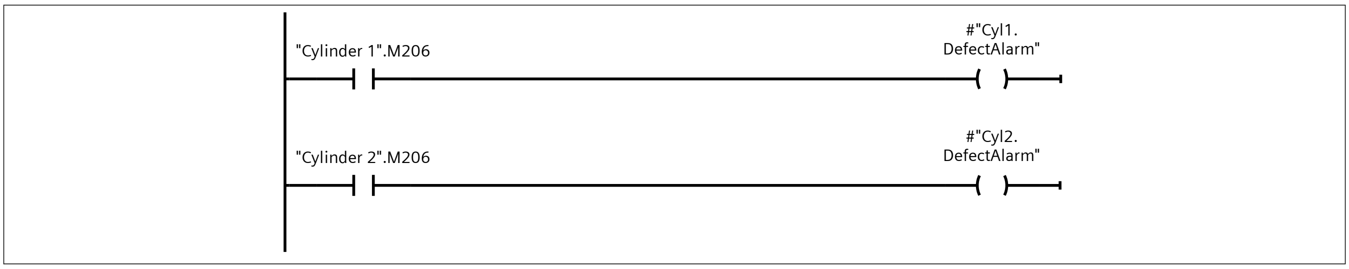
Alarm Codes Properties					
General					
Name	Alarm Codes	**Number**	2	**Type**	FC
Language	LAD	**Numbering**	Automatic		
Information					
Title	Alarms	**Author**	Alireza.Qadiri	**Comment**	The instruction related to the different errors and system failure is written in this function.
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Default value	Comment
Input			
Leak Test	Bool		
Wear Test	Bool		
Emergency.Stop	Bool		
Manometer.HPS	Int		
Manometer.LPS	Int		
Cyl1.RearChamber.Pressure	Real		
Cyl1.FrontChamber.Pressure	Real		
Cyl2.RearChamber.Pressure	Real		
Cyl2.FrontChamber.Pressure	Real		
Output			
LPS.PressureAlarm	Bool		
HPS.PressureAlarm	Bool		
Cyl1.DefectAlarm	Bool		
Cyl2.DefectAlarm	Bool		
Cyl1.RearCham.LeakageAlarm	Bool		
Cyl2.RearCham.LeakageAlarm	Bool		
Cyl1.FrontCham.LeakageAlarm	Bool		
Cyl2.FrontCham.LeakageAlarm	Bool		
Emergency.StopAlarm	Bool		
InOut			
Temp			
Constant			
Return			
Alarm Codes	Void		

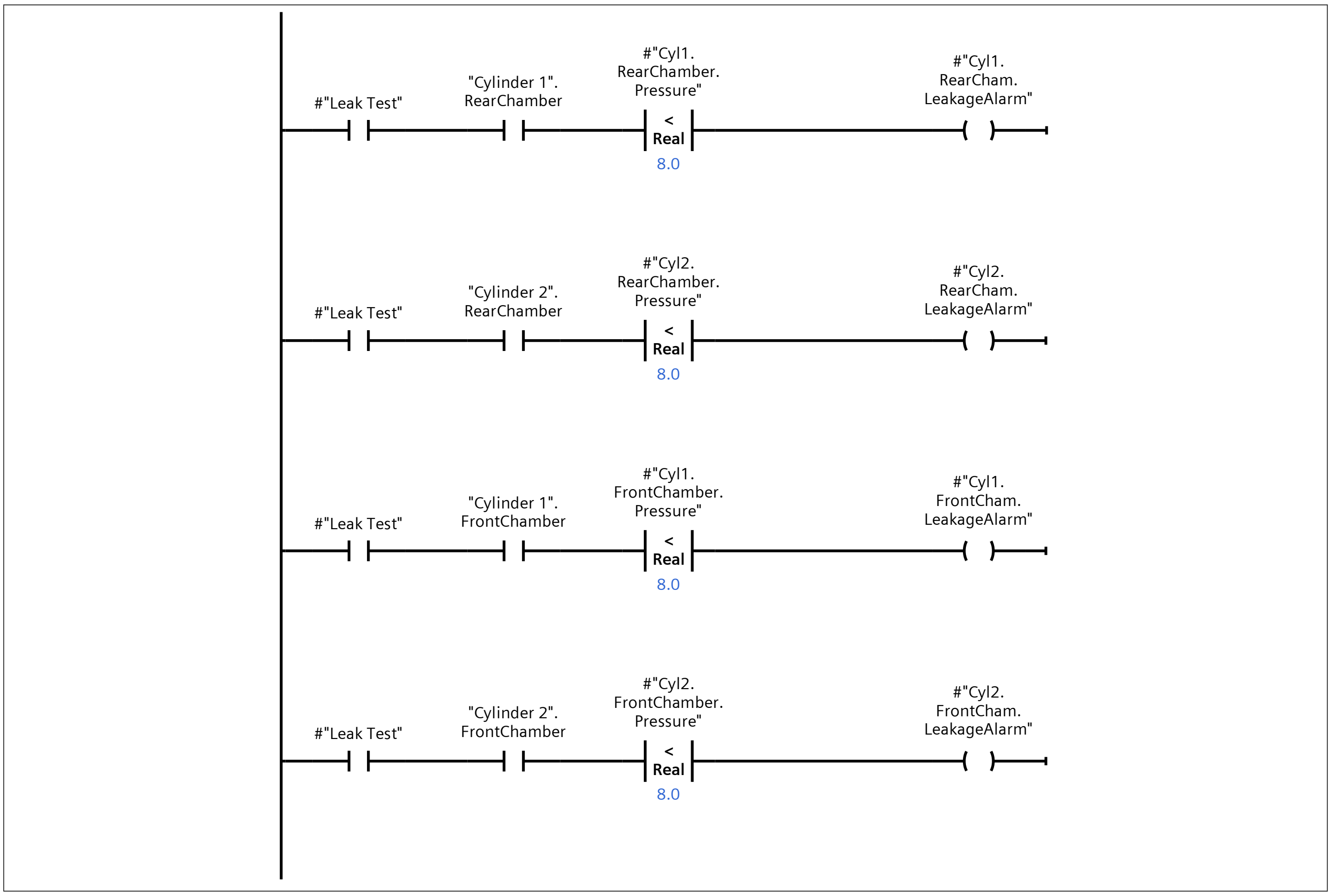
Network 1: Supply Pressure Alarms



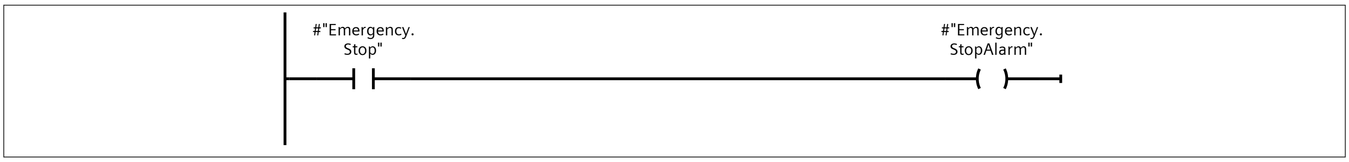
Network 2: Cylinder Defect Alarms



Network 3: Leak Test Pressure Drop Alarms



Network 4: Emergency Stop Alarm



Trace.Codes [DB1]

Trace.Codes Properties					
General					
Name	Trace.Codes	**Number**	1	**Type**	DB
Language	DB	**Numbering**	Automatic		
Information					
Title	Global Data Block	**Author**	Alireza.Qadiri	**Comment**	This global data block is ceated to provide static variables which is used in FC1.
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Static									
U	Real	-2.308	False	True	True	True	False		
V	Real	1.0E-05	False	True	True	True	False		

Cylinder 1 [DB8]

Cylinder 1 Properties					
General					
Name	Cylinder 1	**Number**	8	**Type**	DB
Language	DB	**Numbering**	Automatic		
Information					
Title	Cylinder 1	**Author**	Alireza.Qadiri	**Comment**	
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
Leak Test	Bool	false	False	True	True	True	False		
Wear Test	Bool	false	False	True	True	True	False		
Full Test	Bool	false	False	True	True	True	False		
Sensor.in	Bool	false	False	True	True	True	False		
Sensor.out	Bool	false	False	True	True	True	False		
Reset	Bool	false	False	True	True	True	False		
Start	Bool	false	False	True	True	True	False		
Emergency Stop	Bool	false	False	True	True	True	False		
Manometer	Int	0	False	True	True	True	False		
▼ Output									
HPV	Bool	false	False	True	True	True	False		
LPV	Bool	false	False	True	True	True	False		
AAV	Bool	false	False	True	True	True	False		
PAV	Bool	false	False	True	True	True	False		
DV	Bool	false	False	True	True	True	False		
InOut									
▼ Static									
M1	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
M2	Bool	false	False	True	True	True	False		
M3	Bool	false	False	True	True	True	False		
M4	Bool	false	False	True	True	True	False		
M5	Bool	false	False	True	True	True	False		
M6	Bool	false	False	True	True	True	False		
M7	Bool	false	False	True	True	True	False		
M8	Bool	false	False	True	True	True	False		
M9	Bool	false	False	True	True	True	False		
M10	Bool	false	False	True	True	True	False		
M11	Bool	false	False	True	True	True	False		
M12	Bool	false	False	True	True	True	False		
M13	Bool	false	False	True	True	True	False		
M14	Bool	false	False	True	True	True	False		
RearChamber	Bool	false	False	True	True	True	False		
FrontChamber	Bool	false	False	True	True	True	False		
M206	Bool	false	False	True	True	True	False		
Normalized value	Real	0.0	False	True	True	True	False		
Scaled value	Int	0	False	True	True	True	False		
▼ Full Test Counter	CTU_DINT		False	True	True	True	True		
CU	Bool	false	False	True	True	True	False		
CD	Bool	false	False	True	True	True	False		
R	Bool	false	False	True	True	True	False		
LD	Bool	false	False	True	True	True	False		
QU	Bool	false	False	True	True	True	False		
QD	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
PV	DInt	0	False	True	True	True	False		
CV	DInt	0	False	True	True	True	False		
▼ Wear Test Counter	CTU_DINT		False	True	True	True	True		
CU	Bool	false	False	True	True	True	False		
CD	Bool	false	False	True	True	True	False		
R	Bool	false	False	True	True	True	False		
LD	Bool	false	False	True	True	True	False		
QU	Bool	false	False	True	True	True	False		
QD	Bool	false	False	True	True	True	False		
PV	DInt	0	False	True	True	True	False		
CV	DInt	0	False	True	True	True	False		
▼ Wear Test Cycle Counter	CTU_DINT		True	True	True	True	True		
CU	Bool	false	True	True	True	True	False		
CD	Bool	false	True	True	True	True	False		
R	Bool	false	True	True	True	True	False		
LD	Bool	false	True	True	True	True	False		
QU	Bool	false	True	True	True	True	False		
QD	Bool	false	True	True	True	True	False		
PV	DInt	0	True	True	True	True	False		
CV	DInt	0	True	True	True	True	False		
Total Number of Wear Sequence	DInt	0	True	True	True	True	False		
▼ Timer 1	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
Q	Bool	false	False	True	False	True	False		
▼ Timer 2	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 3	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 4	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 5	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 6	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
Q	Bool	false	False	True	False	True	False		
▼ Timer 7	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

Cylinder 2 [DB9]

Cylinder 2 Properties					
General					
Name	Cylinder 2	**Number**	9	**Type**	DB
Language	DB	**Numbering**	Automatic		
Information					
Title	Cylinder 2	**Author**	Alireza.Qadiri	**Comment**	
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
Leak Test	Bool	false	False	True	True	True	False		
Wear Test	Bool	false	False	True	True	True	False		
Full Test	Bool	false	False	True	True	True	False		
Sensor.in	Bool	false	False	True	True	True	False		
Sensor.out	Bool	false	False	True	True	True	False		
Reset	Bool	false	False	True	True	True	False		
Start	Bool	false	False	True	True	True	False		
Emergency Stop	Bool	false	False	True	True	True	False		
Manometer	Int	0	False	True	True	True	False		
▼ Output									
HPV	Bool	false	False	True	True	True	False		
LPV	Bool	false	False	True	True	True	False		
AAV	Bool	false	False	True	True	True	False		
PAV	Bool	false	False	True	True	True	False		
DV	Bool	false	False	True	True	True	False		
InOut									
▼ Static									
M1	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
M2	Bool	false	False	True	True	True	False		
M3	Bool	false	False	True	True	True	False		
M4	Bool	false	False	True	True	True	False		
M5	Bool	false	False	True	True	True	False		
M6	Bool	false	False	True	True	True	False		
M7	Bool	false	False	True	True	True	False		
M8	Bool	false	False	True	True	True	False		
M9	Bool	false	False	True	True	True	False		
M10	Bool	false	False	True	True	True	False		
M11	Bool	false	False	True	True	True	False		
M12	Bool	false	False	True	True	True	False		
M13	Bool	false	False	True	True	True	False		
M14	Bool	false	False	True	True	True	False		
RearChamber	Bool	false	False	True	True	True	False		
FrontChamber	Bool	false	False	True	True	True	False		
M206	Bool	false	False	True	True	True	False		
Normalized value	Real	0.0	False	True	True	True	False		
Scaled value	Int	0	False	True	True	True	False		
▼ Full Test Counter	CTU_DINT		False	True	True	True	True		
CU	Bool	false	False	True	True	True	False		
CD	Bool	false	False	True	True	True	False		
R	Bool	false	False	True	True	True	False		
LD	Bool	false	False	True	True	True	False		
QU	Bool	false	False	True	True	True	False		
QD	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Acces-sible from HMI/OPC UA/Web API	Wri ta-ble fro m HMI/OPC UA/Web API	Visible in HMI engi-neer-ing	Set-point	Super-vision	Comment
PV	DInt	0	False	True	True	True	False		
CV	DInt	0	False	True	True	True	False		
▼ Wear Test Counter	CTU_DINT		False	True	True	True	True		
CU	Bool	false	False	True	True	True	False		
CD	Bool	false	False	True	True	True	False		
R	Bool	false	False	True	True	True	False		
LD	Bool	false	False	True	True	True	False		
QU	Bool	false	False	True	True	True	False		
QD	Bool	false	False	True	True	True	False		
PV	DInt	0	False	True	True	True	False		
CV	DInt	0	False	True	True	True	False		
▼ Wear Test Cycle Counter	CTU_DINT		True	True	True	True	True		
CU	Bool	false	True	True	True	True	False		
CD	Bool	false	True	True	True	True	False		
R	Bool	false	True	True	True	True	False		
LD	Bool	false	True	True	True	True	False		
QU	Bool	false	True	True	True	True	False		
QD	Bool	false	True	True	True	True	False		
PV	DInt	0	True	True	True	True	False		
CV	DInt	0	True	True	True	True	False		
Total Number of Wear Sequence	DInt	0	True	True	True	True	False		
▼ Timer 1	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
Q	Bool	false	False	True	False	True	False		
▼ Timer 2	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 3	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 4	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 5	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		
▼ Timer 6	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		

Name	Data type	Start value	Retain	Acces-sible from HMI/OPC UA/Web API	Writa-ble from HMI/OPC UA/Web API	Visible in HMI engi-neer-ing	Set-point	Super-vision	Comment
Q	Bool	false	False	True	False	True	False		
▼ Timer 7	TON_TIME		False	True	True	True	True		
PT	Time	T#0ms	False	True	True	True	False		
ET	Time	T#0ms	False	True	False	True	False		
IN	Bool	false	False	True	True	True	False		
Q	Bool	false	False	True	False	True	False		

Default tag table [108]

PLC tags						
Icon	Name	Data type	Address	Visible in HMI engineering	Accessible from HMI/OPC UA/Web API	Comment
	AAV	Bool	%Q0.2	True	True	
	AlwaysFALSE	Bool	%M1.3	True	True	
	AlwaysTRUE	Bool	%M1.2	True	True	
	Clock_0.5Hz	Bool	%M0.7	True	True	
	Clock_0.625Hz	Bool	%M0.6	True	True	
	Clock_1.25Hz	Bool	%M0.4	True	True	
	Clock_1Hz	Bool	%M0.5	True	True	
	Clock_2.5Hz	Bool	%M0.2	True	True	
	Clock_2Hz	Bool	%M0.3	True	True	
	Clock_5Hz	Bool	%M0.1	True	True	
	Clock_10Hz	Bool	%M0.0	True	True	
	Clock_Byte	Byte	%MB0	True	True	
	Cyl1.DefectAlarm	Bool	%Q1.0	True	True	
	Cyl1.FrontCham.LeakageAlarm	Bool	%Q1.4	True	True	
	Cyl1.FrontChamber.Pressure	Real	%MD22	True	True	
	Cyl1.RearCham.LeakageAlarm	Bool	%Q1.2	True	True	
	Cyl1.RearChamber.Pressure	Real	%MD18	True	True	
	Cyl2.DefectAlarm	Bool	%Q1.1	True	True	
	Cyl2.FrontCham.LeakageAlarm	Bool	%Q1.5	True	True	
	Cyl2.FrontChamber.Pressure	Real	%MD30	True	True	
	Cyl2.RearCham.LeakageAlarm	Bool	%Q1.3	True	True	
	Cyl2.RearChamber.Pressure	Real	%MD26	True	True	
	DiagStatusUpdate	Bool	%M1.1	True	True	
	DV 1	Bool	%Q0.1	True	True	
	DV 2	Bool	%Q0.5	True	True	
	Emergency.Stop	Bool	%I0.7	True	True	

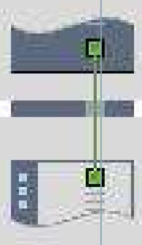
Icon	Name	Data type	Address	Visible in HMI engineering	Accessible from HMI/OPC UA/Web API	Comment
	Emergency.StopAlarm	Bool	%Q1.6	True	True	
	FirstScan	Bool	%M1.0	True	True	
	Full Test	Bool	%I0.6	True	True	
	Full Test Counter Reset	Bool	%M10.6	True	True	
	HPS.PressureAlarm	Bool	%Q0.7	True	True	
	HPV	Bool	%Q0.0	True	True	
	Leak Test	Bool	%I0.0	True	True	
	LPS.PressureAlarm	Bool	%Q0.6	True	True	
	LPV	Bool	%Q0.4	True	True	
	M0	Bool	%M1.4	True	True	
	M206	Bool	%M1.5	True	True	
	Manometer 1	Int	%IW2	True	True	
	Manometer 2	Int	%IW4	True	True	
	Manometer.HPS	Int	%IW8	True	True	
	Manometer.LPS	Int	%IW6	True	True	
	Normalized value 1	Real	%MD0	True	True	
	Normalized value 2	Real	%MD12	True	True	
	PAV	Bool	%Q0.3	True	True	
	Reset	Bool	%I1.1	True	True	
	Scaled value 1	Int	%MW1	True	True	
	Scaled value 2	Int	%MW6	True	True	
	Sensor.in 1	Bool	%I0.2	True	True	
	Sensor.in 2	Bool	%I1.4	True	True	
	Sensor.out 1	Bool	%I0.1	True	True	
	Sensor.out 2	Bool	%I1.3	True	True	
	Start	Bool	%I1.2	True	True	
	System_Byte	Byte	%MB1	True	True	
	Wear Test	Bool	%I0.3	True	True	
	Wear.Counting	Bool	%M8.0	True	True	

Local modules

PLC_2 [CPU 1515F-2 PN]

PLC_2			
General\Project information			
Name	PLC_2	**Author**	Alireza.Qadiri
Comment		**Rack**	0
Slot	1		
General\Catalog information			
Short designation	CPU 1515F-2 PN	**Description**	Fail-safe CPU with display; work memory 750 KB code and 3 MB data; can be used for safety applications; supports consistent safety upload; supports PROFIsafe V2; 30 ns bit operation time; 5-stage protection concept, technology functions: motion control, closed-loop control, counting and measuring; tracing; Runtime options; isochronous mode (central); for all PROFINET interfaces: transport protocol TCP/IP, secure Open User Communication, S7 communication, S7 routing, IP forwarding, Web server, DNS client, OPC UA: Server DA, Client DA, methods, companion specifications; 1st interface: PROFINET IO controller, supports RT/IRT, performance upgrade PROFINET V2.3, 2 ports, I-Device, MRP, MRPD, isochronous mode; 2nd interface: PROFINET IO controller, supports RT, I-Device; firmware V2.8
Article number	6ES7 515-2FM02-0AB0	**Firmware version**	V2.8
General\Identification & Maintenance			
Plant designation		**Location identifier**	
Installation date	2020-12-12 12:56:51.087	**Additional information**	
General\Checksums			
Text lists	FA 70 E8 75 1D 5A 8E 29	**Software**	Not available (compile necessary)
Fail-safe\F-activation			
F-capability activated	0		
Fail-safe\F-parameters			
Central F-source address	1	**Default F-monitoring time for central F-I/O**	150ms
Fail-safe\F-parameters\F-destination address range for PROFIsafe address type 1			
Low limit for F-destination addresses	1	**High limit for F-destination addresses**	99
PROFINET interface [X1]\General			
Name	PROFINET interface_1	**Author**	hamgha0625
Comment			
PROFINET interface [X1]\F-parameters			
Default F-monitoring time for F-I/O of this interface	150ms		
PROFINET interface [X1]\Ethernet addresses\Interface networked with			
Subnet:	PN/IE_1		
PROFINET interface [X1]\Ethernet addresses\IP protocol			
IP configuration	IP address is set directly at the device		

PROFINET interface [X1]\Ethernet addresses\PROFINET			
PROFINET device name is set directly at the device	False	**Generate PROFINET device name automatically**	True
PROFINET device name:	plc_2.profinet interface_1	**Converted name:**	plcxb2.profinetxainterfacexb1fc88
Device number:	0		
PROFINET interface [X1]\Time-of-day synchronization\NTP mode			
Note	Time synchronization for all PROFINET interfaces take place within the settings for time synchronization of the PROFINET interface [X1].	**Enable time synchronization via NTP server**	False
	IP addresses	**Server 1**	0.0.0.0
Server 2	0.0.0.0	**Server 3**	0.0.0.0
Server 4	0.0.0.0	**Update interval**	10s
PROFINET interface [X1]\Operating mode			
IO controller	True	**IO system**	
Device number	0	**IO device**	False
PROFINET interface [X1]\Advanced options\Interface options			
Call the user program if communication errors occur	False	**Support device replacement without exchangeable medium**	True
Permit overwriting of device names of all assigned IO devices	False	**Limit data infeed into the network**	True
Use IEC V2.2 LLDP mode	False	**Keep-Alive connection monitoring:**	30s
PROFINET interface [X1]\Advanced options\Media redundancy			
MRP domain	mrpdomain-1	**Media redundancy role:**	Not device in the ring
PROFINET interface [X1]\Advanced options\Real time settings\IO communication			
Send clock:	1.000ms		
PROFINET interface [X1]\Advanced options\Real time settings\Synchronization			
Sync domain:	Sync-Domain_1	**Synchronization role:**	Unsynchronized
RT class:	RT,IRT		
PROFINET interface [X1]\Advanced options\Real time settings\Real time options			
Calculated bandwidth for cyclic IO data:	0.000ms	**Calculated bandwidth for cyclic IO data:**	0.000%
PROFINET interface [X1]\Advanced options\Port [X1 P1 R]\General			
Name	Port_1	**Author**	hamgha0625
Comment			
PROFINET interface [X1]\Advanced options\Port [X1 P1 R]\Port interconnection\Local port:			
Local port:	PLC_2\PROFINET interface_1 [X1]\Port_1 [X1 P1 R]	**Medium:**	Copper
Cable name:	---		



PROFINET interface [X1]\Advanced options\Port [X1 P1 R]\Port interconnection\Partner port:			
	Monitoring of partner port is not possible	**Alternative partners**	False
Partner port:	Any partner		

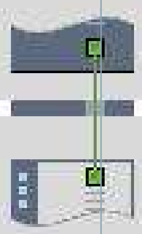
PROFINET interface [X1]\Advanced options\Port [X1 P1 R]\Port options\Activate			
Activate this port for use	True		

PROFINET interface [X1]\Advanced options\Port [X1 P1 R]\Port options\Connection			
Transmission rate / duplex:	Automatic	**Monitor**	False
Enable autonegotiation	True		

PROFINET interface [X1]\Advanced options\Port [X1 P1 R]\Port options\Boundaries			
End of detection of accessible devices	False	**End of topology discovery**	False
End of the sync domain	False		

PROFINET interface [X1]\Advanced options\Port [X1 P2 R]\General			
Name	Port_2	**Author**	hamgha0625
Comment			

PROFINET interface [X1]\Advanced options\Port [X1 P2 R]\Port interconnection\Local port:			
Local port:	PLC_2\PROFINET interface_1 [X1]\Port_2 [X1 P2 R]	**Medium:**	Copper
Cable name:	---		



PROFINET interface [X1]\Advanced options\Port [X1 P2 R]\Port interconnection\Partner port:			
	Monitoring of partner port is not possible	**Alternative partners**	False
Partner port:	Any partner		

PROFINET interface [X1]\Advanced options\Port [X1 P2 R]\Port options\Activate			
Activate this port for use	True		

PROFINET interface [X1]\Advanced options\Port [X1 P2 R]\Port options\Connection			
Transmission rate / duplex:	Automatic	**Monitor**	False
Enable autonegotiation	True		

PROFINET interface [X1]\Advanced options\Port [X1 P2 R]\Port options\Boundaries			
End of detection of accessible devices	False	**End of topology discovery**	False
End of the sync domain	False		

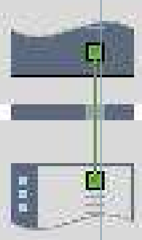
PROFINET interface [X1]\Web server access			
Note	The Web server must also be activated in the properties of the PLC.	**Enable Web server via IP address of this interface**	True

PROFINET interface [X2]\General			
Name	PROFINET interface_2	**Author**	hamgha0625
Comment			

PROFINET interface [X2]\F-parameters			
Default F-monitoring time for F-I/O of this interface	150ms		

PROFINET interface [X2]\Ethernet addresses\Interface networked with			
Subnet:	Not connected		

PROFINET interface [X2]\Ethernet addresses\IP protocol			
IP configuration	IP address is set directly at the device		
PROFINET interface [X2]\Ethernet addresses\PROFINET			
PROFINET device name is set directly at the device	False	**Generate PROFINET device name automatically**	True
PROFINET device name:	plc_2.profinet interface_2	**Converted name:**	plcxb2.profinetxainterfacexb2fdc8
Device number:	0		
PROFINET interface [X2]\Time-of-day synchronization\NTP mode			
Note	Time synchronization for all PROFINET interfaces take place within the settings for time synchronization of the PROFINET interface [X1].	**Enable time synchronization via NTP server**	False
	IP addresses	**Server 1**	0.0.0.0
Server 2	0.0.0.0	**Server 3**	0.0.0.0
Server 4	0.0.0.0	**Update interval**	10s
PROFINET interface [X2]\Operating mode			
IO controller	True	**IO system**	
Device number	0	**IO device**	False
PROFINET interface [X2]\Advanced options\Interface options			
Call the user program if communication errors occur	False	**Support device replacement without exchangeable medium**	True
Permit overwriting of device names of all assigned IO devices	False	**Limit data infeed into the network**	False
Use IEC V2.2 LLDP mode	False	**Keep-Alive connection monitoring:**	30s
PROFINET interface [X2]\Advanced options\Real time settings\IO communication			
Send clock:	1.000ms		
PROFINET interface [X2]\Advanced options\Real time settings\Real time options			
Calculated bandwidth for cyclic IO data:	0.000ms	**Calculated bandwidth for cyclic IO data:**	0.000%
PROFINET interface [X2]\Advanced options\Port [X2 P1]\General			
Name	Port_1	**Author**	hamgha0625
Comment			
PROFINET interface [X2]\Advanced options\Port [X2 P1]\Port interconnection\Local port:			
Local port:	PLC_2\PROFINET interface_2 [X2]\Port_1 [X2 P1]	**Medium:**	Copper
Cable name:	---		



PROFINET interface [X2]\Advanced options\Port [X2 P1]\Port interconnection\Partner port:			
	Monitoring of partner port is not possible	**Alternative partners**	False
Partner port:	Any partner		
PROFINET interface [X2]\Advanced options\Port [X2 P1]\Port options\Activate			
Activate this port for use	True		

PROFINET interface [X2]\Advanced options\Port [X2 P1]\Port options\Connection			
Transmission rate / duplex:	Automatic	**Monitor**	False
Enable autonegotiation	True		
PROFINET interface [X2]\Advanced options\Port [X2 P1]\Port options\Boundaries			
End of detection of accessible devices	False	**End of topology discovery**	False
End of the sync domain	False		
PROFINET interface [X2]\Web server access			
Note	The Web server must also be activated in the properties of the PLC.	**Enable Web server via IP address of this interface**	False
Startup			
Startup after POWER ON	Warm restart - Operating mode before POWER OFF	**Comparison preset to actual configuration**	Startup CPU even if mismatch
Configuration time	60000ms		
Cycle			
Maximum cycle time	150ms		
Enable minimum cycle time for cyclic OBs	True	**Minimum cycle time**	1ms
Communication load			
Cycle load due to communication	50%		
System and clock memory\System memory bits			
Enable the use of system memory byte	False	**Address of system memory byte (MBx)**	1
First cycle		**Diagnostic status changed**	
Always 1 (high)		**Always 0 (low)**	
System and clock memory\Clock memory bits			
Enable the use of clock memory byte	False	**Address of clock memory byte (MBx)**	0
10 Hz clock		**5 Hz clock**	
2.5 Hz clock		**2 Hz clock**	
1.25 Hz clock		**1 Hz clock**	
0.625 Hz clock		**0.5 Hz clock**	
SIMATIC Memory Card\Diagnostics			
Aging of the SIMATIC memory card	False	**Threshold value**	80%
System diagnostics\General			
Activate system diagnostics for this device	True	**Report network faults as maintenance instead of fault**	False
PLC alarms\General			
Central alarm management in the PLC	True		
Web server\General			
Activate web server on this module	True	**Permit access only with HTTPS**	False
Web server\Automatic update			
Enable automatic update	True	**Update interval**	10s
Web server\User management			

User name	User rights
Everybody	

Web server\User-defined web pages					
Application name	**HTML source path**	**Default HTML page**	**Files with dynamic content**	**Web DB number**	**Fragment DB number**
		index.htm	.htm;.html	333	334

Web server\Overview of interfaces		
Device	**Interface**	**Enabled web server access**
PLC_2	PROFINET interface_1	True
PLC_2	PROFINET interface_2	False

Display\General\Display standby mode			
Time to standby mode	30 minutes		
Display\General\Energy saving mode			
Time to energy saving mode	15 minutes		
Display\General\Display language			
Default language on display	English		
Display\Automatic update			
Time to update	5 seconds		
Display\Password\Display protection			
Enable write access	True	**Enable display protection**	True
Password	••••••••	**Confirm password**	••••••••
Time until automatic logoff	15 minutes		
Display\User-defined logo			
User logo activated	False	**Adapt logo**	False
Resolution	240x260	**Company logo**	---

User interface languages	
Assign project language	**User interface languages**
English (United States)	German
English (United States)	English
English (United States)	French
English (United States)	Spanish
English (United States)	Italian
English (United States)	Japanese
English (United States)	Chinese (simplified)
English (United States)	Korean
English (United States)	Russian
English (United States)	Turkish
English (United States)	Portuguese (Brazil)

Time of day\Local time			
Time zone	(UTC) Dublin, Edinburgh, Lisbon, London		
Time of day\Daylight saving time			
Activate daylight saving time	True	**Difference between standard and daylight saving time**	60mins
Time of day\Daylight saving time\Start of daylight saving time			
Selection of the week	Last	**Selection of the weekday**	Sunday
of	March	**at**	01:00 a.m.
Time of day\Daylight saving time\Start of standard time			
Selection of the week	Last	**Selection of the weekday**	Sunday
of	October	**at**	02:00 a.m.

Protection			
Level of protection	Full access (no protection)	**Password**	••••••••
Confirm password	••••••••		

Protection\Connection mechanisms	
Permit access with PUT/GET communication from remote partner	False

Protection\Security event			
Summarize security events in case of high message volume	True	**Length of an interval**	20
Unit	seconds		

OPC UA\Accessibility of the server	
Activate OPC UA server	False

System power supply\General	
General	Connection to supply voltage L+

System power supply\Power segment overview

Module	Slot	Supply/consumption
PLC_2	1	12.00W
DI 16x24VDC HF_1	2	-1.10W
DQ 16x24VDC/0.5A HF_1	3	-1.10W
AI 4xU/I/RTD/TC ST_1	4	-0.70W
	Summary	9.10W

Advanced configuration\DNS configuration	
No DNS server address is configured.	

Advanced configuration\IP Forwarding\Configuration IPv4 Forwarding	
Enable IPv4 forwarding for interfaces of this PLC	False

Advanced configuration\Configuration control\Configuration control for central configuration	
Allow reconfiguration of device via the user program	False

Connection resources\

	Station resources - Reserved - Maximum	Station resources - Reserved - Configured	Station resources - Dynamic - Configured	Module resources - PLC_2 [CPU 1515F-2 PN] - Configured
Maximum number of resources:		10	98	108
	Maximum	Configured	Configured	Configured
PG communication:	4	-	-	-
HMI communication:	4	0	0	0
S7 communication:	0	-	0	0
Open user communication:	0	-	0	0
Web communication:	2	-	-	-
OPC UA client/server communication:	0	-	-	-
Other communication:	-	-	0	0
Total resources used:		0	0	0
Available resources:		10	98	108

Overview of addresses\Overview of addresses\Overview of addresses			
Inputs	True	**Outputs**	True
Address gaps	False	**Slot**	True

Type	I	**Addr. from**	0	**Addr. to**	1	**Module**	DI 16x24VDC HF_1
PIP	Automatic update	**OB**	-	**Device name**	PLC_2 [CPU 1515F-2 PN]	**Device number**	-
Size	2 Bytes	**Master / IO system**	-	**Rack**	0	**Slot**	2
Type	O	**Addr. from**	0	**Addr. to**	1	**Module**	DQ 16x24VDC/ 0.5A HF_1
PIP	Automatic update	**OB**	-	**Device name**	PLC_2 [CPU 1515F-2 PN]	**Device number**	-
Size	2 Bytes	**Master / IO system**	-	**Rack**	0	**Slot**	3
Type	I	**Addr. from**	2	**Addr. to**	9	**Module**	AI 4xU/I/RTD/TC ST_1
PIP	Automatic update	**OB**	-	**Device name**	PLC_2 [CPU 1515F-2 PN]	**Device number**	-
Size	8 Bytes	**Master / IO system**	-	**Rack**	0	**Slot**	4

Runtime licenses\OPC UA\Runtime licenses			
Type of required license	None	**Type of purchased license**	No license
Runtime licenses\ProDiag\Supervisions			
Number of used supervisions	0		
Runtime licenses\ProDiag\Runtime licenses			
Number of required licenses	None (<= 25 supervisions)	**Used ProDiag licenses**	No license
Runtime licenses\Energy Suite\Energy objects			
Number of configured energy objects	0		
Runtime licenses\Energy Suite\Runtime licenses			
Total number of licensed energy objects	0		
Runtime licenses\Energy Suite\Runtime licenses\Number of purchased licenses			
License type '5 energy objects'	No license	**License type '10 energy objects'**	No license

Local modules

DI 16x24VDC HF_1

DI 16x24VDC HF_1			
General\Project information			
Name	DI 16x24VDC HF_1	**Author**	Alireza.Qadiri
Comment		**Rack**	0
Slot	2		
General\Catalog information			
Short designation	DI 16x24VDC HF	**Description**	Digital input module DI16 x DC24V; grouping 16; input delay 0.05..20ms; input type 3 (IEC 61131); configurable diagnostics; hardware interrupts; value status; integrated counter for channel 0 and 1; isochronous mode
Article number	6ES7 521-1BH00-0AB0	**Firmware version**	V2.2
General\Identification & Maintenance			
Plant designation		**Location identifier**	
Installation date	2020-12-15 12:07:43.631	**Additional information**	
Module parameters\General\Startup			
Comparison preset to actual module	From CPU		
Module parameters\Channel template\Inputs\Apply to all channels that use the template\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Module parameters\Channel template\Inputs\Apply to all channels that use the template\Input parameters			
Input delay	3.2ms		
Module parameters\DI Configuration\Configuration of submodules			
Module distribution	None		
Module parameters\DI Configuration\Value status (Quality Information)			
Value status	False		
Module parameters\DI Configuration\Copy of module for Shared Device (MSI)			
Copy of module:	None		
Module parameters\DI Configuration\Counter configuration\Counter configuration on channel 0 and channel 1 enabled			
Counter configuration on channel 0 and channel 1 enabled	False		
Input 0 - 15\General			
Name	DI 16x24VDC HF_1	**Comment**	
Input 0 - 15\Inputs\General\Module failure			
Input values with module failure	Input value 0		
Input 0 - 15\Inputs\Channel 0-7\Channel 0			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 0\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 0-7\Channel 0\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 0\Counter configuration			
Reaction to violation of a counting limit		**Edge selection**	
Set output DQ		**High counting limit**	
Low counting limit		**Start value**	
Comparison value			
Input 0 - 15\Inputs\Channel 0-7\Channel 0\Hardware interrupts			
Comparison event for DQ has occurred	0	**CountRidPrefixEvent**	49248

Event name:		**Hardware interrupt**	0
Rising edge0	Count0	**Channel number**	0
CountHwEventType	5		
Input 0 - 15\Inputs\Channel 0-7\Channel 0\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49152
Event name:		**Hardware interrupt**	0
Rising edge0	Rising edge0	**Channel number**	0
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 0\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49280
Event name:		**Hardware interrupt**	0
Falling edge0	Falling edge0	**Channel number**	0
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 0-7\Channel 1			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 1\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 0-7\Channel 1\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 1\Counter configuration			
Reaction to violation of a counting limit		**Edge selection**	
Set output DQ		**High counting limit**	
Low counting limit		**Start value**	
Comparison value			
Input 0 - 15\Inputs\Channel 0-7\Channel 1\Hardware interrupts			
Comparison event for DQ has occurred	0	**CountRidPrefixEvent**	49249
Event name:		**Hardware interrupt**	0
Rising edge1	Count1	**Channel number**	1
CountHwEventType	5		
Input 0 - 15\Inputs\Channel 0-7\Channel 1\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49153
Event name:		**Hardware interrupt**	0
Rising edge1	Rising edge1	**Channel number**	1
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 1\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49281
Event name:		**Hardware interrupt**	0
Falling edge1	Falling edge1	**Channel number**	1
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 0-7\Channel 2			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 2\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 0-7\Channel 2\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 2\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49154
Event name:		**Hardware interrupt**	0

Rising edge2	Rising edge2	**Channel number**	2
HwEventTypeRisin-gEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 2\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdg-eEvent**	49282
Event name:		**Hardware interrupt**	0
Falling edge2	Falling edge2	**Channel number**	2
HwEventTypeFallin-gEdge	2		
Input 0 - 15\Inputs\Channel 0-7\Channel 3			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 3\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 0-7\Channel 3\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 3\Hardware interrupts			
Enable rising edge de-tection	0	**RidPrefixRisingEdgeE-vent**	49155
Event name:		**Hardware interrupt**	0
Rising edge3	Rising edge3	**Channel number**	3
HwEventTypeRisin-gEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 3\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdg-eEvent**	49283
Event name:		**Hardware interrupt**	0
Falling edge3	Falling edge3	**Channel number**	3
HwEventTypeFallin-gEdge	2		
Input 0 - 15\Inputs\Channel 0-7\Channel 4			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 4\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 0-7\Channel 4\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 4\Hardware interrupts			
Enable rising edge de-tection	0	**RidPrefixRisingEdgeE-vent**	49156
Event name:		**Hardware interrupt**	0
Rising edge4	Rising edge4	**Channel number**	4
HwEventTypeRisin-gEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 4\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdg-eEvent**	49284
Event name:		**Hardware interrupt**	0
Falling edge4	Falling edge4	**Channel number**	4
HwEventTypeFallin-gEdge	2		
Input 0 - 15\Inputs\Channel 0-7\Channel 5			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 5\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 0-7\Channel 5\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 5\Hardware interrupts			
Enable rising edge de-tection	0	**RidPrefixRisingEdgeE-vent**	49157

Event name:		Hardware interrupt	0
Rising edge5	Rising edge5	Channel number	5
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 5\Hardware interrupts			
Enable falling edge detection	0	RidPrefixFallingEdgeEvent	49285
Event name:		Hardware interrupt	0
Falling edge5	Falling edge5	Channel number	5
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 0-7\Channel 6			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 6\Diagnostics			
No supply voltage L+	False	Wire break	False
Input 0 - 15\Inputs\Channel 0-7\Channel 6\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 6\Hardware interrupts			
Enable rising edge detection	0	RidPrefixRisingEdgeEvent	49158
Event name:		Hardware interrupt	0
Rising edge6	Rising edge6	Channel number	6
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 6\Hardware interrupts			
Enable falling edge detection	0	RidPrefixFallingEdgeEvent	49286
Event name:		Hardware interrupt	0
Falling edge6	Falling edge6	Channel number	6
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 0-7\Channel 7			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 0-7\Channel 7\Diagnostics			
No supply voltage L+	False	Wire break	False
Input 0 - 15\Inputs\Channel 0-7\Channel 7\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 0-7\Channel 7\Hardware interrupts			
Enable rising edge detection	0	RidPrefixRisingEdgeEvent	49159
Event name:		Hardware interrupt	0
Rising edge7	Rising edge7	Channel number	7
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 0-7\Channel 7\Hardware interrupts			
Enable falling edge detection	0	RidPrefixFallingEdgeEvent	49287
Event name:		Hardware interrupt	0
Falling edge7	Falling edge7	Channel number	7
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 8			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 8-15\Channel 8\Diagnostics			
No supply voltage L+	False	Wire break	False
Input 0 - 15\Inputs\Channel 8-15\Channel 8\Input parameters			
Input delay	3.2ms		

Input 0 - 15\Inputs\Channel 8-15\Channel 8\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49160
Event name:		**Hardware interrupt**	0
Rising edge8	Rising edge8	**Channel number**	8
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 8\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49288
Event name:		**Hardware interrupt**	0
Falling edge8	Falling edge8	**Channel number**	8
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 9			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 8-15\Channel 9\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 8-15\Channel 9\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 8-15\Channel 9\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49161
Event name:		**Hardware interrupt**	0
Rising edge9	Rising edge9	**Channel number**	9
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 9\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49289
Event name:		**Hardware interrupt**	0
Falling edge9	Falling edge9	**Channel number**	9
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 10			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 8-15\Channel 10\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 8-15\Channel 10\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 8-15\Channel 10\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49162
Event name:		**Hardware interrupt**	0
Rising edge10	Rising edge10	**Channel number**	10
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 10\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49290
Event name:		**Hardware interrupt**	0
Falling edge10	Falling edge10	**Channel number**	10
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 11			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 8-15\Channel 11\Diagnostics			
No supply voltage L+	False	**Wire break**	False

Input 0 - 15\Inputs\Channel 8-15\Channel 11\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 8-15\Channel 11\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49163
Event name:		**Hardware interrupt**	0
Rising edge11	Rising edge11	**Channel number**	11
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 11\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49291
Event name:		**Hardware interrupt**	0
Falling edge11	Falling edge11	**Channel number**	11
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 12			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 8-15\Channel 12\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 8-15\Channel 12\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 8-15\Channel 12\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49164
Event name:		**Hardware interrupt**	0
Rising edge12	Rising edge12	**Channel number**	12
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 12\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49292
Event name:		**Hardware interrupt**	0
Falling edge12	Falling edge12	**Channel number**	12
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 13			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 8-15\Channel 13\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 8-15\Channel 13\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 8-15\Channel 13\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49165
Event name:		**Hardware interrupt**	0
Rising edge13	Rising edge13	**Channel number**	13
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 13\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49293
Event name:		**Hardware interrupt**	0
Falling edge13	Falling edge13	**Channel number**	13
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 14			
Parameter settings	From template		

Input 0 - 15\Inputs\Channel 8-15\Channel 14\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 8-15\Channel 14\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 8-15\Channel 14\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49166
Event name:		**Hardware interrupt**	0
Rising edge14	Rising edge14	**Channel number**	14
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 14\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49294
Event name:		**Hardware interrupt**	0
Falling edge14	Falling edge14	**Channel number**	14
HwEventTypeFallingEdge	2		
Input 0 - 15\Inputs\Channel 8-15\Channel 15			
Parameter settings	From template		
Input 0 - 15\Inputs\Channel 8-15\Channel 15\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Input 0 - 15\Inputs\Channel 8-15\Channel 15\Input parameters			
Input delay	3.2ms		
Input 0 - 15\Inputs\Channel 8-15\Channel 15\Hardware interrupts			
Enable rising edge detection	0	**RidPrefixRisingEdgeEvent**	49167
Event name:		**Hardware interrupt**	0
Rising edge15	Rising edge15	**Channel number**	15
HwEventTypeRisingEdge	1		
Input 0 - 15\Inputs\Channel 8-15\Channel 15\Hardware interrupts			
Enable falling edge detection	0	**RidPrefixFallingEdgeEvent**	49295
Event name:		**Hardware interrupt**	0
Falling edge15	Falling edge15	**Channel number**	15
HwEventTypeFallingEdge	2		
Input 0 - 15\I/O addresses\Input addresses			
Start address	0.0	**End address**	1.7
Isochronous mode	False	**Organization block**	0
Process image	0		

Local modules

DQ 16x24VDC/0.5A HF_1

DQ 16x24VDC/0.5A HF_1			
General\Project information			
Name	DQ 16x24VDC/0.5A HF_1	**Author**	Alireza.Qadiri
Comment		**Rack**	0
Slot	3		
General\Catalog information			
Short designation	DQ 16x24VDC/0.5A HF	**Description**	Digital output module DQ16 x DC24V / 0,5A; grouping 8; 4A per group; configurable diagnostics; configurable substitute value for output; isochronous mode; switching cycle counter
Article number	6ES7 522-1BH01-0AB0	**Firmware version**	V1.1
General\Identification & Maintenance			
Plant designation		**Location identifier**	
Installation date	2020-12-15 12:08:13.527	**Additional information**	
Module parameters\General\Startup			
Comparison preset to actual module	From CPU		
Module parameters\Channel template\Outputs\Apply to all channels that use the template\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Short circuit to ground	False	**Maintenance switching cycles**	False
Module parameters\Channel template\Outputs\Apply to all channels that use the template\Output parameters			
Reaction to CPU STOP	Shutdown	**Switching cycle counter**	False
Switching cycle limit			
Module parameters\DQ configuration\Configuration of submodules			
Module distribution	None		
Module parameters\DQ configuration\Value status (Quality Information)			
Value status	False		
Module parameters\DQ configuration\Copy of module for shared device (MSO)			
Copy of module:	None		
Output 0 - 15\General			
Name	DQ 16x24VDC/0.5A HF_1	**Comment**	
Output 0 - 15\Outputs\Channel 0 - 7\Channel 0			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 0\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Short circuit to ground	False	**Maintenance switching cycles**	False
Output 0 - 15\Outputs\Channel 0 - 7\Channel 0\Output parameters			
Reaction to CPU STOP	Shutdown	**Switching cycle counter**	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 0 - 7\Channel 1			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 1\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Short circuit to ground	False	**Maintenance switching cycles**	False
Output 0 - 15\Outputs\Channel 0 - 7\Channel 1\Output parameters			
Reaction to CPU STOP	Shutdown	**Switching cycle counter**	False

Switching cycle limit			
Output 0 - 15\Outputs\Channel 0 - 7\Channel 2			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 2\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 0 - 7\Channel 2\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 0 - 7\Channel 3			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 3\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 0 - 7\Channel 3\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 0 - 7\Channel 4			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 4\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 0 - 7\Channel 4\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 0 - 7\Channel 5			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 5\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 0 - 7\Channel 5\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 0 - 7\Channel 6			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 6\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 0 - 7\Channel 6\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 0 - 7\Channel 7			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 0 - 7\Channel 7\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False

Output 0 - 15\Outputs\Channel 0 - 7\Channel 7\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 8			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 8\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 8\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 9			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 9\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 9\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 10			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 10\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 10\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 11			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 11\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 11\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 12			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 12\Diagnostics			
No supply voltage L+	False	Wire break	False
Short circuit to ground	False	Maintenance switching cycles	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 12\Output parameters			
Reaction to CPU STOP	Shutdown	Switching cycle counter	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 13			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 13\Diagnostics			
No supply voltage L+	False	Wire break	False

Short circuit to ground	False	**Maintenance switching cycles**	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 13\Output parameters			
Reaction to CPU STOP	Shutdown	**Switching cycle counter**	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 14			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 14\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Short circuit to ground	False	**Maintenance switching cycles**	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 14\Output parameters			
Reaction to CPU STOP	Shutdown	**Switching cycle counter**	False
Switching cycle limit			
Output 0 - 15\Outputs\Channel 8 - 15\Channel 15			
Parameter settings	From template		
Output 0 - 15\Outputs\Channel 8 - 15\Channel 15\Diagnostics			
No supply voltage L+	False	**Wire break**	False
Short circuit to ground	False	**Maintenance switching cycles**	False
Output 0 - 15\Outputs\Channel 8 - 15\Channel 15\Output parameters			
Reaction to CPU STOP	Shutdown	**Switching cycle counter**	False
Switching cycle limit			
Output 0 - 15\I/O addresses\Output addresses			
Start address	0.0	**End address**	1.7
Isochronous mode	False	**Organization block**	0
Process image	0		

Local modules

AI 4xU/I/RTD/TC ST_1

AI 4xU/I/RTD/TC ST_1			
General\Project information			
Name	AI 4xU/I/RTD/TC ST_1	**Author**	Alireza.Qadiri
Comment		**Rack**	0
Slot	4		
General\Catalog information			
Short designation	AI 4xU/I/RTD/TC ST	**Description**	Analog input module AI4 x U/I/RTD/TC 16-bit; grouping 4; 2 channels with RTD measurement; common mode voltage 10 V; configurable diagnostics; hardware interrupts
Article number	6ES7 531-7QD00-0AB0	**Firmware version**	V1.0
General\Identification & Maintenance			
Plant designation		**Location identifier**	
Installation date	2021-01-08 22:42:10.113	**Additional information**	
Module parameters\General\Startup			
Comparison preset to actual module	From CPU		
Module parameters\Channel template\Inputs\Apply to all channels that use the template\Diagnostics			
No supply voltage L+	False	**Overflow**	False
Underflow	False	**Common mode error**	False
Reference junction	False	**Wire break**	False
Current limit for wire break diagnostics			
Module parameters\Channel template\Inputs\Apply to all channels that use the template\Measuring			
Measurement type	Voltage	**Measuring range**	+/- 10V
Temperature coefficient		**Temperature unit**	
Reference junction		**Fixed reference temperature**	
Interference frequency suppression	50Hz	**Smoothing**	None
Module parameters\AI configuration\Configuration of submodules			
Module distribution	None		
Module parameters\AI configuration\Value status (Quality Information)			
Value status	False		
Module parameters\AI configuration\Copy of module for Shared Device (MSI)			
Copy of module:	None		
Input 0 - 3\General			
Name	AI 4xU/I/RTD/TC ST_1	**Comment**	
Input 0 - 3\Inputs\Channel 0			
Parameter settings	From template		
Input 0 - 3\Inputs\Channel 0\Diagnostics			
No supply voltage L+	False	**Overflow**	False
Underflow	False	**Common mode error**	False
Reference junction	False	**Wire break**	False
Current limit for wire break diagnostics			
Input 0 - 3\Inputs\Channel 0\Measuring			
Measurement type	Voltage	**Measuring range**	+/- 10V
Temperature coefficient		**Temperature unit**	

Reference junction		**Fixed reference temperature**	
Interference frequency suppression	50Hz	**Smoothing**	None
Input 0 - 3\Inputs\Channel 0\Hardware interrupts			
High limit 1		**Low limit 1**	
High limit 2		**Low limit 2**	
Input 0 - 3\Inputs\Channel 0\Hardware interrupts			
Hardware interrupt high limit 1	0	**RidPrefixFallingEdgeEvent**	49272
Event name:		**Hardware interrupt:**	0
UpperLimitOne0	UpperLimitOne0	**Channel number**	0
HwEventTypeLimit1Overrun	4		
Input 0 - 3\Inputs\Channel 0\Hardware interrupts			
Hardware interrupt low limit 1	0	**RidPrefixFallingEdgeEvent**	49288
Event name:		**Hardware interrupt:**	0
LowerLimitOne0	LowerLimitOne0	**Channel number**	0
HwEventTypeLimit1Underrun	3		
Input 0 - 3\Inputs\Channel 0\Hardware interrupts			
Hardware interrupt high limit 2	0	**RidPrefixFallingEdgeEvent**	49264
Event name:		**Hardware interrupt:**	0
UpperLimitTwo0	UpperLimitTwo0	**Channel number**	0
HwEventTypeLimit2Overrun	6		
Input 0 - 3\Inputs\Channel 0\Hardware interrupts			
Hardware interrupt low limit 2	0	**RidPrefixFallingEdgeEvent**	49280
Event name:		**Hardware interrupt:**	0
LowerLimitTwo0	LowerLimitTwo0	**Channel number**	0
HwEventTypeLimit2Underrun	5		
Input 0 - 3\Inputs\Channel 1			
Parameter settings	From template		
Input 0 - 3\Inputs\Channel 1\Diagnostics			
No supply voltage L+	False	**Overflow**	False
Underflow	False	**Common mode error**	False
Reference junction	False	**Wire break**	False
Current limit for wire break diagnostics			
Input 0 - 3\Inputs\Channel 1\Measuring			
Measurement type	Voltage	**Measuring range**	+/- 10V
Temperature coefficient		**Temperature unit**	
Reference junction		**Fixed reference temperature**	
Interference frequency suppression	50Hz	**Smoothing**	None
Input 0 - 3\Inputs\Channel 1\Hardware interrupts			
High limit 1		**Low limit 1**	
High limit 2		**Low limit 2**	
Input 0 - 3\Inputs\Channel 1\Hardware interrupts			
Hardware interrupt high limit 1	0	**RidPrefixFallingEdgeEvent**	49273
Event name:		**Hardware interrupt:**	0
UpperLimitOne1	UpperLimitOne1	**Channel number**	1

HwEventTypeLimit1Overrun	4		
Input 0 - 3\Inputs\Channel 1\Hardware interrupts			
Hardware interrupt low limit 1	0	**RidPrefixFallingEdgeEvent**	49289
Event name:		**Hardware interrupt:**	0
LowerLimitOne1	LowerLimitOne1	**Channel number**	1
HwEventTypeLimit1Underrun	3		
Input 0 - 3\Inputs\Channel 1\Hardware interrupts			
Hardware interrupt high limit 2	0	**RidPrefixFallingEdgeEvent**	49265
Event name:		**Hardware interrupt:**	0
UpperLimitTwo1	UpperLimitTwo1	**Channel number**	1
HwEventTypeLimit2Overrun	6		
Input 0 - 3\Inputs\Channel 1\Hardware interrupts			
Hardware interrupt low limit 2	0	**RidPrefixFallingEdgeEvent**	49281
Event name:		**Hardware interrupt:**	0
LowerLimitTwo1	LowerLimitTwo1	**Channel number**	1
HwEventTypeLimit2Underrun	5		
Input 0 - 3\Inputs\Channel 2			
Parameter settings	From template		
Input 0 - 3\Inputs\Channel 2\Diagnostics			
No supply voltage L+	False	**Overflow**	False
Underflow	False	**Common mode error**	False
Reference junction	False	**Wire break**	False
Current limit for wire break diagnostics			
Input 0 - 3\Inputs\Channel 2\Measuring			
Measurement type	Voltage	**Measuring range**	+/- 10V
Temperature coefficient		**Temperature unit**	
Reference junction		**Fixed reference temperature**	
Interference frequency suppression	50Hz	**Smoothing**	None
Input 0 - 3\Inputs\Channel 2\Hardware interrupts			
High limit 1		**Low limit 1**	
High limit 2		**Low limit 2**	
Input 0 - 3\Inputs\Channel 2\Hardware interrupts			
Hardware interrupt high limit 1	0	**RidPrefixFallingEdgeEvent**	49274
Event name:		**Hardware interrupt:**	0
UpperLimitOne2	UpperLimitOne2	**Channel number**	2
HwEventTypeLimit1Overrun	4		
Input 0 - 3\Inputs\Channel 2\Hardware interrupts			
Hardware interrupt low limit 1	0	**RidPrefixFallingEdgeEvent**	49290
Event name:		**Hardware interrupt:**	0
LowerLimitOne2	LowerLimitOne2	**Channel number**	2
HwEventTypeLimit1Underrun	3		
Input 0 - 3\Inputs\Channel 2\Hardware interrupts			
Hardware interrupt high limit 2	0	**RidPrefixFallingEdgeEvent**	49266
Event name:		**Hardware interrupt:**	0

UpperLimitTwo2	UpperLimitTwo2	**Channel number**	2
HwEventTypeLi-mit2Overrun	6		
Input 0 - 3\Inputs\Channel 2\Hardware interrupts			
Hardware interrupt low limit 2	0	**RidPrefixFallingEdg-eEvent**	49282
Event name:		**Hardware interrupt:**	0
LowerLimitTwo2	LowerLimitTwo2	**Channel number**	2
HwEventTypeLi-mit2Underrun	5		
Input 0 - 3\Inputs\Channel 3			
Parameter settings	From template		
Input 0 - 3\Inputs\Channel 3\Diagnostics			
No supply voltage L+	False	**Overflow**	False
Underflow	False	**Common mode error**	False
Reference junction	False	**Wire break**	False
Current limit for wire break diagnostics			
Input 0 - 3\Inputs\Channel 3\Measuring			
Measurement type	Voltage	**Measuring range**	+/- 10V
Temperature coeffi-cient		**Temperature unit**	
Reference junction		**Fixed reference tem-perature**	
Interference frequen-cy suppression	50Hz	**Smoothing**	None
Input 0 - 3\Inputs\Channel 3\Hardware interrupts			
High limit 1		**Low limit 1**	
High limit 2		**Low limit 2**	
Input 0 - 3\Inputs\Channel 3\Hardware interrupts			
Hardware interrupt high limit 1	0	**RidPrefixFallingEdg-eEvent**	49275
Event name:		**Hardware interrupt:**	0
UpperLimitOne3	UpperLimitOne3	**Channel number**	3
HwEventTypeLi-mit1Overrun	4		
Input 0 - 3\Inputs\Channel 3\Hardware interrupts			
Hardware interrupt low limit 1	0	**RidPrefixFallingEdg-eEvent**	49291
Event name:		**Hardware interrupt:**	0
LowerLimitOne3	LowerLimitOne3	**Channel number**	3
HwEventTypeLi-mit1Underrun	3		
Input 0 - 3\Inputs\Channel 3\Hardware interrupts			
Hardware interrupt high limit 2	0	**RidPrefixFallingEdg-eEvent**	49267
Event name:		**Hardware interrupt:**	0
UpperLimitTwo3	UpperLimitTwo3	**Channel number**	3
HwEventTypeLi-mit2Overrun	6		
Input 0 - 3\Inputs\Channel 3\Hardware interrupts			
Hardware interrupt low limit 2	0	**RidPrefixFallingEdg-eEvent**	49283
Event name:		**Hardware interrupt:**	0
LowerLimitTwo3	LowerLimitTwo3	**Channel number**	3
HwEventTypeLi-mit2Underrun	5		
Input 0 - 3\I/O addresses\Input addresses			
Start address	2	**End address**	9

Organization block	0	**Process image**	0