POLITECNICO DI TORINO
Master's Degree in Mechatronics Engineering

Master's Degree Thesis

Electro-pneumatic closed-loop leakage detecting system for pneumatic cylinders (Test bench)

Supervisor

Prof. Luigi Mazza

Candidate

Ibrahim Tahoun

June 2022.

Abstract

The massive spread of electro-pneumatic systems in all areas of industry has made it imperative for us to ensure reliability and the efficiency of all the components used in creating these systems, and one of the main components used is the linear actuator (double-acting cylinder) which can be affected by leakage problems during its life cycle due to worn or damaged seals, this requires an ongoing air leak detection and repair system.

This thesis proposes using "Automation Studio 7.0" to design, simulate, and animate our system. The main objective of this thesis is to automatically test these cylinders used in our system to detect the presence of the leak and its analysis so that it can be repaired before the cylinder breaks, and this is done using the RASPBERRY PI device as our controller and CODESYS software for programming it. This thesis also shows how to make a connection between the two software programs and how to use them for leak detection and analysis.[2]

Table of Contents

Table of figures

List of tables

1 Introduction

Pneumatic cylinders are one of the most common and important components in the industry world. They are powerful and quick linear actuators that can be controlled using different kinds of valves. The power is supplied by compressed air which can give a huge amount of power that can provide thousands of pounds of push and pull force. The structure of cylinders consists of many parts as shown in the photo below, but the most important part is the piston of the cylinder that performs most of the work by using the power of the compressed air. The basic working principle of these cylinders is that when the compressed air goes from the left side inlet of the cylinder, it pushes the piston to move to the other end of the cylinder. On the other end, there will be also another inlet for compressed air or spring to push the piston back again to its initial position and this process is called "Extension and Retraction". The role of valves is to regulate the flow of compressed air that can change the speed of the piston, so they are the controllers of the cylinders.[5]

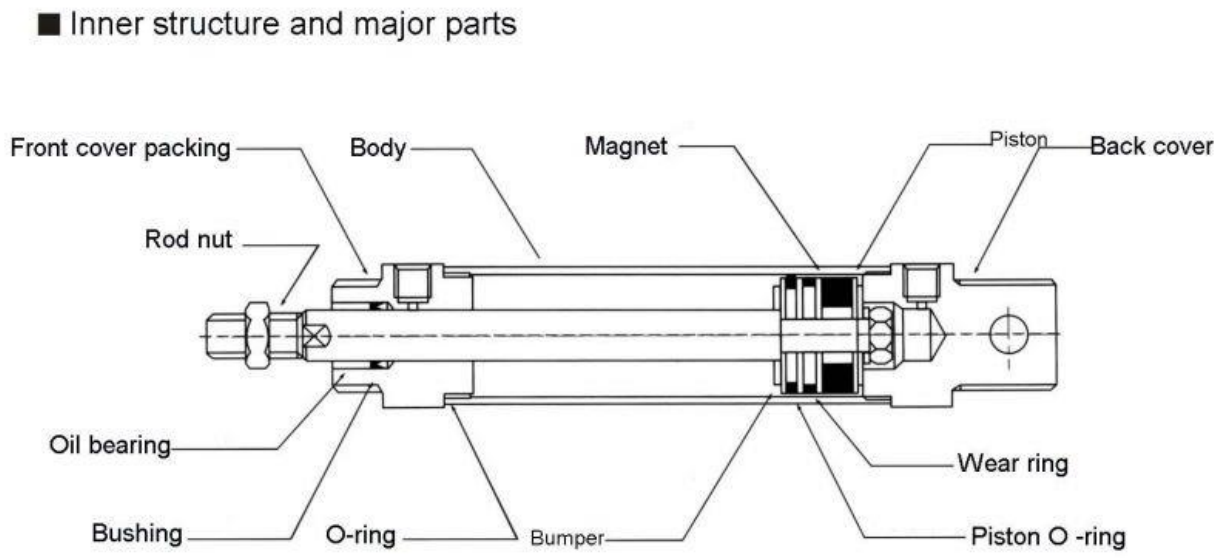

Figure 1: Inner structure and major parts of a pneumatic cylinder

Nowadays pneumatic cylinders are widely spread in most industrial fields, and the reason for that is because they are very simple to work, they can give high power and speed, they are strong enough to resist the harsh environments, they are efficient and reliable, and they can provide high performance with low cost. Due to these advantages of pneumatic cylinders, they are used in many industrial applications, for example in manufacturing industries, they are often used for taking things on or off conveyor belts, lifting heavy objects, and opening valves or doors. Also in the automotive industry, they are used in cars and trucks to be used for suspension and brakes.

While using these pneumatic cylinders, there are some mistakes that we must avoid, for example, mounting a load on the side of the piston rod and not on the middle can lead to a bending rod especially if we have a heavy load because the operation of the piston is extension and retraction (linear movement). Moreover, if we mounted a heavy load perpendicularly to the piston's rod axis, this also could lead to a bending rod, and the seals of the cylinder can be damaged, and cause leakage. However, the leakage can happen even if we are under the normal conditions, because after a certain period of operating time of the pneumatic cylinders, air leakage will occur, and that is because the internal seals of the cylinders can be worn due to the friction phenomenon, and at that time they must be substituted with new ones. Because of the disasters that could happen if the cylinders are suddenly stopped or broken, we must check the pressure of the cylinders periodically.

For the reasons discussed before, we designed and simulated a test bench that can work for any no. of cylinders, and this designed and simulated system allows the users to have a complete HMI (Human Machine Interface) system that allows the user to easily and automatically make a test for any cylinder or even all of them in parallel, they can check the values of the pressure reading of the cylinder live on a monitor, they can be warned up in case of any leakage detection, and also they can have a complete SCADA (Supervisory Control and Data Acquisition) system that can be used for monitoring and supervisory functions, and also to collect the data of the pressure readings and analyze them.

For our test bench to be working effectively and at the same time with a very cheap cost, we decided to use "Raspberry Pi" as a controller instead of using a PLC controller because, in our design and simulation, it can do the same functions exactly as PLC controller, but it is also much cheaper than it.

Raspberry Pi boards are specially designed low-cost computers that plug into a computer monitor or TV and use a standard keyboard and mouse, and it was designed in the beginning for educational purposes. It is a device that enables people of all ages to explore computing and learn how to program in languages like Scratch and Python. It can do almost everything the desktop computer can do like browsing the internet, making spreadsheets, and many other things.

Nowadays robust and affordable Raspberry Pi technology has been deployed in tens of thousands of applications in a variety of industries across the world like 3D printing, FPGA, Industrial automation, IoT, and many other applications, and that is because of the powerful performance that allows it to compute solutions to fit a wide range of applications. These reasons were a very powerful reason for me to use the Raspberry Pi as a controller for the project, and with the help of CODESYS software, we can program the Raspberry Pi with a programming language of PLC controllers which is the “Ladder” programing language.

2 AUTOMATION STUDIO 7.0

2.1 What is AUTOMATION STUDIO (AS)!?

- AS is a software created and developed by Famic Technologies company in 1986 as a training and teaching platform.
- It is a completely integrated software package that allows users to design, simulate, and animate circuits consisting of various automation technologies.
- It is a very special and unique software solution that offers intuitive design, animation, simulation, and system analysis features in a versatile and very user-friendly environment that can be used by teachers, students, and engineers.
- In addition to its standard pneumatic and Ladder Logic libraries that we mainly used in the thesis project, AS further supports the following technologies: Hydraulics, Pneumatics, Digital Electronics, and a lot of human-machine interfaces (HMI) features.
- Also, you must know that it isn't free software, so you must have a license from "Famic Technologies" company to be able to use it.[1][2]

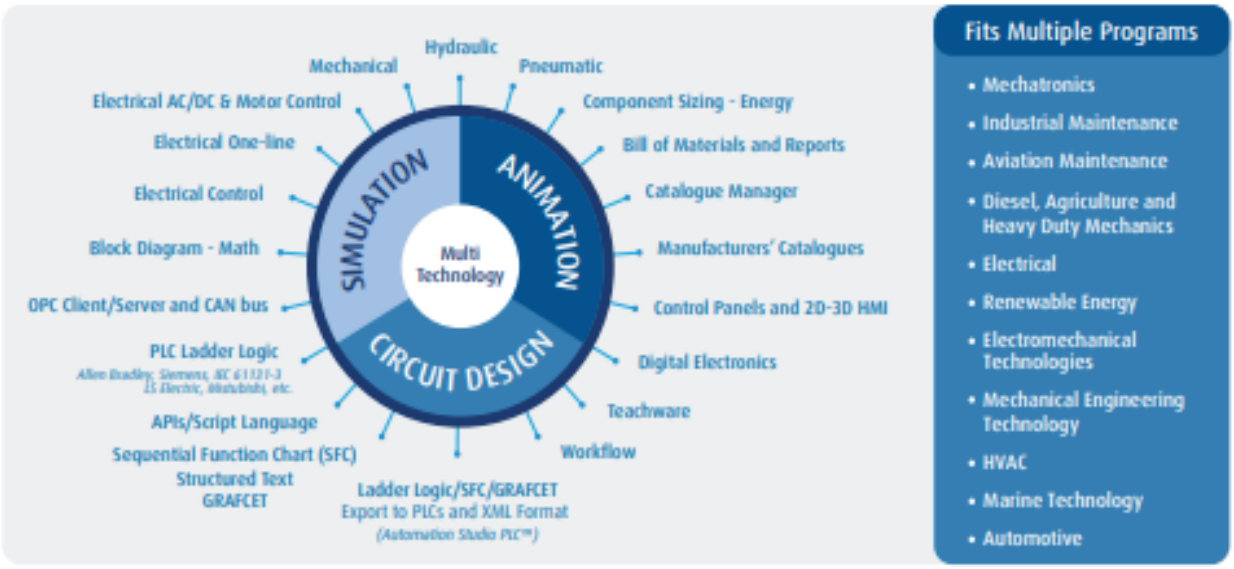

Figure 2: Features of Automation Studio software

- Here there are some photos from the project that show some of (AS) features and capabilities:

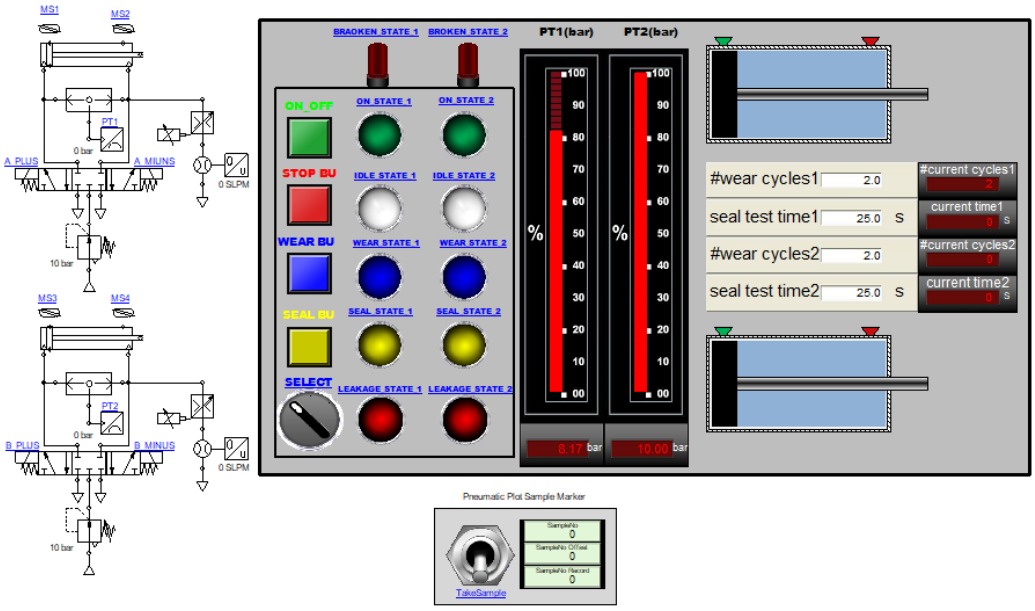

Figure 4: Electro-pneumatic and HMI systems of the project in Automation Studio software

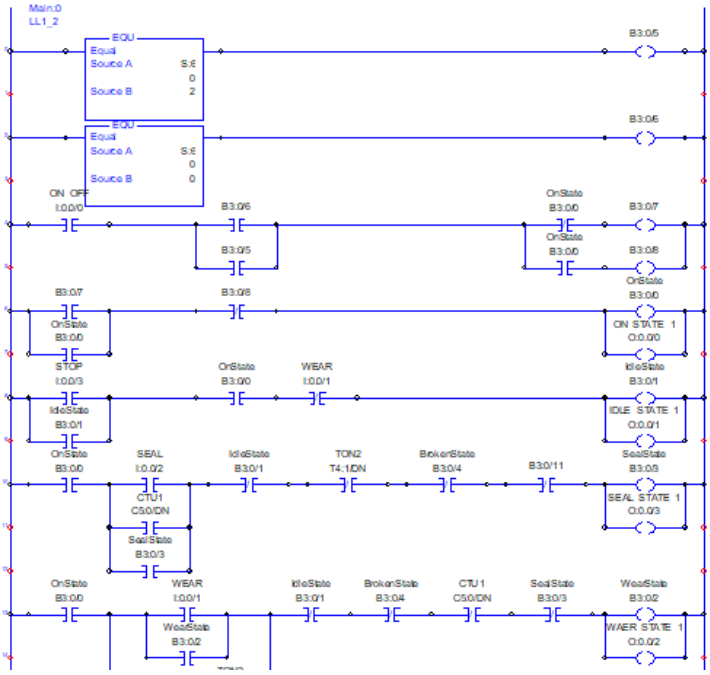

Figure 3: Part of the Ladder program of the project in Automation Studio software

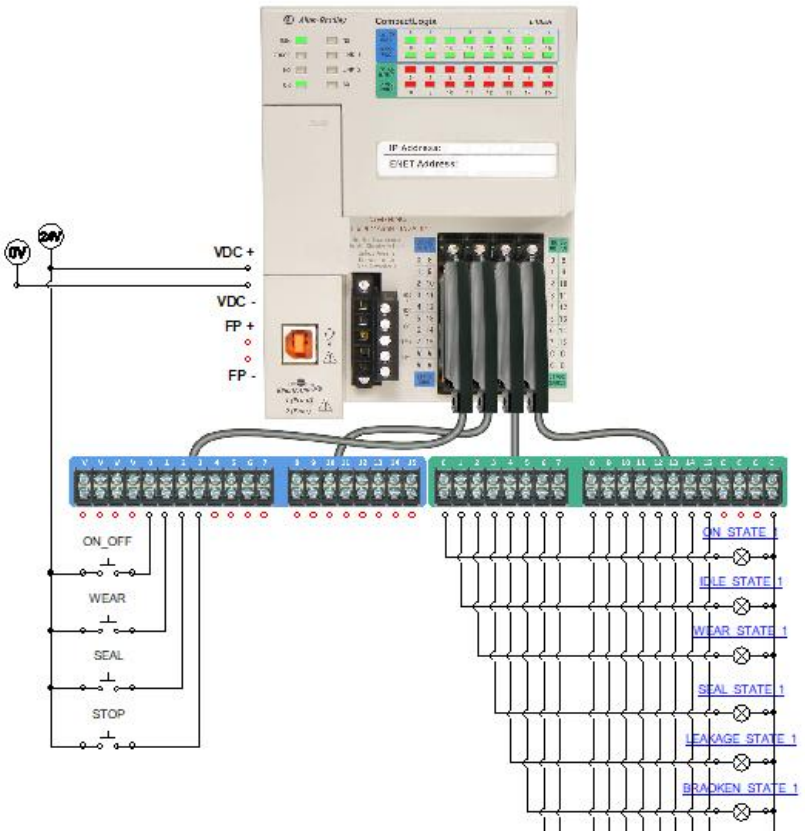

Figure 6: The Allen Bradley PLC simulation of the project in Automation Studio software

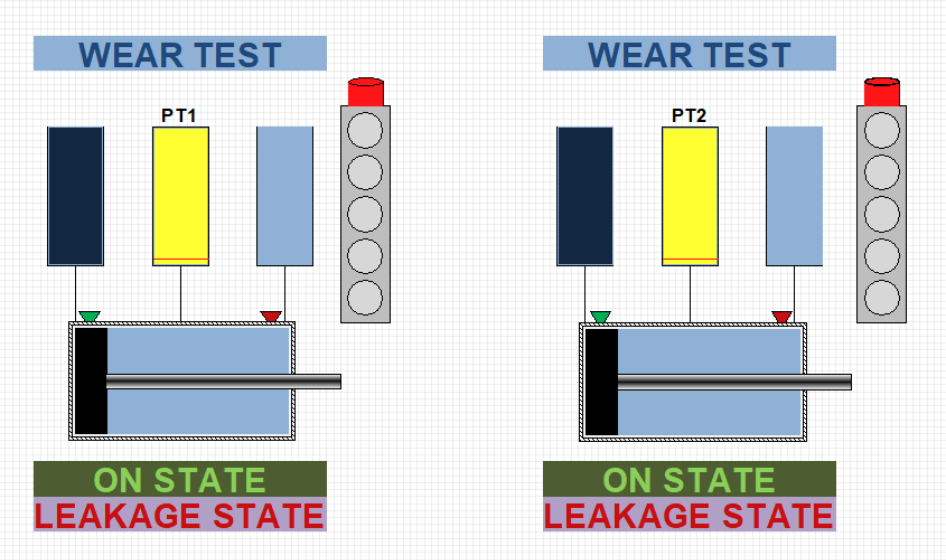

Figure 5: The supervisory system of the project in Automation Studio

2.2 How could we use AS to simulate the project in detail!?

2.2.1 Electro-pneumatic system

- First, we will briefly explain the electro-pneumatic system as it is discussed before in detail in the second chapter:
  The electro-pneumatic system is a small test bench for two cylinders (you can add as many cylinders as needed with just a little modification in the SW program), what we are doing here is a leakage or seal test for these two cylinders, we know that as the cylinder do more cycles (extension and retraction) which is called "wear phenomena", the sealing capability of the gaskets will be affected over time and so the leakage happens.
- In our project we can choose either to do the test for only one of the two cylinders or both together at the same time.
- Now we are going to explain in detail the electro-pneumatic system for only one cylinder as they are completely identical and how we could do that using AS software:

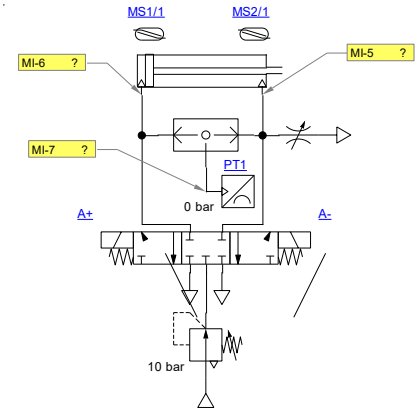

Figure 7: The electro-pneumatic system in AS software

Component Name	Quantity
Direct Exhaust	3
Directional Valve	1
Double-Acting Cylinder	1
Magnetic Sensor	2
Pneumatic Pressure Source	1
Pneumatic Pressure Transducer with Analog Output	1
Shuttle Valve	1
Variable Pressure Reducing Valve	1
Variable Throttle Valve	1

Figure 8: The electro-pneumatic system components in AS software

As shown in the previous two photos, we can see the electro-pneumatic system and a table of all the components used and their quantity.

Now we will know how to make that electro-pneumatic System using Automation Studio software:

1) When we open the SW, this window shown under this paragraph will appear in front of us, we can see on the left of that window, the generic libraries in which we can find all the generic components we need for our project. We used two libraries for the electro-pneumatic system: “pneumatic and proportional pneumatic” libraries.

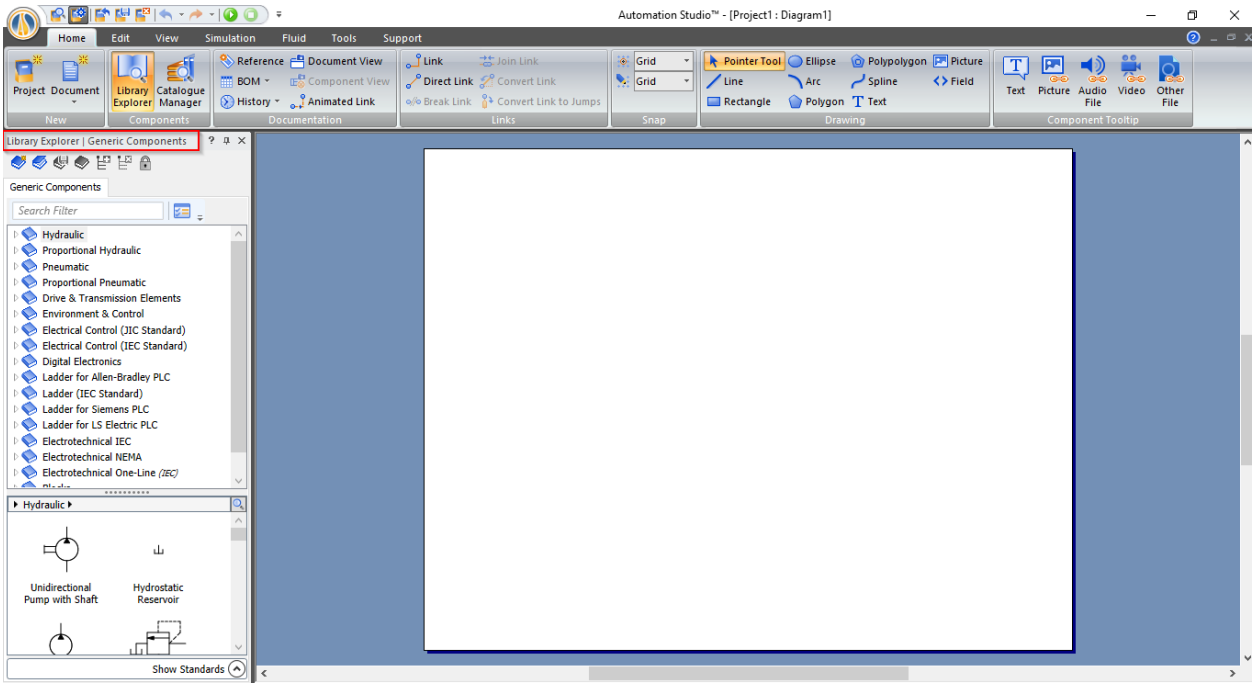

Figure 9: The main window of AS and where to find the generic components

2) We used a 5/3 way normally closed directional control valve that can be electrically activated in the project. Two springs can return the directional control valve to the initial or idle position. To use this component, we go to the pneumatic library -> directional valves -> 5/3-way valves and we can pick up the valve we need by just dragging it into the diagram, and it will directly appear in the project document as shown on the right photo below:

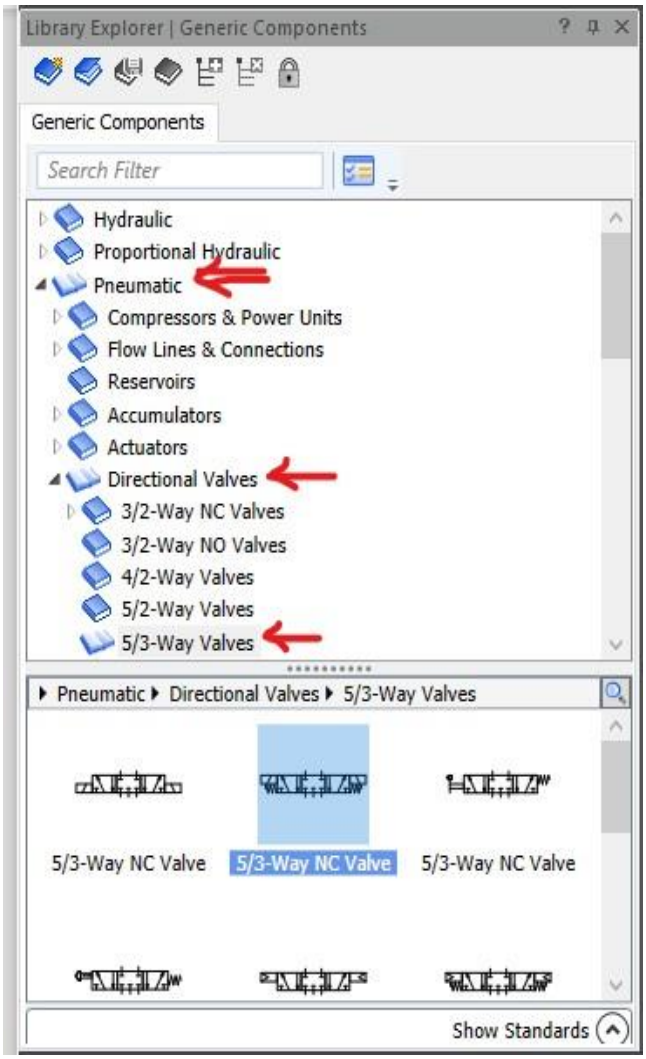

Figure 11: Adding a 5/3-way normally closed valve in AS software

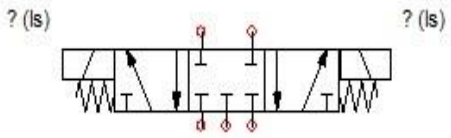

Figure 10: The 5/3-way normally closed valve component in AS software

Note that you could change all the features of any component you choose by just double click on it using the right button of the mouse, like our component here, if you made a double click on it, this window shown in the photo below will appear, and by selecting “data” from the left part of the window, you can see all the properties of the component written in front of it the default value and you can change it to the value you like, and you can do the same for any other component:

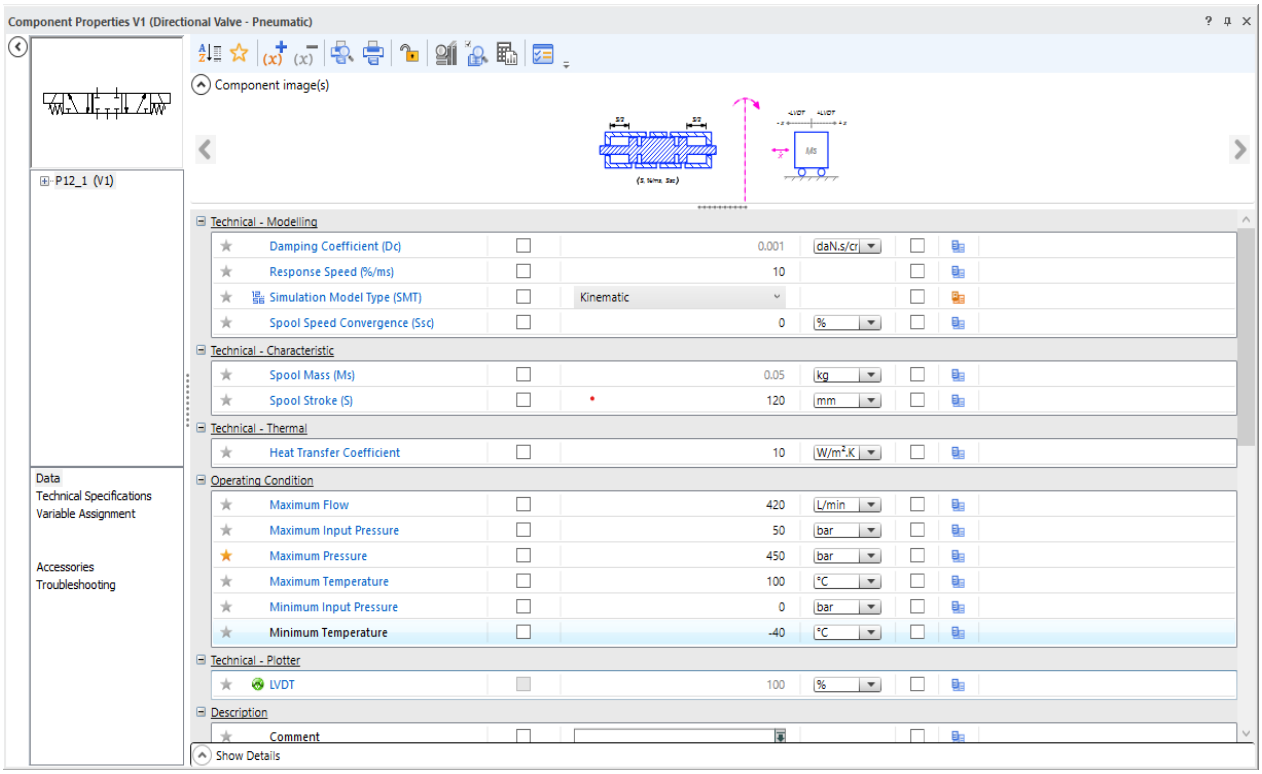

Figure 12: The properties of the directional control valve

Also if you would like to know any information about your component (technical, operating condition, commercial...etc.), it is very easy, you just go and make a right-click on that component, and choose the last option in the list which is “context help” as shown below in the photo, a window (like in the photo below) will automatically show up containing all the information you may need about that component in:

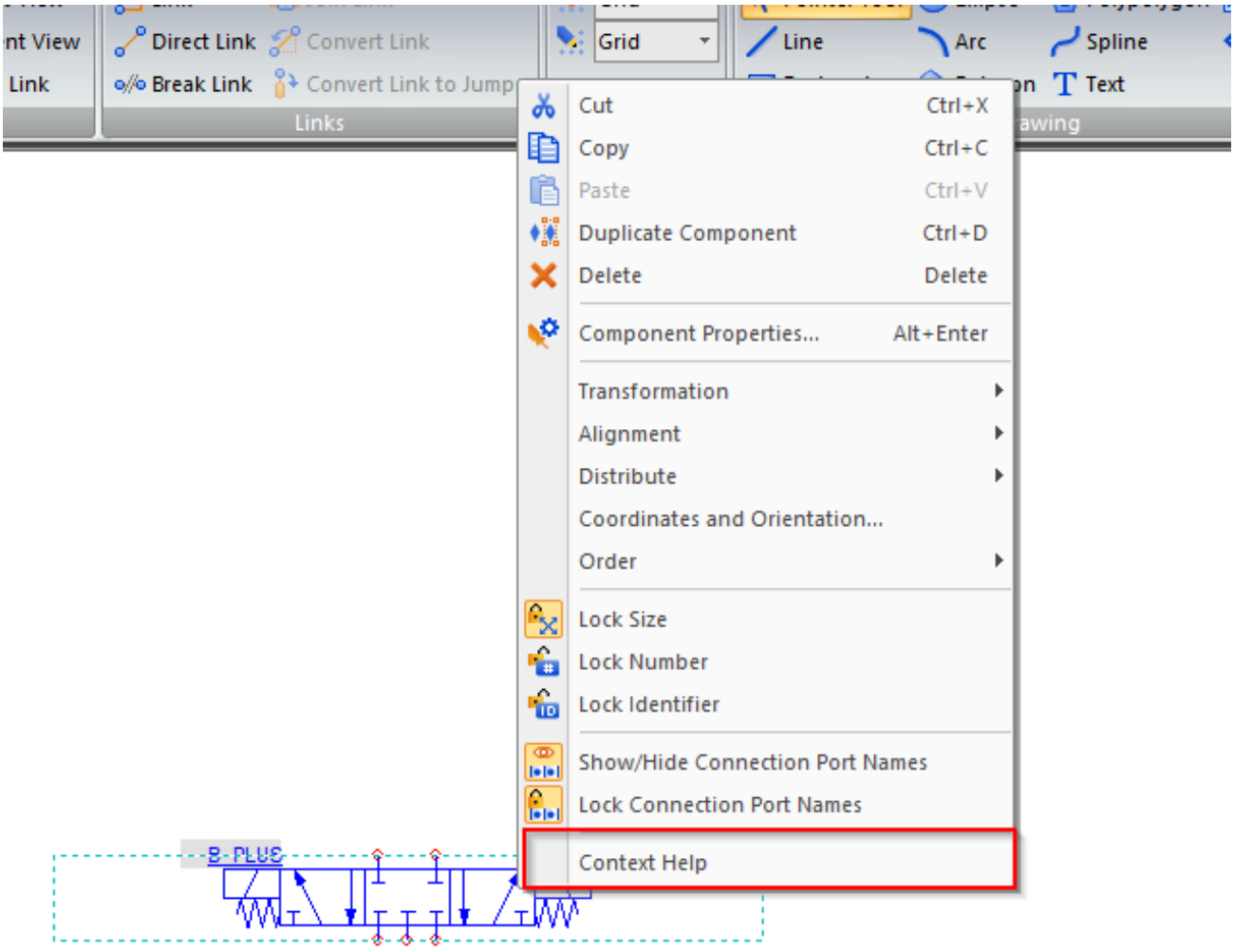

Figure 13: How to get any info. about any of the AS components

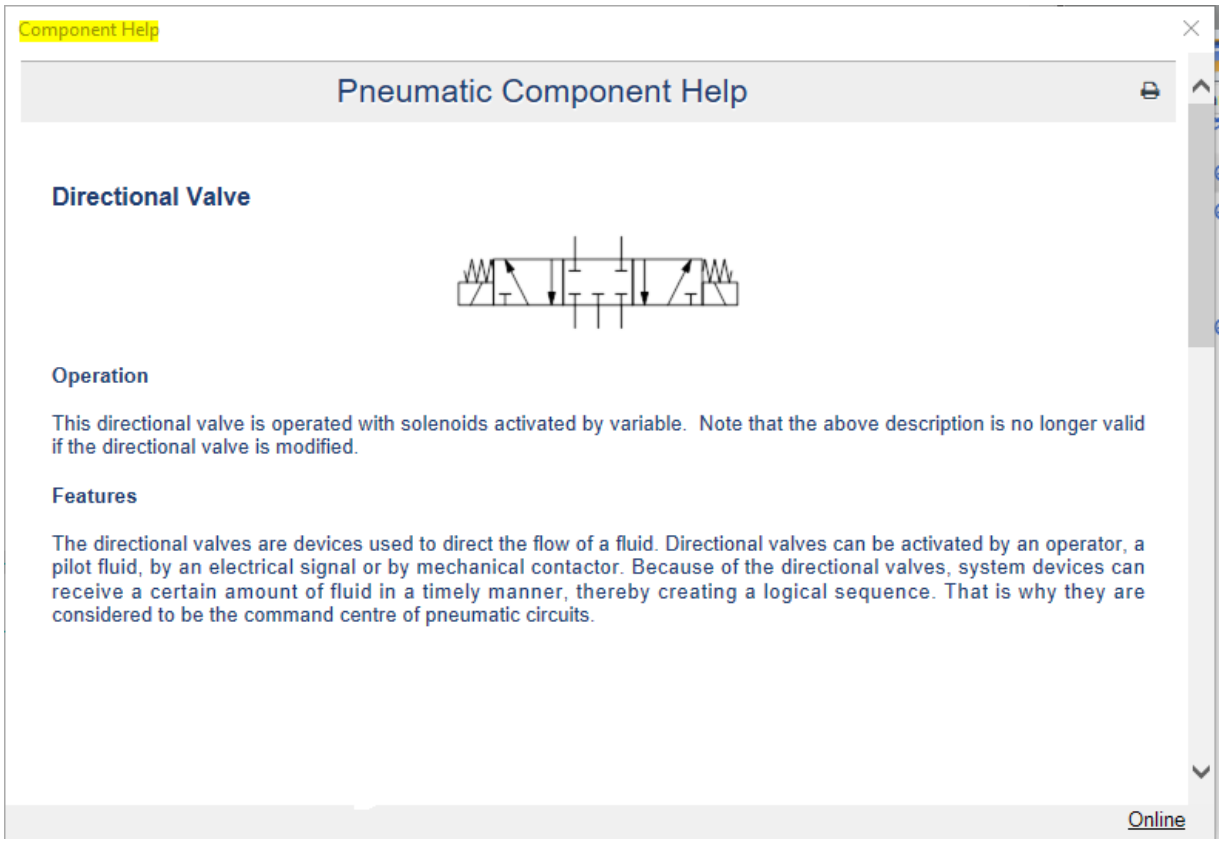

Figure 14: Help info. about the directional valve in AS software

By the way, if go by your mouse pointer to any part of the program and wait for a second, a few words will appear in a small window that can help you know what you are pointing to.

3) By doing the same we can get our double-acting cylinder but that time we go to the actuators section as shown in the photos below:

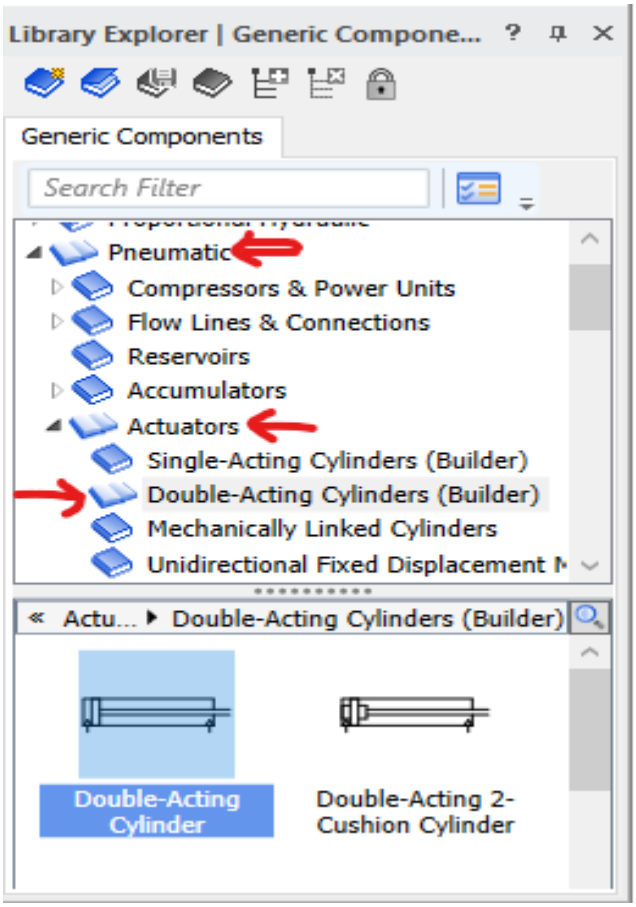

Figure 16: Adding a double-acting cylinder in AS software

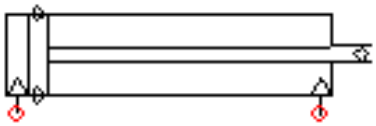

Figure 15: Double-acting cylinder component in AS software

4) We will use also a “variable pressure reducing valve” so that we can maintain the air pressure at a certain value which is “10 bar” in our project. You can find this component in the same library (pneumatic), and you can get it as shown in the below photo:

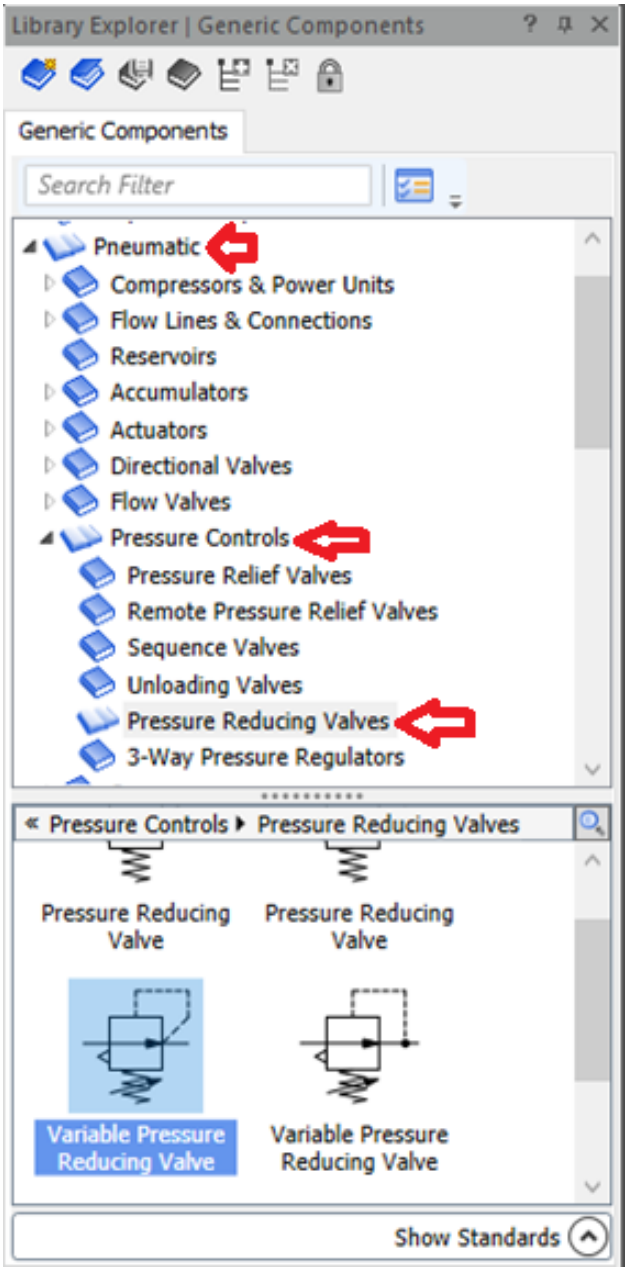

Figure 18: Adding a variable pressure reducing valve in AS software

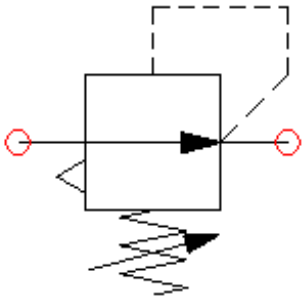

Figure 17: Variable pressure reducing valve component in AS software

Note that you can rotate or flip any of the components you use so that you can use them in the needed orientation in your project and that can be done easily either by using “ctrl+H” or by clicking right on the component and from the transformation option, you can rotate or flip your component in any direction as shown in the photo below:

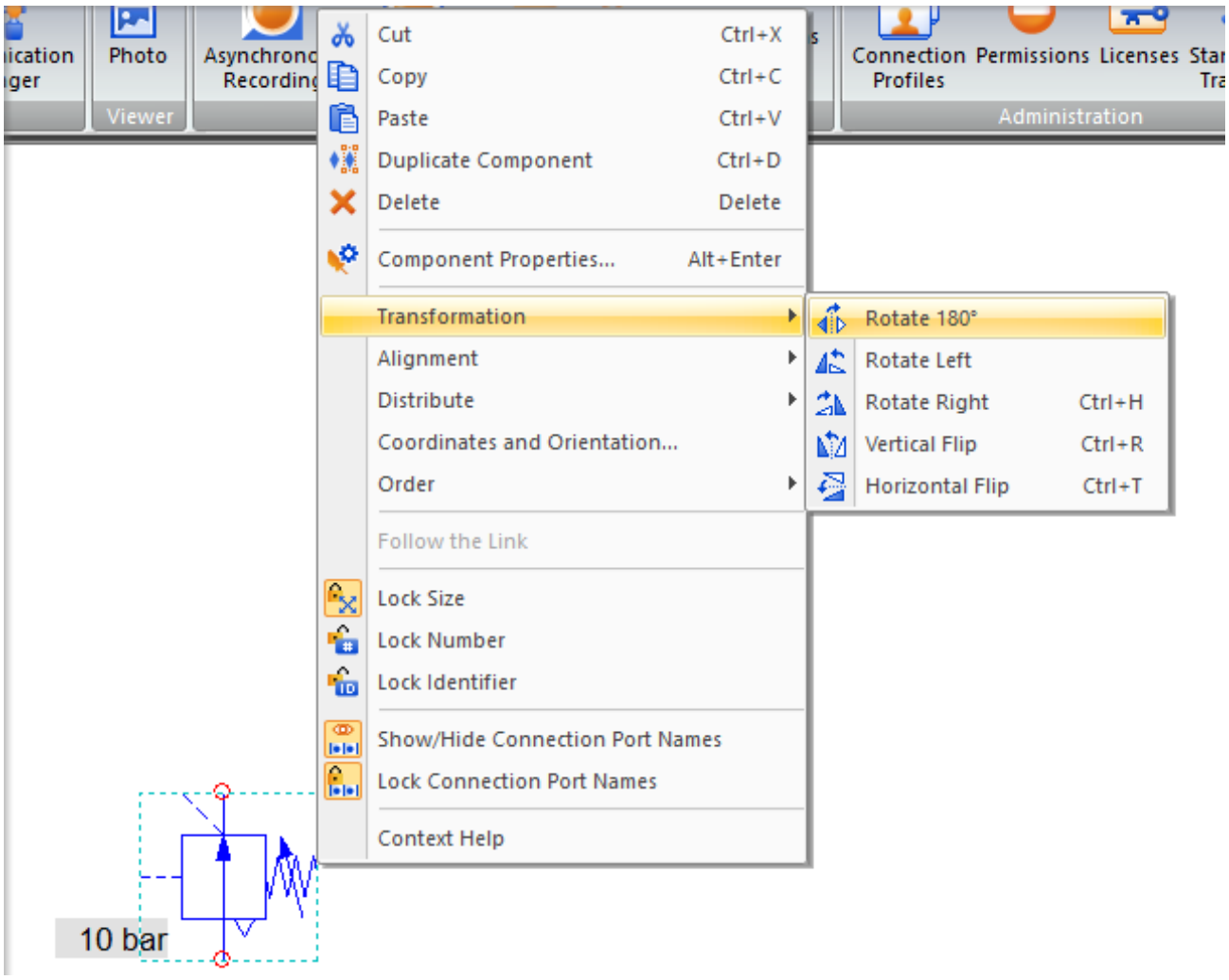

Figure 19: How to rotate a component in AS software

Or you can just select the component you would like to transform and go to the “edit” part of the toolbar, you will find a position menu containing all the needed transformations as shown in the photo below:

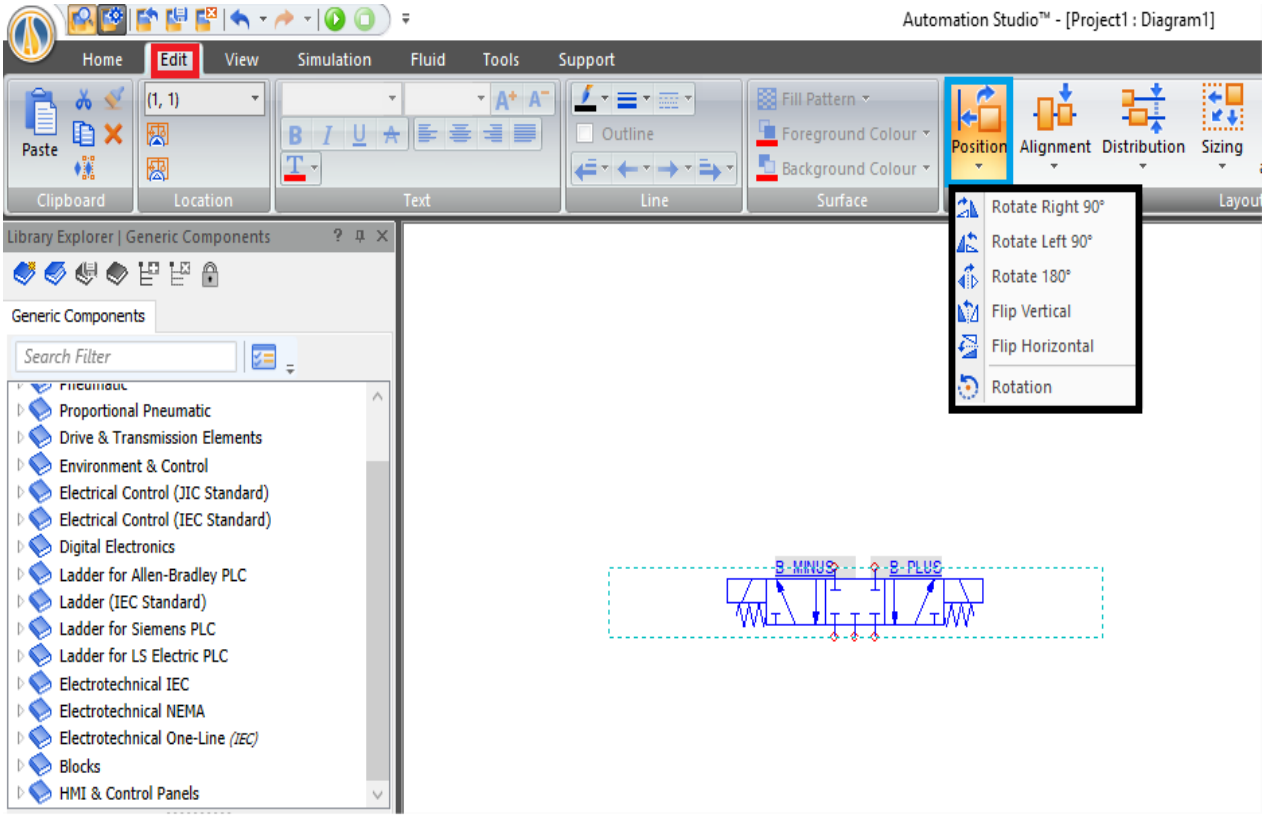

Figure 20: Another way for rotating a component in AS software

In the project, we want to use a pressure of 10 bar, it is easy to do that, you just go to the component properties as shown before by double-clicking on it, and then you change the “setting pressure” from 5 bar (the default value) to the value you need which is 10 bar in our case. If you want to show this value next to the component, you must check the small box next to the “setting pressure” label as shown in the photo below, and you can do the same for other properties:

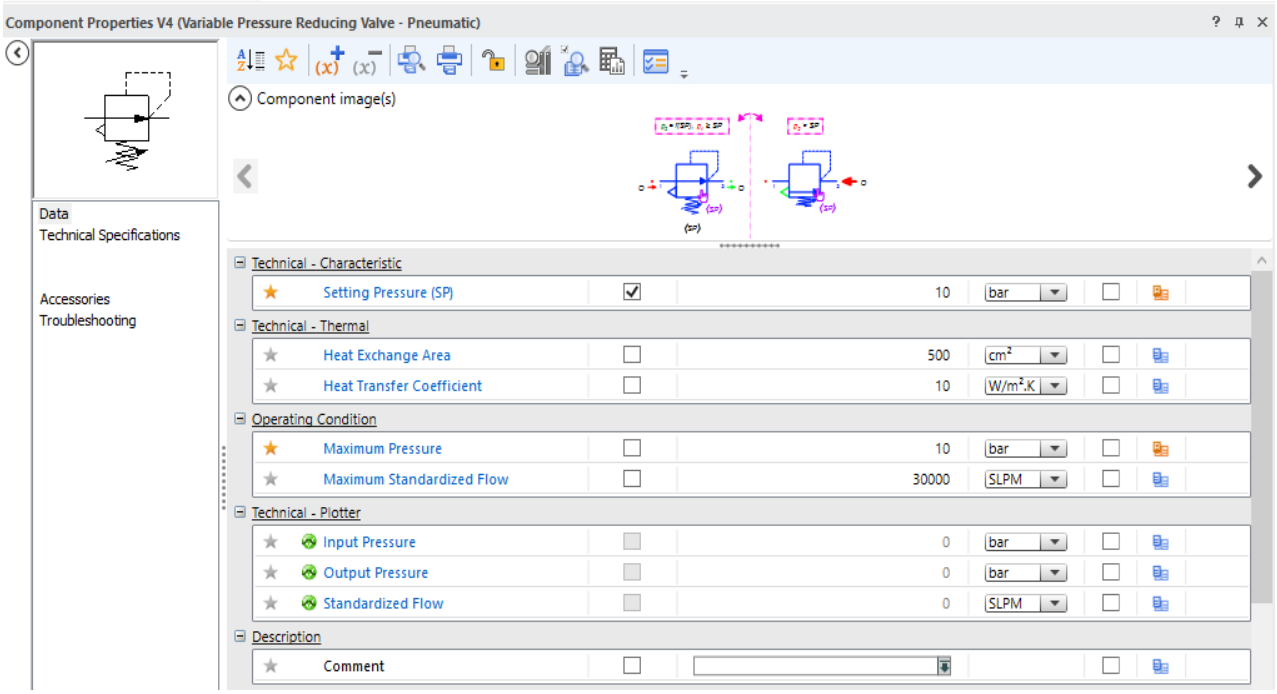

Figure 21: The properties of the variable pressure reducing valve in AS software

5) We will also need two sensors to indicate the extinction and retraction of the cylinder, and in the project, we use two magnetic sensors to do that. You can find these sensors in the same library (pneumatic), and you can get them as shown in the photo below:

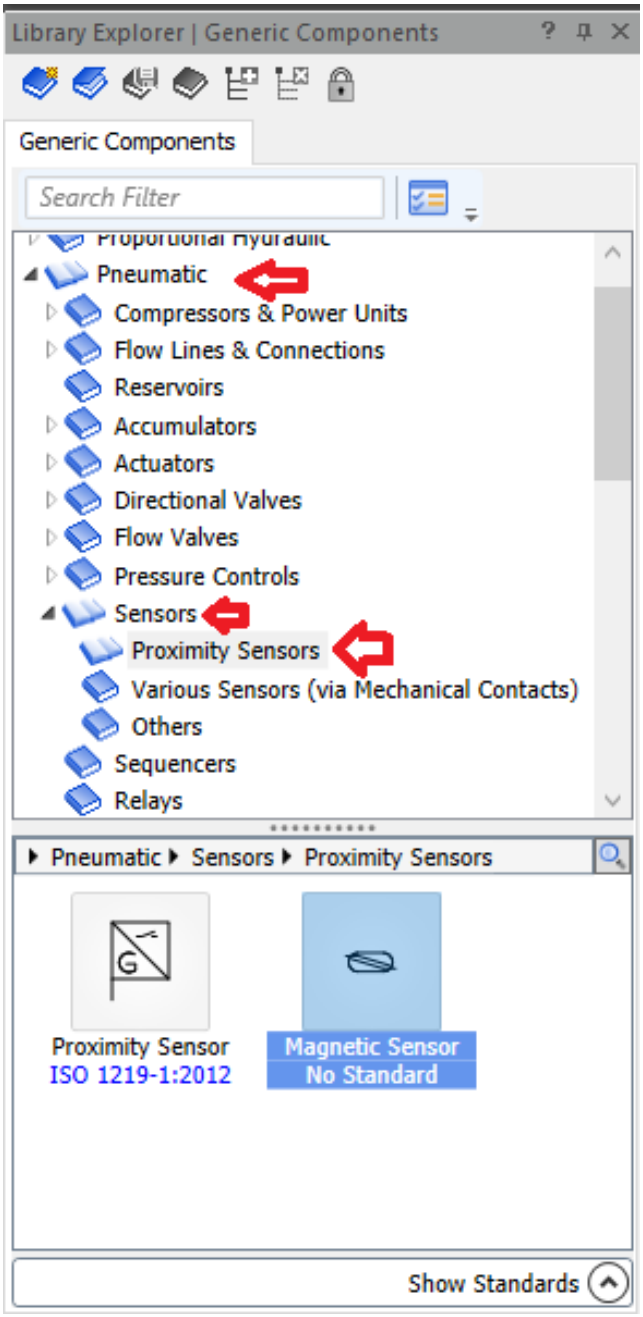

Figure 23: Adding a magnetic sensor in AS software

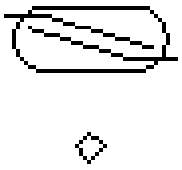

Figure 22: Magnetic sensor component in AS software

Note that for some components like the magnetic sensor, when being dragged into your diagram, a small window automatically shows up asking you to write an “alias” for your component which is the name you prefer to use for that component for example. We have chosen “MS1” for the first magnetic sensor, and you can choose whatever you want, but it is better to choose something related to the real component name so that you can remember it easily, because later you will use that alias to search for your component, and we will see that later. If you do not want to choose the alias right after you choose the component, there is another way to do that by just closing this window that showed up automatically and choosing the alias of your component later, and you can do that easily as shown in the photo below, you just must go to the data section of your component as shown before, then just next to this small green led, you make a double click with the left button of the mouse, and you be allowed to write whatever you like (the alias of the component):

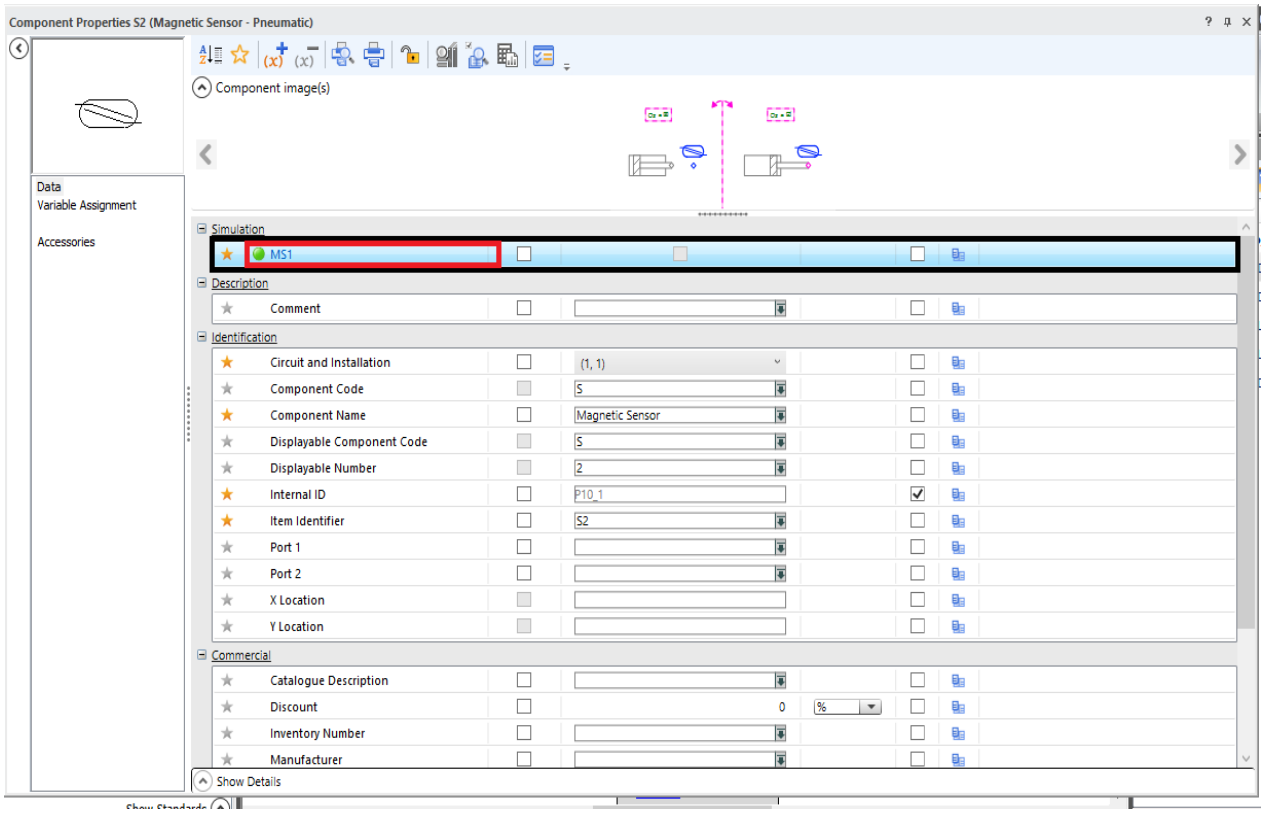

Figure 24: How to write the alias of a component in AS software

6) We also use a “pneumatic pressure transducer with analog output” to measure the cylinder’s pressure getting in and out. We use only one pressure transducer for each cylinder. However, we want to measure the pressure on two points, and we will see later the reason why we use only one instead of two. You can get this component easily, but this time we will use a new library which is “proportional pneumatic” as shown in the photo below:

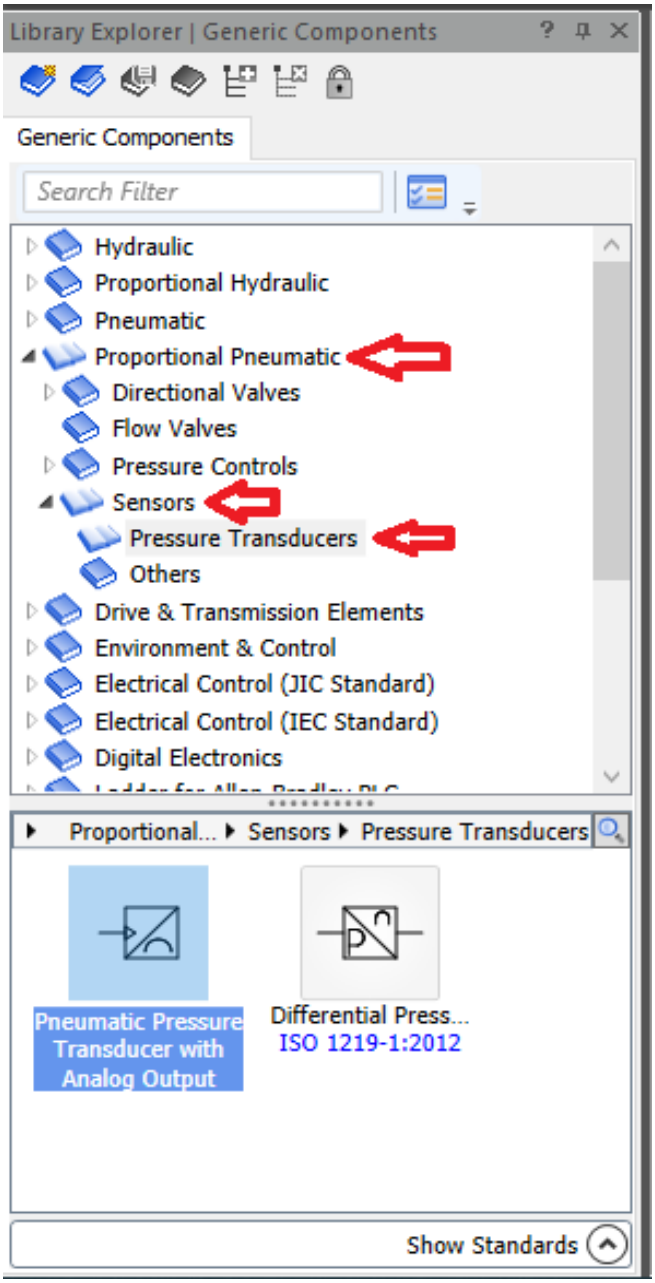

Figure 26: Adding a pneumatic pressure transducer in AS software

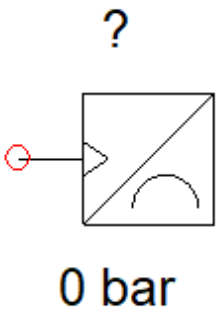

Figure 25: Pneumatic pressure transducer in AS software

7) We use also a “shuttle valve”, and the importance of that valve in our project is that it allows you to use only one sensor instead of two sensors to measure the pressure on the two points of the cylinder. As this valve has 3 connection points, we can connect the 2 points (on the right and left) to the 2 points of the cylinder, and connect the pressure transducer to the third position of the valve (the middle position) as we will see at the end of explaining the electro-pneumatic part, and so when the air pressure is going into one side of the cylinder, this valve closes completely the other side, and so allows the pressure transducer to read that pressure passing through that way, and vice versa. You can get that component easily by going to the “other flow valves” in the “pneumatic” library as shown in the photo below:

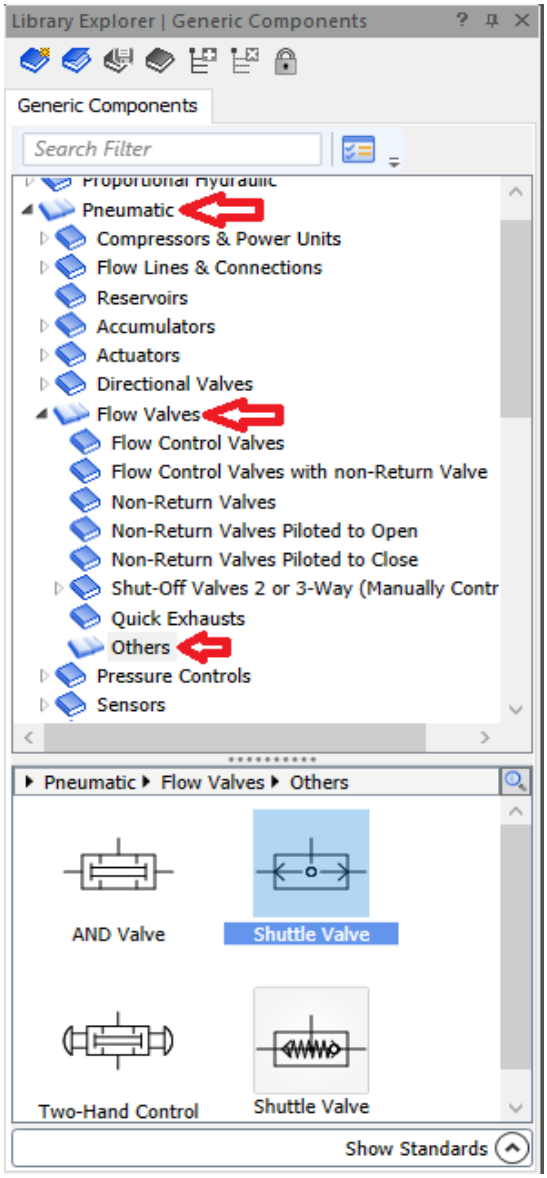

Figure 28: Adding a shuttle valve in AS software

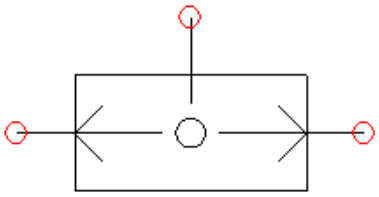

Figure 27: Shuttle valve component in AS software

8) We also use a “proportional flow regulator valve” to simulate the leakage. As we discussed before in detail, our project’s main goal is to make a leakage or seal test on the cylinder. We use this proportional flow valve to simulate the leakage and see what happens during that leakage and how the system is being affected by that leakage. You can find and use this component easily by just going to the “flow valves” in the “proportional pneumatic” library as shown below:

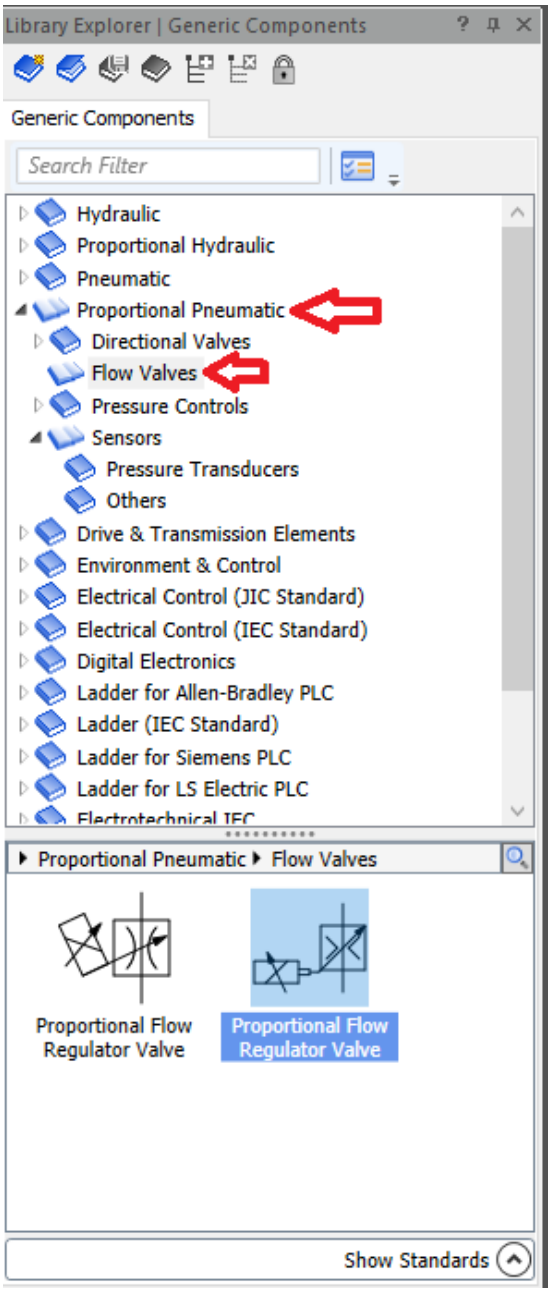

Figure 30: Adding a proportional flow regulator valve in AS software

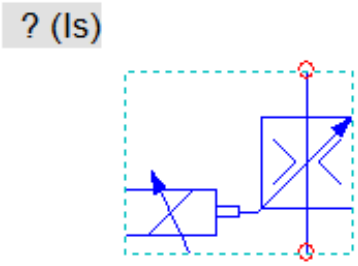

Figure 29: Proportional flow regulator valve component in AS software

9) We also use a “pneumatic flow transducer with analog output” to measure the flow passing through the “proportional flow regulator valve”, and so that to know the leakage flow of our cylinder. It is easy to find and use this component by just going to “other sensors” in the “proportional pneumatic” library as shown in the photo below:

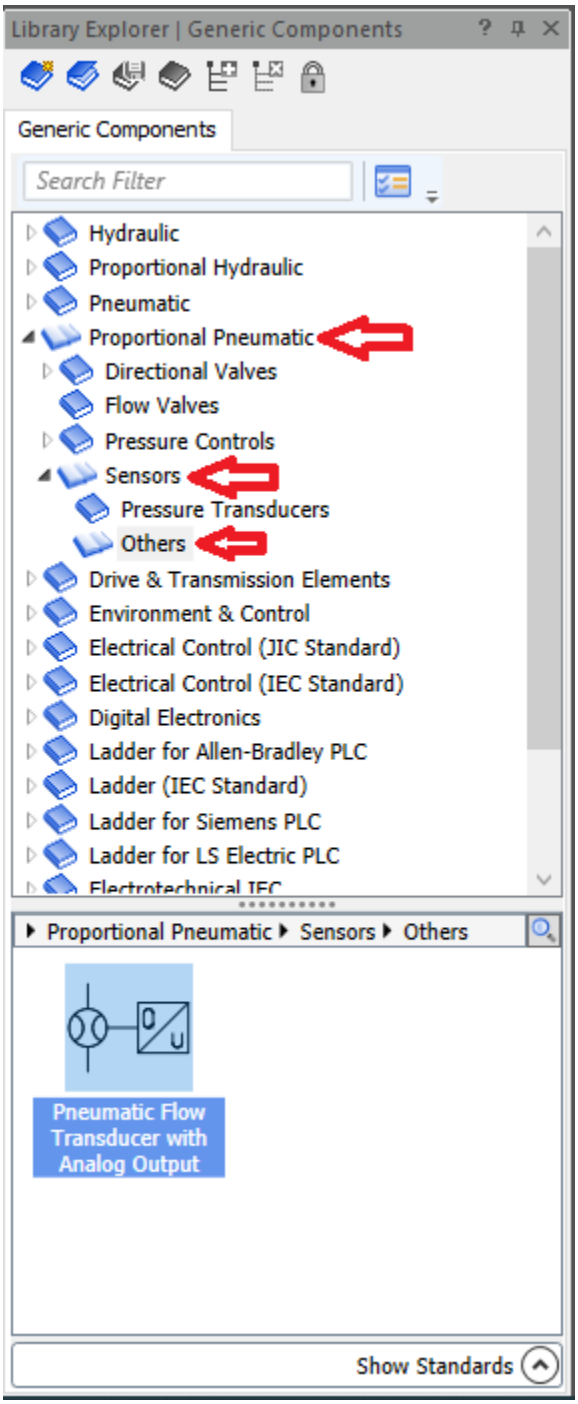

Figure 32: Adding a pneumatic flow transducer in AS software

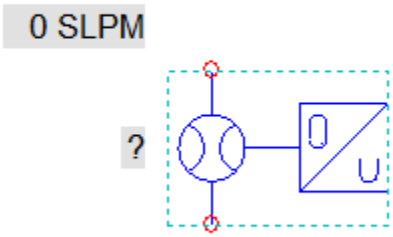

Figure 31: Pneumatic flow transducer component in AS software

10) Now we have all the needed components for our electro-pneumatic system (for one cylinder only) except for the pneumatic source and the exhaust, and these 2 components can be found and used easily with just one left mouse click on the “pneumatic” library as shown in the photo below:

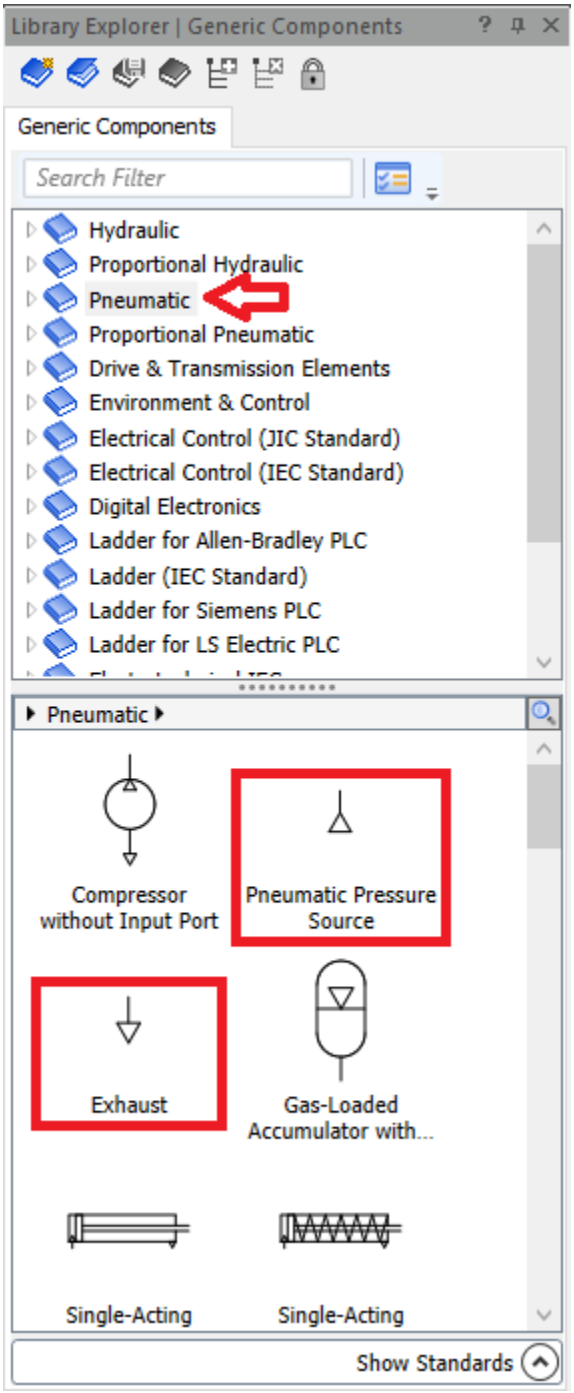

Figure 34: Adding a pneumatic pressure source and exhaust in AS software

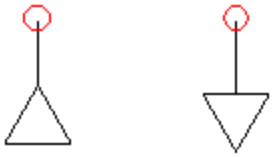

Figure 33: Pneumatic pressure source and exhaust components in AS software

11) There is only one step left to finish our electro-pneumatic system which is the connection between all these components we already had, and that will be very easy. You for sure noticed that each component we used has one or more red circles except for the “magnetic sensor” that we will see how to connect later, these red circles are the connecting points of your component with any other component, you need just to go to one red circle and press on time on it by the left button of your mouse and just move up, down, left or right with your mouse, and you can find the connecting line move with you wherever you need. When you reach the red circle of the other component you want to connect with, you just press the mouse’s left button one time, and now the connection is done. Note that if you made any 90 degrees angle in your connection line (like an elbow connector in real systems), you must press one time on the left button of the mouse, so that to confirm that step, for example, if you were going up from one component, and then you want to go right (making 90 degrees) and then go up again to connect to the other component, then you have to press on the left button of your mouse at the endpoint of your horizontal line and then go up again (shown by the blue arrow in the photo below). You also can move this horizontal line up and down by just moving your mouse up and down.

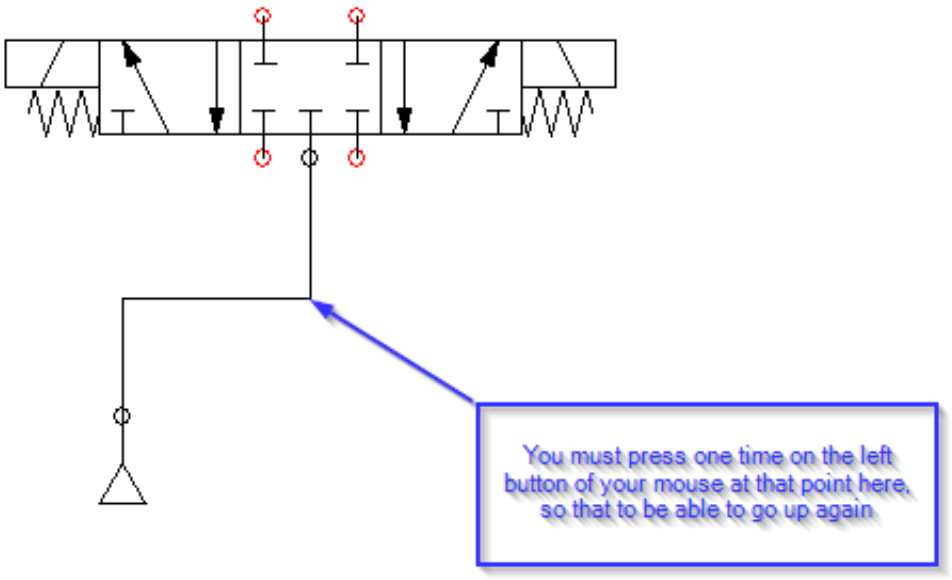

Figure 35: How to make an elbow in the connection lines in AS software

12) For the magnetic sensor connection with the cylinder, as we can see in the photo below, the magnetic sensor has a small shape under it which is an inclined parallelogram, and we will also find the same shape at the top left of the cylinder as also shown in the photo below:

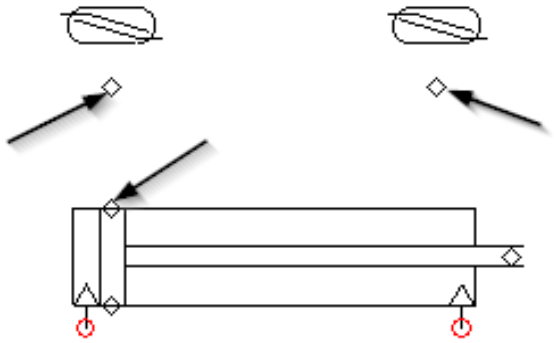

Figure 36: How to connect the magnetic sensors with the cylinder in AS software

For the left magnetic sensor, we can connect it easily by just moving it until the inclined parallelogram of it is exactly above the one related to the cylinder and seem like they are one inclined parallelogram as shown in the photo below (the left magnetic sensor):

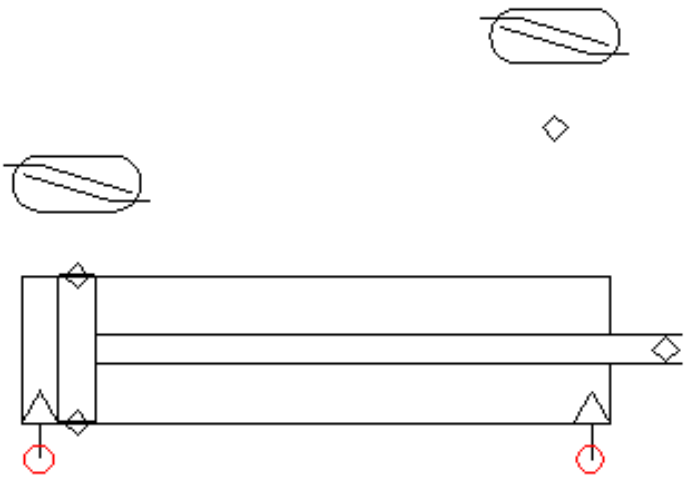

Figure 37: How to connect the magnetic sensors with the cylinder in AS software

For the right magnetic sensor, we must do one extra step to connect it with the cylinder which is that we must extend the cylinder to its final position (100% extension), and we can do that by double-clicking on the left button of the mouse (on the cylinder), and we go to the data section of the cylinder as we explained before, then we change the value of the extension from 0% to 100% as shown in the photo below:

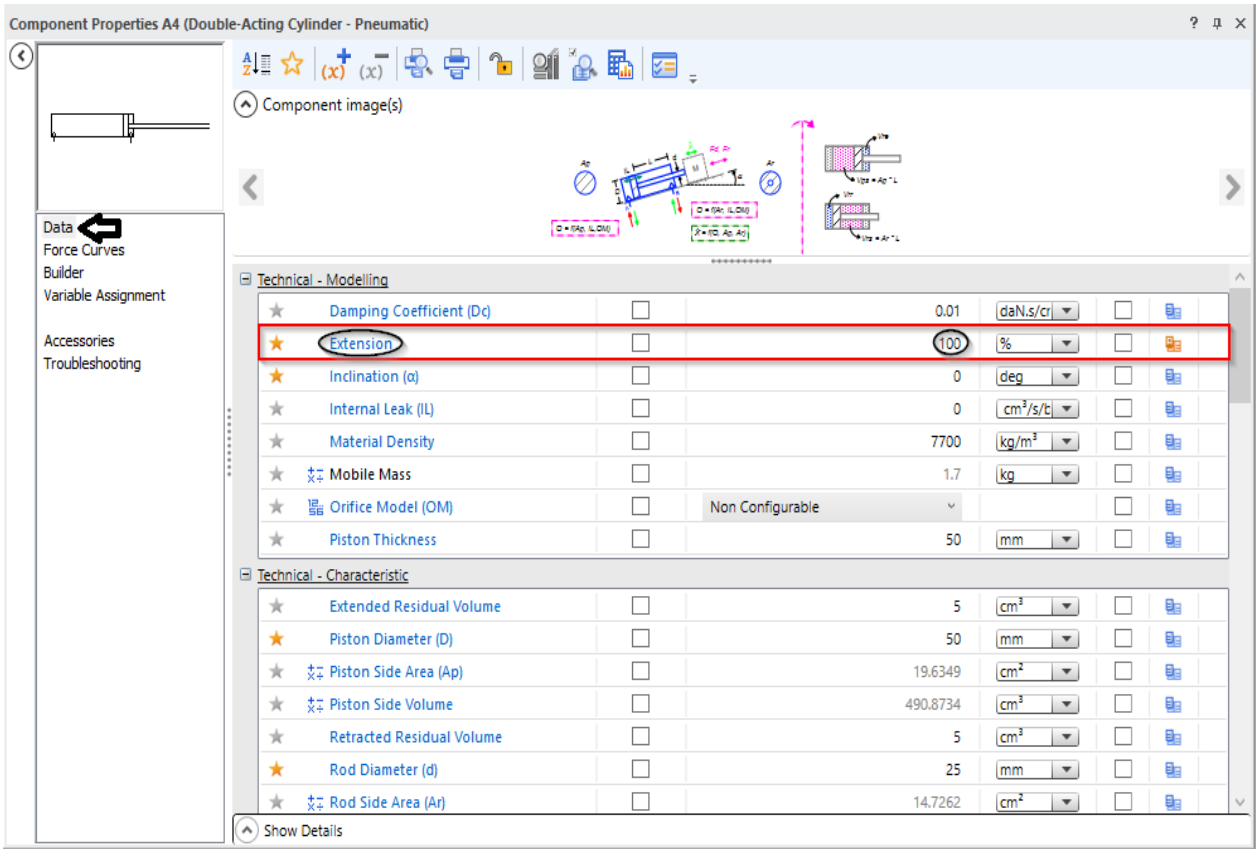

Figure 38: How to make an extension to the piston of the cylinder manually in AS software

After we do that, we do the same we did with the first sensor as shown in the photo below:

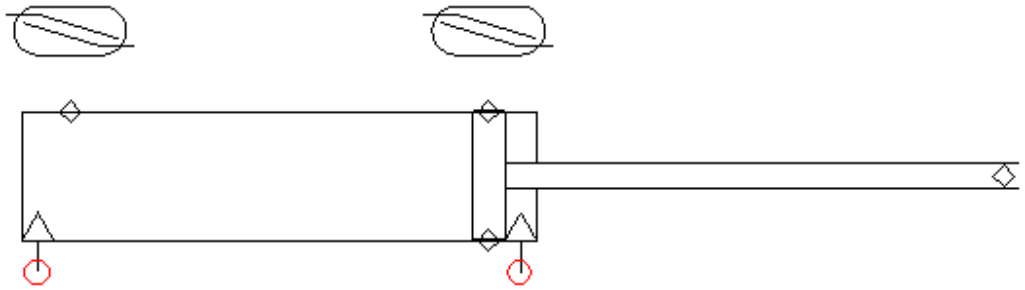

Figure 39: Connecting a magnetic sensor to the cylinder in AS software

And then we move the cylinder back to its initial position by changing the “extension” again from 100% to 0%.

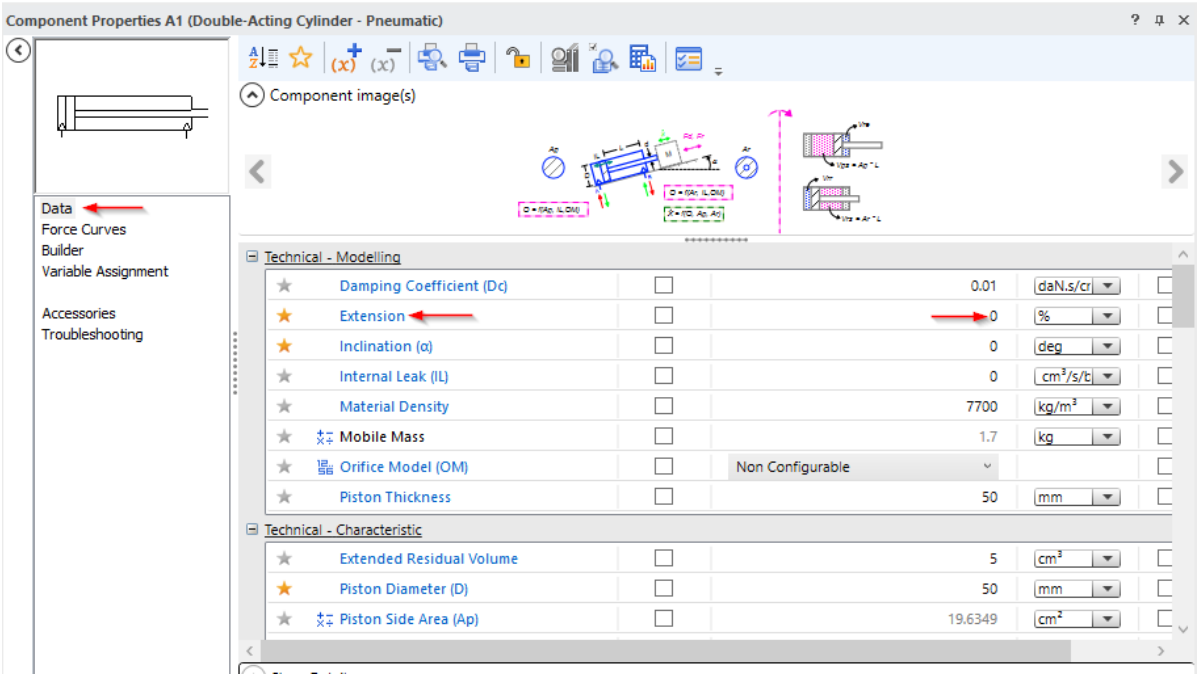

Figure 40: Returning the piston of the cylinder to its normal position in AS software

13) If you did all the previous steps, you would have an electro-pneumatic system like the one shown in the photo below:

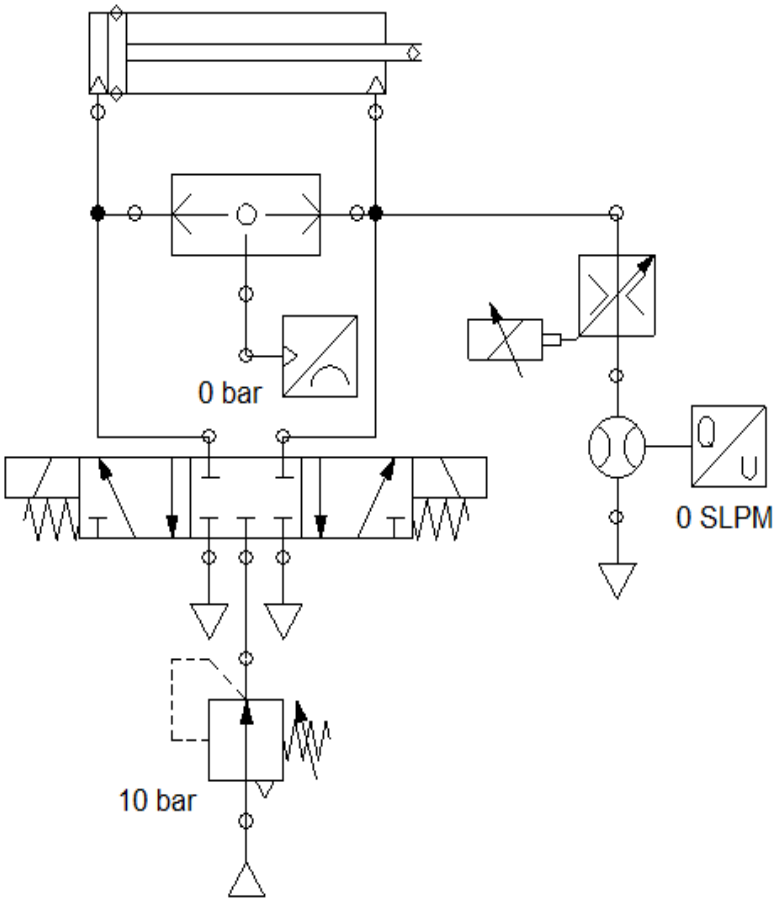

Figure 41: The complete design of the electro-pneumatic system in AS software

This is the complete electro-pneumatic system of our project for only one cylinder, and it is the same for the second cylinder. In the next step, we will explain in detail how we could make a ladder program to control this electro-pneumatic system and make it work as we need in the project to achieve our goal by testing our cylinder's leakage over time to analyze the results.

2.2.2 HMI (Human Machine Interface) & Control Panel

- We will explain in detail how we can find and use the components related to HMI, and the reason for using these components in our project, we must note that these components will be only for one cylinder:

1) We will use 4 pushbuttons, and by the way, these pushbuttons will be common for both cylinders (and for any number of cylinders we want to add):
    a) "ON/OFF" pushbutton which will be just for starting our system or making it ready to go for work in the sequence of our ladder program that we will see and discuss later.
    b) "STOP BU" pushbutton which we will use at any time to force the whole system to go to its initial position and stop, so it can be used in emergencies or stuff like that.
    c) "WEAR BU" which is used when we want to start our wear test (extension and retraction) like it will be in the real world so that we can make the leakage test after it.
    d) "SEAL BU" which is used to start the seal or leakage test at any time even before or during the wear test, but just for your knowledge, the seal test will automatically start after the wear test as we will see later in the ladder program.

You can find these pushbuttons easily in the “HMI & Control Panels” library as shown in the photo below:

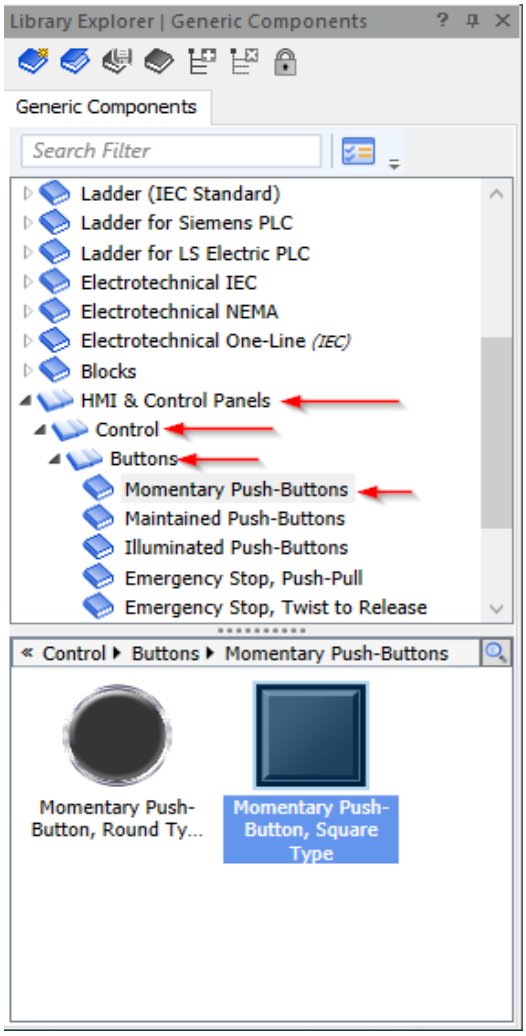

Figure 42: Adding a momentary push button in AS software

Figure 43: Momentary push button component in AS software

We use in our project four “momentary pushbutton, square type”, you can also use the other type (round type), they will be the same. As soon as you drag any of the pushbuttons to your diagram, you can immediately write the alias of it as we said before or close the window (that showed up automatically) and use the other way discussed before by going to the element data and writing the alias you want for your element.

After you do the same for all the pushbuttons and have them all in your diagram, you may need to change the dimensions of any or all of them, it is very easy to do so. When you press one time on any one of these pushbuttons using the left button of your mouse, some small squares will appear in each corner and also in the right and left middle as shown in the photo below, if you go with your mouse pointer to any of these squares, a left-right arrow will appear that allows you to drag this point in the directions of that arrow by just keep pressing on the left button of your mouse and drag. By doing that you will be able to change the dimensions of your element very easily.

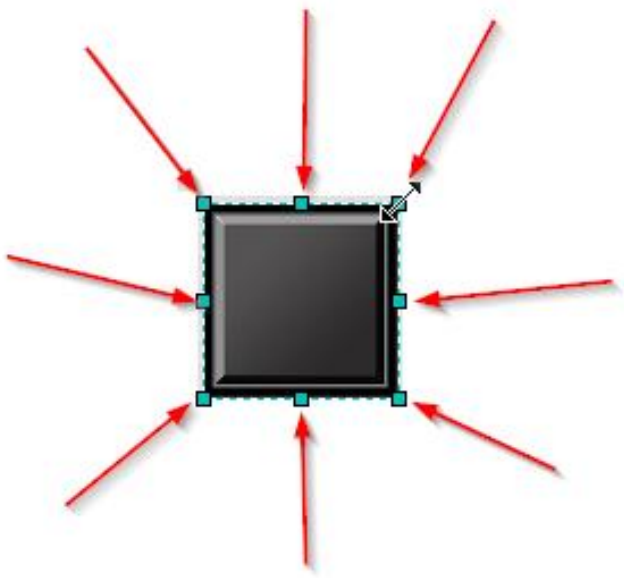

Figure 44: Changing the dimensions of a button in AS software

You can see in the previous photo that we are pointing to the top right square, and so this left-right arrow showed up.

You must take into account that you don't have to change the dimensions for each push button. In case you want them all to be identical as we did in our project, you just have to change the dimensions of one of them, and after you finish, you just highlight it by pressing on it using the left button of your mouse as you did to change its dimensions, and then you press on the "shift" button of your keyboard and keep pressing on it through the whole procedure, and then you have to select the other elements or pushbuttons you would like them to have the same dimensions by pressing on each of them using the left button of your mouse, then you can release the "shift" button, and you must go to the "Edit" section of your toolbar, then go to the sizing option and choose "Width and Height" as shown in the photo below:

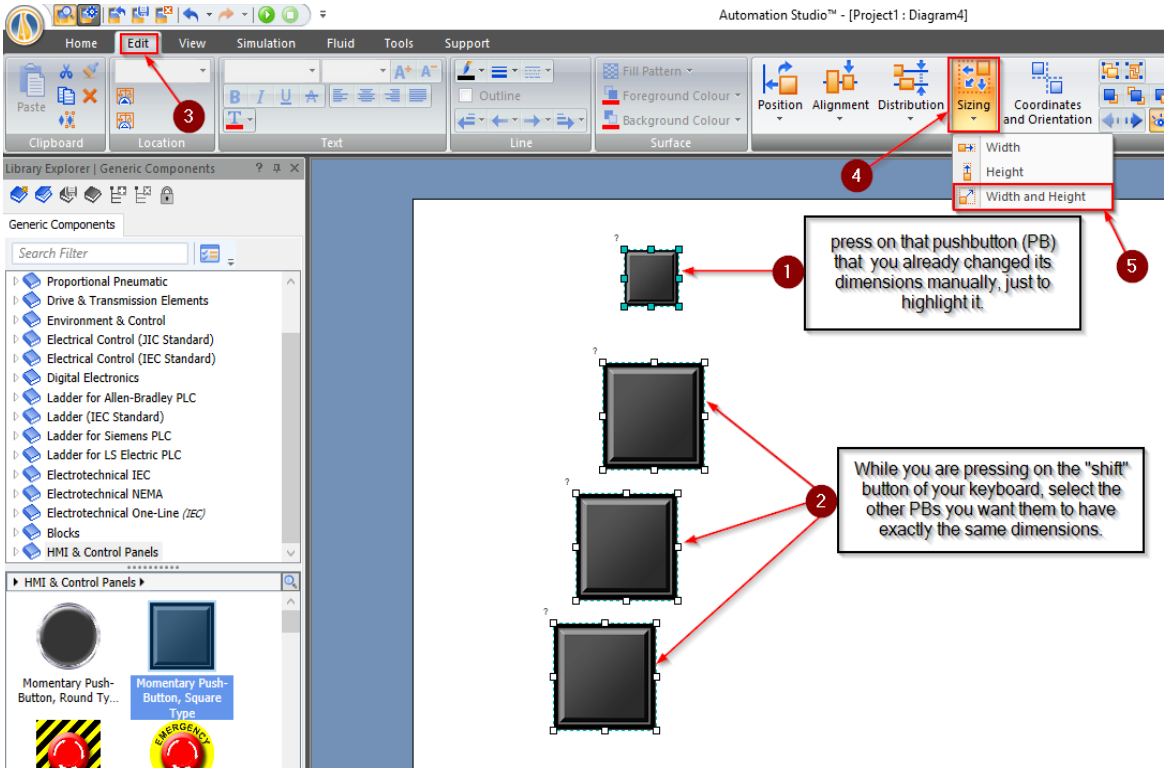

Figure 45: Giving the same width and height for many buttons according to a reference one in AS software

After you do these steps, you will have the same dimensions for all the selected pushbuttons without the need of doing that manually for all of them.

Also, you may need to align these elements or any other elements to be at the same line, you can do that easily by just selecting all the elements you need to be aligned, and going to the "Alignment" option in the "Edit" section of the toolbar as shown in the photo below and choose "Align Middle":

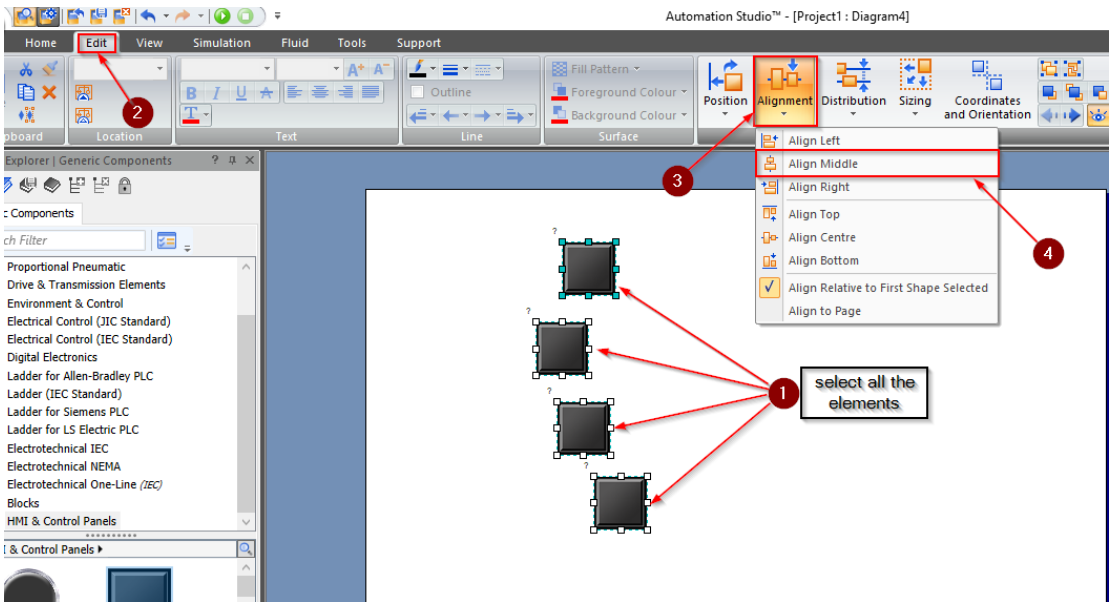

Figure 46: How to make alignment for some components in AS software

You can choose another alignment option as you can see in the previous photo, we have chosen "Align Middle" to make all the pushbuttons arranged on the same vertical line. Note that the question mark shown at the left top of each pushbutton is because we didn't write the alias of these components until now.

You may need also to have equal space between each element and the other, it is also very easy to do by first selecting all the elements, and then you go to the "Distribution" option in the "Edit" section of the toolbar and choose "Distribute vertically" as shown in the photo below:

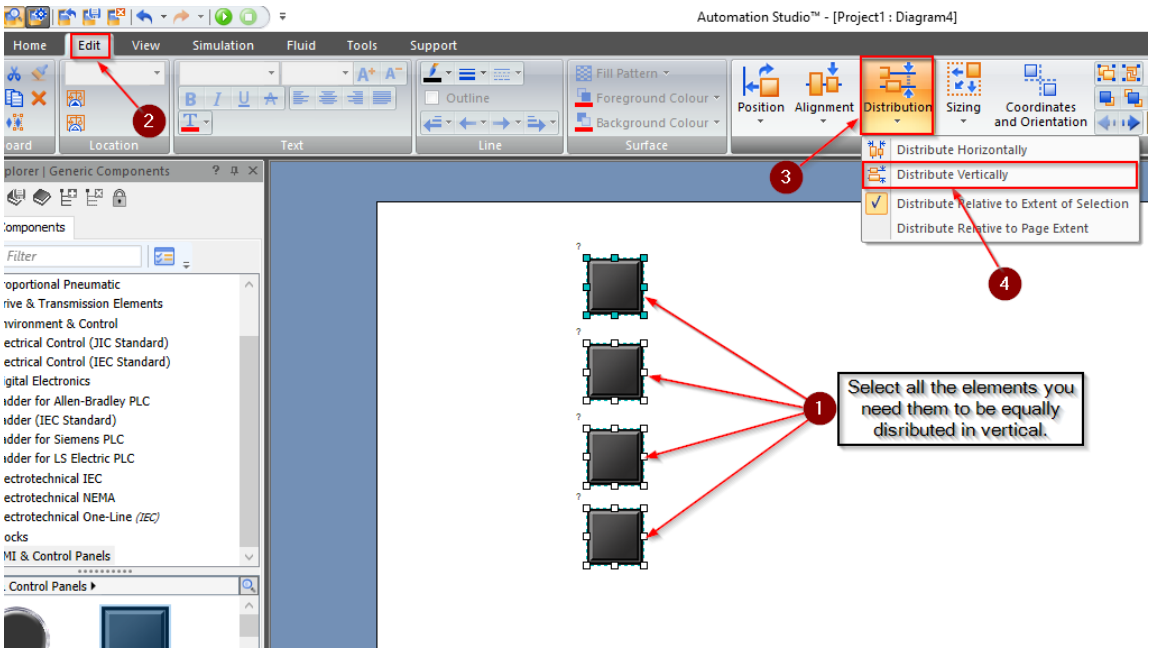

Figure 47: How to distribute vertically some of the components in AS software

You can choose other distribution options as needed for your project, we used "Distribute vertically" in our project because we wanted our pushbuttons to be equally distributed vertically as shown in the photo below:

Figure 48: Vertically distributed buttons in AS software

2) We will need also a “selector” which will be used to choose whether we want to take the test for the first cylinder, the second one, or both. The selector will have 3 positions:
   a) First position (which is the initial position) will be for selecting the first cylinder only, and it will be the default position.
   b) The Second position will be for the second cylinder only.
   c) The Third position will be for both cylinders.

   If you have more than two cylinders, you can easily increase the positions of your selector as we will see later. We will program our selector to do the job we determined by using a ladder program as we will see later in detail. You can find this element in the “HMI & Control Panels” library, and you can get it into your diagram by just going to the “Control” section, and then pressing on “switches” (not opening the tree), and you will find it as shown in the photo below:

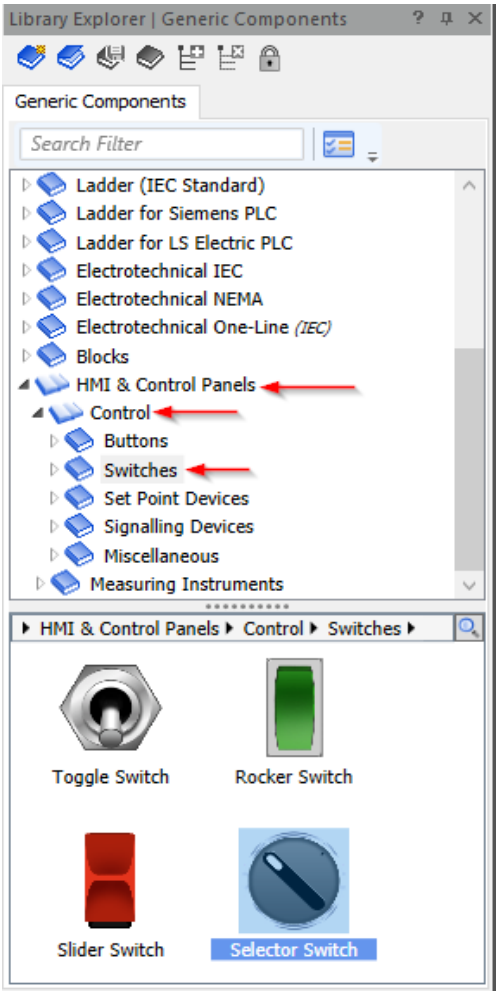

Figure 50: Adding a selector switch in AS software

Figure 49: Selector switch component in AS software

Note that the default number of the selector's positions is two. To change this number to three as we need in our project, we must go to the data section of our component, and change the number in front of "Number of Positions" as shown in the photo below:

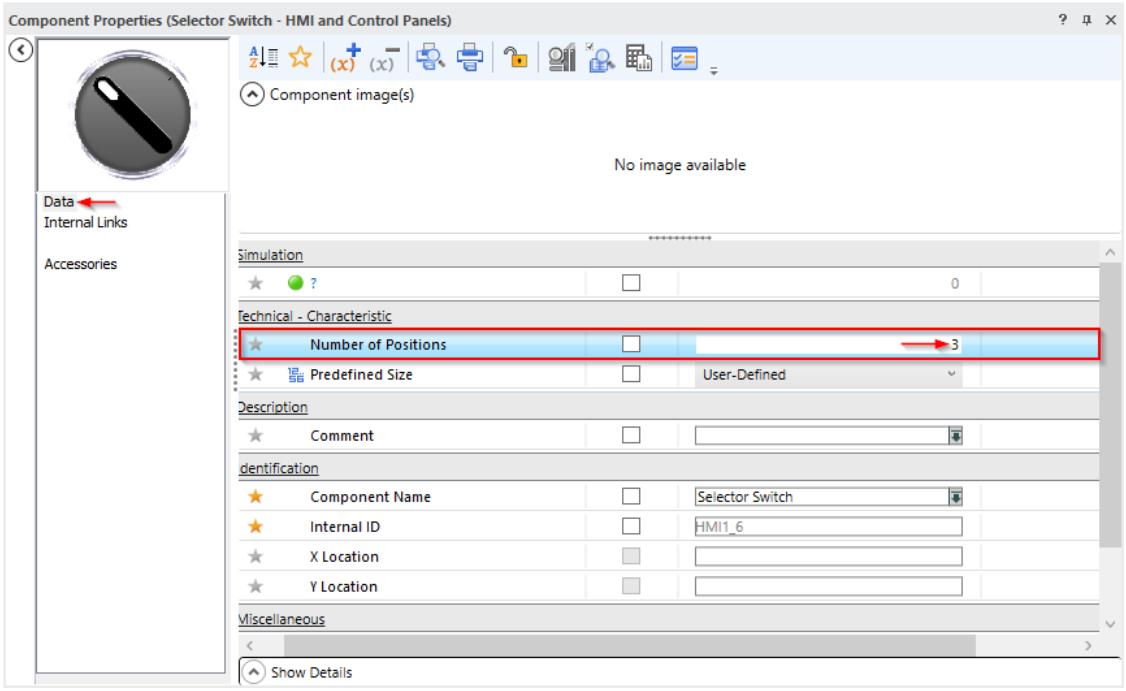

Figure 51: How to determine the no. of the selector positions in AS software

After you change the number, you press the "Enter" button on your keyboard. When you close this window, you will find the positions of the selector changed to three instead of two.

Note that you can change the font, type, and color of the alias (which chosen by you) written over the element in your diagram by just double-clicking on that alias using the left button of the mouse, and you will find a window showing up allowing you to change all of what we mentioned, and more features as shown in the photo below:

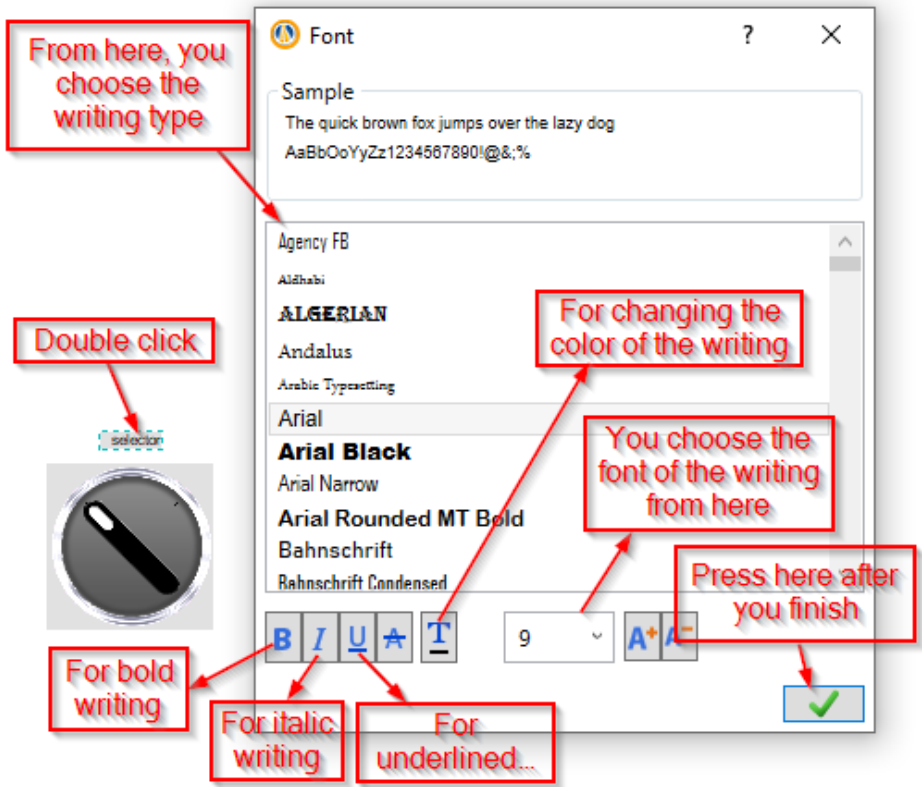

Figure 52: How to change the font, type, and color of the selector alias

You can see in the previous photo an example of how to change the alias of our selector which we named also “selector”. After you finish changing the features you prefer, you must press one time on the right mark shown in the previous photo at the right bottom of the window to save the changes, you can see in the photo below how the writing would be after finishing some features:

Figure 53: The change of the font and color of the selector alias in AS software

3) We will also use 5 pilot lights (for each cylinder), and these lights will be an indication of the five states we have in our system as discussed before in the previous chapter, we will briefly mention these five states here again:
   a) The “green” light (ON_STATE_1), which will be an indication of the on-state for the first cylinder, means that the user is already started the system (ON_OFF pushbutton is pressed), and we are ready to start the test for the first cylinder by pressing on the “WEAR BU”, and this light will be on during the whole operation.
   b) The “white” light (IDLE_STATE_1), which will be an indication of the idle state for the first cylinder, means that the user is already stopped the system (STOP BU is pressed), and the system is now at the idle state (every component is at its initial position or is going back to its initial position).
   c) The “blue” light (WEAR_STATE_1), which will be an indication of the wear state for the first cylinder, means that the system is already at the “on state”, and the user already started the wearing test (making the cylinder doing a determined number of cycles entered by him as we will see later) by pressing on the “WEAR BU”.
   d) The “yellow” light (SEAL_STATE_1), which will be an indication of the seal or leakage test for the first cylinder, means that the cylinder has already finished the wearing test, and automatically moved to the seal or leakage test. It could be also that the system is still doing the wearing test, but the user wanted to move to the seal test manually before the wearing test finishes, so he/she pressed on the “SEAL BU”. Or it could be that the system is at the on state, and the user wanted to start manually the seal test without passing through the waring test.
   e) The “red” light (LEAKAGE_STATE_1), will be an indication of a leakage from the first cylinder, and that is happening when the “pressure transducer” measures a value less than a determined value in our ladder program which is “9.98 bar” during the “SEAL TEST”.

You can find these pilot lights easily in the “HMI & Control Panels” library as shown in the photo below:

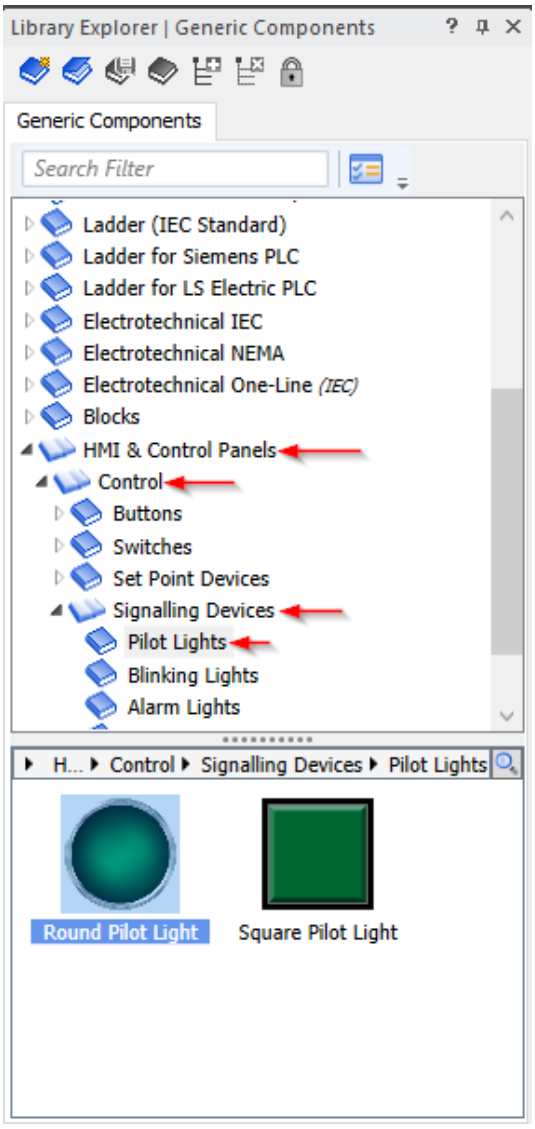

Figure 54: Adding a round pilot light in AS software

Figure 55: Round pilot light component in AS software

We use in our project five “ Round Pilot Light”, you can also use the other type (square pilot light), they will do the same job. As soon as you drag any of these lights to your diagram, you will notice that the small window asking you to write an alias (as we saw with pushbuttons) willn’t show up and you will not be able to write an alias for that component, because it is a signaling device for output, so it must be connected to an output, and it takes the same alias of the output that will be connected with. We will see later that we will connect these lights with the “indicator lights” connected to the simulated PLC we use in our project.

After you have all the needed pilot lights in your diagram, you may need to change the dimensions of any or all of them, it is very easy to do so. You can use the same methods we used to do the same with pushbuttons, after doing that we will have them as shown in the photo below:

Figure 56: Some of the lights in random positions in AS software

Also to make "alignment & equal distribution" for these lights, we can use the same method used to do the same with pushbuttons, and after you do that, you will have them as shown in the photo below:

Figure 57: Aligned and equally distributed lights in AS software

You may need to give each pilot light a certain color as we did in our project. It is very easy to do that, you just must go to the data section of the element as we saw before, then you will find a feature called “component color 4”, which you can use to change the color of your component. The green color is the default one, and you have four other different colors as shown in the photo below:

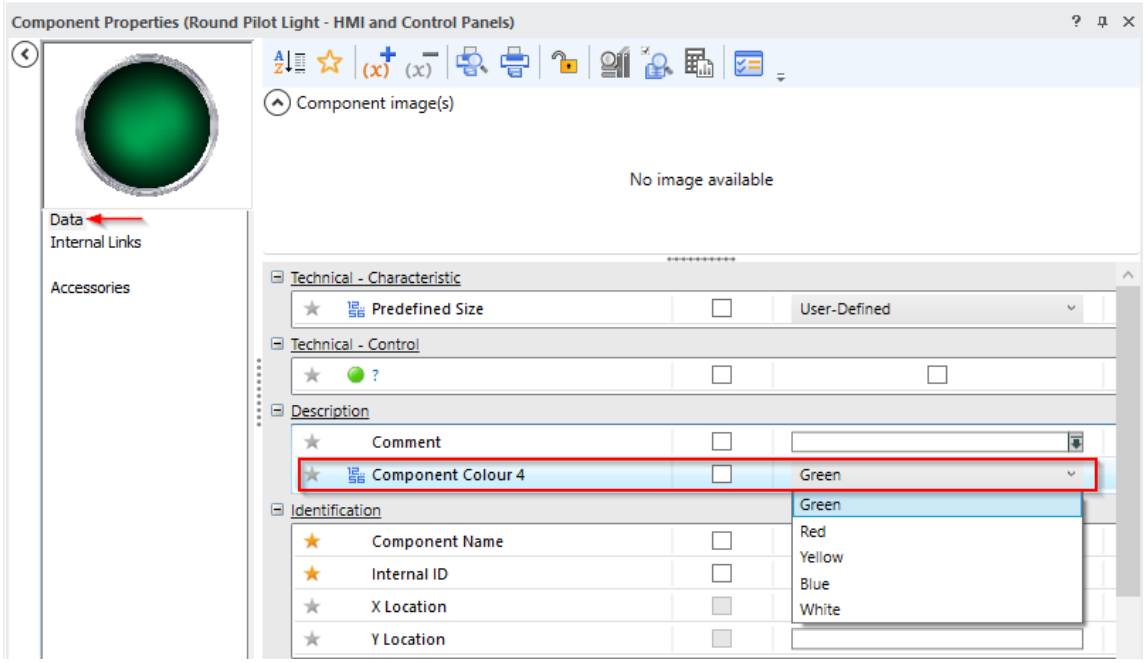

Figure 58: How to change the color of light in AS software

Note that you can do the same for pushbuttons (changing their colors) by using the same method we used here for the pilot lights.

4) We will also need an alarm that will be activated in case we have a very high-pressure drop reading (very high leakage), and we can determine exactly the value of that pressure drop that will immediately activate the alarm by using our ladder program as we will see. We are going to use a "Flashing Beacon Light" to alarm us in case of a very high leakage as we said, and we can also find it in the "Signaling Devices" section as shown in the photo below:

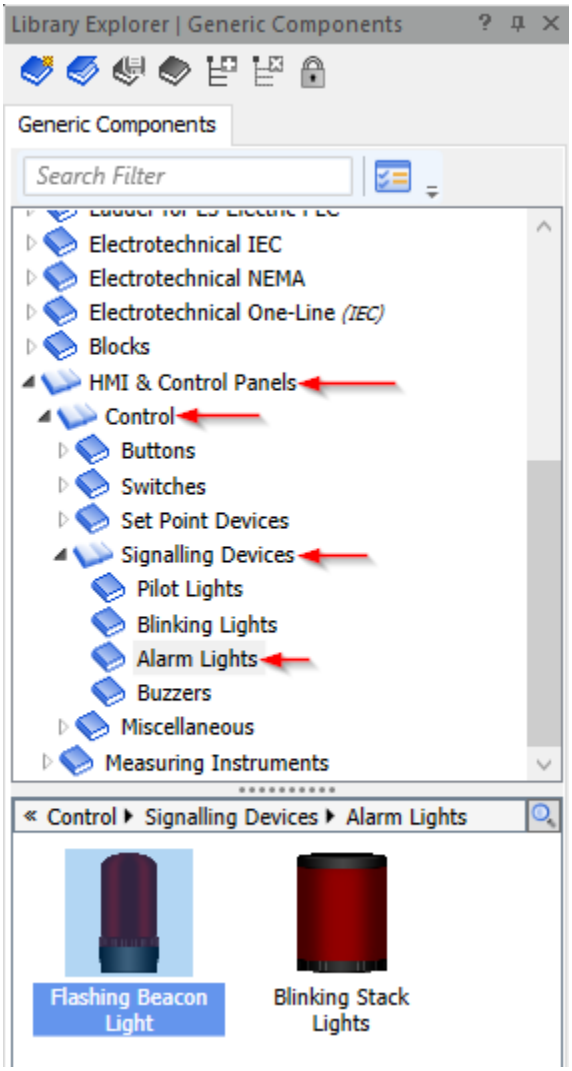

Figure 60: Adding a flashing beacon light in AS software

Figure 59: Flashing beacon light component in AS software

And this element will be connected to an output, and it will take its alias, as we said for the pilot lights, and we will see how to do that later.

5) We will need also a small monitor that allows the user to enter or write the number of cycles he/she prefers for the wear test, and this small monitor is called the “MMI Numeric Command Box” in our SW, and we can find it easily in the “HMI & Control Panels” library as shown in the photo below:

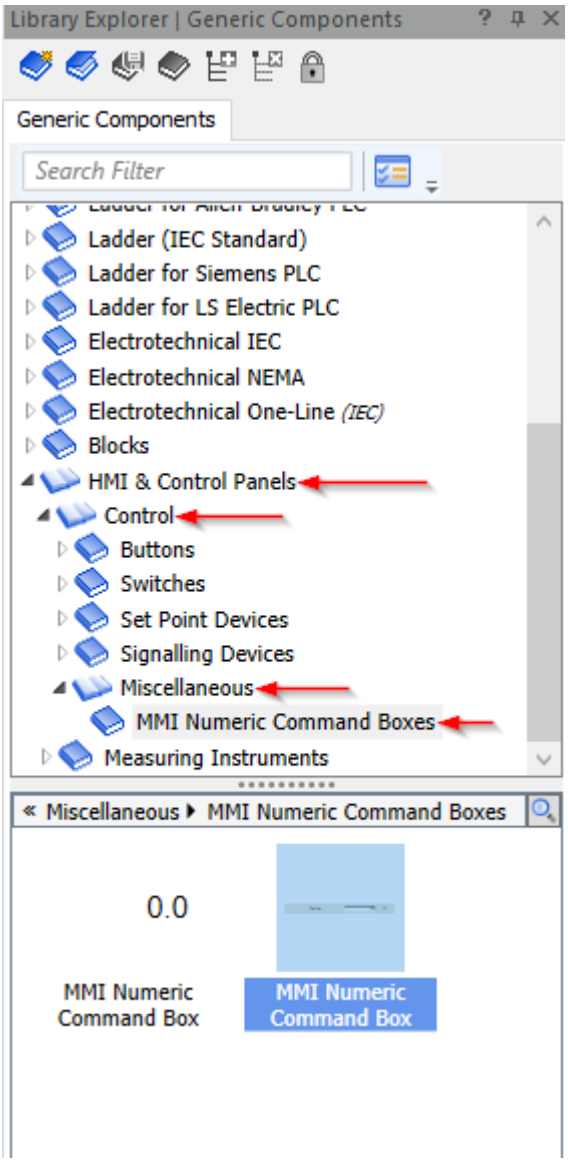

Figure 62: Adding MMI numeric command box in AS software

Figure 61: MMI numeric command box component in AS software

and we will use our ladder program to program our “MMI Numeric Command Box” to do what we just explained, and we will see that later.

6) We will also need another “MMI Numeric Command Box”, but this time it will be used to allow the user to enter or write the time (in seconds in our case) he/she prefers for the seal or leakage test, and we can find it in our SW exactly as we said in the previous step. We may need to change the label of these numeric boxes to allow the user to know what and where to write, and we can do that easily by just going to the element data, and then to the appearance section, and we can find “Displayed Label” with a default value (Value), and “Displayed Unit” with a default value “V”, so we can change them by just doing to these default values, and make a double click using the left button of the mouse, then we will be allowed to write whatever I want, then we just close the window shown in the photo below:

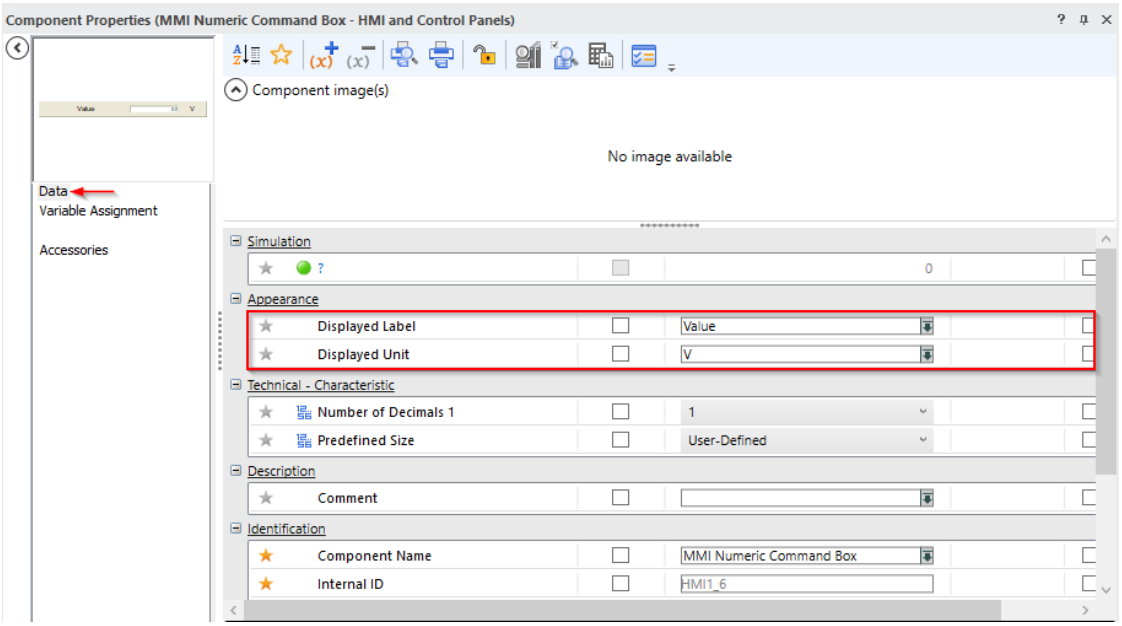

Figure 63: How to display a certain label and unit on the MMI numeric command box in AS software

We changed the label and unit in our project (for the first cylinder) as shown below in the photo, and note that you can leave any of them empty and display nothing like what you can see for the wear cycles as it is just a number, we don’t need a unit:

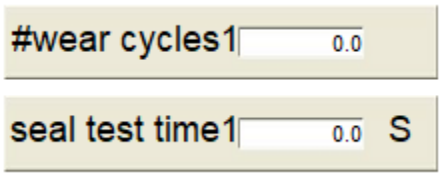

Figure 64: The changed label and unit on the MMI numeric command box in AS software

7) We will also need another two monitors, but that time for displaying the current value for the two “MMI Numeric Command Boxes” (the current wear cycles number and the current seal test time) to let the user know what is going on step by step (this will be programmed using our ladder program), these displaying monitors are called “MMI Numeric Display Box” in our SW, and we can find them in the same library (HMI & Control Panels) as shown in the photo below:

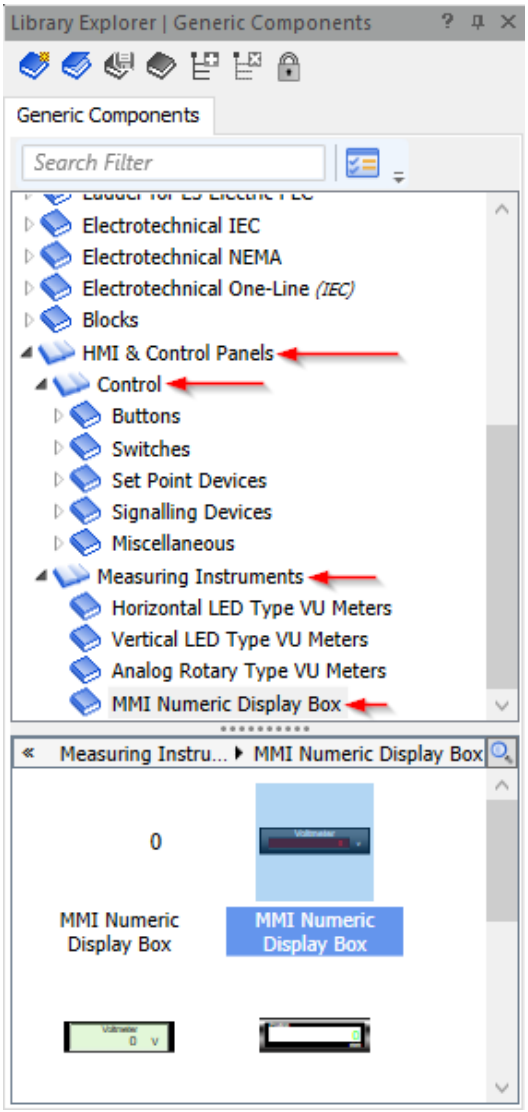

Figure 66: Adding MMI numeric display box in AS software

Figure 65: MMI numeric display box component in AS software

And you are also able to change or remove the “Displaying Label & Displaying unit” by using the same way we used with the “MMI Numeric Command Box”.

8) We will also need to display a very important value which is the measured value by the pressure transducer we use in our electro-pneumatic system, and this value will be programmed to be displayed after each sealing or leakage test so that to let the user know if there is any leakage in our system and how much exactly it is. We will use the same "MMI Numeric Display Box" we used before to display a numeric value of the measured pneumatic pressure, and the numeric box will be programmed as we said to show the value of the pressure after each seal test, and we will see later how to connect this numeric box to our ladder and make it do the needed job. Also, we are going to use another displaying way which will be a percentage of the maximum pressure value which is 10 bar in our project, and it will be displayed like a led displaying column which is called "Led Type VU Meter" in our SW, and we will see below how to find it:

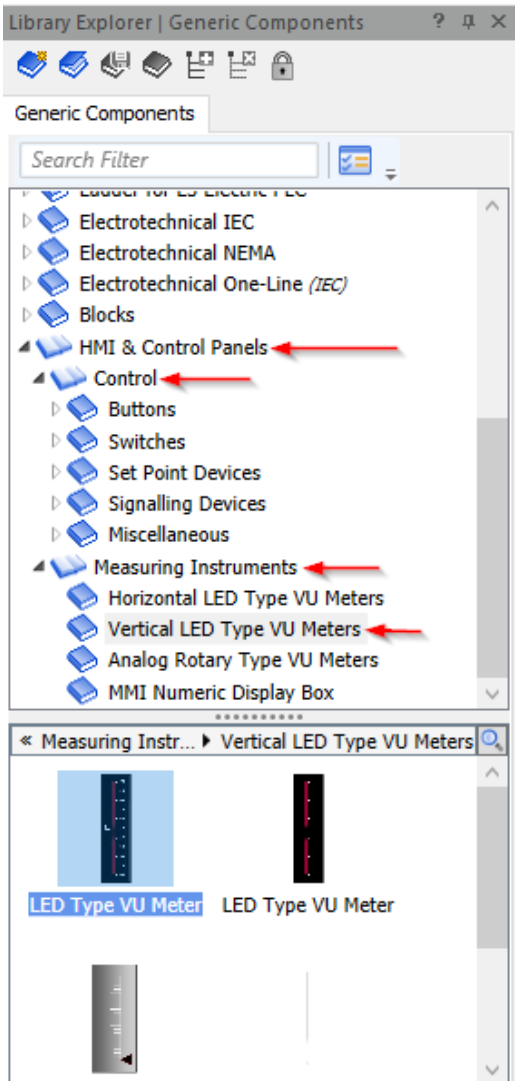

Figure 67: Adding a led type VU meter in AS software

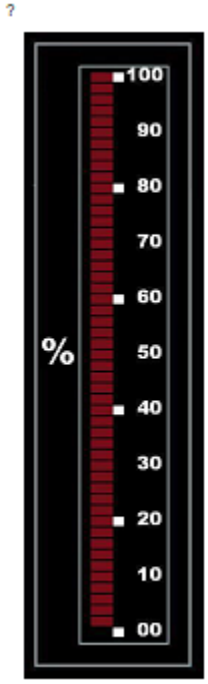

Figure 68: Led type VU meter component in AS software

There is an important step you should do for that “Led Type VU Meter” which is to determine the minimum and maximum variable input which is in our case from “0 bar” to “10 bar” so that to have the right percentage of your measured pressure, and we can do that easily by going to the element data, then we go to “Operating Condition” and change these values to the needed one as shown in the photo below:

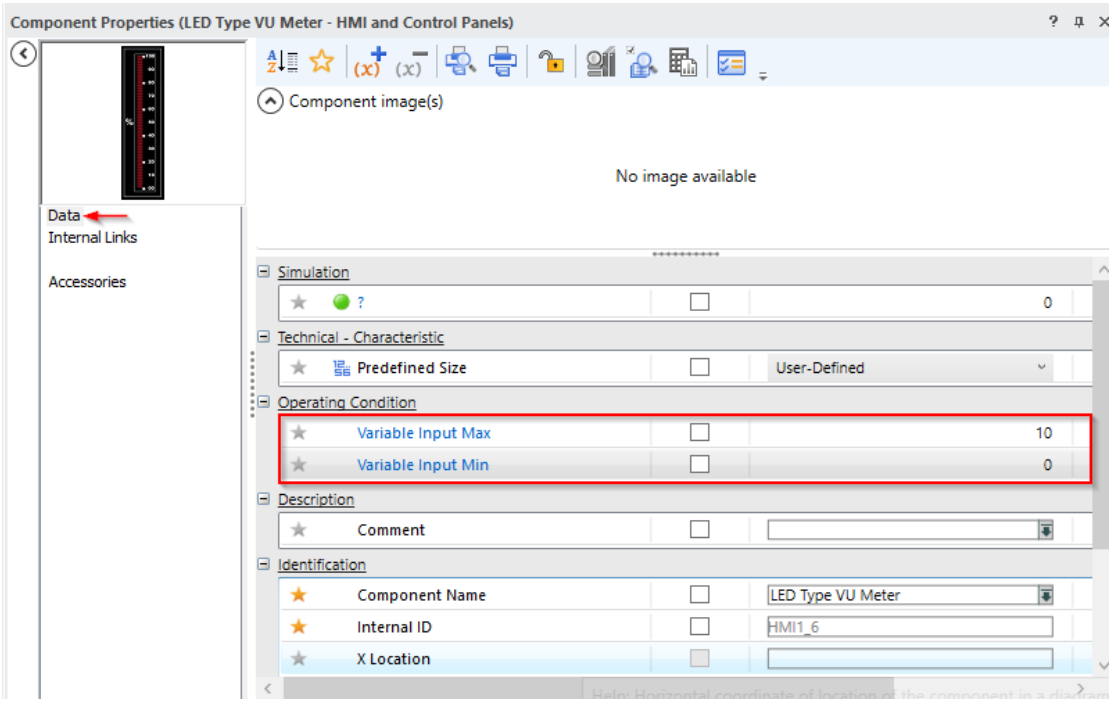

Figure 69: How to choose the min. and max. value for the led type VU meter in AS software

Also, you may need to name your element to let the user know what you are measuring, but you must take into account that this name isn't the alias of your element, it is just something like a comment for the element, and you will find this feature also in the "data" section of the element, then you go to "Description", and write your comment, and to let your comment show up over the element in your diagram, you must check that small box shown in the photo below:

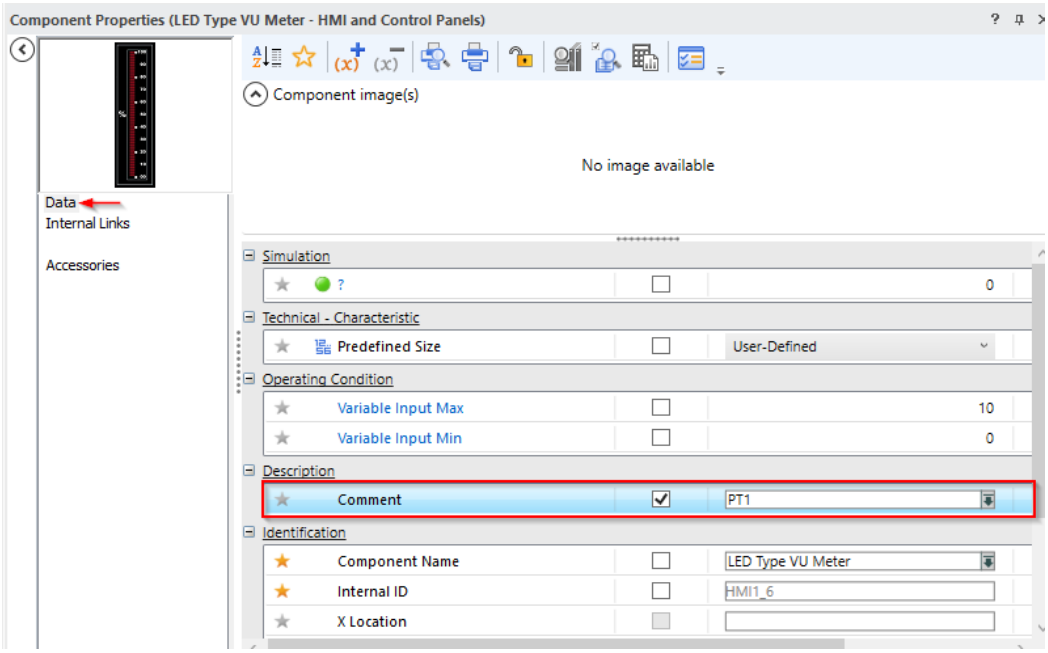

Figure 70: How to write a comment on a component in AS software

You can see in the previous photo that we wrote "PT1" as a comment and checked the small box to let the comment appear in the diagram, so will have the element in the diagram as shown in the photo below (we just changed the font features as discussed before):

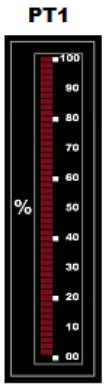

Figure 71: A written comment on the led type VU meter in AS software

9) After doing all the steps discussed before, we will have all the needed HMI & control panel components for our cylinder, and we can do the same steps for the second cylinder, and finally we will have a system like the one shown below photo:

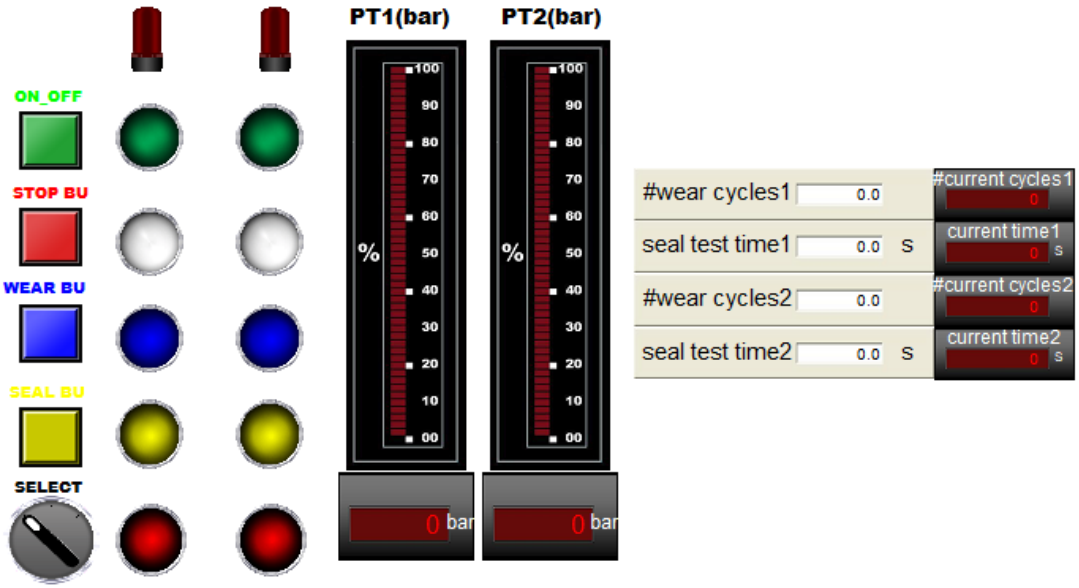

Figure 72: All the components of the HMI system in AS software

Note that we wrote labels ended by “1” for the first cylinder, and labels ended by “2” for the second cylinder, and you can notice that the common components for both cylinders are the four pushbuttons and the selector (which allow us to choose each cylinder will be tested or both of them). Also, you can see in the photo that all the signaling LEDs have no alias, and that is because as we said before the pilot and alarm LEDs will take the same alias as the outputs that will be linked, and that will be explained later.

10) After we finish adding and modifying all the HMI components for our project (for the first and the second cylinder) as we explained, we may need to enhance our a little bit our HMI system for example:
    a) If we want to add these four components (in the photo below) together, i.e. letting them be in one group so that we can deal with them like a single element (but they are still four elements, and not a single block, because we have another option that can make them as a real single block, not just grouped, and we will see that later), for example moving all of them at one time, reducing their dimensions at one time using the same way we discussed before, or doing any other modifications on all of them at once. You can do the same for any other components in your diagram using the same way we will explain.

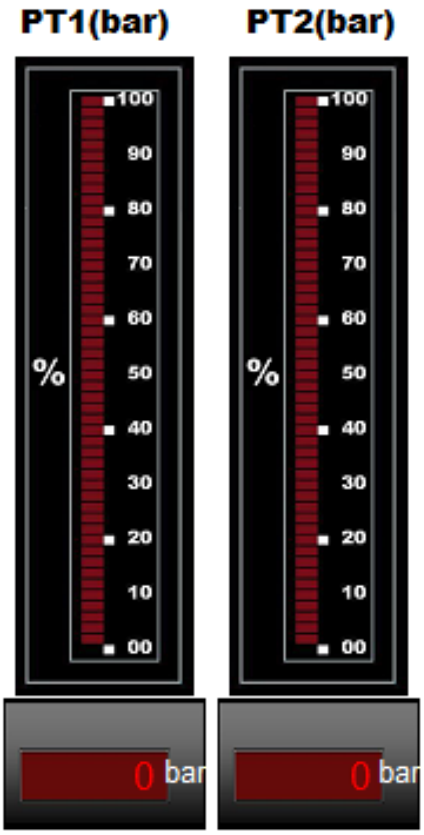

Figure 73: Adding some of the HMI components in one group in AS software

To do that, we have first to group all the needed components to be grouped which are in our case the four components in the previous photo, we can group them as we said before even by using our mouse or using the shift button of our keyboard, then we go the “Edit” section in our toolbar and press one time on that symbol here shown in the photo below which allows you to create a group with the selected objects, and that is all.

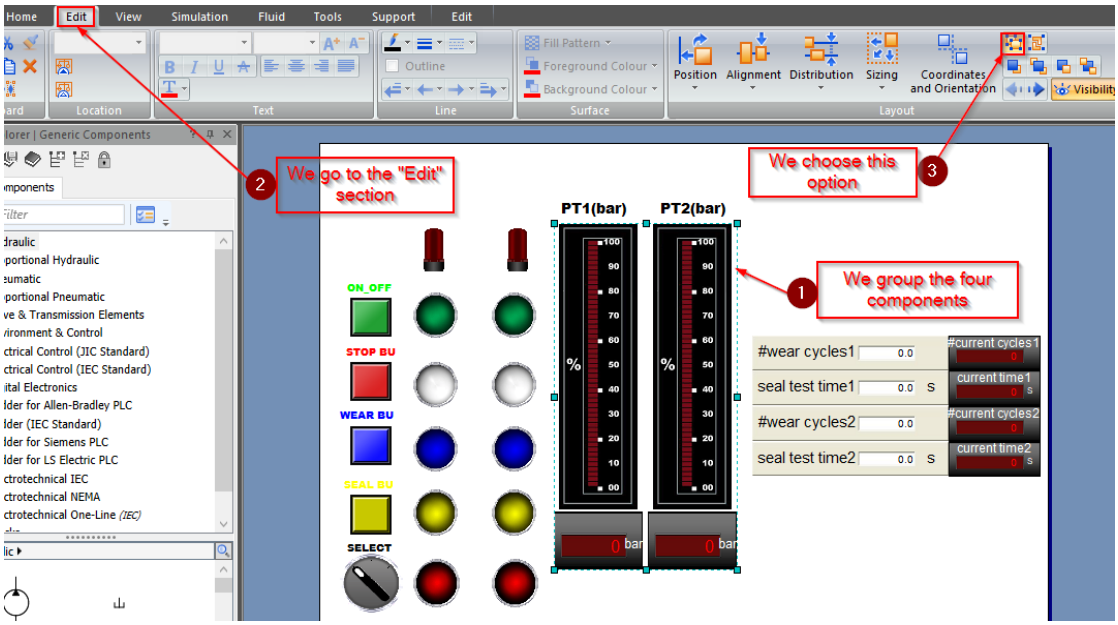

Figure 74: How to Add some of the HMI components in one group in AS software

You may notice that the symbol used to group these components isn’t active until you group more than one component, and also the symbol next to it, and it will be explained later.

b) If we want to let some components be bordered with lines or with a rectangle like they are in a single box, and change the color or thickness of these borders or the color of the background, like what we did in the photo below:

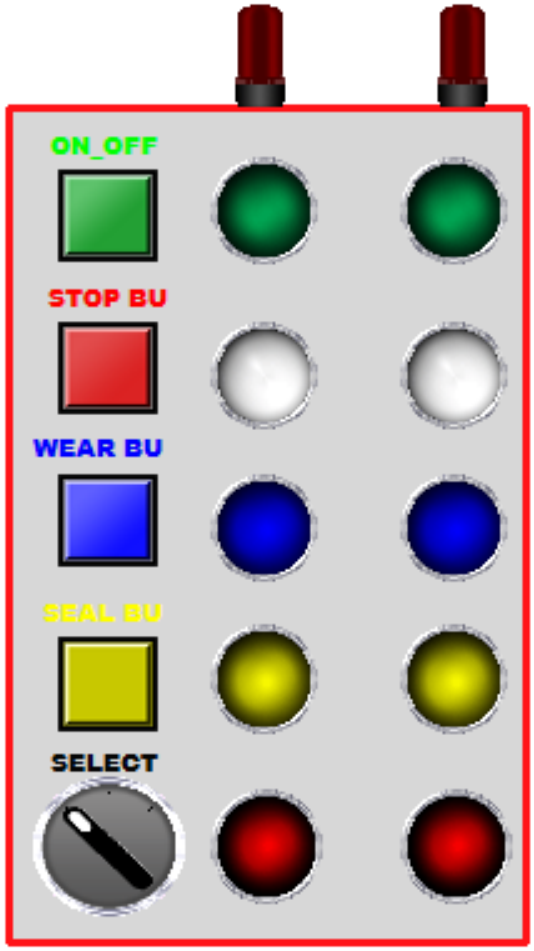

Figure 75: Putting some components of the HMI system in a box in AS software

Also, it is possible to use dashed lines or any kind of lines toucan imagine, and we will see how to do that step by step in detail.

1. We will start first to know how to make borders or put some components in something like a box, and how to change the color of the lines surrounding our components. First, we go to the “Home” section in our toolbar, then we will find a special part for “drawing”, you can choose to draw a line, an arc, an ellipse, a rectangle, and many other shapes, or you can even choose to write a text or to add a photo in your diagram. We will choose “Rectangle” in our case, once you choose it and go to your diagram using the mouse pointer, you will find yourself allowed to draw rectangles, so you can group the components with a rectangle like in the previous photo after arranging them as we explained before. The photo below clarifies what we just explained:

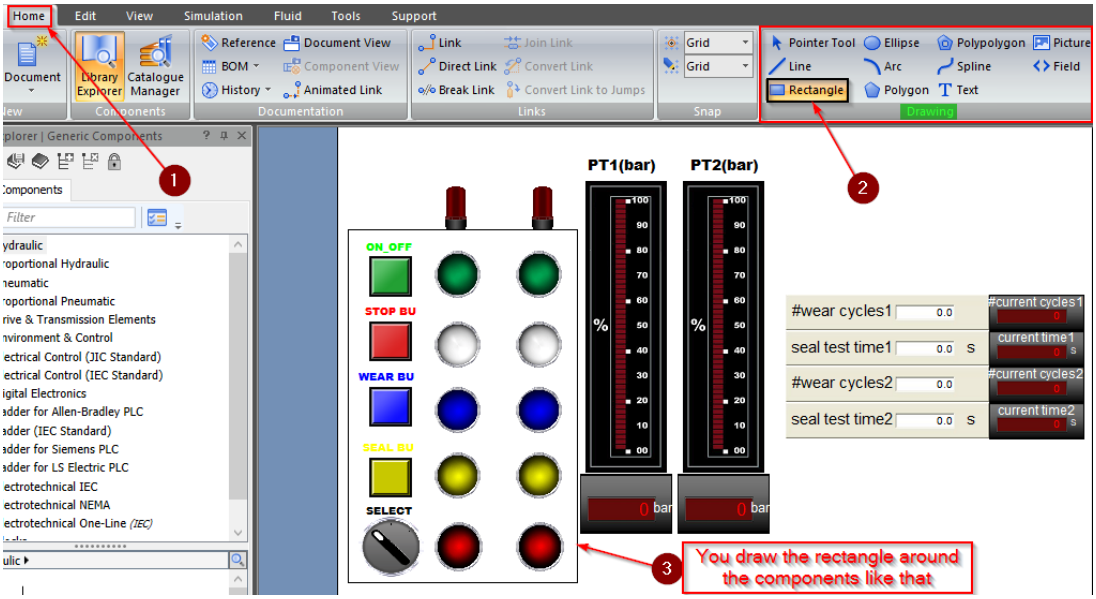

Figure 76: How to Put some components of the HMI system in a box in AS software

You can also change the dimensions of the rectangle you have drawn by just pressing on it using the left button of your mouse and then changing its dimensions easily exactly as we explained for other components like pushbuttons and lights.

2. To change the color and thickness of your lines, we first must highlight our rectangle by pressing on it using the left button of the mouse, then we go to the "Edit" section in our toolbar, and we will find a part related to the "Line" features, and this part will only be activated after we highlight our rectangle. At the top left of that part, we can find an option that allows us to change the color of our rectangle, and exactly next to it another option allows us to change the thickness of our rectangle, if we pressed on any of them using the left button of the mouse, we will see many options that we can choose as shown in the photo below:

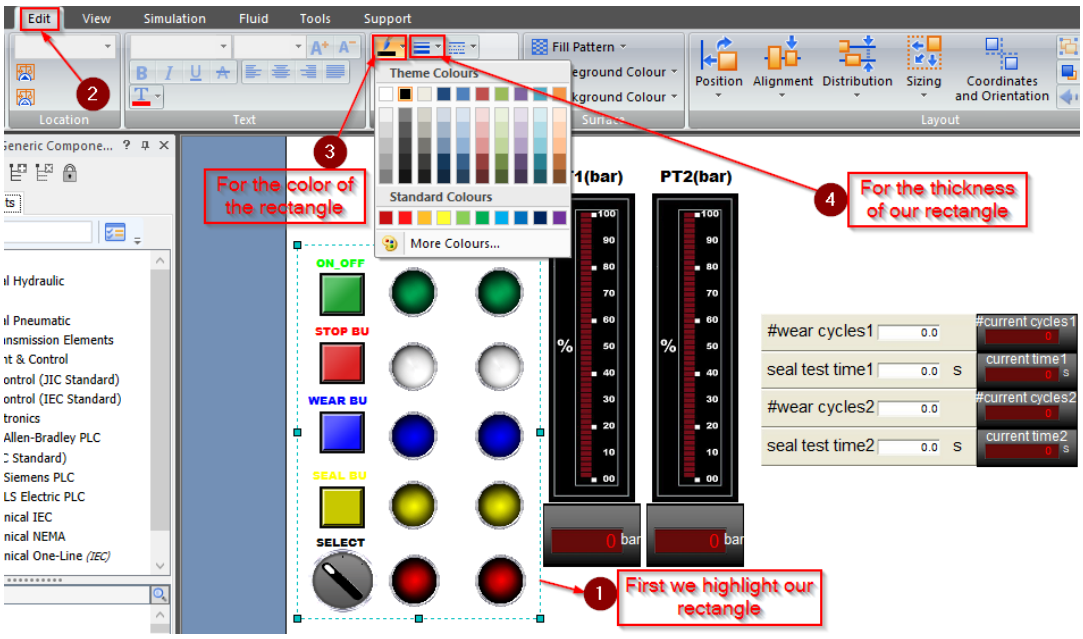

Figure 77: How to change the thickness and color of a box in AS software

We have also another feature which is the line type (if you want it a solid line, dashed, or any other type), and it is next to the other two features clarified in the previous photo (at the top right of the part related to line features).

3. To change the color of the background, you must also highlight your rectangle, and this time we go to the part related to the "Surface" features instead of the part related to the "line" features, also you will notice that these features will not be activated unless you highlighted your rectangle. You will find three features you can change, the first one is "Fill Pattern" which is used to fill your box or rectangle with the pattern you choose, after choosing one of the patterns, you will find that all the components disappeared, so you have to send this pattern to the background as it covers your components, and you can do that by moving to the "Layout" features which are in the same section (edit), and choose the symbol (send to back) ( ) which will move your pattern to the background and allow your components to show up, and you can choose the other two options ("Foreground Colour" & "Background Colour") to change the color of your foreground and background. Note that most of the time we change only the "Foreground Colour" except for some special patterns that allow us to use both features (Foreground & Background). The photo below clarifies more what we just explained:

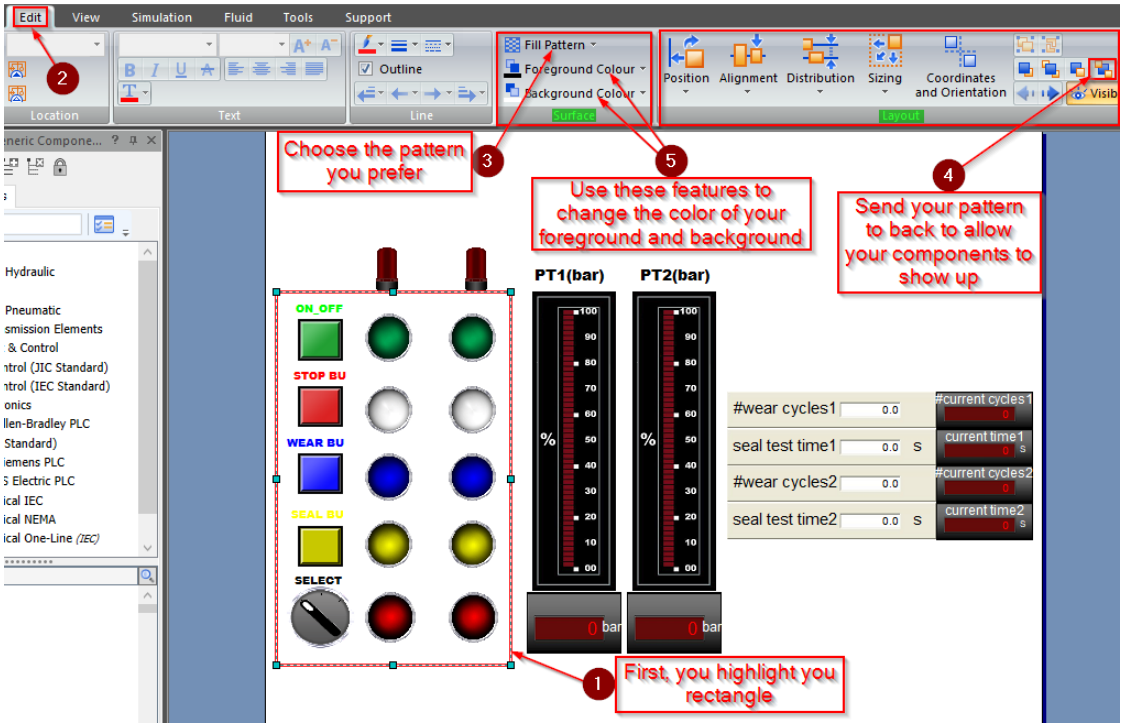

Figure 78: How to change the color of the background of a box of components in AS software

11) After doing all the previous steps as we explained, we now have this “HMI & Control Panel” shown in the photo below:

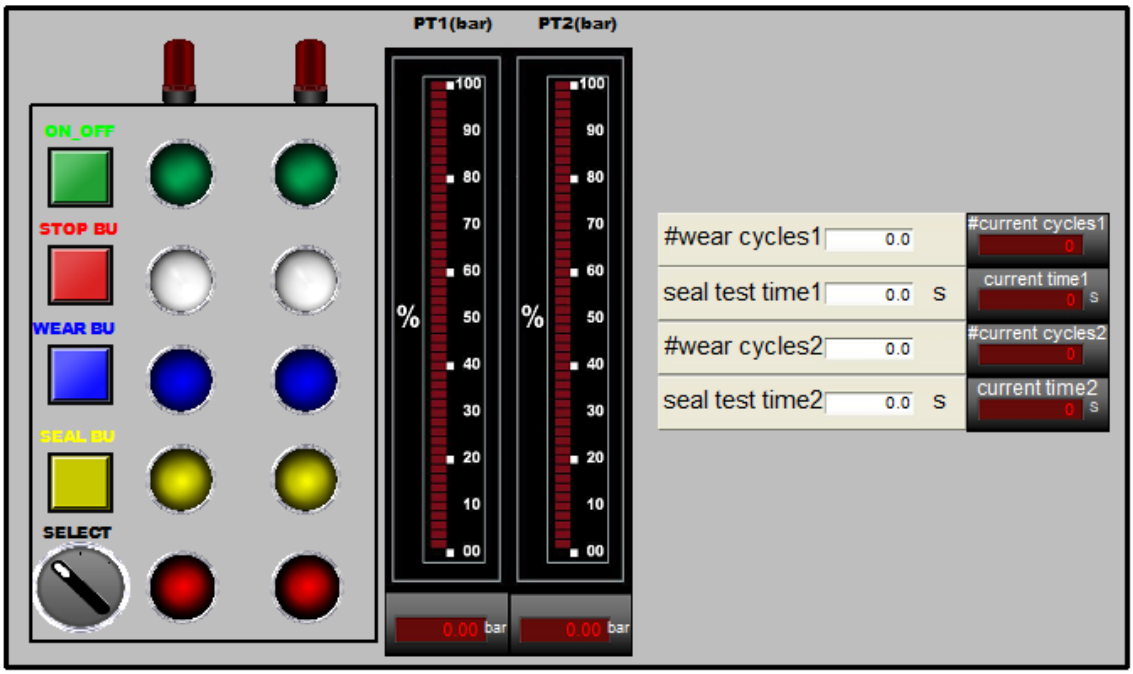

Figure 79: The final version of the HMI system in AS software

Our next step now will be to show how to add a plc to our diagram, how to power it, how to connect it with our components, and how to make a variable assignment with its outputs and our “HMI & control panel” system.

2.2.3 PLC

- In that part, we are going to explain in detail how could we add a “CompactLogix Allen-Bradly PLC” that we are going to use in our project to our diagram since we don’t have the library which contains such PLC devices, so we will show how to get this library, add it to our SW, and use it.
- Also we will see how to make all the connections related to our compactlogix PLC (how to power it up & how to connect our input and output components to it).
- We will explain also how to make a connection (between the components connected to the outputs of our PLC & the similar components we used in our “HMI & control panel” and also in our “Electro-pneumatic system”).
- We can see on the next page a photo of what we will have at the end of this explanation, and we can see that we have four components connected to the inputs of the PLC (from input “0 to 3”) which are the four pushbuttons we have in our system, and they are common between the two cylinders as we already know, also we have sixteen components connected to the outputs of the PLC (from output “0 to 15”), and you can notice that the first eight components are related to the first cylinder (connected to the PLC from output “0 to 7”), and these eight components are divided into six components for the lights used in our system as we saw before and two components for the two solenoids of the 5/3 way electro-pneumatic valve used to control our first cylinder.
- We have other eight components for the second cylinder connected to the outputs of the PLC (from “8 to 15”), and they are the same as the others for the first cylinder, so we are going to explain only the needed steps for the first cylinder, and we can do the same for the second cylinder.[7]

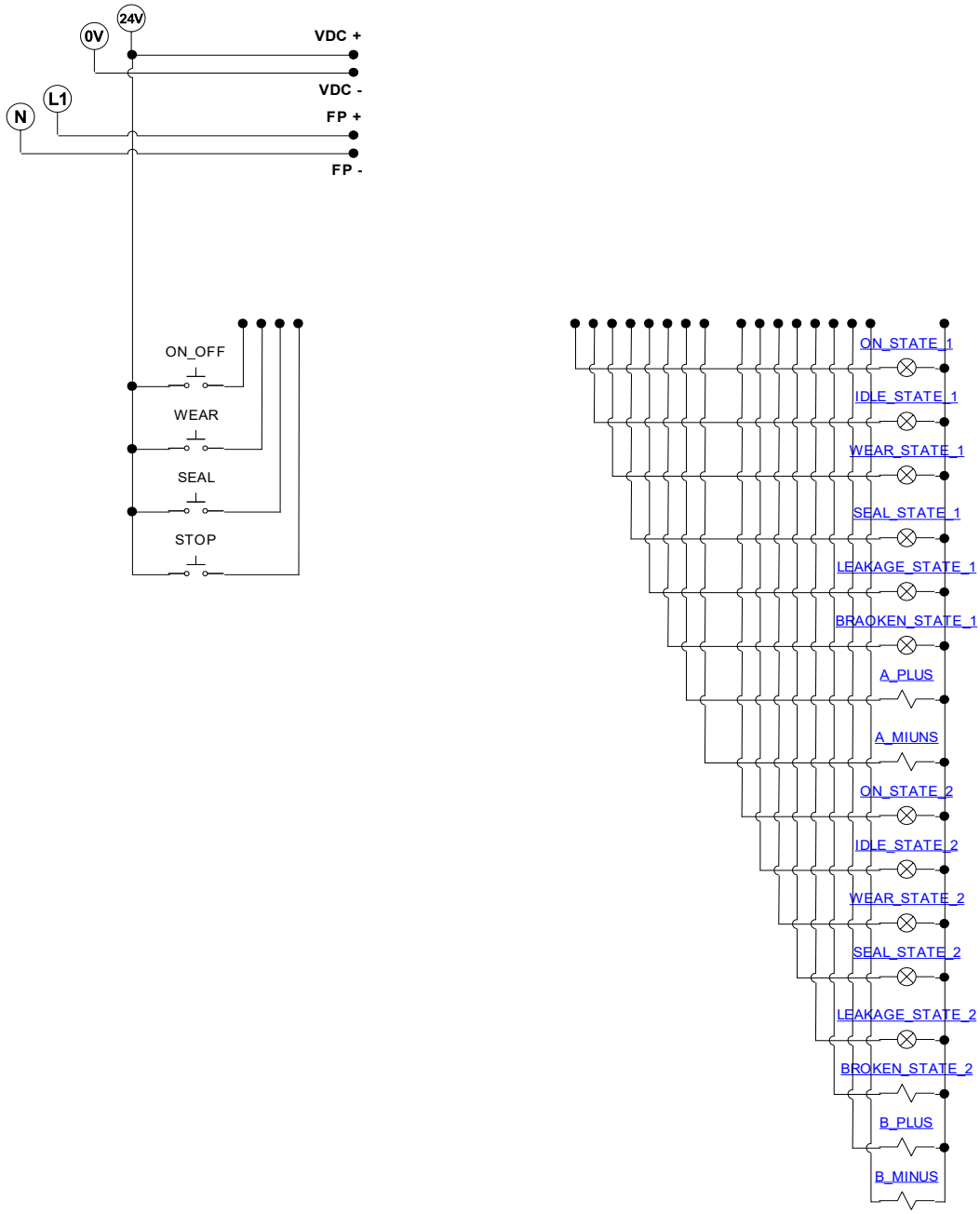

Figure 80: The simulated PLC of the project in AS software

1) First we have to download the PLC library so that we can use it in our SW. To do that, you must be a client of the "Famic Technologies" company, we can do that easily by going to the "Famic Technologies" website using that link (https://www.famictech.com/en/) or by searching on the internet for it, then on the website, we can find all the libraries including the PLC library we need, so download it or ask the support if we couldn't, but for sure you must be a client of the company to be able to download it. After you succeeded to download it, you must copy it to that location (C:\ProgramData\Famic Technologies\Automation Studio E7.0\Libraries), You may have a different location, so what you can do is to open your project on AS, and make "add a new library" ( ) like explained in the photo below, or you can make "open library" ( ), and when you do that, a window will show up contains the library you have, so you can copy this location, and close that window, then use the search for location option in your PC, and copy your downloaded library to that location:

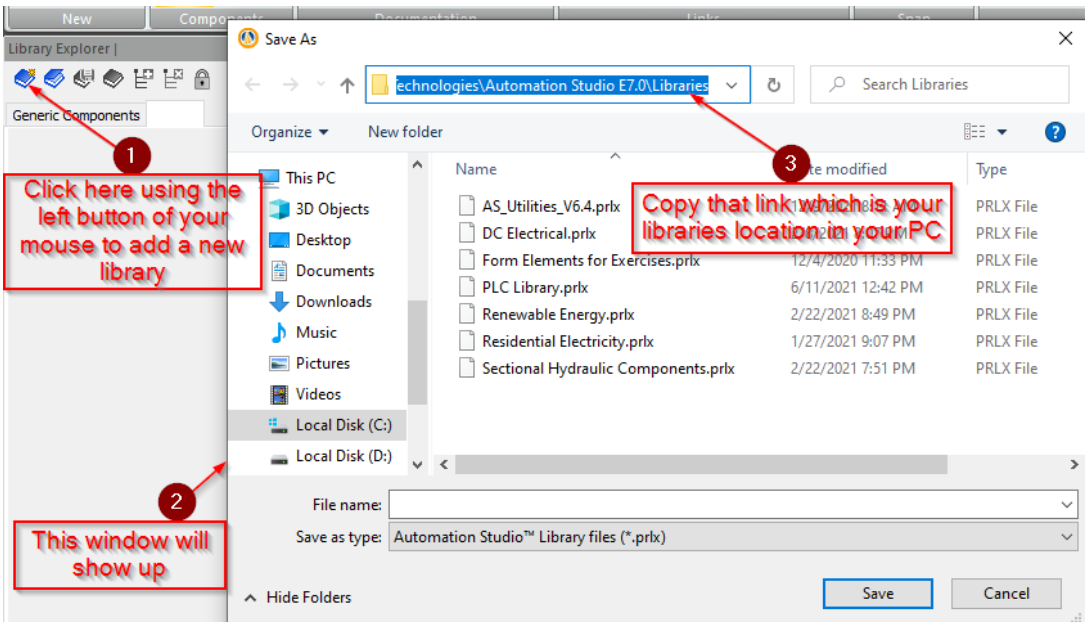

Figure 81: How to get the path of the location where we should put the new downloaded library in AS software

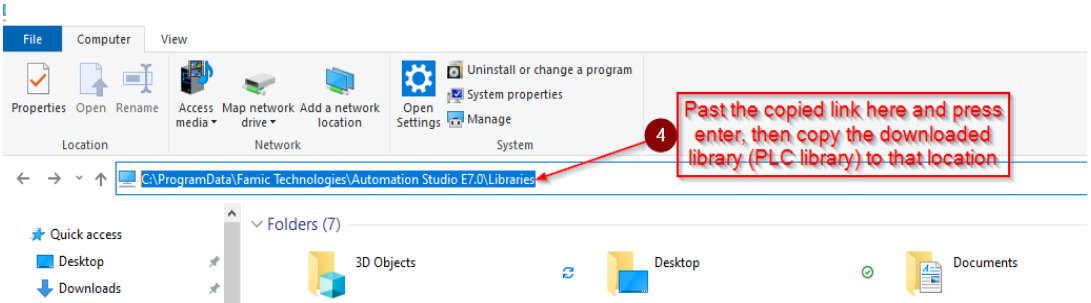

Figure 82: The location where we should add the downloaded library to be used in AS software

2) After downloading the PLC library, copy it to a certain location on our PC as we explained, we want now to open this library to use it in our SW, and to do that, we will make an "open library" ( ) by pressing on that symbol as we did before, then a window will show up which contains all the libraries we have, we can choose the "PLC" library we need to use, then we press on open using the left button of our mouse, or we just make a double click on our library using the same button as shown in the photo below:

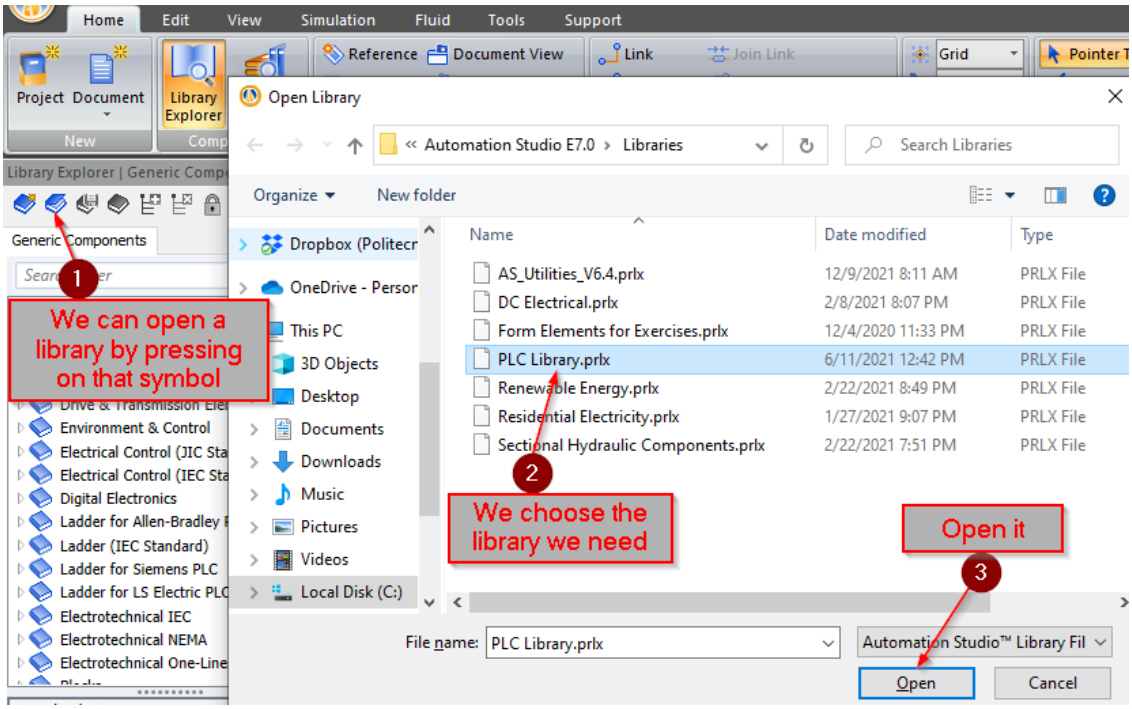

Figure 83: How to add a new library (PLC library) to AS software

After doing these steps, our new library will open immediately, and you will now have two main libraries in your "Library Explorer" section which are (the "Generic components library" & "PLC library"). If you would like to close that library you just opened, you must press "Close Library" ( ). Note that if you go by your mouse pointer to any of these symbols, the SW will immediately show you what these symbols do, or even any other part in your SW window as we explained before.

3) After opening the PLC library, we will find many types of PLC manufacturers like (Siemens, Allen Bradley, Koyo, ..etc), and also we will have two kinds of PLC:

1. **A modular PLC** → is a PLC built with several components that are plugged into a common rack or bus with extendable I/O capabilities. It contains a power supply module, CPU, and other I/O modules that are plugged together in the same rack, which are from the same manufacturers or other manufacturers. These modular PLCs come in different sizes with variable power supply, computing capabilities, I/O connectivity, etc.
2. **An integrated or Compact PLC** → is built by several modules within a single case. Therefore, the I/O capabilities are decided by the manufacturer, but not by the user. Some of the integrated PLCs allow connecting additional I/Os to make them somewhat modular. We will use an "Allen Bradley" PLC of type "compact" in our project (CompactLogix_L18ER_FP_PLC), and we can get it in our diagram easily as shown in the photo below:

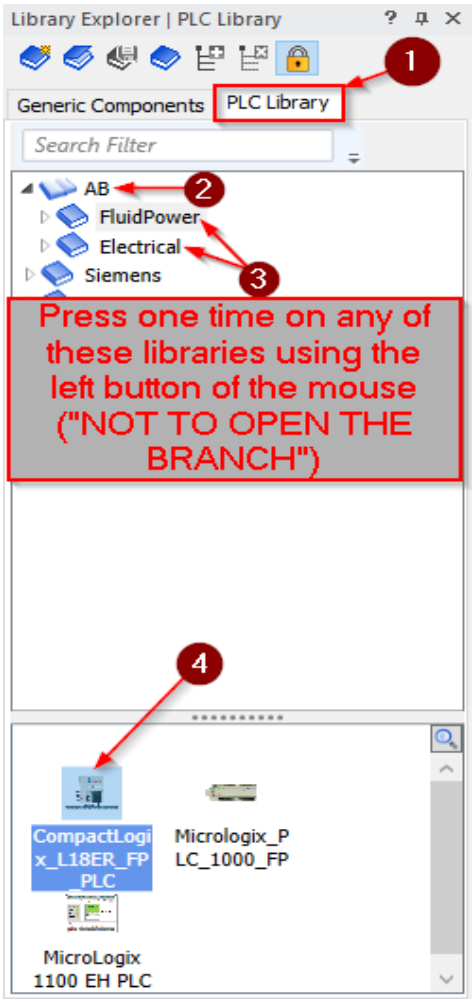

Figure 84: How to add the compact Logix Allen Bradley PLC to AS software

4) After we dragged our PLC to our diagram, we will explain now how to power it, and how to connect it with the needed I/Os in our project. We will use the "Electrical Control" library to make all the connections with our PLC, as it has all the needed components, and we will find two different types of this library (two standards):

1. Electrical Control "JIC standard" → This library contains specific component symbols which will be used in the fluid power electrical control circuits and other circuits, and these symbols are standard "JIC" (Joint "Industrial" Council) symbols as approved and adopted by the NMTBA (National Machine Tool Builders Association).
2. Electrical Control "IEC standard" → This library contains specific component symbols which will be used also in the fluid power electrical control circuits and other circuits, and these symbols are standard "IEC" (the International Electrotechnical Commission) which is an organization that prepares and publishes international standards for all electrical, electronic, and related technologies. You can use any of them, but in our project, we are going to use the "JIC" standard:

a. We will start now with powering our Allen Bradley PLC, and we will use a line and neutral to do that, and also we will power it with 24 volts to energize our inputs. We can get these power symbols from the “Power Sources” section inside the “Electrical Control (JIC standard)” library, and we connect them with PLC like any other components as shown in the photos below:

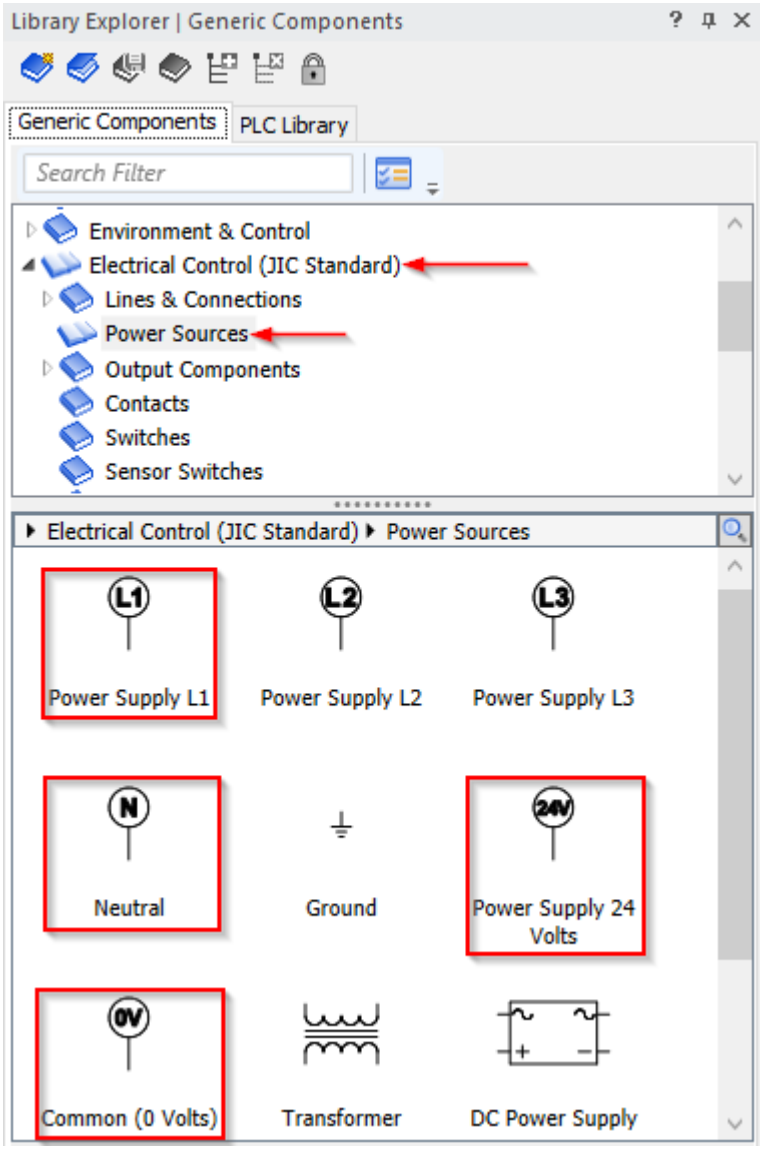

Figure 85: How to add the components needed for powering up the compact Logix Allen Bradley PLC to AS software

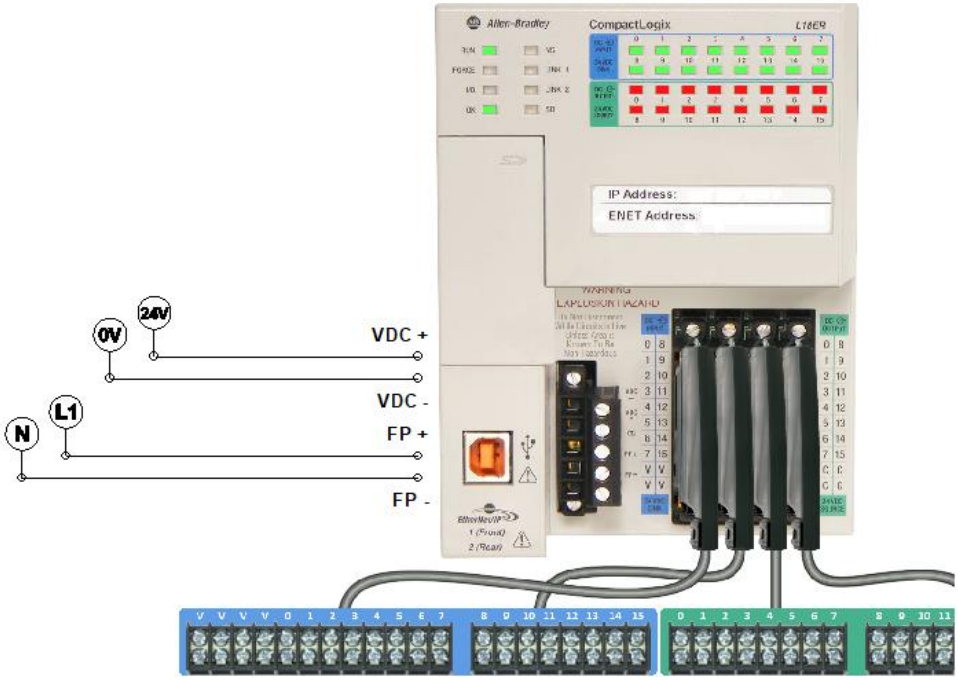

Figure 86: How to Power up a compact Logix Allen Bradley PLC in AS software

As you can see in the previous photo, these are the needed connections to power up our PLC and energize our inputs, and to make these connections, you can do exactly as we explained before to connect any components.

b. As we said before, we need to connect our PLC input card with four normally open push-buttons, and we will find them in the "switches" section inside our "Electrical Control (JIC)" library as we can see in the photo below. These switches will be connected as shown in the second photo below, we must connect one side of them with the 24 volts to be energized, and the other side of them will be connected to the PLC input card from 0 to 3, as we have four push-buttons:

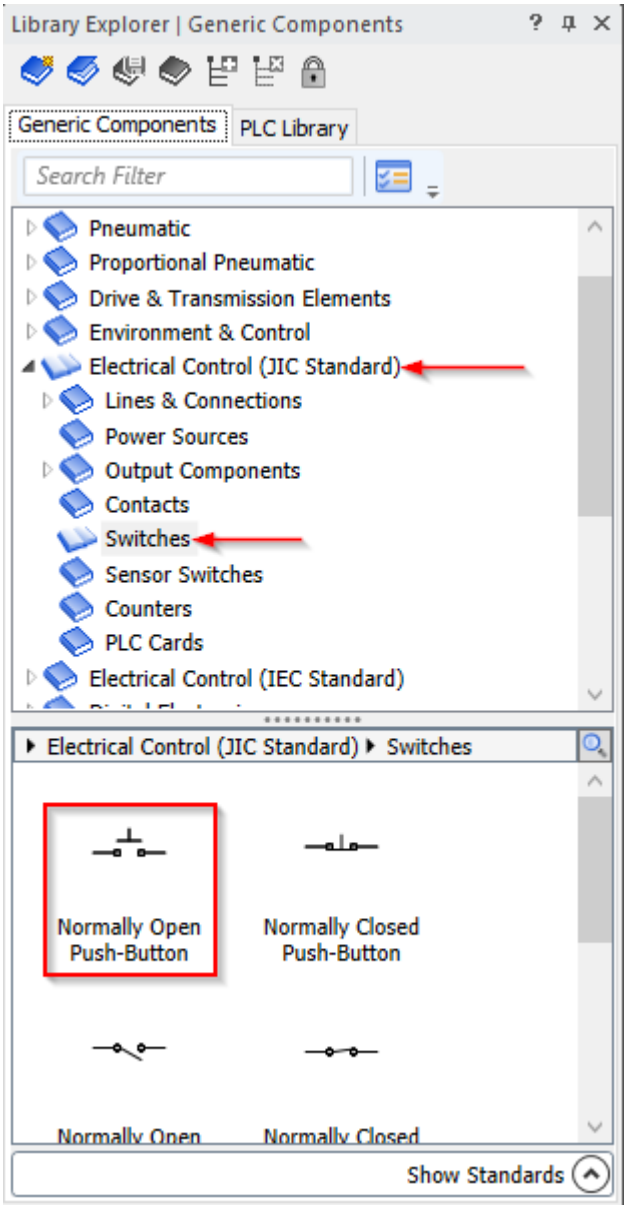

Figure 88: Adding a normally open push button in AS software

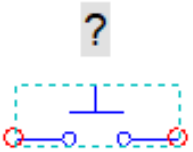

Figure 87: Normally open push button component in AS software

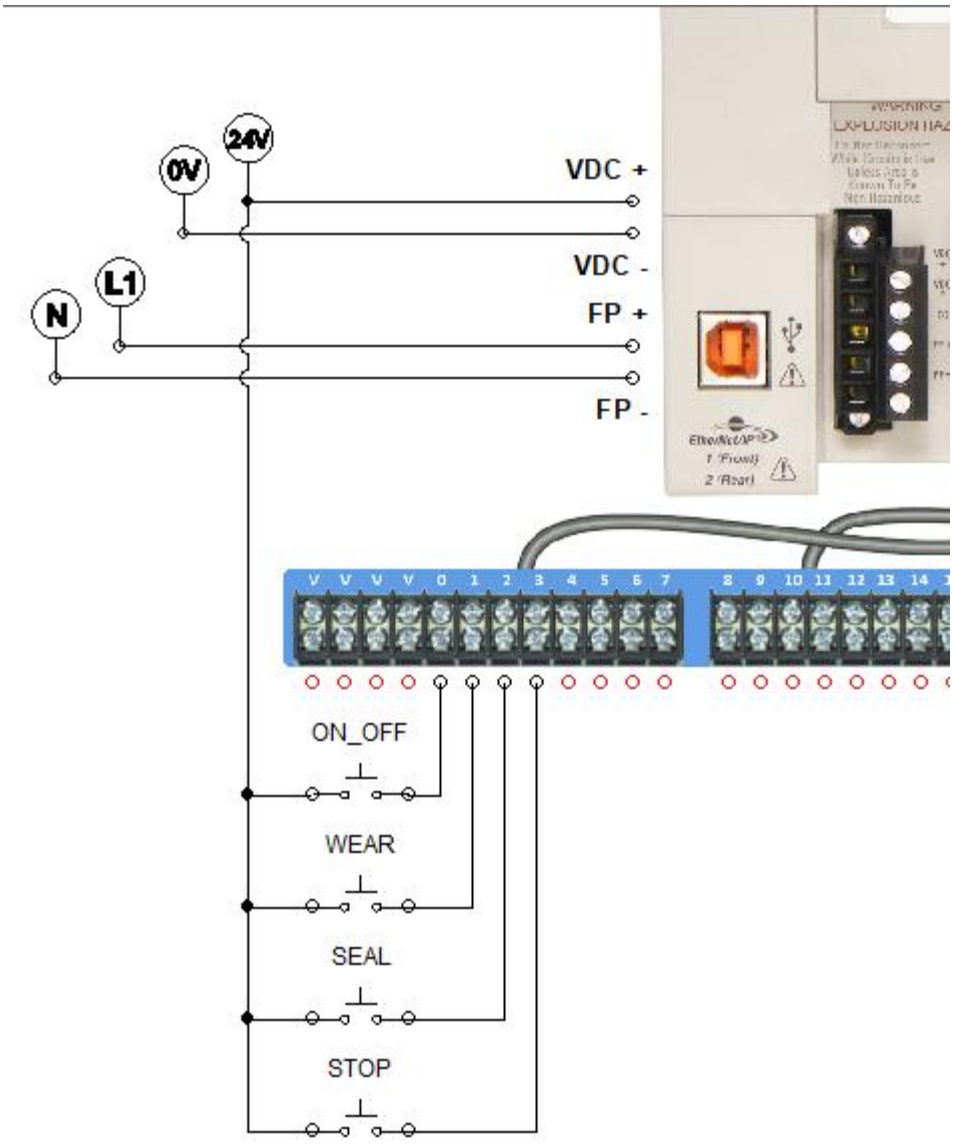

Figure 89: How to connect buttons to the compact Logix Allen Bradley PLC in AS software

As shown in the previous photo, we gave our normally open push buttons almost the same aliases as the others in the HMI & control panel system, and that is better so that you didn't be confused if you have a more complex system with so many components, so it is better to get used in doing that with the similar or connected components.

c. As we also said before, we need eight components for each cylinder to be connected to our PLC output card, and these eight components will be divided as we said before to:

   1. Six “Signalling Devices” will be communicated with (the five pilot lights and the flashing beacon light) of our control panel system as we will see later.

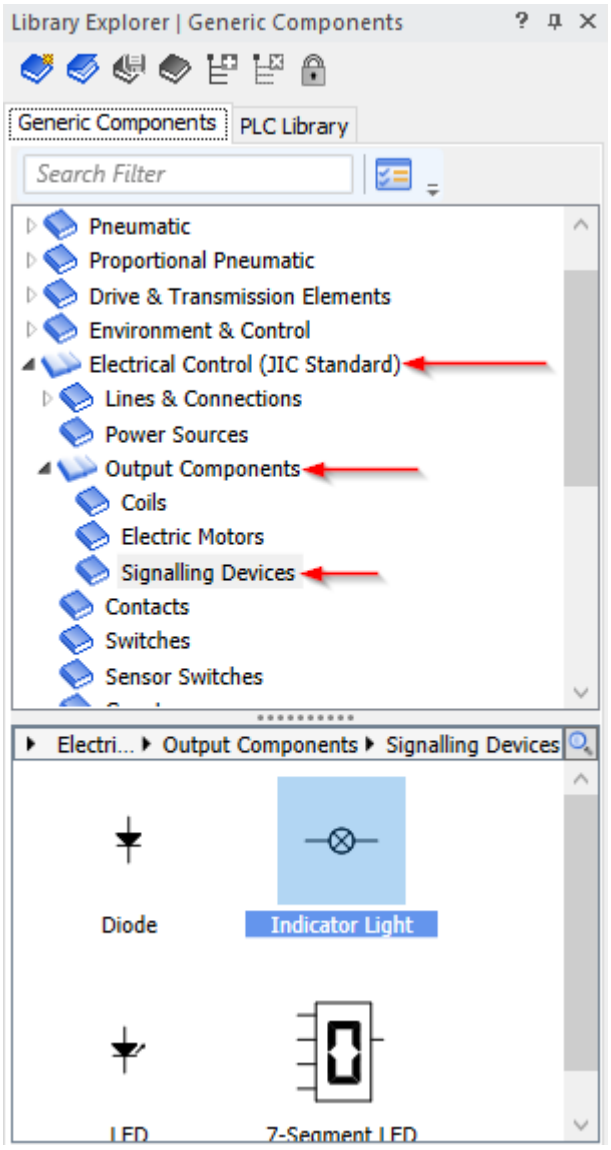

Figure 91: Adding an indicator light in AS software

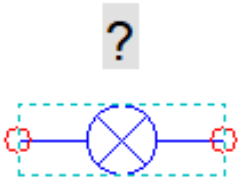

Figure 90: Indicator light component in AS software

2. Two solenoids will be communicated with the other two solenoids of the 5/3-way valve in our electro-pneumatic system. These components can be found in the “Output Components” inside our “Electrical Control (JIC)” library as shown in the photos below:

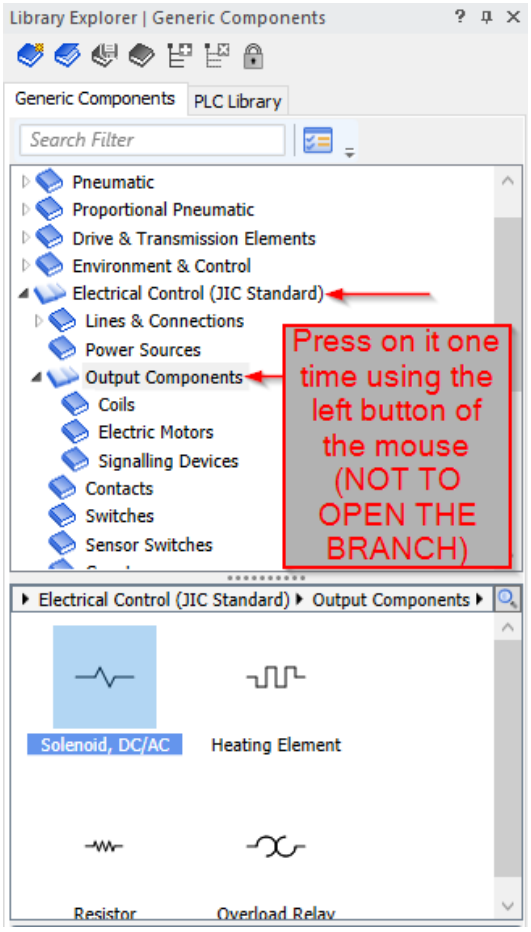

Figure 93: Adding a solenoid (DC/AC) in AS software

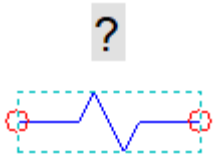

Figure 92: Solenoid (DC/AC) component in AS software

After we got all the needed components in our diagram, we can arrange them, equally, distribute them, or do whatever to enhance our design using the "Layout" options in the "Edit" section of our "Toolbar" as we explained before so that we can have all the components arranged and connected in a good shape as we can see in the photo below:

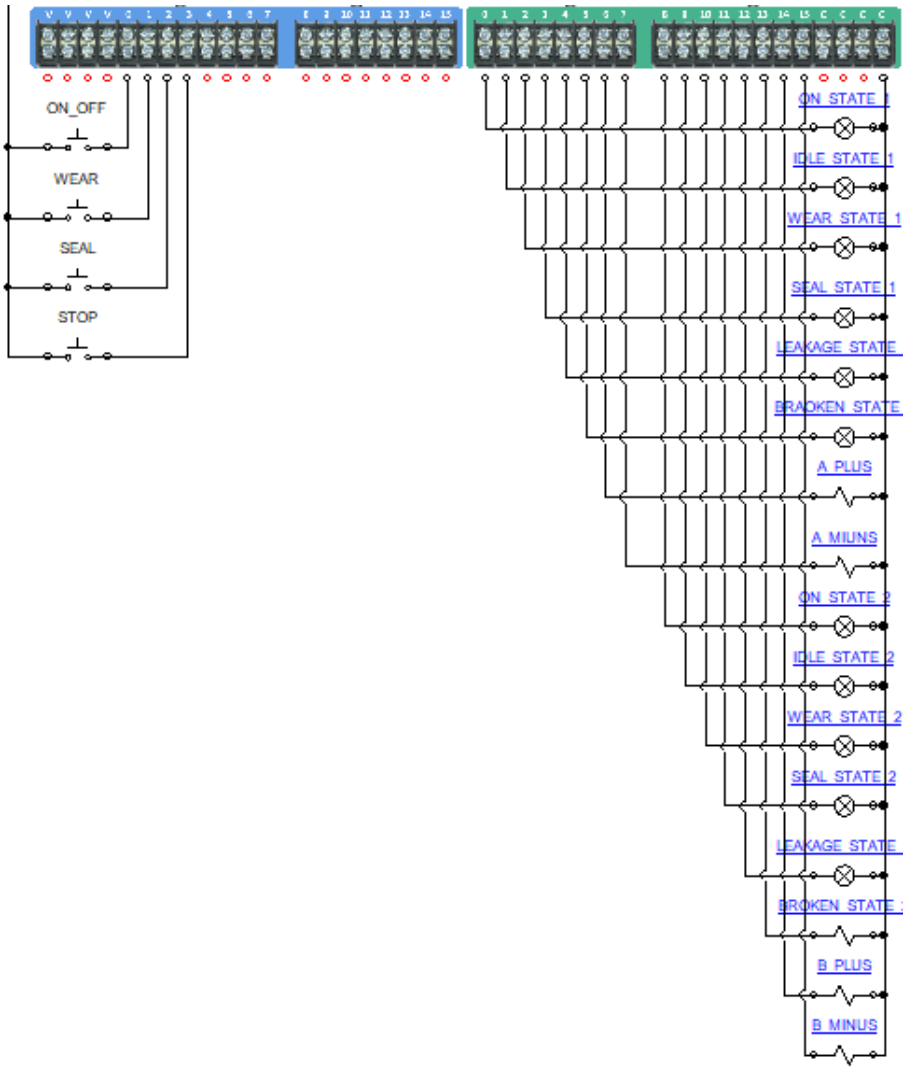

Figure 94: The complete connection of the compact Logix Allen Bradley PLC in AS software

:

You will not have the aliases of your output components in blue like in the previous photo, because they aren't assigned yet to their similar components in the HMI & control panel system (they aren't communicated with each other), we will see in the next step how to make this assignment.

5) Now we will see how to make a variable assignment between the components connected to the PLC output card and their similar components in the HMI & control panel system:

1. For the lights:

   a. we must go to our "HMI & Control Panel" system, then we double click (using the left button of the mouse) on one of the lights, for example, the green one used for the "on state" of the first cylinder as shown in the photo below:

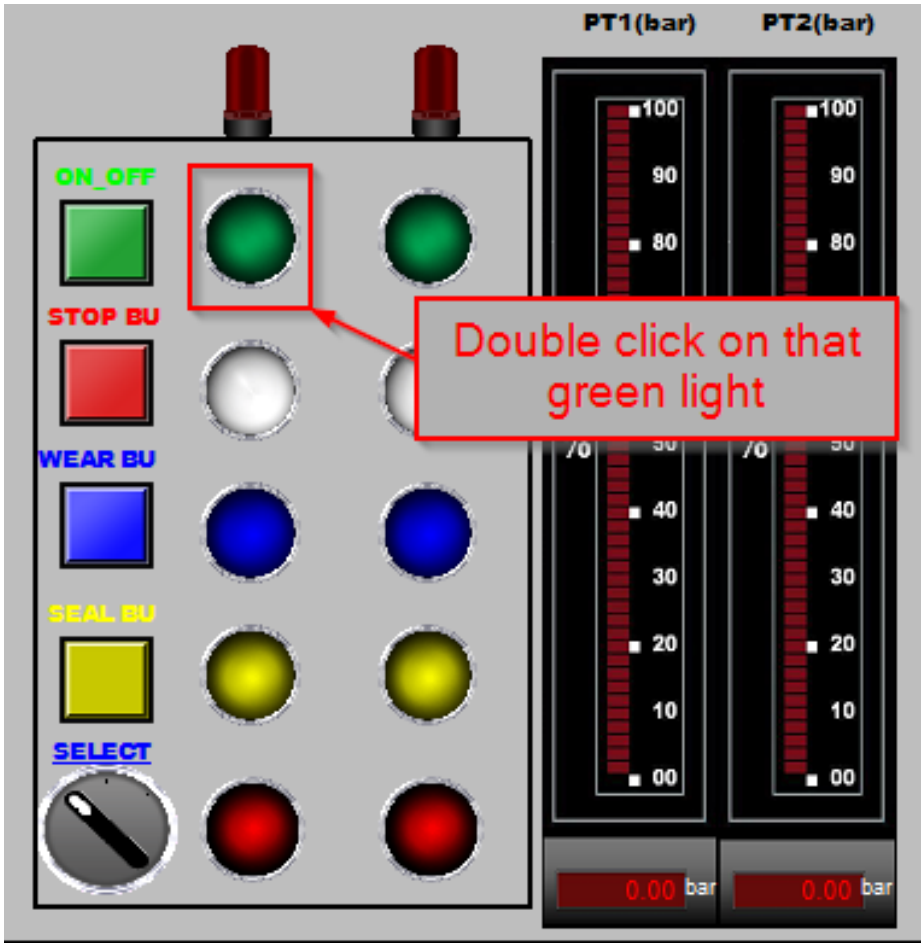

Figure 95: selecting the component needed from the HMI system to be assigned with the corresponding element from the components connected to the PLC in AS software

b. A window will show up, on the left of that window, we must choose “Internal links”, then in the “Compatible Simulation Variables” section, we search for the component connected to our PLC (indicator light) that we want it to be linked to the “green pilot led”, and we will do that search using the alias of the component which is “ON_STATE_1” in that case. After we found it, we must highlight it, and then create a read or write association using that symbol ( ) as shown in the photo below:

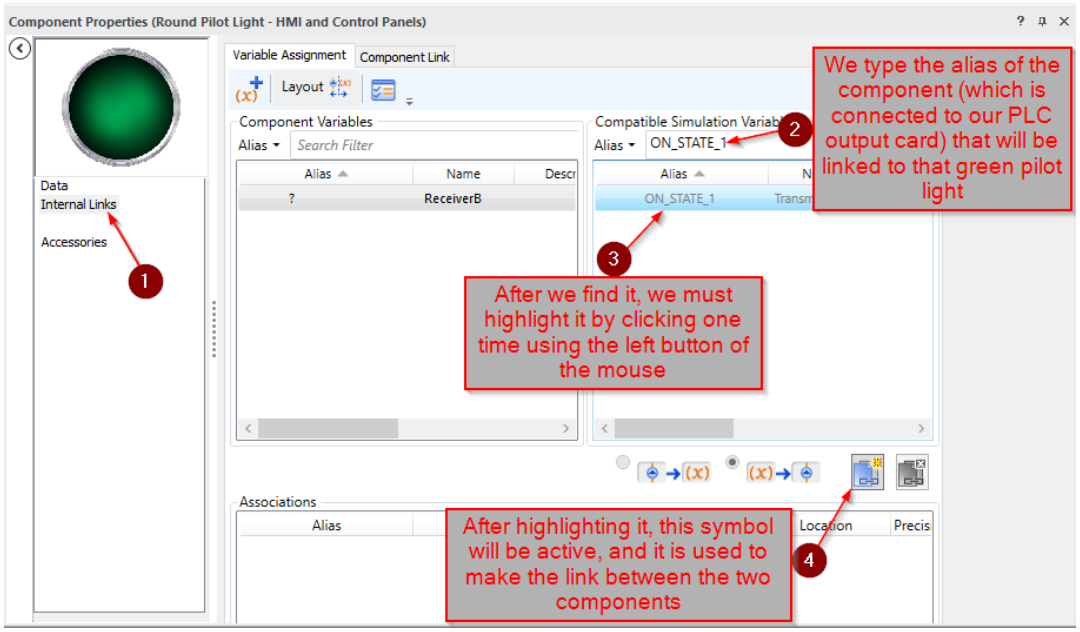

Figure 96: How to make an assignment or communication between two components in AS software

c. After doing the previous step, the link will be established and shown in the "Associations" section, and also you can see that the alias of the linked component (connected to the PLC) is now in blue, and the same alias will be written over the "green pilot light" in your HMI & control panel system as shown in the photos below, and this means that the linkage is already established successfully:

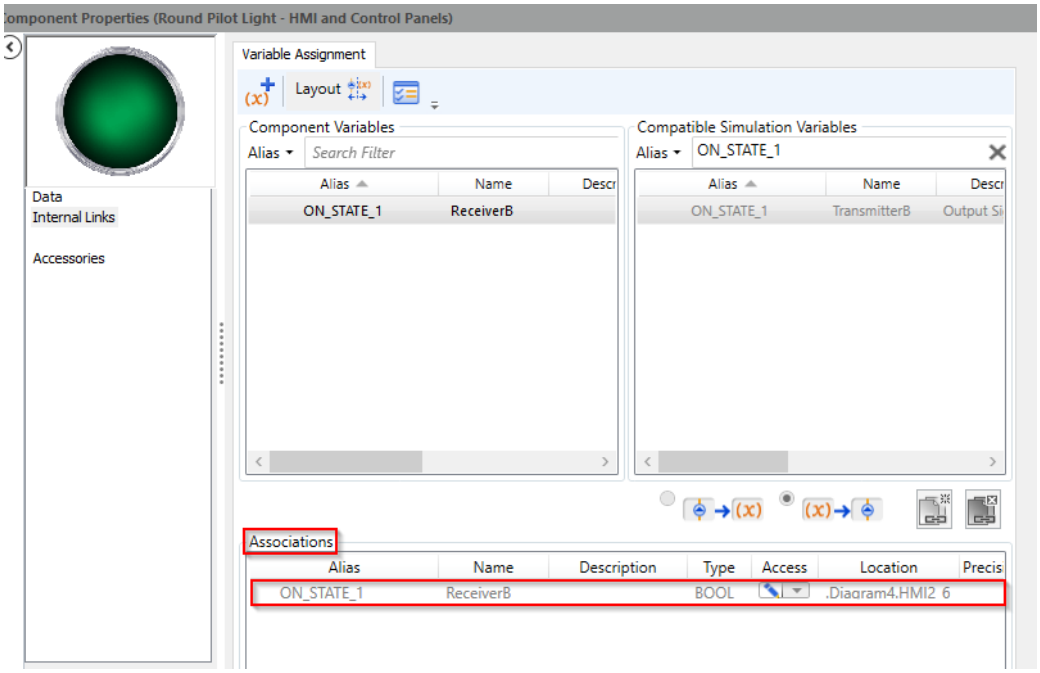

Figure 97: How to make sure that the assignment or communication between two components in AS software has already happened

Figure 99: Indication of the assignment (the blue alias) in AS software

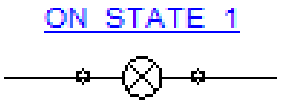

Figure 98: Indication of the assignment (the blue alias) in AS software

You can do the same steps for linking other lights together ("indicator lights" of the plc & "round pilot lights and flashing beacon lights" of the HMI&control panel system).

2. For the solenoids:

   a. we must go to our electro-pneumatic system, by double-clicking on the directional valve used to control our first cylinder, a window will show up, then we go to “Variable Assignment”, and we highlight one of the solenoids for example the left one (Sol1), and we can highlight it either by pressing one time on the left solenoid symbol drawn on the top left of the shown window, or by opening the branch under the valve symbol, and choose “Sol1” as shown in the photo below:

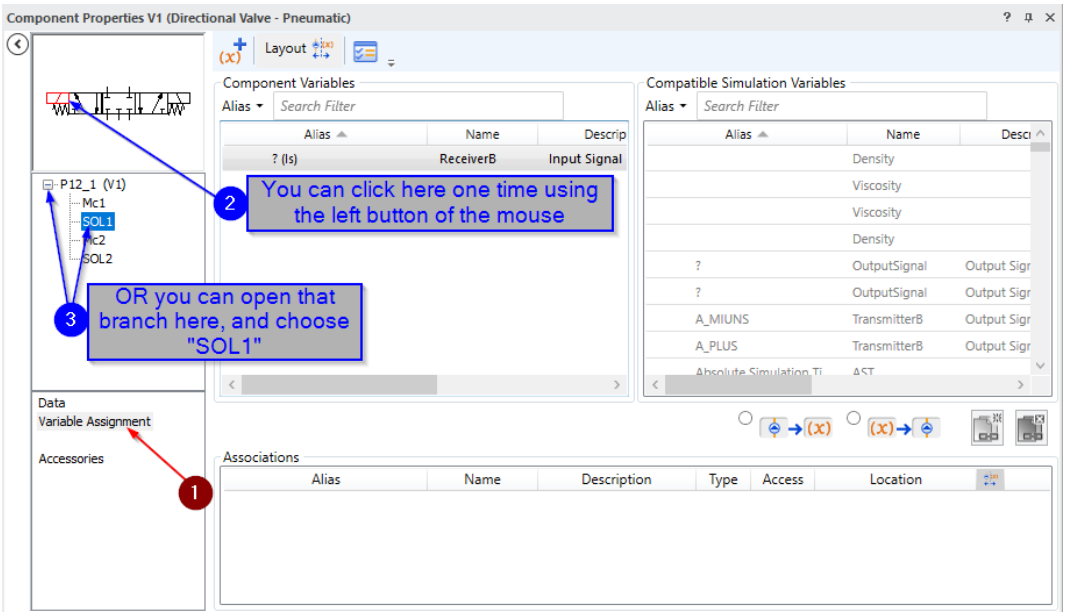

Figure 100: How to make an assignment to the solenoids of the electro-pneumatic cylinder in AS software

b. Then we can do the same steps that we did before, so we type the alias of the component (solenoid) which is connected to our PLC (A_PLUS) in the "search" bar of the "Compatible Simulation Variables", or we can find it directly without searching, then we select it, and we establish the linkage by pressing on that symbol ( ) as shown in the photo below:

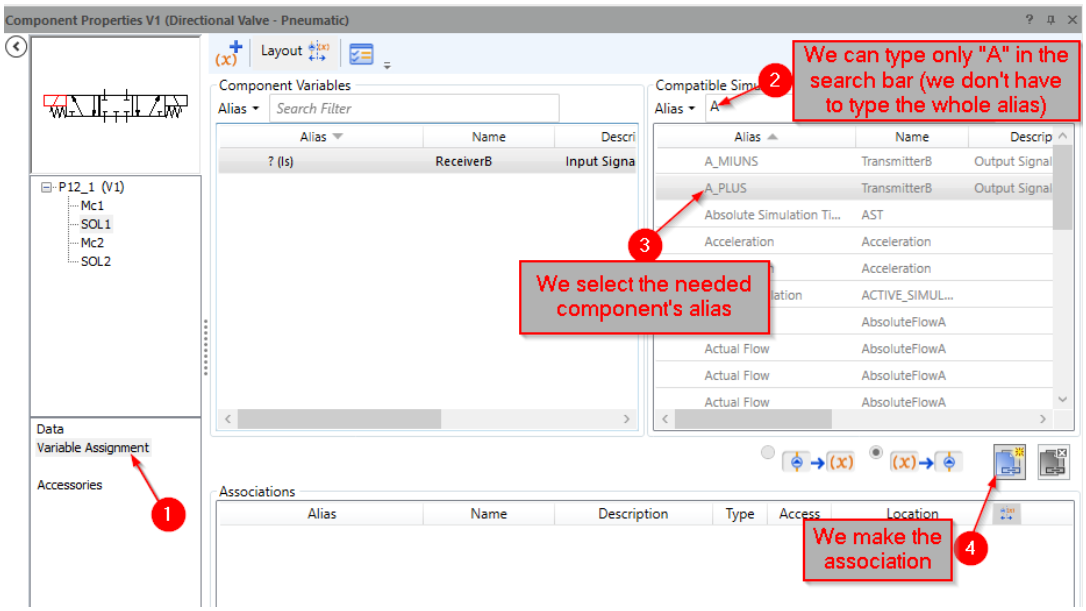

Figure 101: How to choose the component that will be assigned to the left solenoid of the electro-pneumatic cylinder in AS software

Note that you have to make sure that you are searching with the alias because if you are searching with for example the name of the component, you will get nothing, as the name of the component is different from its alias.

c. After doing the previous steps, the link will be established and shown in the "Associations" section, and also you can see that the alias of the linked component (connected to the PLC) is now in blue, and the same alias will be written over the left solenoid of the valve and also in blue as shown in the photos below, and this means that the association is done successfully:

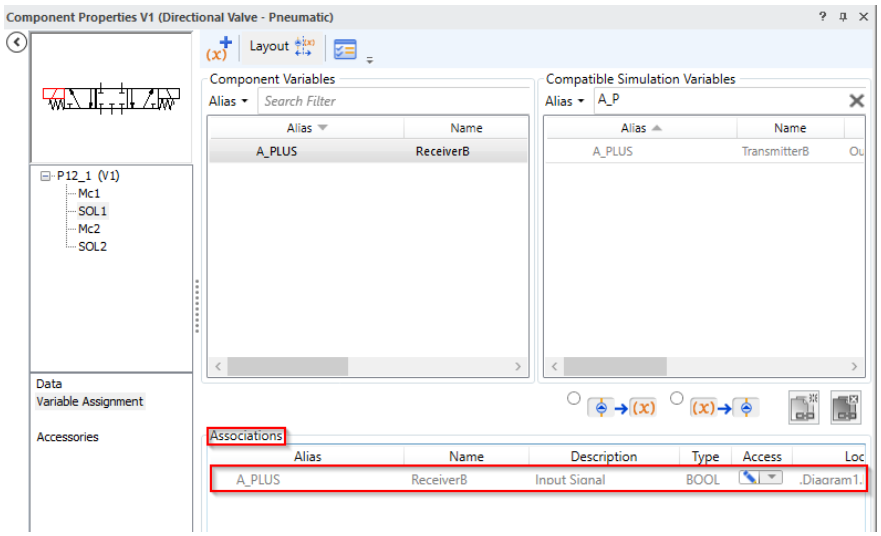

Figure 102: How to make sure that the assignment or communication between two components in AS software has already happened

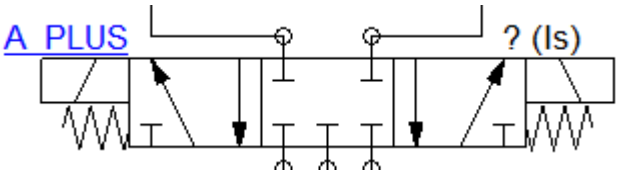

Figure 103: Another way How to make sure that the assignment or communication between two components in AS software has already happened (blue alias)

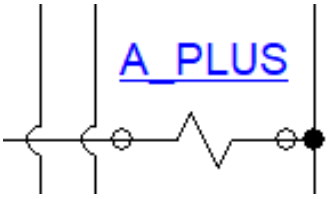

Figure 104: Another way How to make sure that the assignment or communication between two components in AS software has already happened (blue alias)

We can do the same steps to make the association between the right solenoid of the valve and the solenoid connected to the output of the PLC which we called "A_MINUS".

2.2.4 Ladder program

- First, we will briefly explain what we want the ladder program to do, or how our project is supposed to work:
  a. First, we use the selector to choose if we want to take the test on the first cylinder, the second cylinder, or both.
  b. We want to use the "ON_OFF" button to turn on or off the system, If we pressed it one time, the system will be on, and the green light of the selected cylinder or cylinders will be on. If we pressed another time on the same button, the system will be off, and the green light or lights will be off.
  c. When the system is on, we can start the automatic test sequence by pressing the "WEAR" button, which will let the cylinder do several cycles that can be determined by the user through our HMI system, during this operation, we need a certain led to be on (the blue led) to indicate the state of the system, we need also to show up the counting number of cycles.
  d. After the cylinder finishes the determined number of cycles, we need the system to automatically start the seal or leakage test, turn off the blue led, and let the yellow one be on to indicate the new state which is the "SEAL" state, in that state we need first the pressure to be stable so we keep the "A_MINUS" active for a certain time determined by me (that allows the pressure to get stable or reach its maximum if there is no leakage) until the pressure transducer measures "10 bar" which is the maximum pressure of the system. If the transducer read a value less than "9.98 bar", this means that there is a leakage and the red led will be on automatically. In any case, after the determined time passes, the valve will go to its initial position (at the middle), and the leakage test will start automatically during a certain time that can be determined by the user and displayed on a monitor while counting through the HMI system, during that time we need to check the measured pressure to see if there is a leakage or not.
  e. During the seal test, if the value of the measured pressure is between "9.98 bar" & "0.5", this means that there is a leakage in the system and it will be indicated by the red led which will turn on as soon as the measured pressure is less than "9.98 bar", at the end of the determined time of the seal test the final value of the measured pressure will be displayed to the user through the HMI system be using a "Numeric Display box" that shows up the exact final value and also by using a "Led Type VU meter" that will show a percentage of the final pressure out of the 100% which represents "10 bar" in our project.
  f. During the time of the leakage or seal test, if the measured pressure became less than "0.5 bar", we need the system to be stopped and an alarm light gets on indicating to the user that the cylinder is broken, and it needs to be replaced immediately.

g. We need also to use a “Proportional Flow Regulator Valve” for simulating the leakage in the system automatically, which means that at the beginning the proportional valve will be closed, after the system does the first wear & seal test, the proportional valve opens a little bit to simulate very small leakage after the system does the second wear & seal test, the proportional valve opens a little bit more to simulate more leakage and so on so forth.
h. In case of a leakage state, we need to record the values of the measured pressure during the seal test, so that we can analyze them to be able to know more about the leakage.
i. We need also to have a “STOP” button to be used when we need to stop the system for any reason like an emergency, and the system can continue again by pressing the “WEAR” button.

- Second, we will explain how to write our ladder program in Automation Studio software:
  1) We should first add a new document which is “Ladder Diagram” by going to the “Home” section and then “Document” as shown in the photo below:

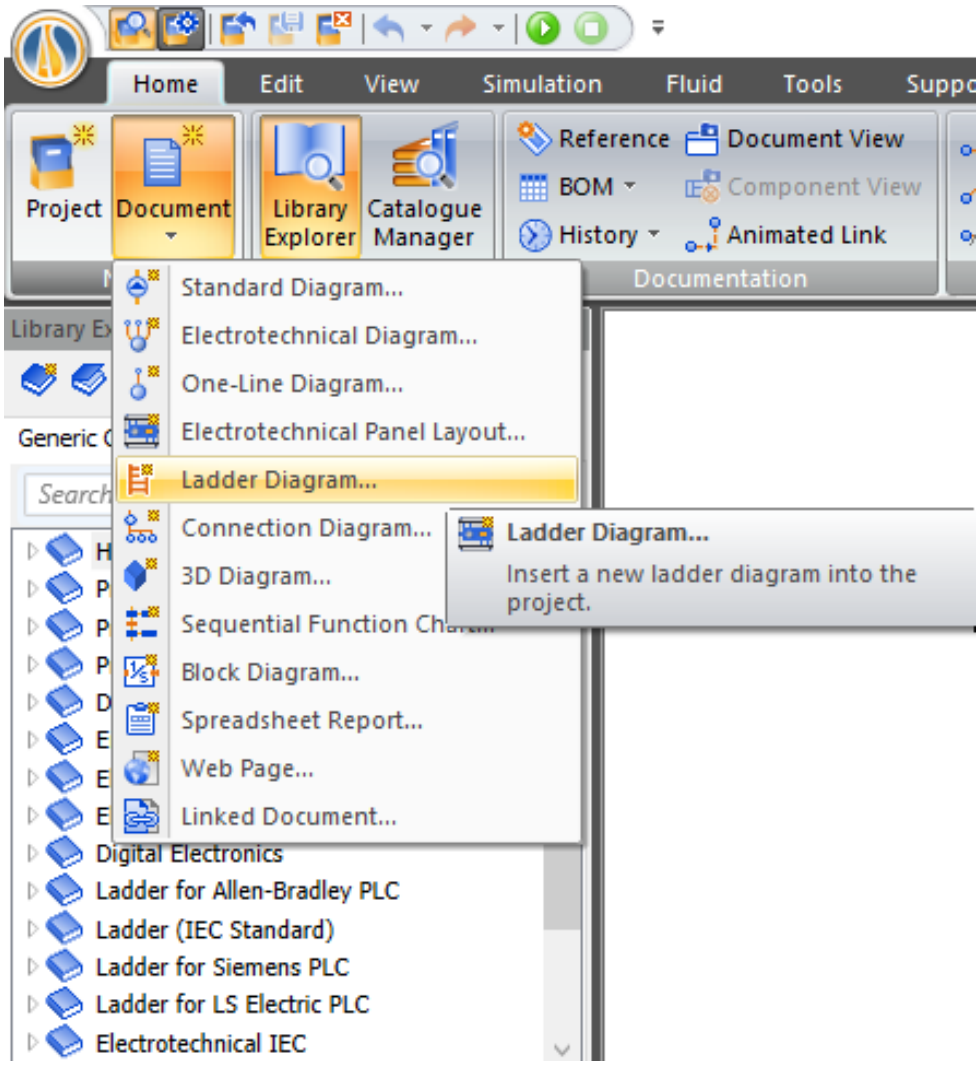

Figure 105: How to add a new Ladder diagram in AS software

We are going to work on Allen Bradley 500 software, so when we add a new ladder diagram as shown before, we must choose “AB500.lad” and press on “OK” button as shown in the photo below:

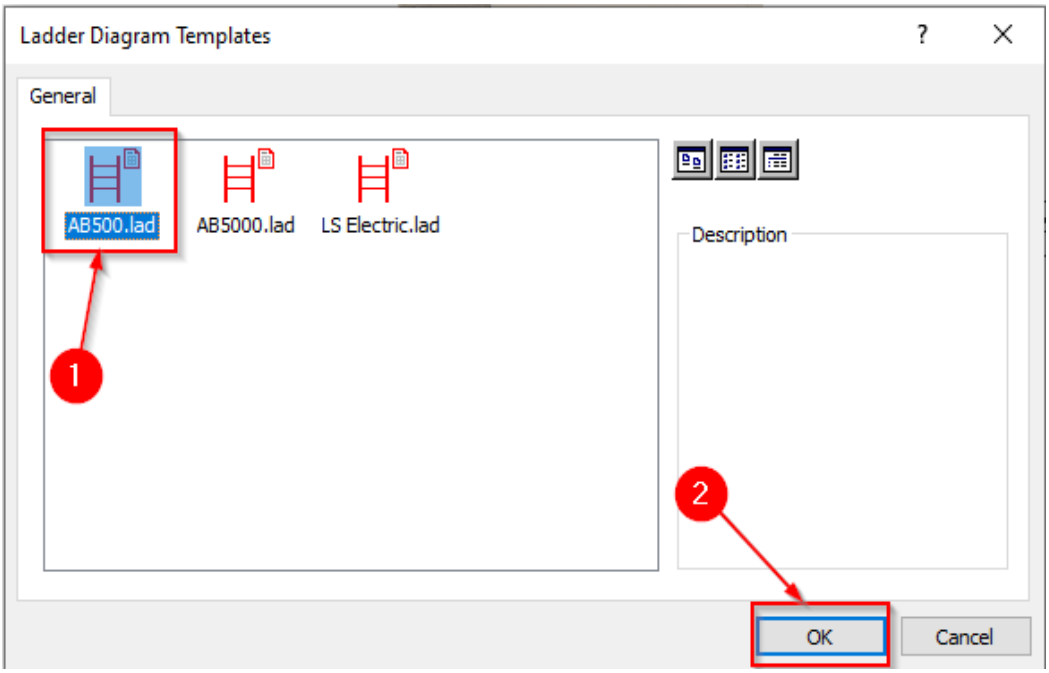

Figure 106: Choosing the Ladder diagram for AB500 PLC in AS software

When we do these previous steps, a new document will show up, and we can see a new section has been added in the toolbar called “Ladder” which has some operations that can be done on the ladder program as shown in the photo below:

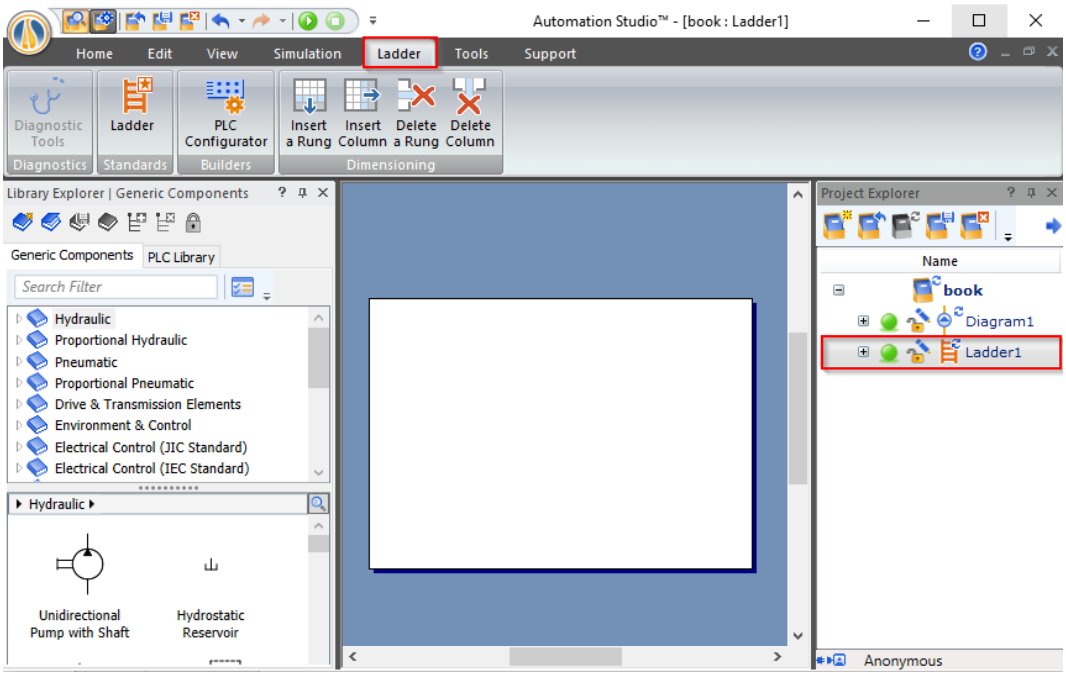

Figure 107: The newly added section called “Ladder” of the newly added diagram for the AB PLC in AS software

As shown also in the previous photo, it has been added a new diagram in the “Project Explorer” part which is the “Ladder Diagram”, is called automatically “Ladder 1” as it is the first ladder diagram added and you can change the name of this diagram or the others easily like we do with any normal file on PC, so you make a double click using the left button of the mouse on the default name, and you will be allowed to write whatever you want.

2) Now we can start writing our ladder program in that new ladder diagram, and we should first add the “Power Rail” as shown in the photo below:

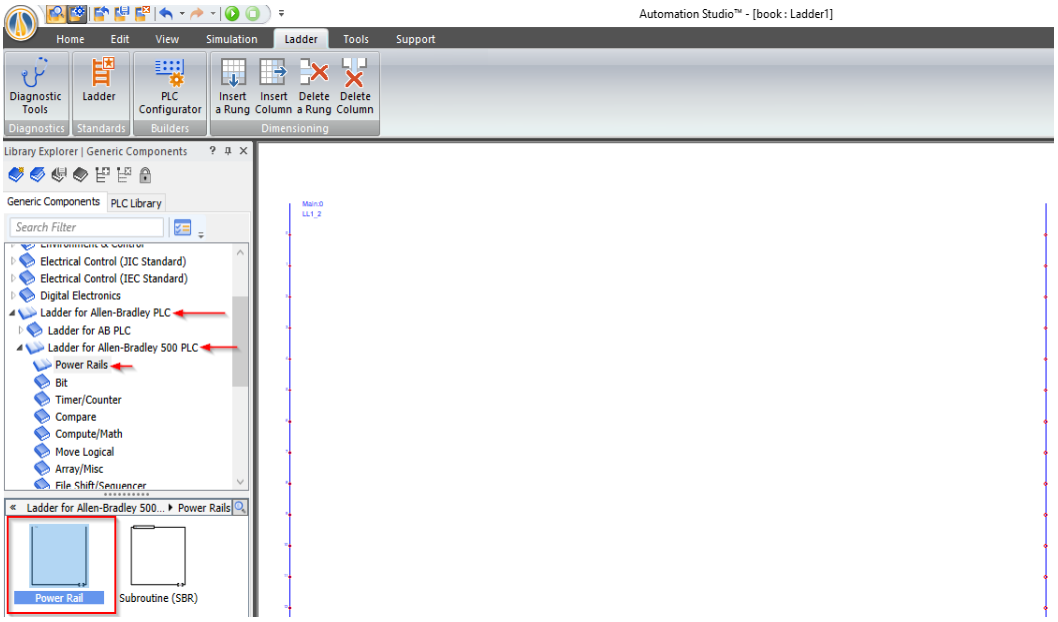

Figure 108: How to add a new rung for the ladder program in AS software

As shown in the previous photo, we will use the “Ladder for Allen-Bradley 500 PLC” library for our ladder program, and in this library, you can find all the needed components for your ladder program, and you can add them easily as we did before with other components in the other libraries.

3) For writing the ladder program, we started writing it using a sheet and a pencil so that we can easily modify the ladder as shown in the photos below:

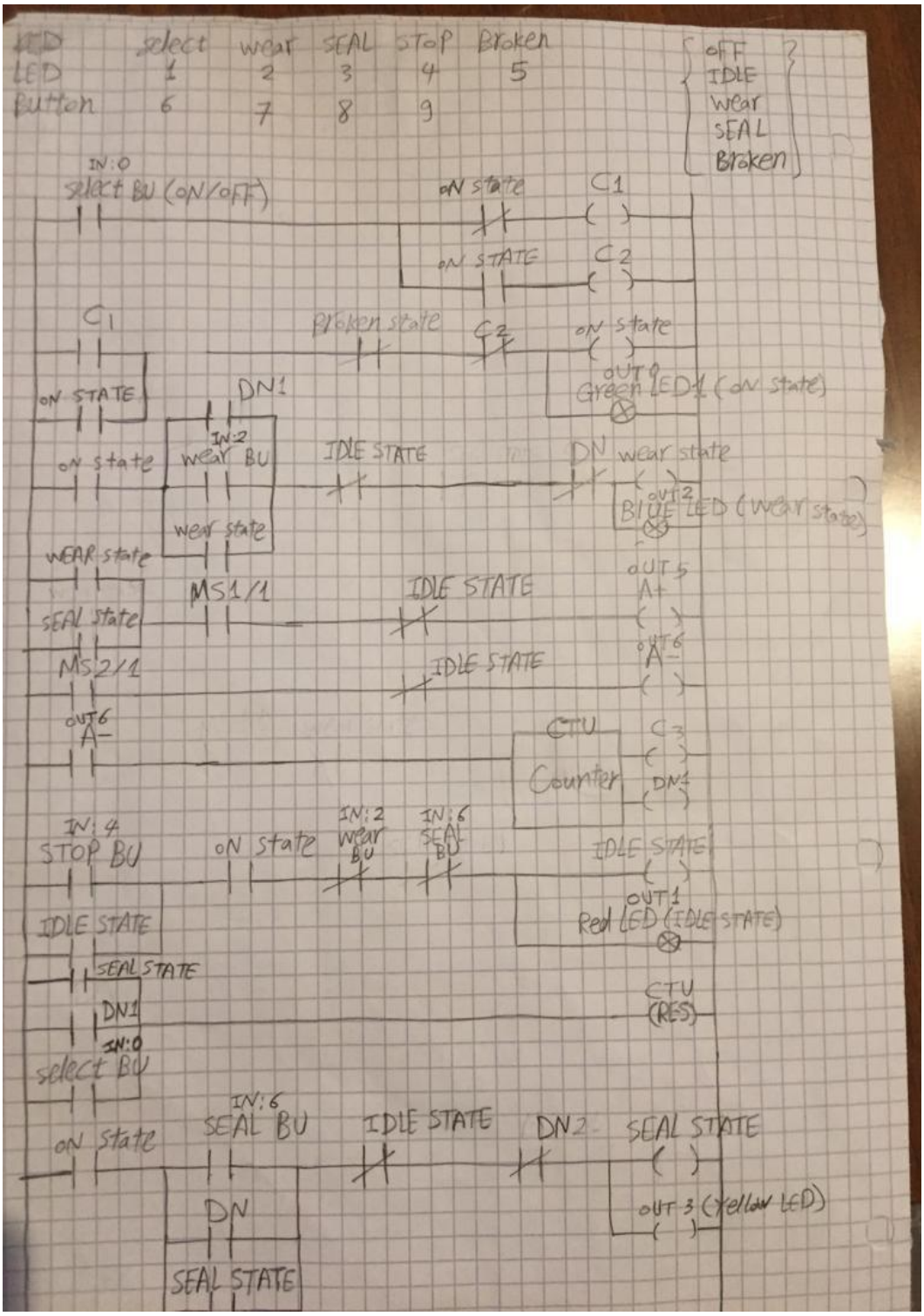

Figure 109: Part of the Ladder program written on a sheet

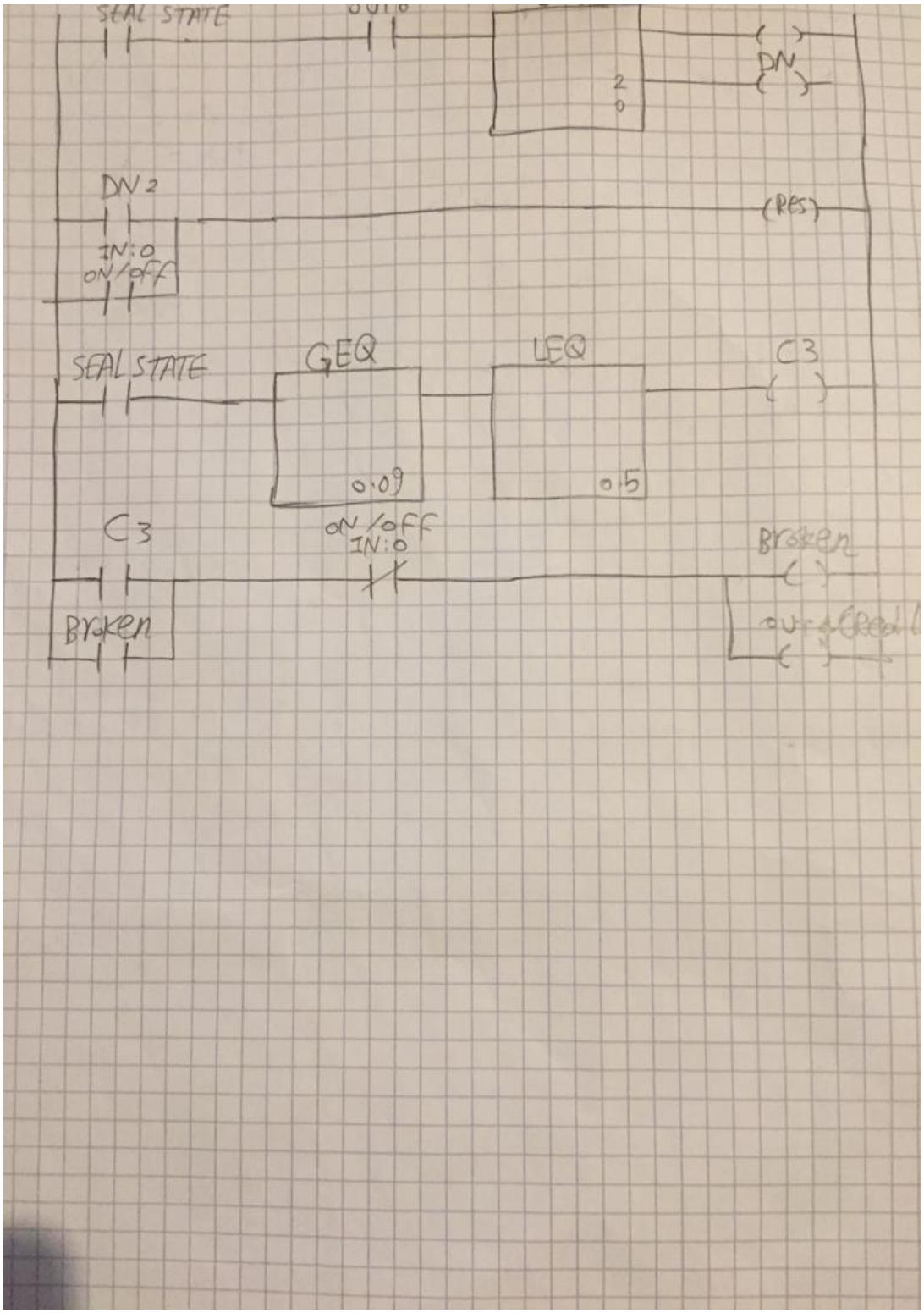

Figure 110: Part of the Ladder program written on a sheet

4) In the end, we wrote and simulated the program in Automation Studio software, and we kept enhancing it until we reached the final goal as shown in the photos below which are for the first cylinder:

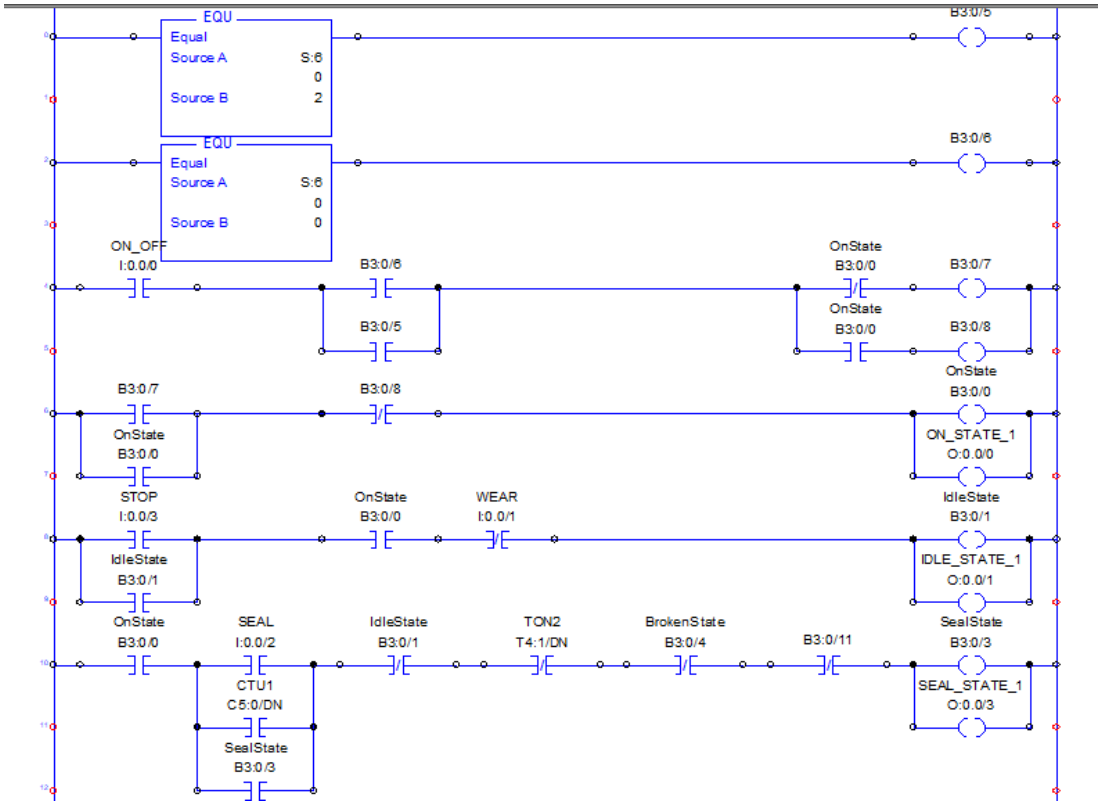

Figure 111: First part of the Ladder program in AS software

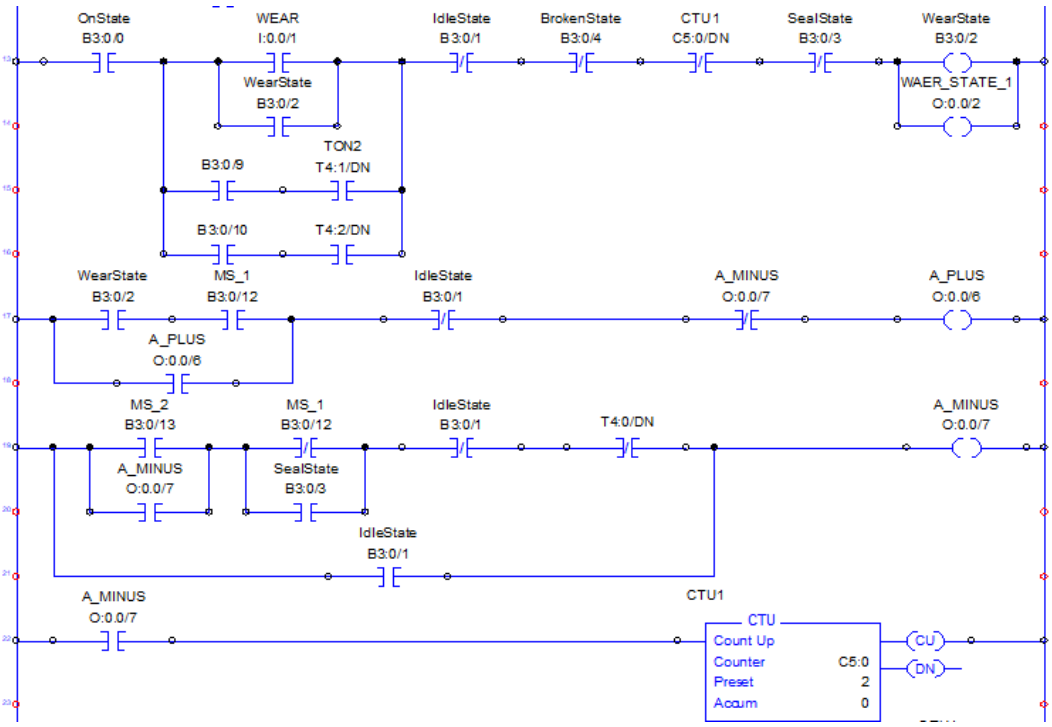

Figure 112: Second part of the Ladder program in AS software

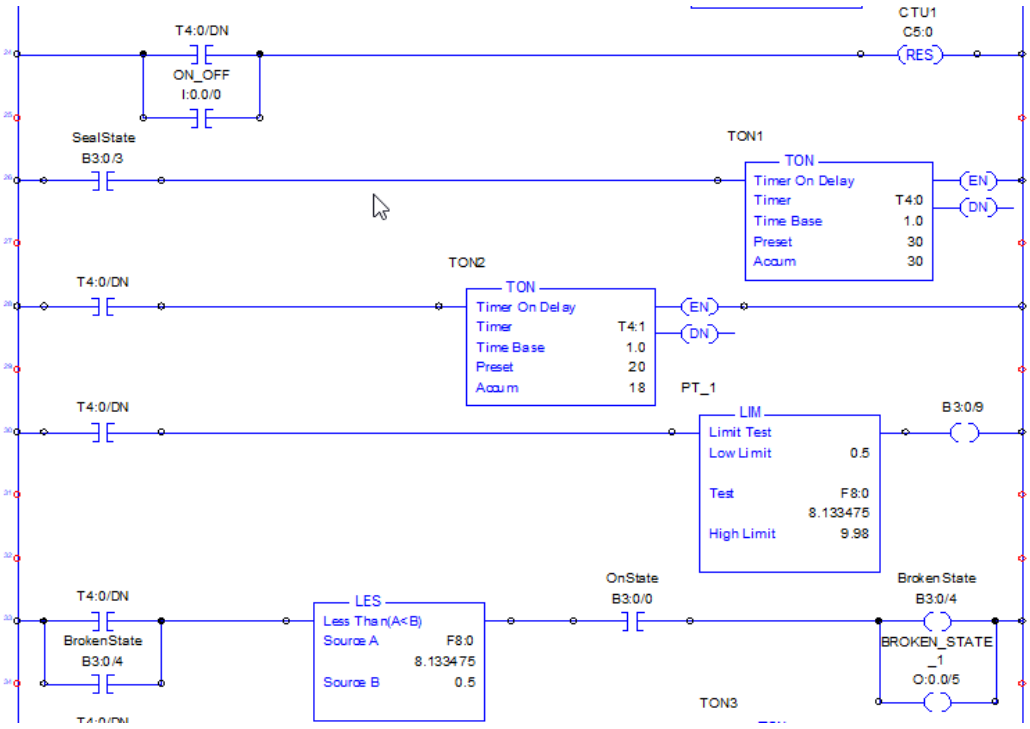

Figure 113: Third part of the Ladder program in AS software

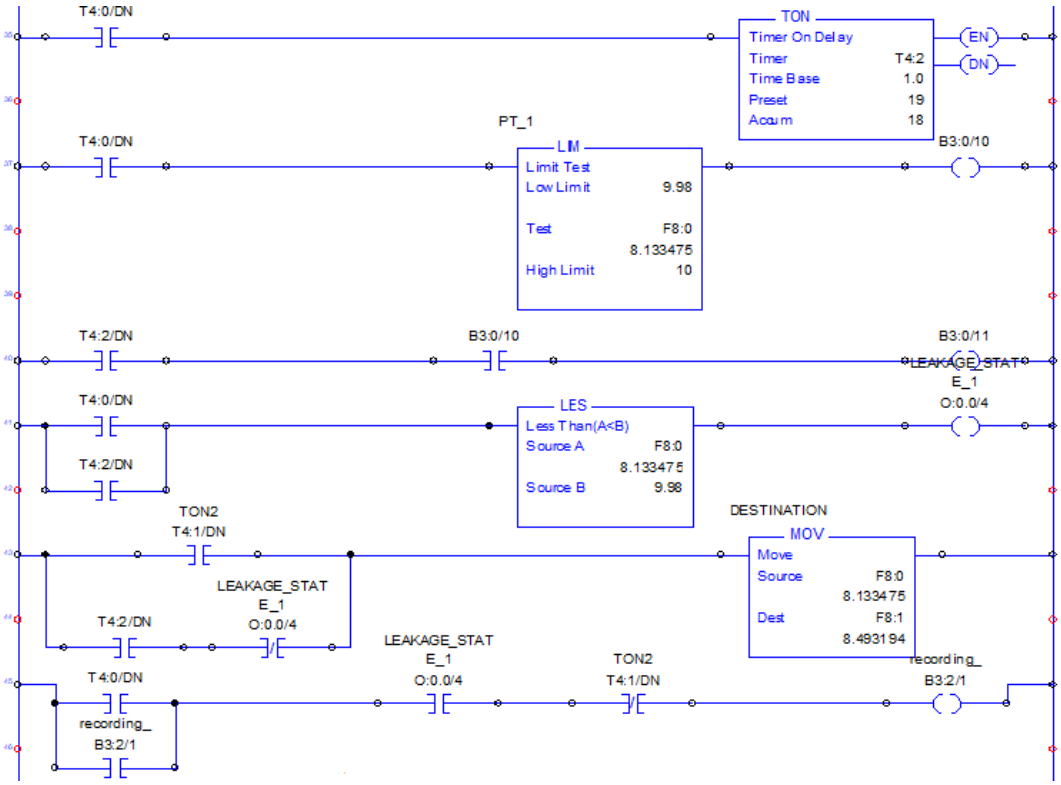

Figure 114: Forth part of the Ladder program in AS software

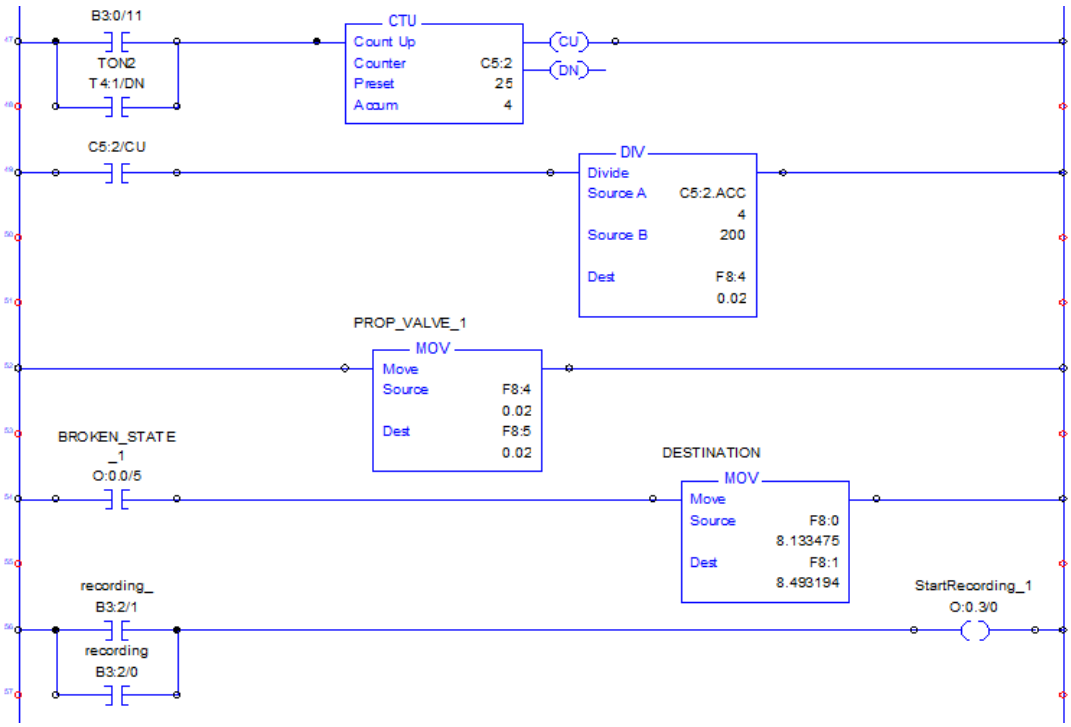

Figure 115: Fifth Part of the Ladder program in AS software

- Now, we are going to explain in detail each part of the ladder program and its function in the project, and as we did in the previous discussion, we will explain the Ladder program of the first cylinder, like the one for the second is almost the same, but first let's clarify the way of addressing inputs, outputs, auxiliary outputs, timers, counters that we will use in our Ladder program (Allen Bradly 500):

Inputs	I:0.0 → I:9.250
Outputs	O:0.0 → O:9.250
Auxiliary outputs	B3:0 → B3:250
Timers	T4:0 → T4:255
Counters	C5:0 → C5:255

Table 1: How to address inputs, outputs, auxiliary outputs, timers, and counters of the LD program in AS software

Now we can start explaining our Ladder program:

1) The first two lines in the Ladder program are linked to the selector in the HMI system, so the selector is in the normal position which is the first one on the left that equals "0" in the Ladder program, then as we see in the second line the "Source A=0" as "Source B", and so this line will be active or on, and only the first cylinder will work, in case that we want to let the two cylinders work at the same time, we will put the selector on the third position that equals "2", and so the first line will be active, and one of these lines must be active to let the rest of the Ladder program for the first cylinder work, and as shown in the photo below, we use two auxiliary coils (not a real outputs), so that they can be used later in the Ladder program as a condition.[6]

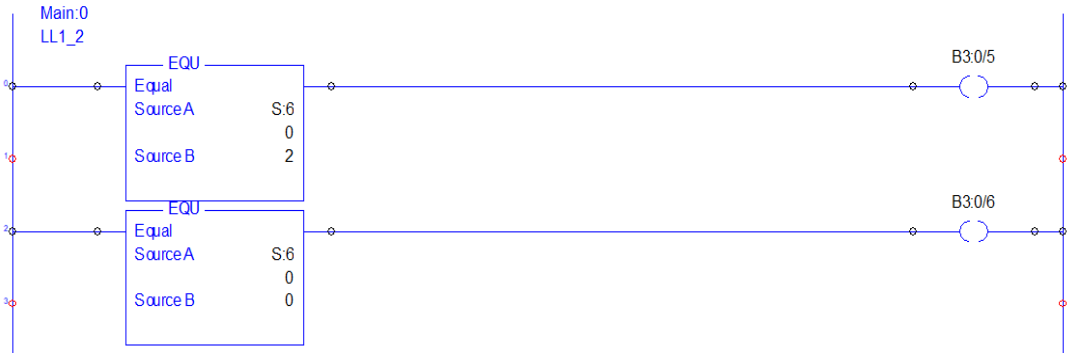

Figure 116: The equalization comparison in the Ladder program

2) These two lines in the photo below are used for the “on/off” function for the whole project, and we use only one button for that (ON_OFF), and as shown below the auxiliary coil (OnState) will be used below as a condition for activating the rest of the program, the output coil (ON_STATE_1) is used to activate the green led in the HMI system to indicate that the system is on and ready.[4]

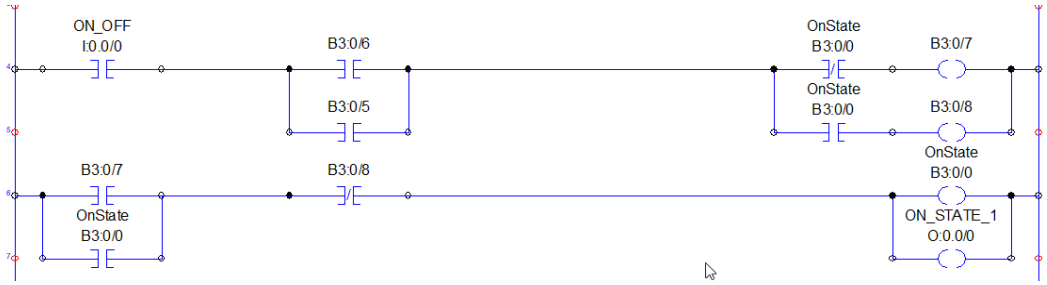

Figure 117: ON/OFF state for the project in LD program

3) This blow line here is used for the idle state, as we explained before when we press the “STOP” button, I want the system to go to the idle state and stop, and we use here an output (IDLE_STATE_1) that is connected to the white light in the HMI system to indicate the idle state of the system.

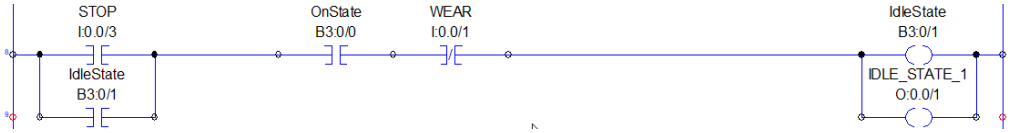

Figure 118: Idle state in LD program

4) The line below is used to start the wear state (the extraction and retraction of the cylinder) according to a determined no. by the user, and to do that we use the button "WEAR", and also here as shown below we use an auxiliary coil for helping me later on in the program, and an output coil to turn on the blue light in the HMI system to indicate the "Wear State". The contact called "WearState" is used for latching, and the other parallel lines with the "WEAR" button are used to let the system work automatically, so when the "Seal" test that we will see later finishes, one of these lines will be active, and let the "Wear" state to work again automatically, also for the closed contacts used for other three states, they are used to not allow the other three states (Idle, Seal, and Broken states) to work during the Wear state, so that we don't have an overlap, finally for the "CTU1" contact, it is used to stop the "Wear" state when the no. of cycles are finished, so it is connected to the first counter.

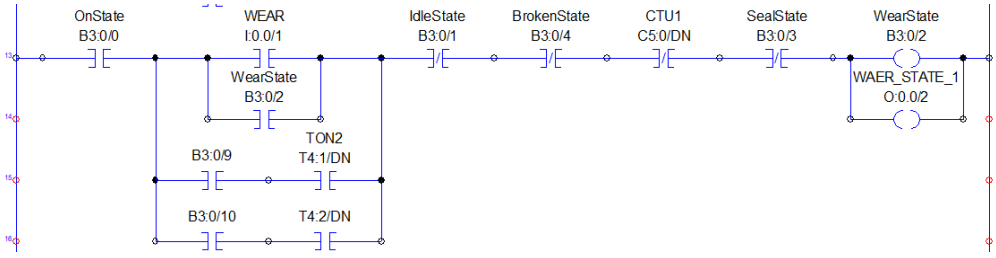

Figure 119: The wear state in LD program

5) The line below is for the "Seal" state (Leakage test state), we use the same idea as we used for the "Wear" state, the difference is that we use here "CTU1" contact in parallel with the button "SEAL" to allow the seal state to work automatically when the counter used for the wear state cycles is done, so it is a done bit, the "TON2" contact which is the done bit for the second timer is used to make the seal state stop when the time ends, the closed contact "B0:0/11" is used to end up the seal state (Leakage test) even without letting it finish, and that is because at a certain time, the system didn't discover any leakage, so we don't need to wait so long time to finish this state, and later we will explain exactly what is activating this contact.

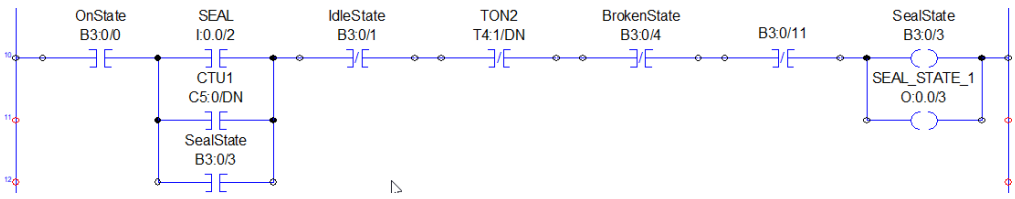

Figure 120: The activation of the "Seal" state in the LD program

6) These lines are used for extension and retraction of the cylinder, we use “A_PLUS” for extension, and “A_MINUS” for retraction, and we can recognize that “A_PLUS” can be activated when the wear button is pressed (we are in the wear state), the magnetic sensor (MS_1) is on, and we aren’t in the idle state, and we use a normally closed contact (A_MIUNS) which is related to the output coil of retraction to prevent overlapping. For “A_MINUS”, it can be activated in two ways, the first is automatically when the piston of the cylinder is fully extended because in that case the second sensor “MS_2” will be active and the normally closed contacts in that line will not be active, the second way is that when we press on the “STOP” button (activate the idle state), the retraction will be activated as we want the cylinder to become in the idle state and then stop. When the retraction ends, the sensor (MS_1) will be active again and this will deactivate “A_MINUS” and activate “A_PLUS” and so on until the cycles are finished. For the normally closed contact “T4:0/DN”, it is used when the wear cycles end to keep the pressure way for retraction open until a certain amount of time passes, and then when the contact is activated, the directional valve returns to its idle position in the middle.

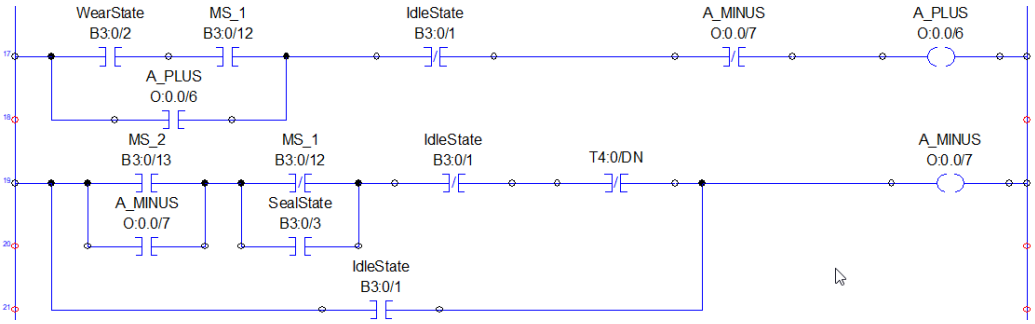

Figure 121: Extension and retraction of the cylinder in the LD program

7) These two lines below are for the counter used in our LD program to count the cycles done by the cylinder, and as shown below that every time the "A_MINUS" contact is activated, the counter is incrementing by one, in the second line, we make reset to the counter when the done bit of the first-timer (T4:0/DN) is activated, and this is the timer used for the pressure to become stable before the "Seal" test starts, and this part will be discussed later, or we can use the "ON/OFF" button to reset the counter, as we want to start our program again.

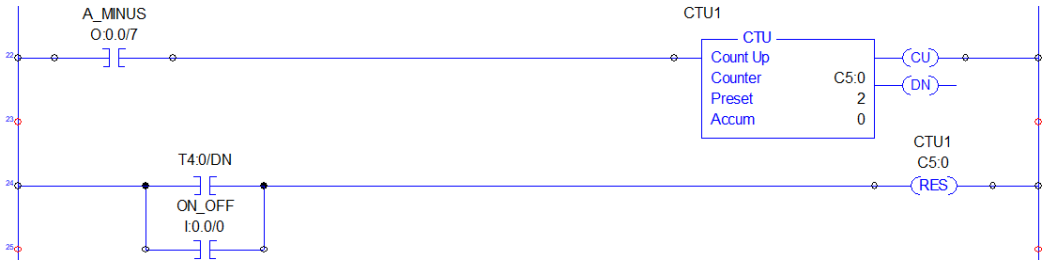

Figure 122: The counter function and how to rest

8) These two lines are for determining the time for the seal (Leakage) test, the first-timer is used to give the time for the pressure to be stable at the maximum limit (10 bar), and when this time ends the other timer in the second line starts, and that is the time during which we check if there is any leakage in the system, and this time can be determined by the user as we explained before in the part related to the HMI system.

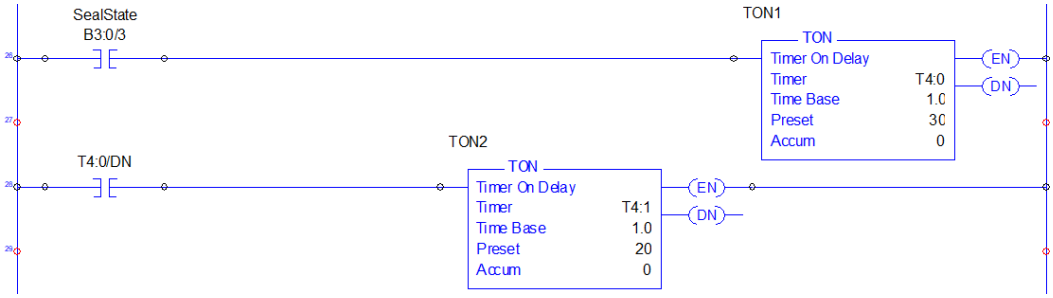

Figure 123: How to determine the seal (Leakage) test in the LD program

9) In the line below, we use a limit comparator to check the leakage in the system during the seal test, the done bit (T4:0/DN) is used, because we want to check the leakage during the seal test which starts after the system become stable (after the first timer ends), the two limit points here are (0.5 bar) which is the lowest limit of the pressure before the cylinder get broken due to the very high leakage, and (9.98 bar) which is almost the maximum pressure which is exactly (10 bar), the "Test" value in the limit component is the value coming from the pneumatic pressure transducer in our electro-pneumatic system, and if the value is between the limits, this means that there is a leakage in the system, and the auxiliary output coil will be activated to indicate that leakage.

Figure 124: Detecting the leakage using the LD program

10) This line is for indicating the broken state which can happen as we said before when the pressure of the system becomes less than (0.5 bar), and that is why we used less than component, and this check happens at the same time as for checking the leakage that is why we used "T4:0/DN" here again, we use here an auxiliary output (BrokenStatee), and a real output connected to a red alarm in the HMI system to warn the user about this state, and if that state happened, the whole system will immediately stop as we will see later.

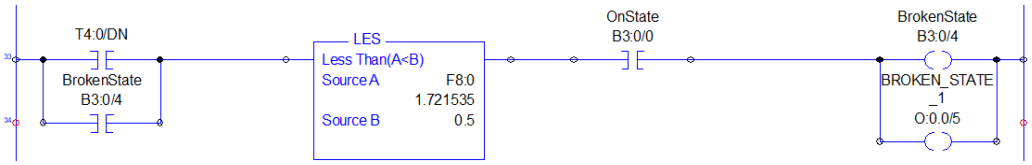

Figure 125: The broken state in the LD program

11) We used these two lines in the photo below to save the time during the seal or leakage test, so at the same time the program checks for the leakage, we use another timer with a certain time which can be much less than the time for leakage test, and at that time if it is detected that there is no leakage (the pressure is between 9.98 bar & 10 bar), then there is no need to continue the seal test, and we can save time, and I use the third line to have an auxiliary output indicating that during a certain time, there is no leakage so that we can use it later.

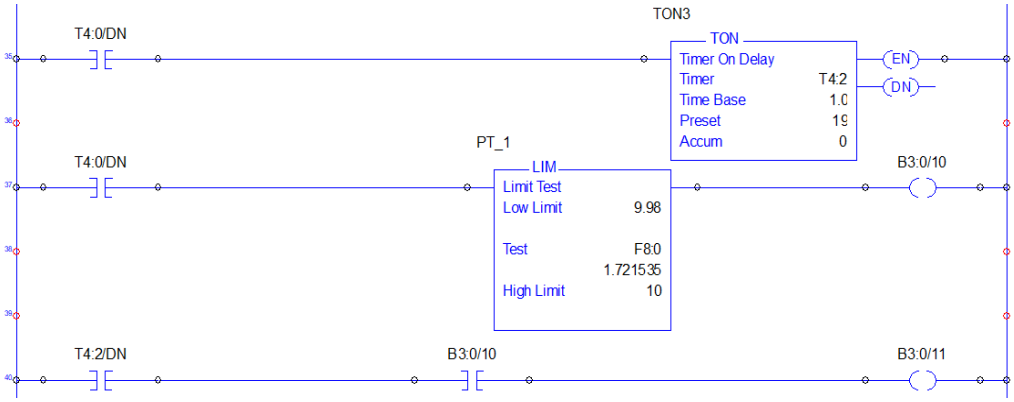

Figure 126: Saving the time of the seal test

12) The first line in the photo below is used to turn on the red light which indicates that there is a leakage in the system (LEAKAGE_STATE_1) during the pressure stability time or the seal checking time. The second line is used for moving the value of the pressure from the pressure transducer to a monitor in the HMI system to show the final value of the pressure after the end of the seal test.

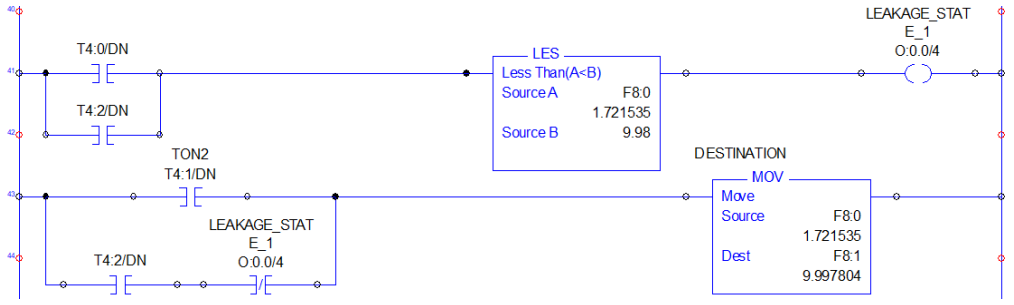

Figure 127: Indicating the leakage state in the HMI system and showing the final value of the pressure on a monitor

13) This line in the photo below is used for automatically activating a trigger signal for indicating the pressure values we need to record (the pressure values during the “Seal” test).

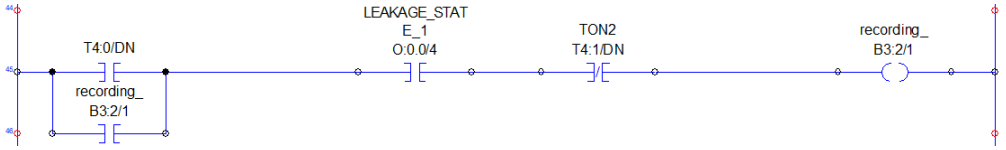

Figure 128: Recording the pressure values in the LD program

14) These lines in the photo below are used for automatically controlling the opening area of the “Proportional Flow Regulator Valve” which is used for simulating the leakage, the idea for doing that is to send ascending numbers every cycle to open the valve a little bit more every time, and so we use the counter for only getting an incrementing integer no., but because it will open the valve too much, we used a component (DIV) to divide the integer no. over a large value (200) to decrease the value of the automatically generated numbers, and then we move this value to the valve so that in each cycle, it opens a little bit more than the previous one.

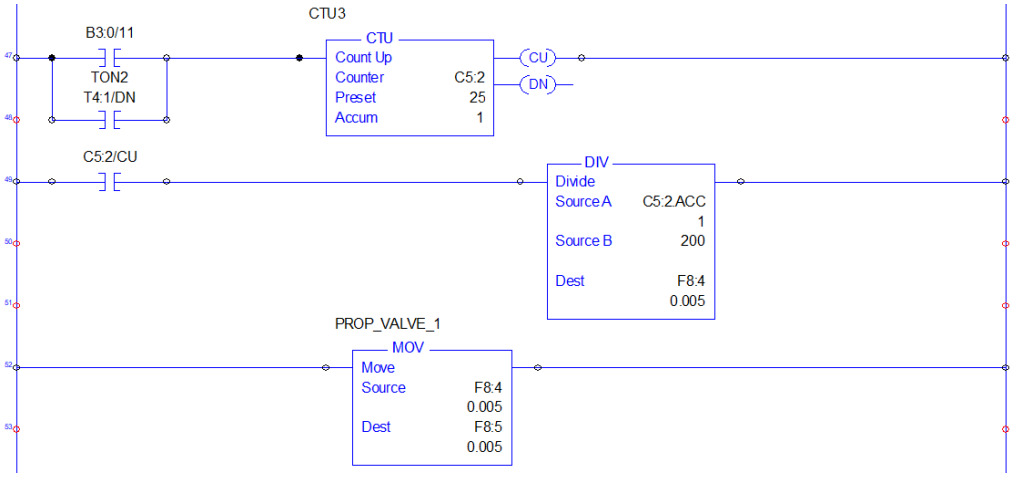

Figure 129: Simulating the leakage automatically in the LD program

15) This line in the photo below is used to show the value of the pressure transducer when the system is in the "Broken" state, and so we use a moving component to move the value of the pressure transducer to a monitor in the HMI system to show this value.

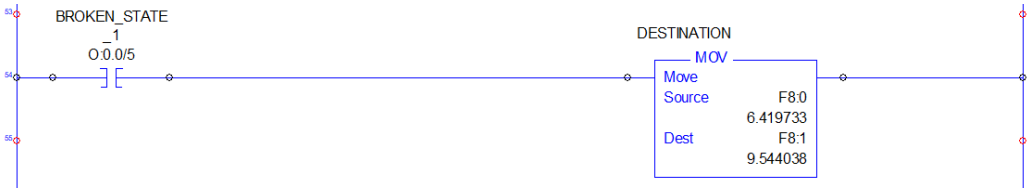

Figure 130: Showing the value of the pressure when the system is in the broken state

16) The line in the photo below is just used to activate the recording automatically, so we use the two normally open contacts related to the two auxiliary outputs for recording the pressure values of the first and the second cylinders, and we use them to activate a real output (StartRecording_1) which in his turn activates a trigger signal we use in the plotter so that we can extract the pressure values during the time of the trigger activation which is also the time of the "Seal" test according to our Ladder program.

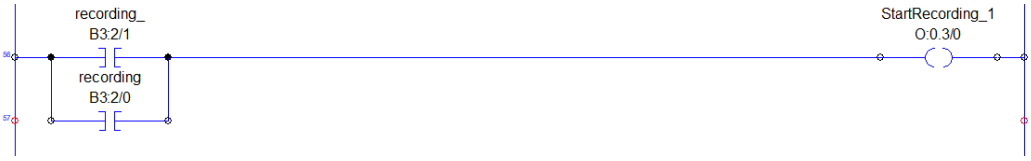

Figure 131: Automatic recording during the seal test

- In this part we will show how to plot the values of the pressure reading coming from the pneumatic pressure transducer, and how you get the recorded values from the plotter, saving, and analyze them:
  1) For plotting the values of the pneumatic pressure transducer readings, we just should go to “Simulation”, then we press on “y(t) Plotter”, and it will be added immediately. Now we should tell the plotter the variable we need to plot, and so what we should do is drag our pneumatic pressure transducer to the plotter, then we choose the value we want to plot which is the “Pressure” as shown in the photo below:

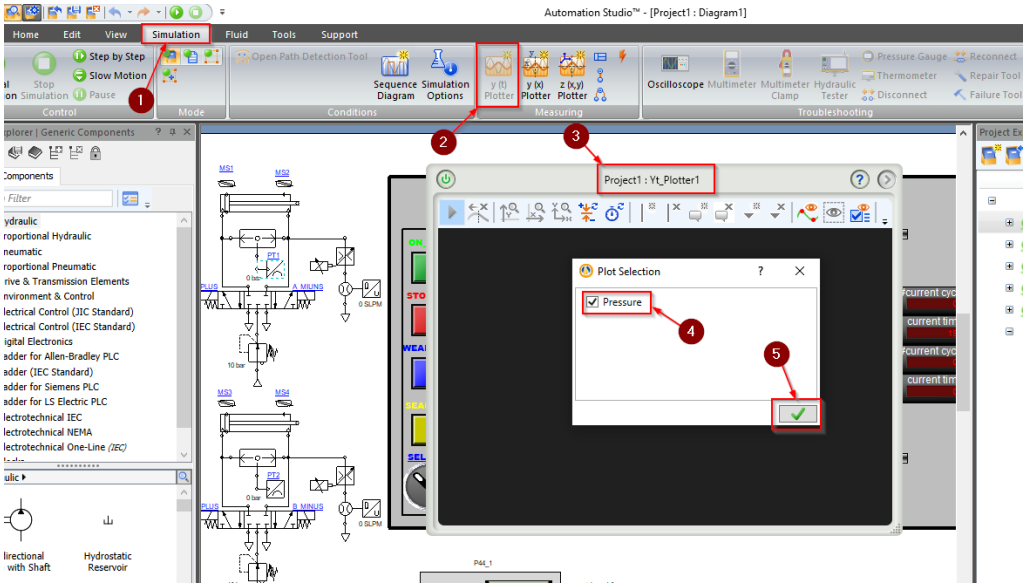

Figure 132: How to add a plotter of the pressure values coming from the pressure transducer in AS software

We need to add another variable in our plotter which is a “Trigger” that is used for recording purposes to let the plotter know when exactly we want to record during the simulation, it can be added exactly in the same way done before, and this trigger can be controlled automatically as we programmed in our Ladder program, it is like a counter that can give you a certain number for the part of the cycles we want to record, and this part is discussed before in detail.

In the end, we will have this plotter shown in the photo below:

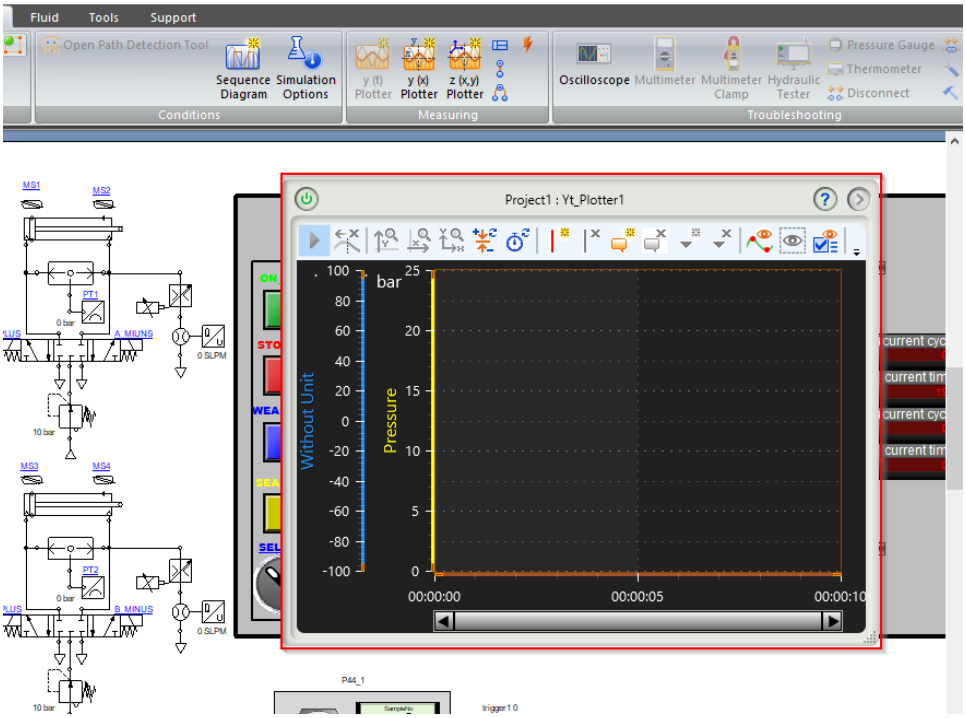

Figure 133: The plotter used for plotting the pressure values and for recording some values in a certain period in AS software

The photo below shows some plotted cycles of the pressure values, and as you can recognize from the photo, there is no leakage in the first cycle, as the pressure goes to the max. value (10 bar) and remains constant until the system starts a new cycle. In the other cycles, we can see a pressure drop, as the pressure values go to the max. value and then drops because of the leakage phenomenon, or even it never reaches the max. value of the pressure because of the high leakage.

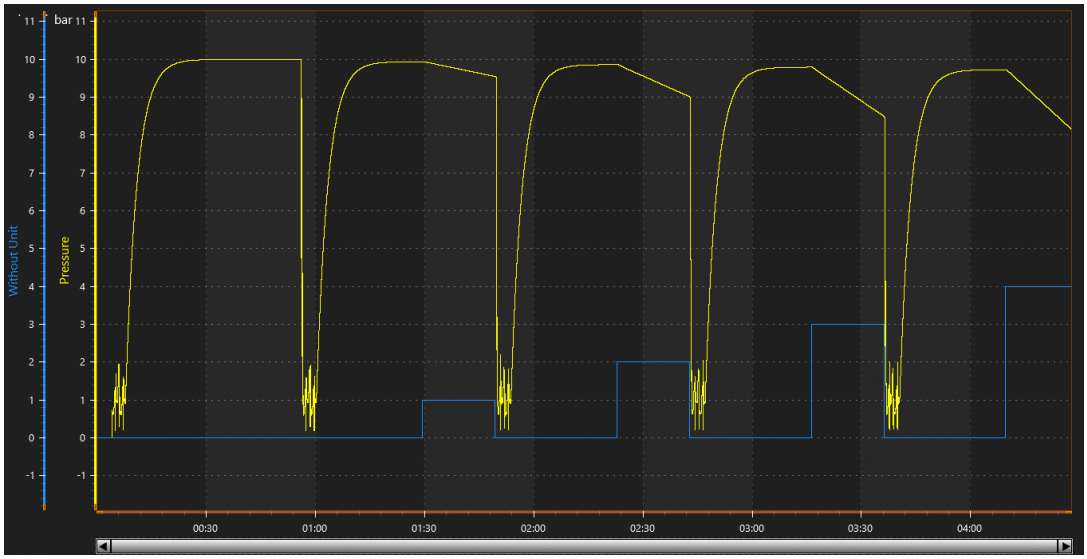

Figure 134: How the cycles of the cylinder plotted in our plotter in AS software

2) As we discussed in the previous point, we use another variable “Trigger” to record the pressure values during the seal test, and it is very easy to extract these values either to be saved or analyzed and to do that, we just must press into the curved arrow of the plotter, then a new window will show up asks you to give a name to your file, and then saving it as a TXT file as shown in the photo below:

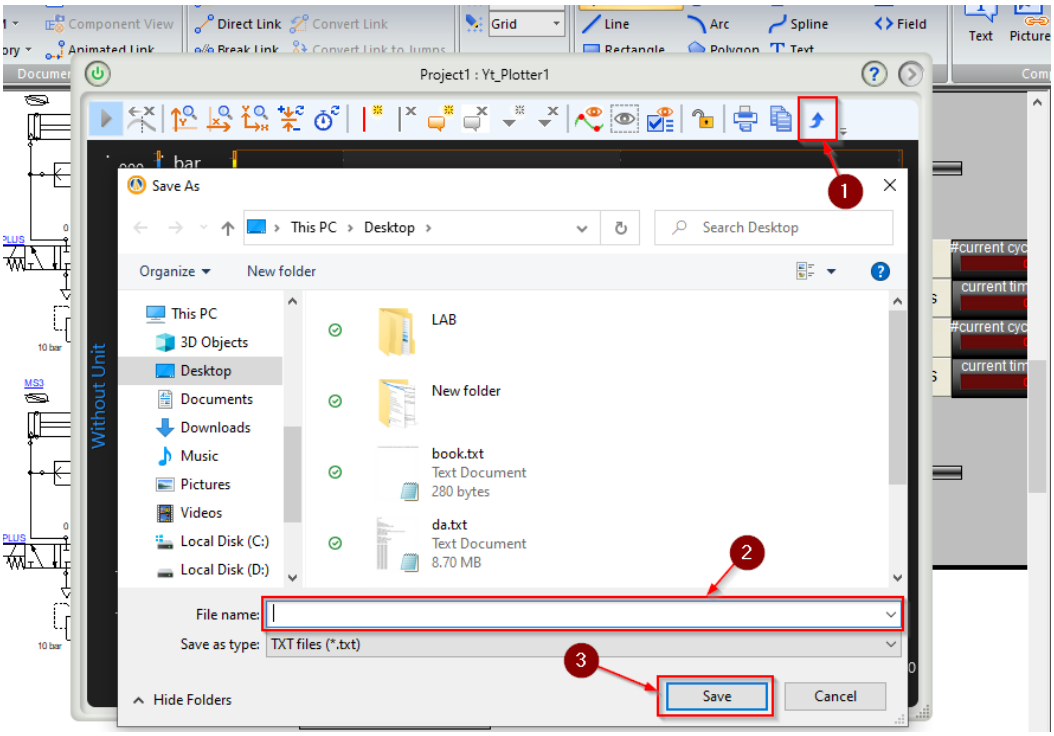

Figure 135: How to extract the recorded values from the plotter to a TXT file

After saving these values in a TXT file, you can easily copy them to any other software like Excel, and analyze these values using this software, for me we did that using “Access” software, and in the photo below you can see some of the recorded valued during the seal test from the first cycle:

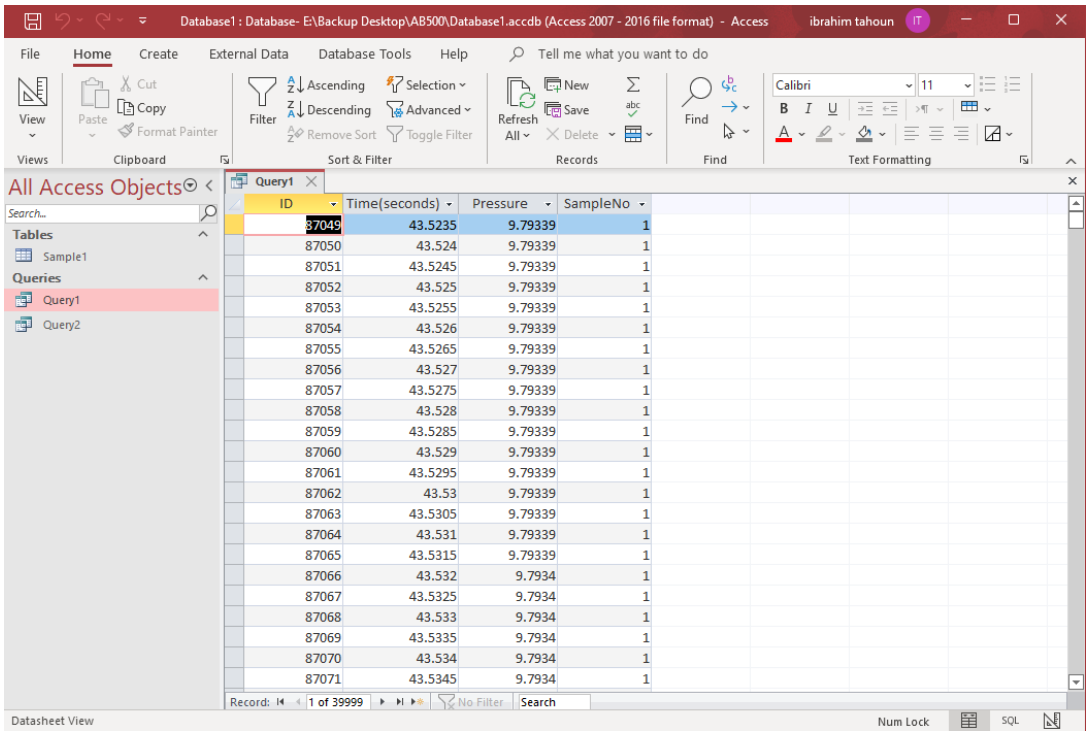

Figure 136: Some of the recorded values copied to the Access software from the TXT file

3 CODESYS software

3.1 What is CODESYS software!?

- CODESYS is an open-source programming language, used in physical and industrial computing and it follows industrial standard IEC 61131-3 means that it is used by hundreds of hardware developers globally.
- It is a complete, integrated development environment (IDE) and includes all programming functions as well as a comprehensive debugging and monitoring functionality.
- The control runtime system of CODESYS can be ported to almost all the operating systems or devices without an operating system.
- The CODESYS runtime system can also be turned into a CODESYS-compatible IEC 61131-3 PLC. In addition, SoftPLC systems are provided for immediate installation on standard platforms.
- Here are some screenshots from the CODESYS software to have a general idea about it:[8]

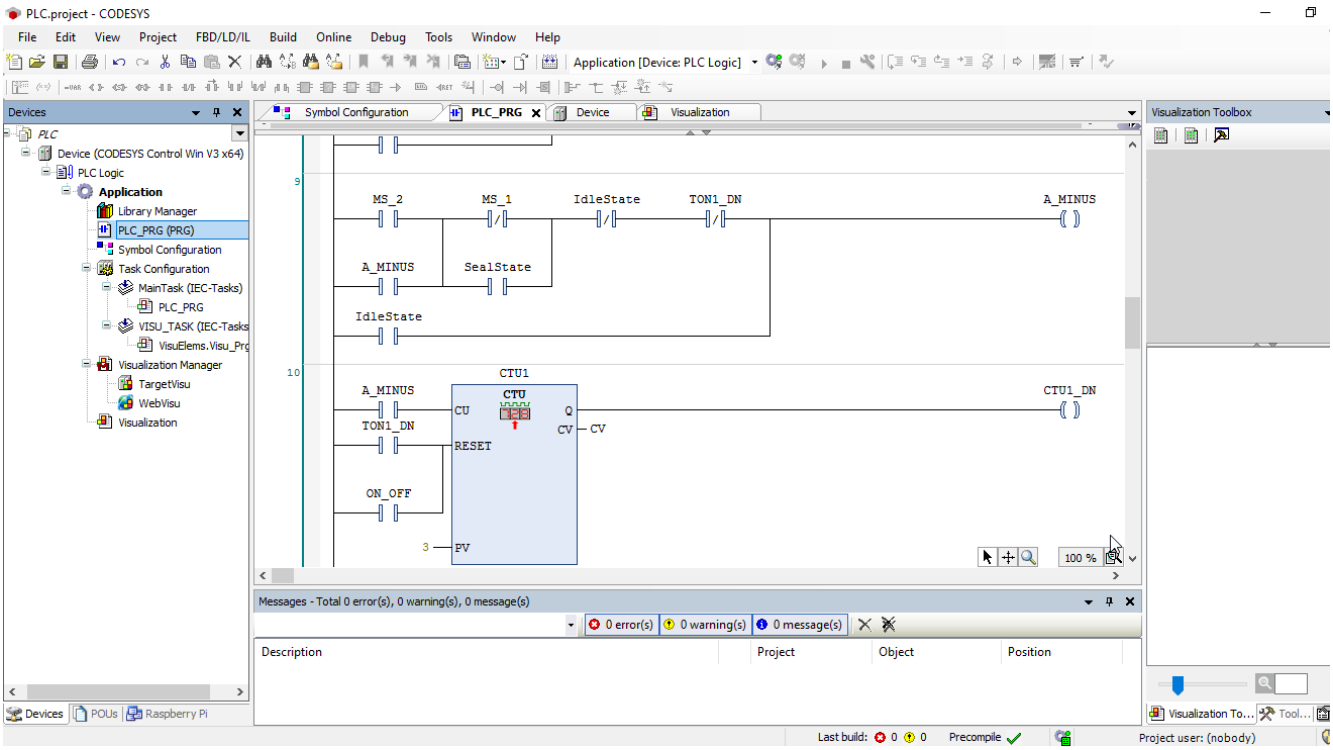

Figure 137: Part of the ladder program in CODESYS software

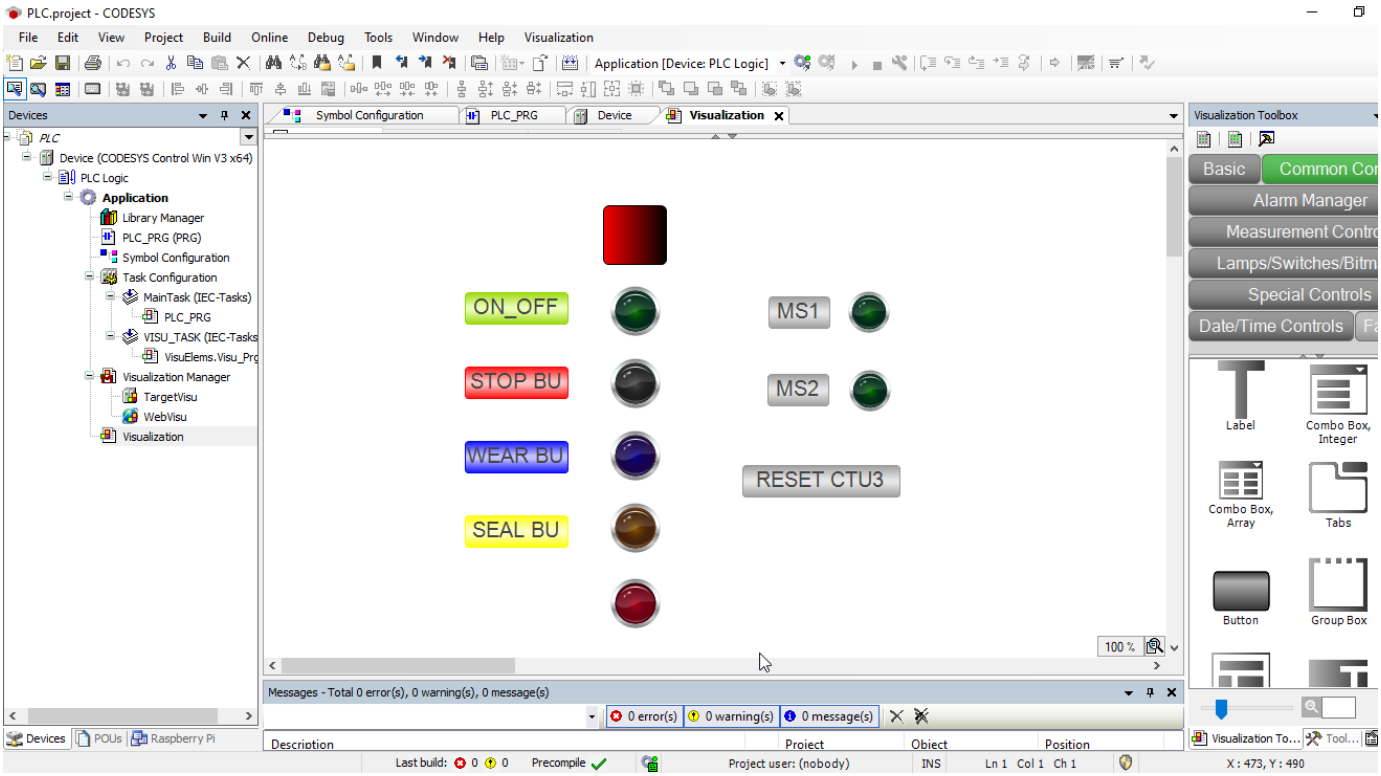

Figure 138: The HMI system in CODESYS software

3.1.1 Why did we decide to use CODESYS software in the project!?

1. The main reason for me to use this software is that it's compatible with RASPBERRY PI, and so I can use it to directly program the controller in the project which is RASPBERRY PI.
2. CODESYS is also free software as we said before, it is easy to use, and has a lot of features.
3. It allows me also to use the ladder programing language which I already used before in Automation Studio software, so I can write the same ladder program I wrote before with very limited modifications which are related to the type of PLC.
4. A very important feature that CODESYS has is that it allows me to simulate the program on a virtual PLC, so I can replace very easily the controller (Raspberry Pi) with PLC.

3.1.2 How could we use CODESYS software for programming the controller (RASPBERRY PI)!?

1. Firstly, we can download the software easily using that link (https://store.codesys.com/de/) as shown in the photo below:

Figure 139: How to download the CODESYS software

2. After downloading the software, we should install the library of the RASPBERRY PI board in the CODESYS software, and for doing that we have to follow these steps which will be illustrated also by photos:
   a) Firstly, we can download the library easily using that link (https://store.codesys.com/en/codesys-control-for-raspberry-pi-sl.html) as shown in the photo below:

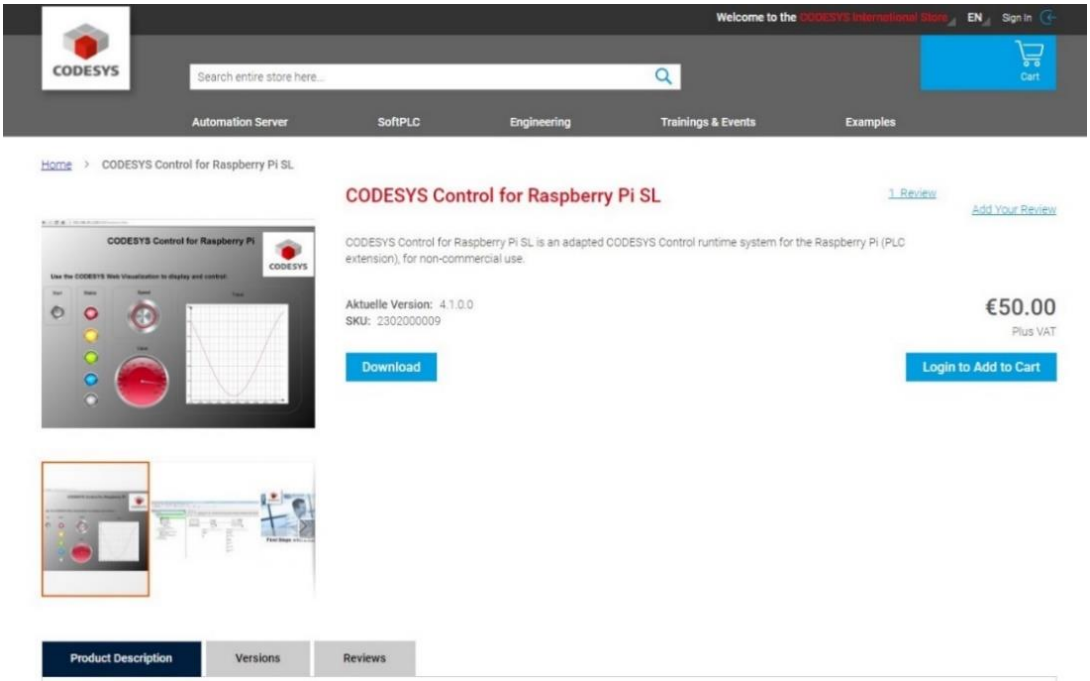

Figure 140: How to download the control library for Raspberry Pi

   b) We should select the "**TOOLS**" option from the main bar of the program.

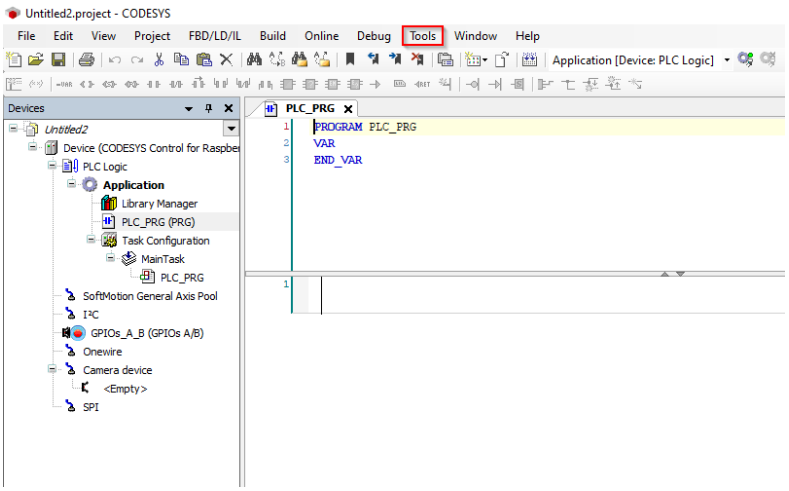

Figure 141: selecting the tools option in CODESYS software

c) From the "**TOOLS**" section, we should choose "**Package manager**" as shown in the photo below:

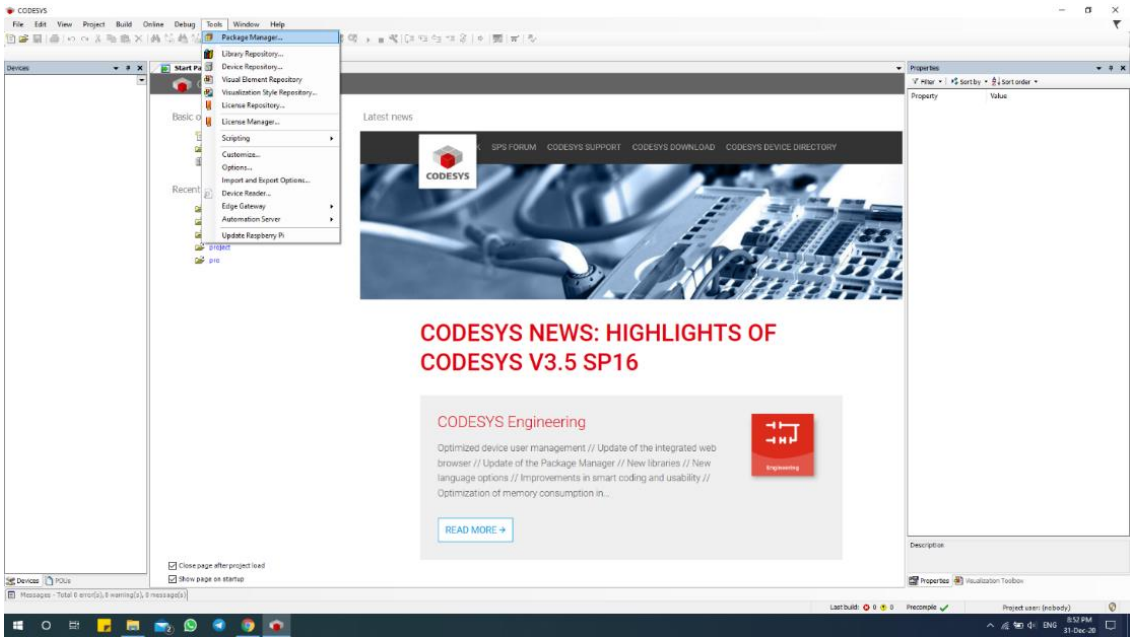

Figure 142: Choosing "Package Manager" in CODESYS software

d) We should press into "**INSTALL**" then select the library we just downloaded, then wait for the installation to be completed.

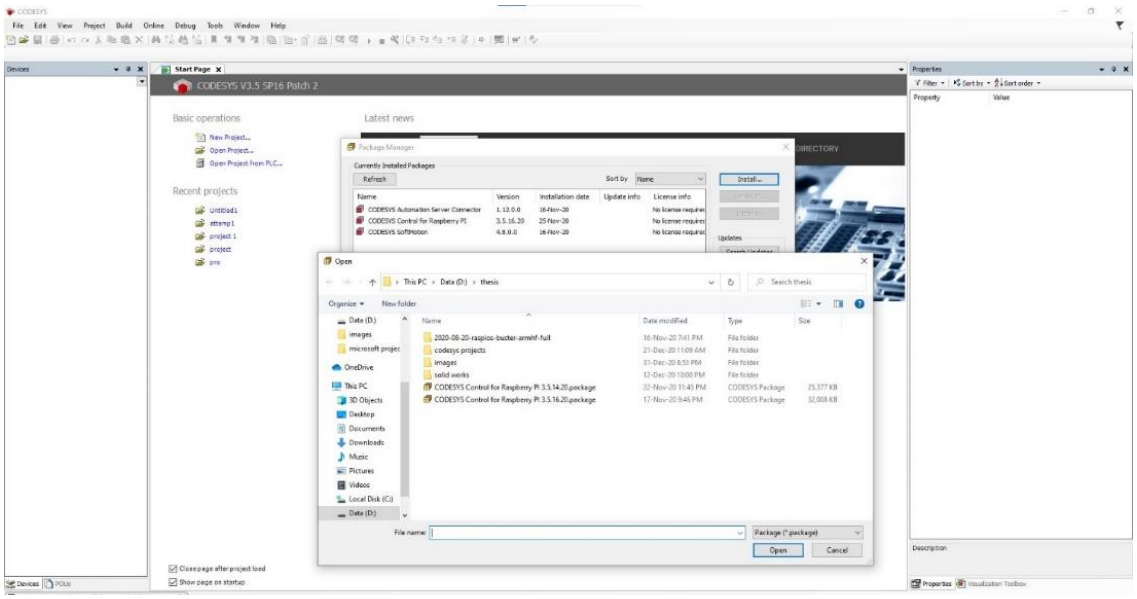

Figure 143: Installing the library we downloaded for controlling the Raspberry Pi in CODESYS software

e) After the installation finishes, we should update the library we just installed, and that is by selecting the "**TOOLS"** section, and then we choose "**Update Raspberry Pi**" as shown in the photo below:

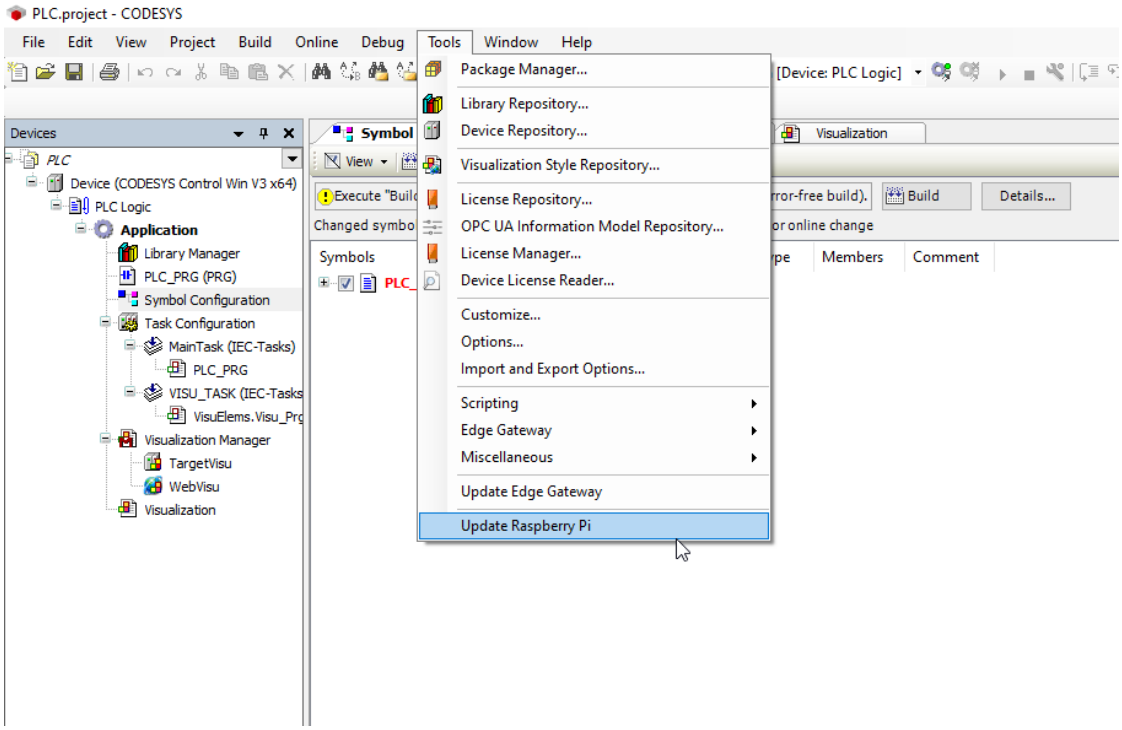

Figure 144: How to update the Raspberry Pi in CODESYS software

3. After doing all the previous steps, we are now ready for writing our program, and as we said before we will write the same program that we wrote before in Automation Studio Software, it is discussed in detail in the previous chapter, we just had to make very limited changes because of the change of the PLC type, so we will use the Ladder programing language also here, and as we said before it is a very simple language to use and deal with, and also we are going to make a simple HMI system to check our program after finishing it, and to do that we should follow these steps:
   a) We should first start a new project by just pressing on "**NEW PROJECT**", then we select "**STANDARD PROJECT**", we can also change the name of our project, and then press on "**OK**" as shown in the photo below:

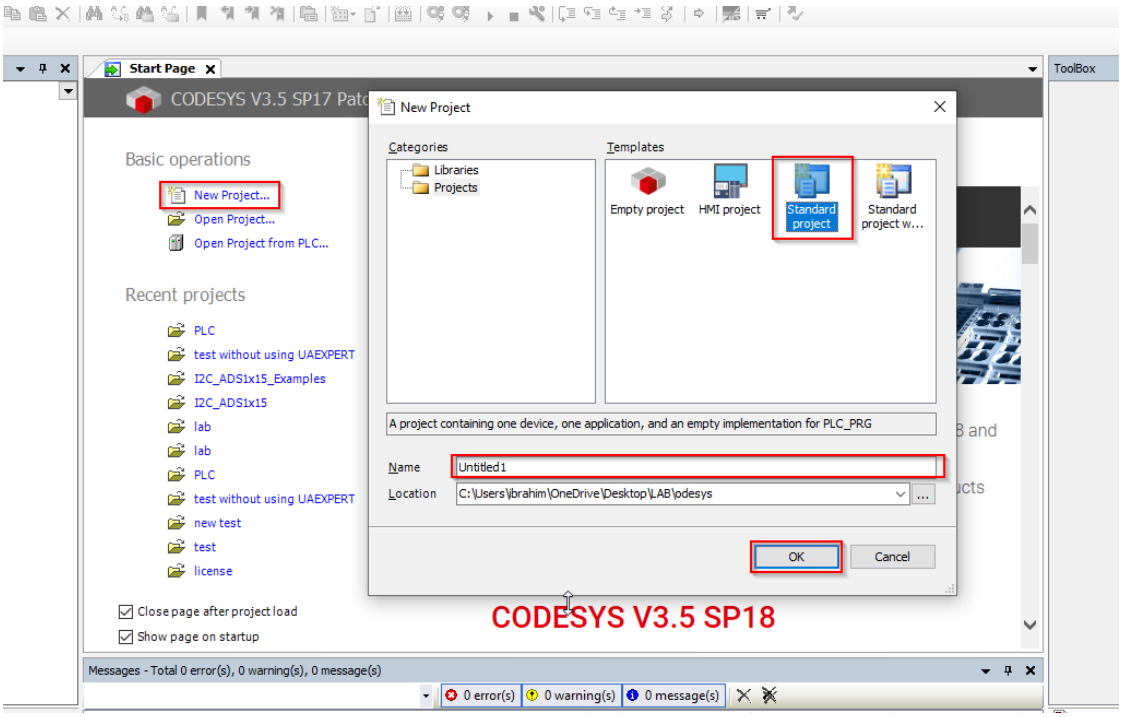

Figure 145: How to start a new project in CODESYS software

b) After that you will have to select the device, which you are going to program (which is the “RASPBERRY PI” in our case) and you must choose the PLC programing language you will use, and we have chosen the Ladder Logic Diagram (LD) as shown in the photos below:

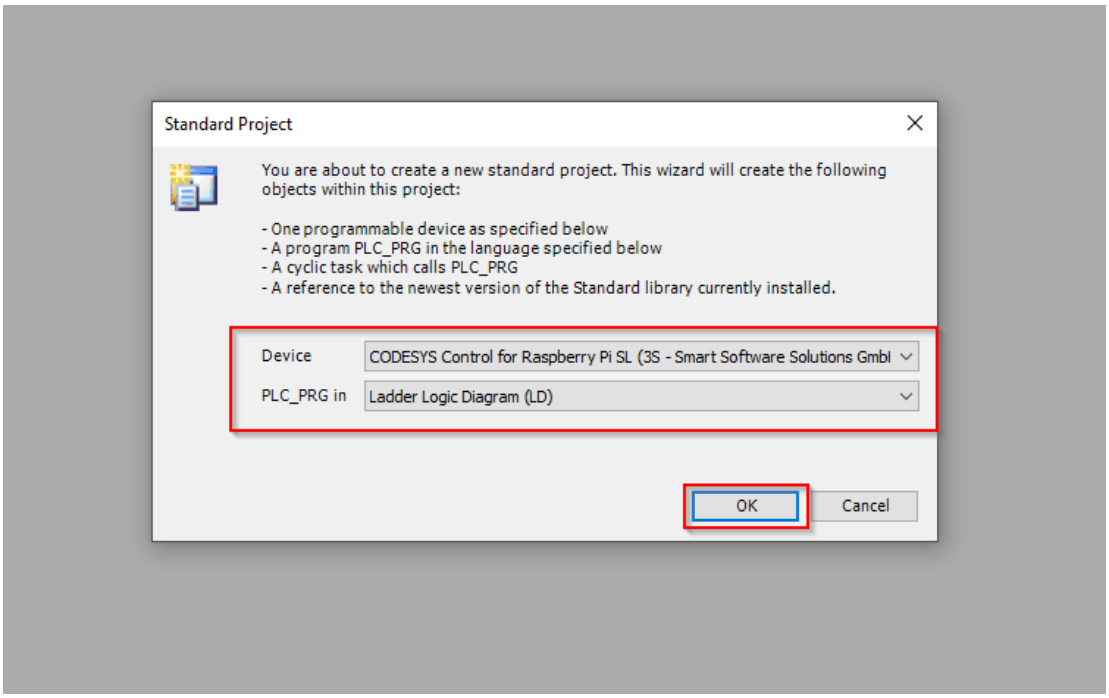

Figure 146: Choosing the device and the programing language that will be used in CODESYS software

c) Now we are ready for programming, we just must select the PLC program, then a window will show up to allow you to write your program as shown in the photo below:

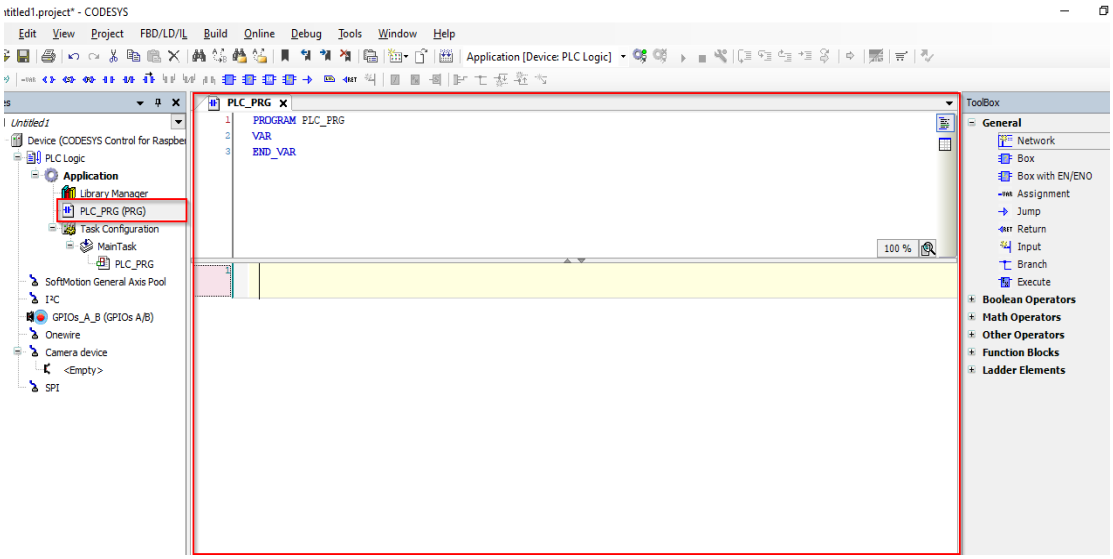

Figure 147: How to start writing a Ladder program in CODESYS software

Then we can start writing our program by just selecting the first line, and then insert an element by selecting the needed element either from the "**TOOLBOX**" on the right of our window or from the toolbar on the top of our window. After inserting the first element, a small bob will show up, we must select it, and insert the new element, and so on so forth as shown in the photos below:

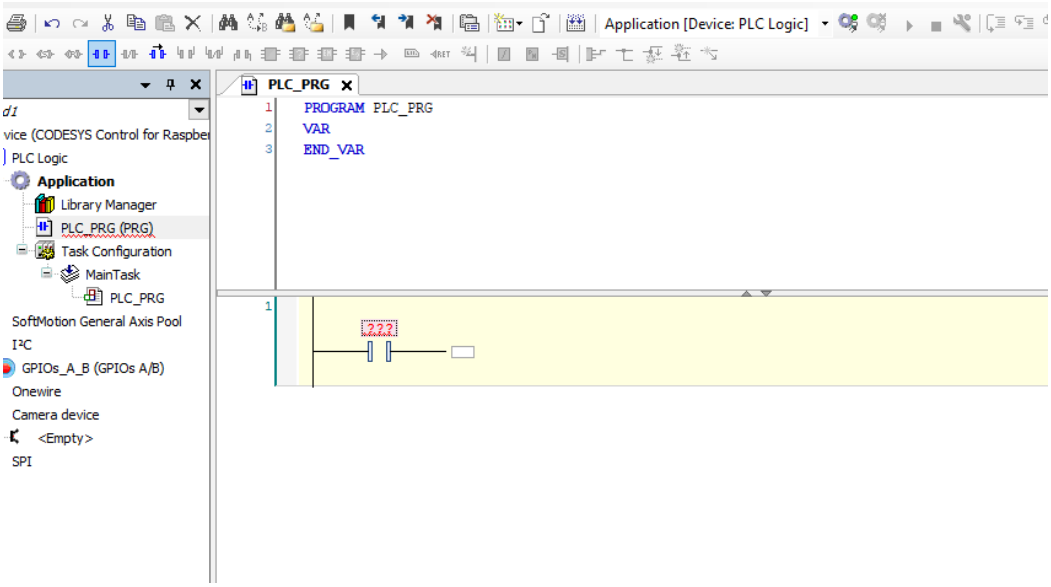

Figure 148: How to insert the first element into the Ladder program in CODESYS software

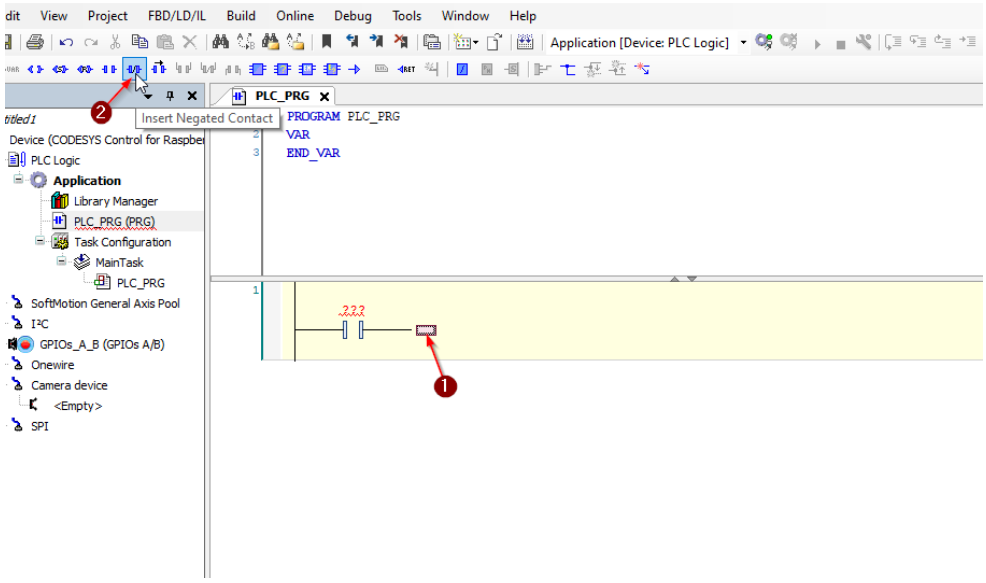

Figure 149: How to insert the other elements into the Ladder program in CODESYS software

d) After inserting all the needed elements, you should give all these elements a name and select their types, and you can do that by double-clicking on the question marks on the top of each of the elements and start writing whatever you want and then press on “Enter”. When you do that, a window will show up, it allows you to choose the type of the data, modify the name you just entered, give an initial value to your element, write a comment, and other features that can be helpful as shown in the photo below:

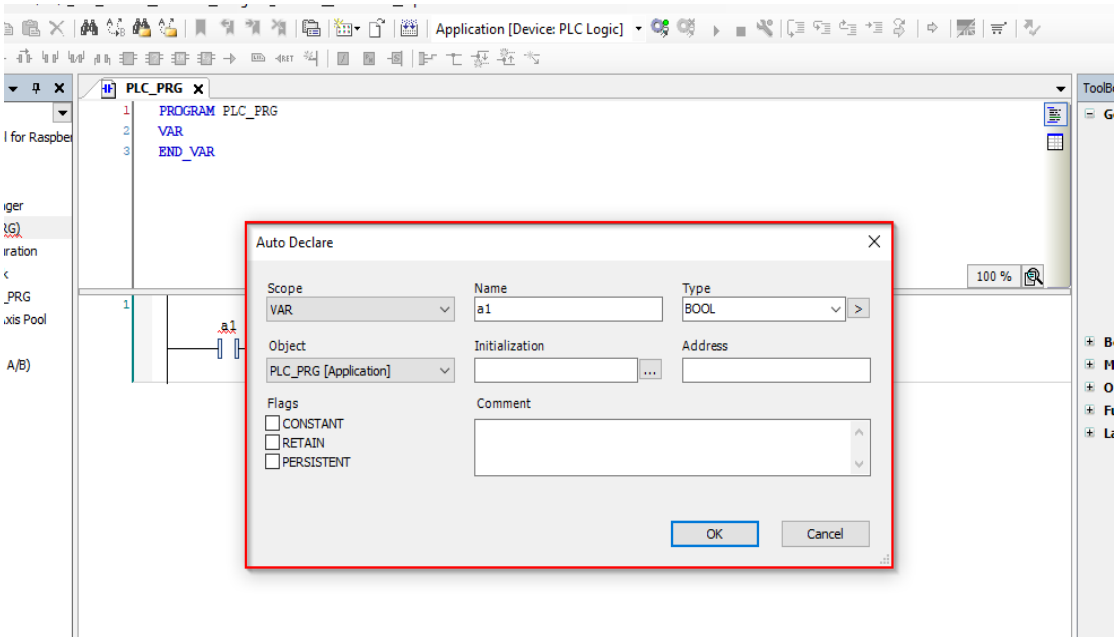

Figure 150: How to write the name of a variable element, choose the type, write the address, and choose other features of the Ladder program elements in CODESYS software

e) After writing our Ladder program, we can now start designing our HMI system, and to do that we have first to create a visualization object by clicking right on “APPLICATION”, then choosing “ADD OBJECT”, and finally we choose “VISUALISATION”, these steps are shown in the photo below:

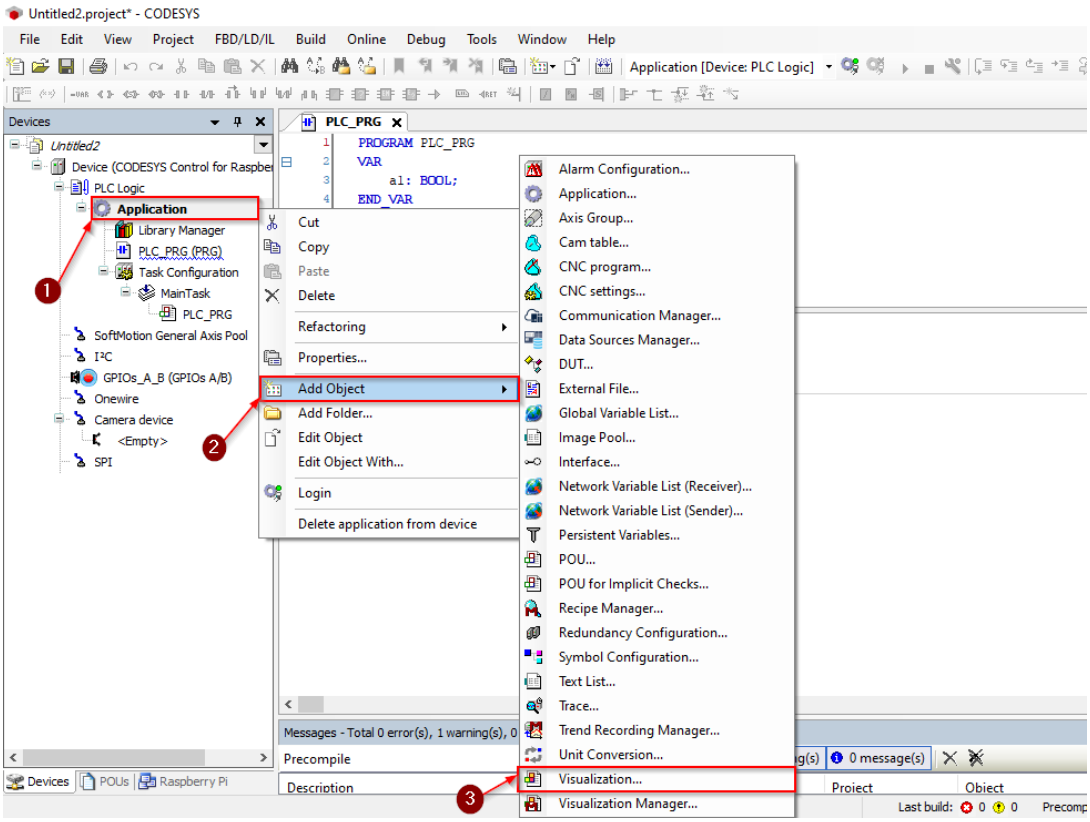

Figure 151: How to add the visualization system in CODESYS software

After doing that, you will find a new window showing up under the name of “VISUALIZATION”, and also a new toolbox under the name of “VISUALIZATION TOOLBOX” as shown in the photo below:

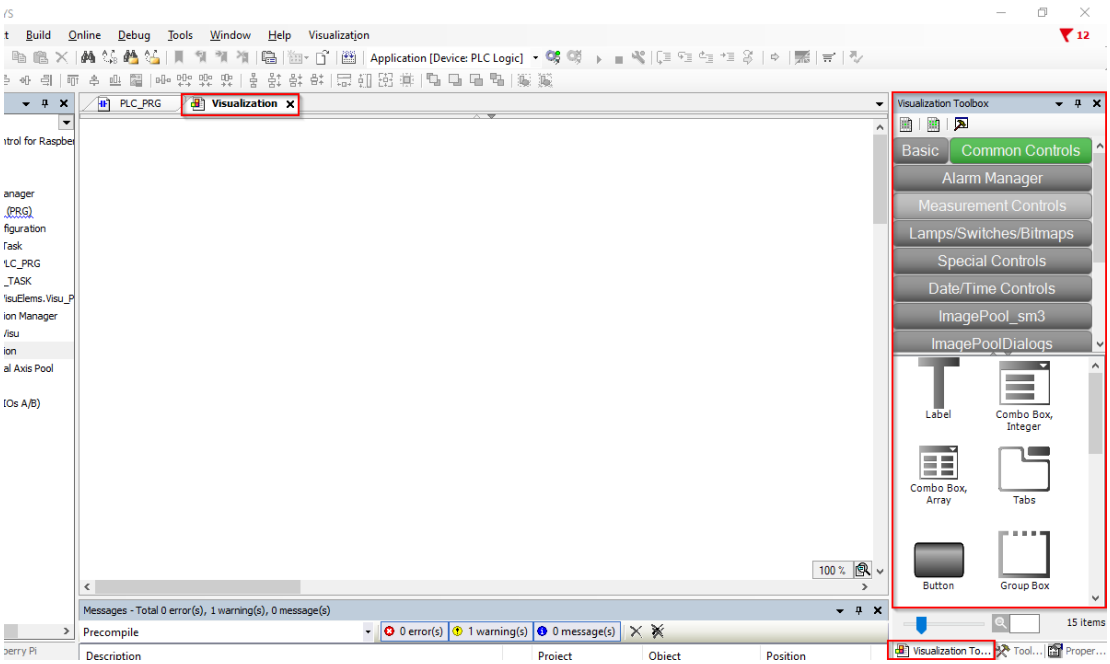

Figure 152: The visualization toolbox in CODESYS software

As you can see in the photo below, we only use buttons and lights, so we will show you in detail how to do that.

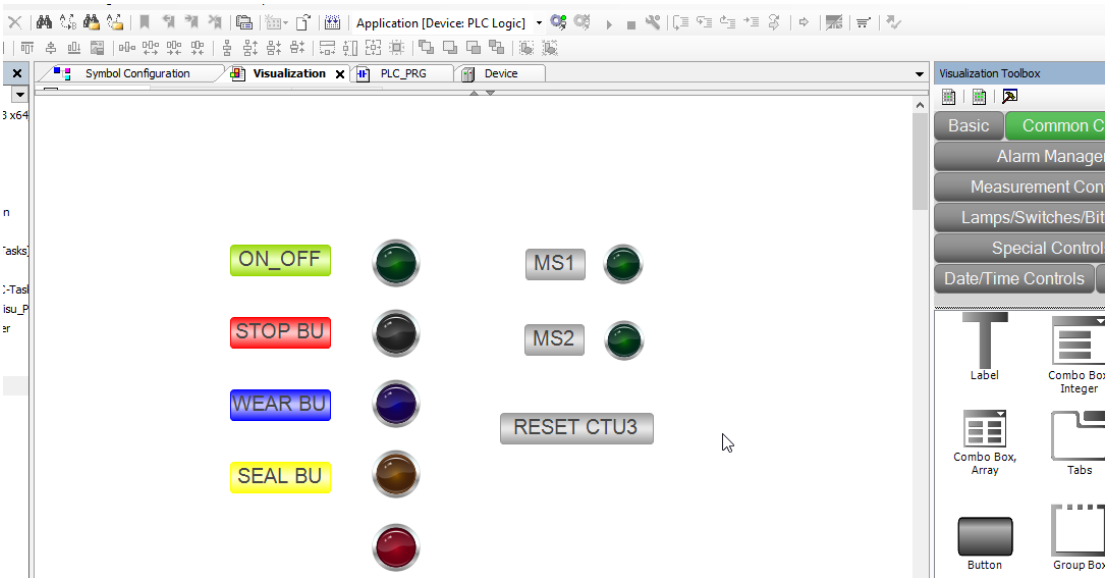

Figure 153: The HMI system in CODESYS software

To add a button, you should go to the “COMMON CONTROLS” section in the “VISUALIZATION TOOLBOX”, you will then find an element called “BUTTON”, so you can easily drag it to your diagram as shown in the photo below:

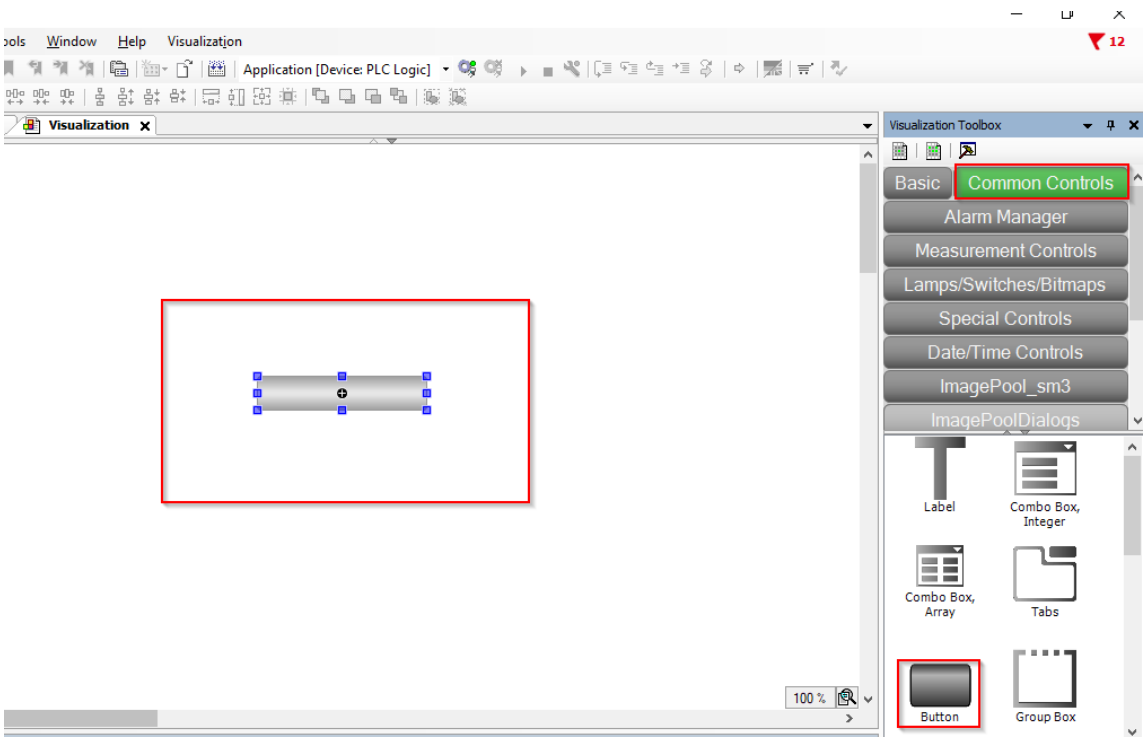

Figure 154: How to add a button to the visualization system in CODESYS software

For giving a name to your element, you should go with your mouse pointer very close to the center and make a double click, after doing that you will be allowed to write whatever you want, the photo below shows what we just discussed:

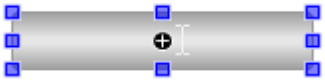

Figure 155: How to give a name to a button of the HMI system in CODESYS software

To change the color of your button, you must highlight it first, then a new window called “Properties” will show up, and from its name, it contains all the properties of your element like color, text properties, position, and many other properties that can be helpful. For the color, you can find a tree property called “Colors”, when opening that tree, you will find another tree called “Color”, you must make a double click on the right part, and then press on the gear icon that will open you a new window to allow you choose any color you prefer, also you can change the transparency of your color, the photo below shows what we just discussed:

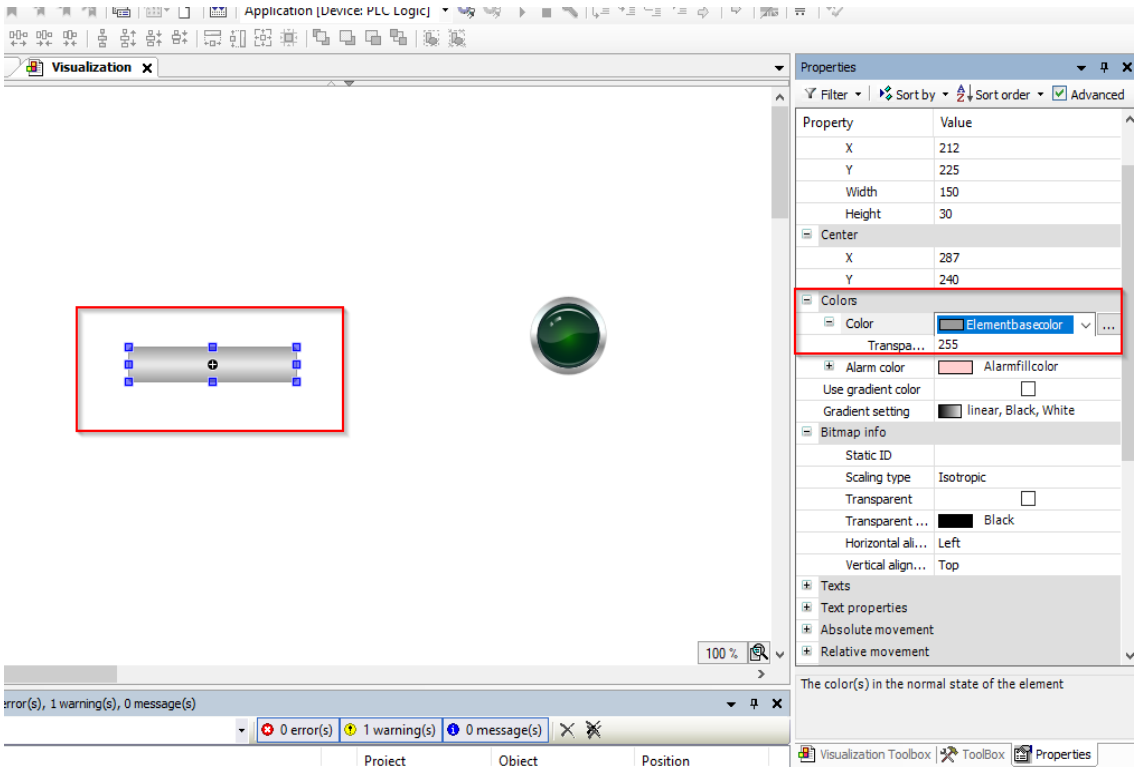

Figure 156: How to change the color of a button in the HMI system in CODESYS software

To add a light, we will follow the same steps as we did to add a button, but this time, we will go to the section called “Lamps/Switches/Bitmaps”, then we choose “Lamp” and drag it to our diagram as shown in the photo below:

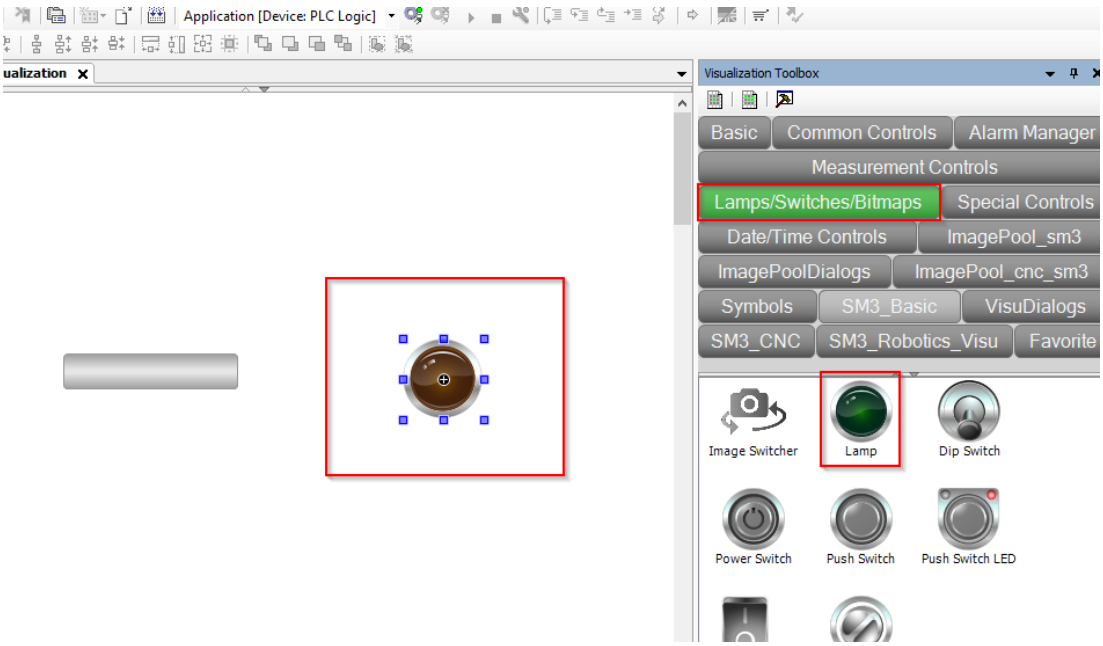

Figure 157: How to add a lamp to the HMI of CODESYS software

To change the color of your lamb, we will do the same steps as we did with the button, but this time we will go to another property called “Background”, and under it, we will find another property called “Image” from which we can change the color by double-clicking on the right part, and choose one of the colors as shown in the photo below, and when we go back to our diagram, the color will be changed into the chosen color.

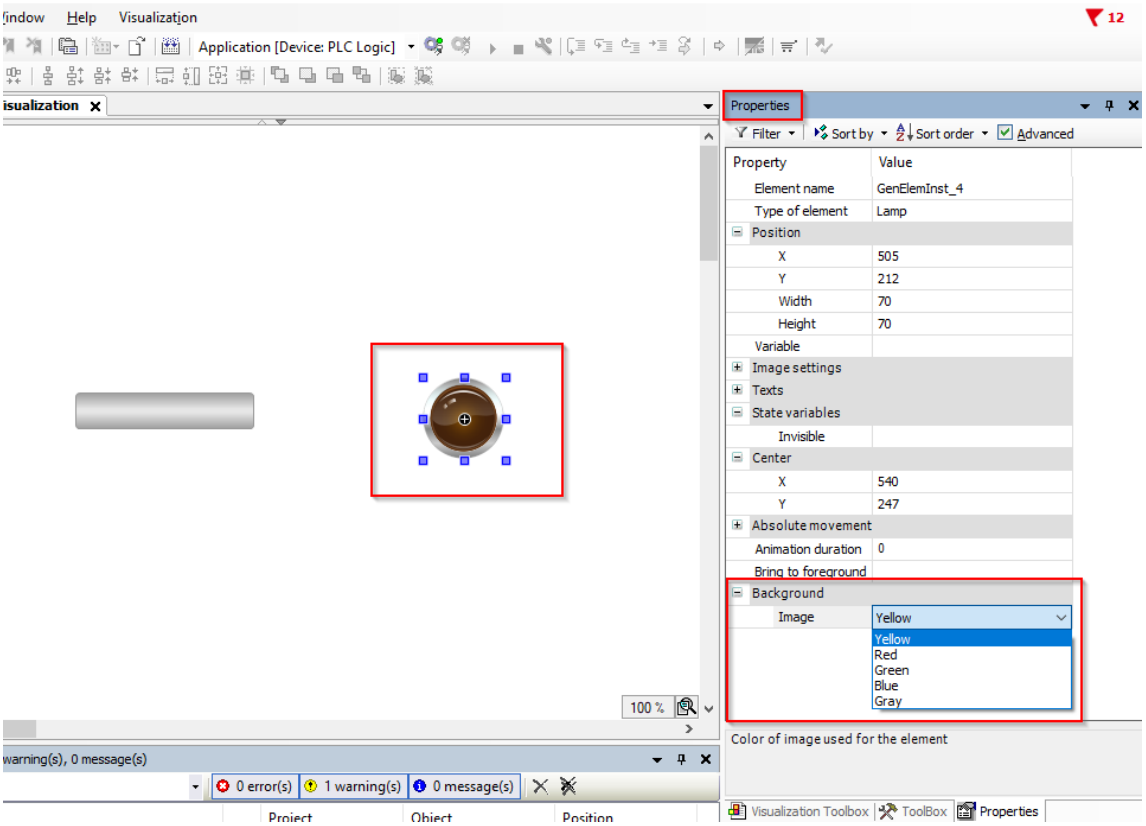

Figure 158: How to change the color of a lamp of CODESYS HMI system

NOTE: We used two buttons called MS1 and MS2 which are connected to the two outputs in the ladders program which are indicating the on-state of the two sensors used for each cylinder to indicate its extension and retraction. We had to do that manually as we can’t simulate these sensors here in CODESYS software as we did in Automation Studio software, and that is why we decided to find a way that will be discussed later to make integration or communication between the two software so that we can get the advantages of each of them. We also use an extra button for resetting counter no.3 manually, just for testing reasons, so it is not mandatory.

f) Now after finishing the design of our HMI, we need to connect all our HMI elements to our ladder program. We will start first with buttons and see how to connect them as needed in our project. In the beginning, you must highlight the button which you are going to connect with the Ladder program, and as we said before, the properties of our button will show up, and so under the "Input Configuration" tree which is the last property, then you will find another tree called "Tap", you should open it, then you will find a property called "Variable" which is the property we need to make the connection, and so as we did before, you must make a double click on the right part which is related to the value of the property, and the press on the gearbox which will show up after doing that, a new window called "Input assistant", then under the tree called "PLC_PRG", you can find all the variables you have in your ladder program, and so you can choose the needed variable to be connected with the highlighted button in your HMI system as shown in the photo below:

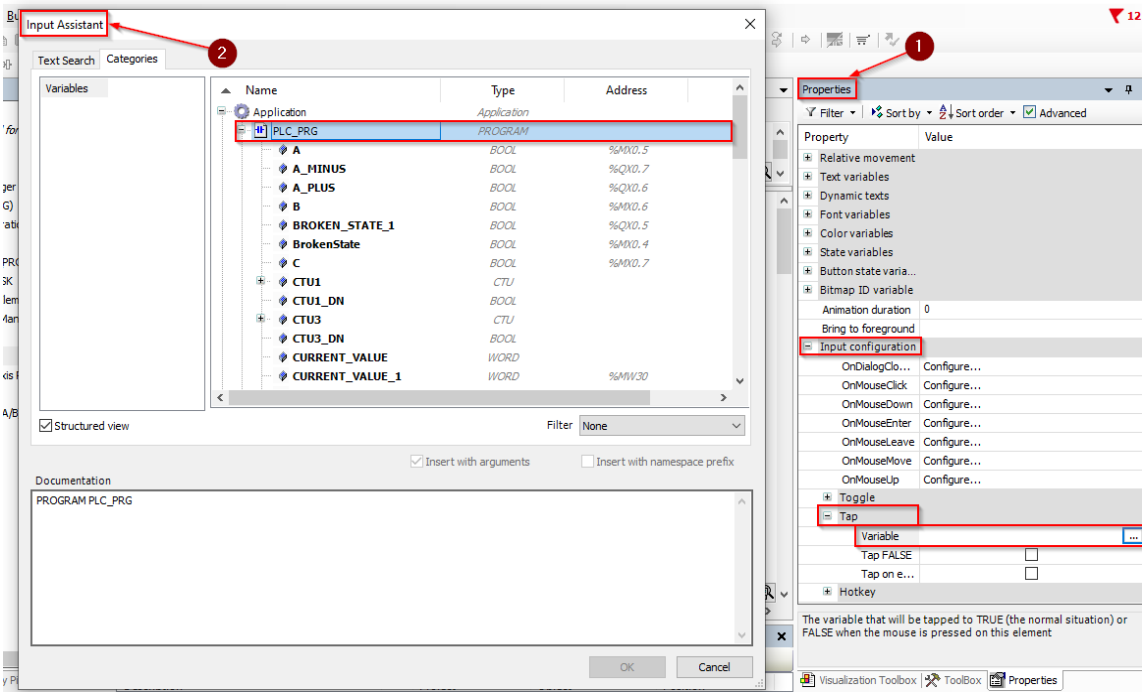

Figure 159: How to make a connection between a button in the HMI system and a variable in the ladder program (CODESYS software)

For lamps, you will do the same except that you will find the “Variable” property under another property called “Position”, and you should do exactly like what we did with buttons as you see in the photo below:

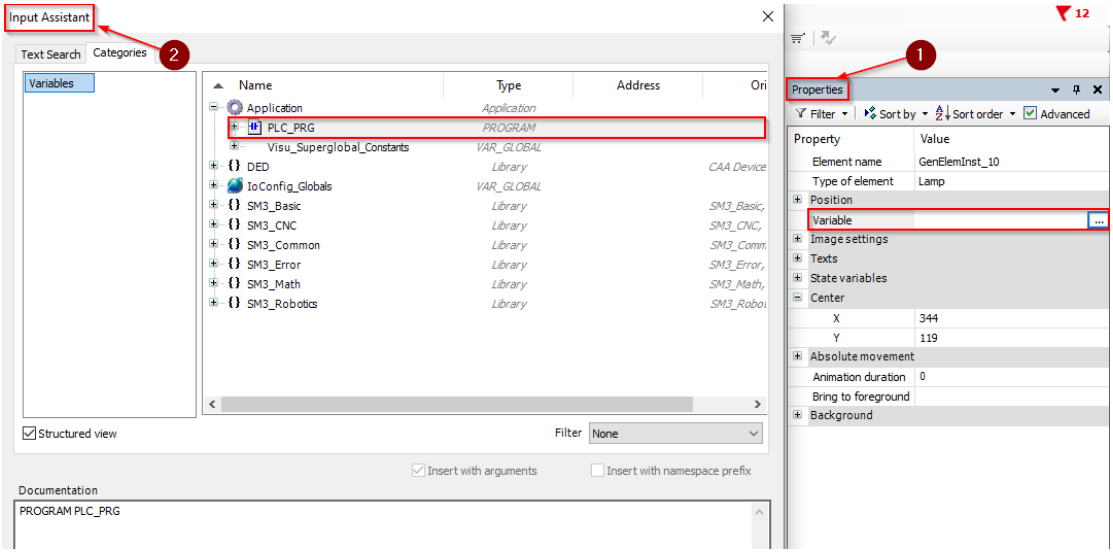

Figure 160: How to make a connection between a lamp in the HMI system and a variable in the ladder program (CODESYS software)

Now we have everything ready to work perfectly, and to check that we don't have any errors in our ladder program, we must press on that symbol ( ) to log in, and if we got zero errors, we could start visualizing our HMI system as shown in the photo below:

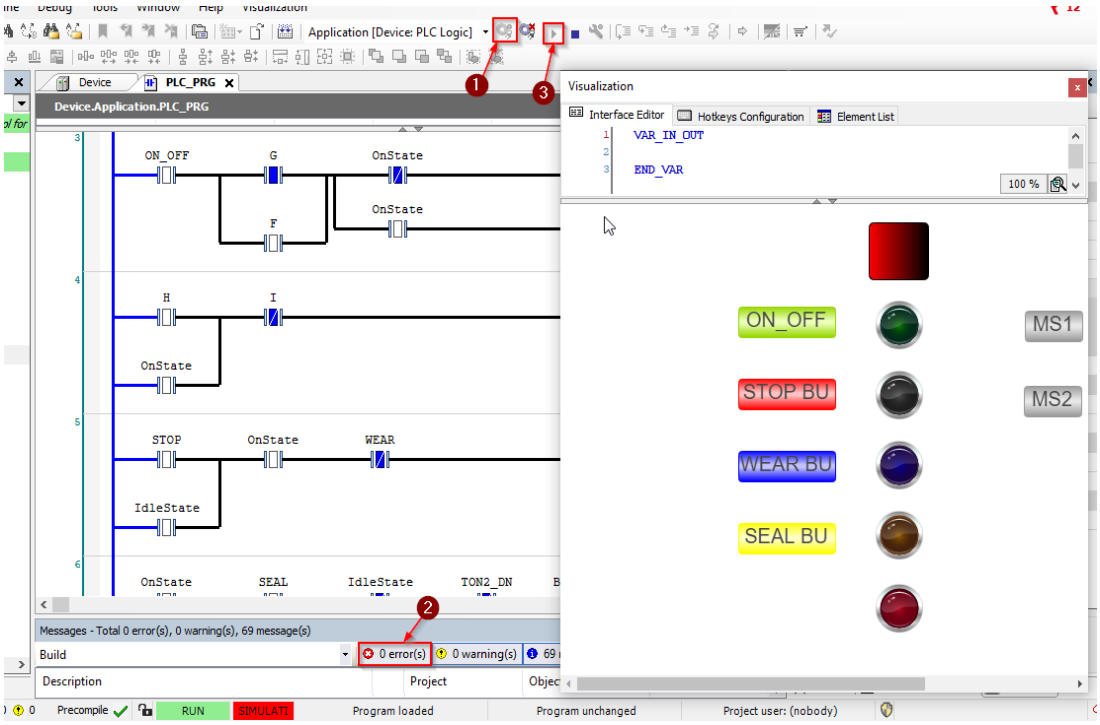

Figure 161: How to debug and simulate a CODESYS project.

This simulation is done on the virtual PLC of the program, we can now connect our Raspberry Pi, and see our project in real, you can make a wireless connection between CODESYS and Raspberry Pi by just going to the section related to Raspberry Pi, the write the username and the password of your Raspberry, then scan for the IP address of your Raspberry, and finally press on “Install”, but take into account that you should do some preparations on Raspberry first before doing that, and we could discuss these preparations later on. The photo below clarifies more what we just said:

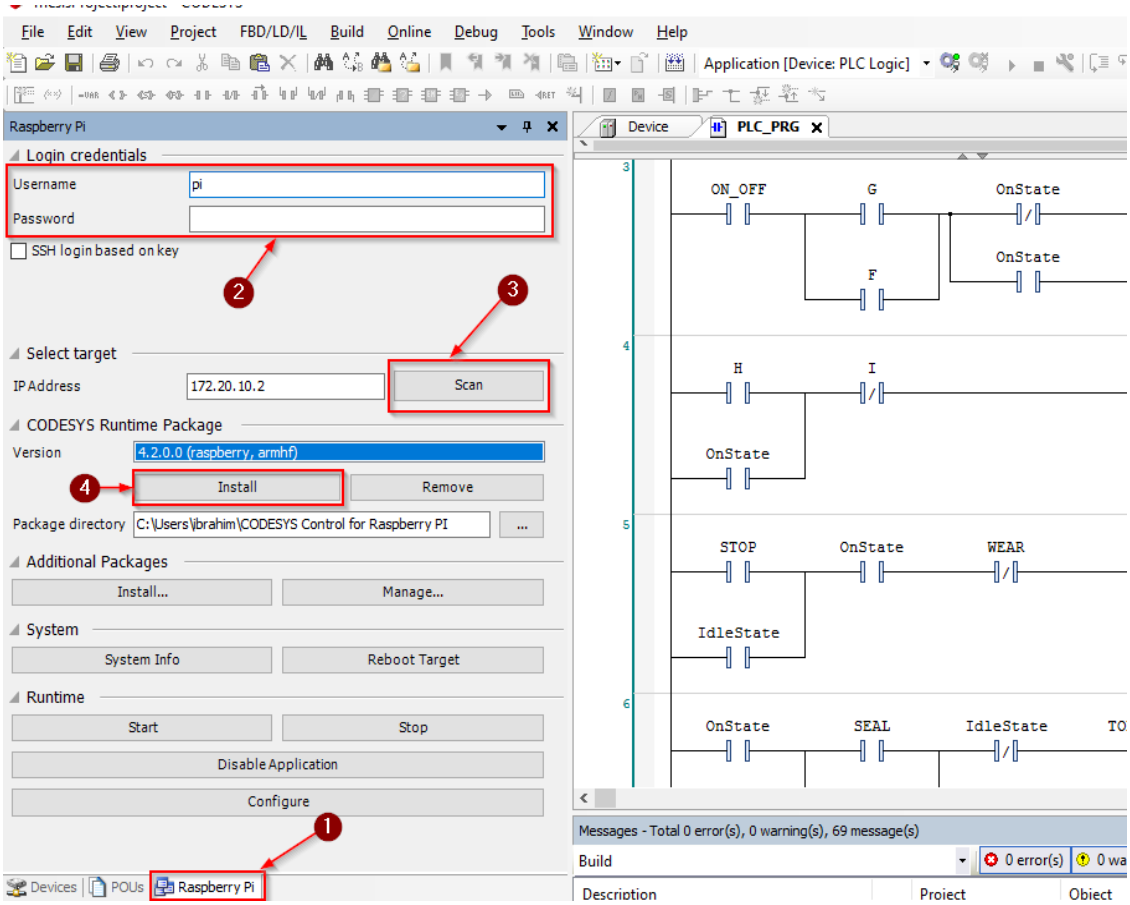

Figure 162: How to make a wireless connection between CODESYS software and Raspberry Pi

In the next chapter, we are going to know how to make an integration between Automation Studio software and CODESYS software and why we decided to do that.

4 Integration between Automating Studio & CODESYS software

4.1 Why did we decide to make an integration between Automating Studio& CODESYS software!?

- The reason why we decided to make this integration is that it is not possible in CODESYS software to design an electro-pneumatic system like we already did in Automation Studio software, so the idea was why not to integrate the two software and get the features and advantages of both of them, and that is what we already did, and that allows me to simulate the Ladder Program (in CODESYS) using the electro-pneumatic system in Automation Studio, and we will explain in detail how to do that.[2]

4.2 How could we make an integration between Automating Studio& CODESYS software!?

1. In CODESYS, the local discovery server in OPC UA uses port 4840 by default, and this port is used to communicate with other applications or software. The problem here is that there is some conflict in communication with Automation Studio, so we must change this number, and to do that you should follow this path on your PC (C:\ProgramData\CODESYS\CODESYSControlWinV3x64\7F8584BF), and note that this no. "7F8584BF" differs according to the installation, so you will have a different no., also note that we have chosen this folder "CODESYSControlWinV3x64" because that is the "Device" we have chosen when creating the project in CODESYS. After doing that, we must open a file which is a configuration file called "CODESYSControl.cfg" by using Notepad or Notepad++ software, and then as we have too many configuration sections, we can search for the needed section which is "CmpOPCUAServer", and then you must remove the semicolon (;) before the first line under this configuration, by doing that, CODESYS will use "NetworkPort= 4841" instead of "4840", then you have to save the file, these previous steps are shown again in the photo below:

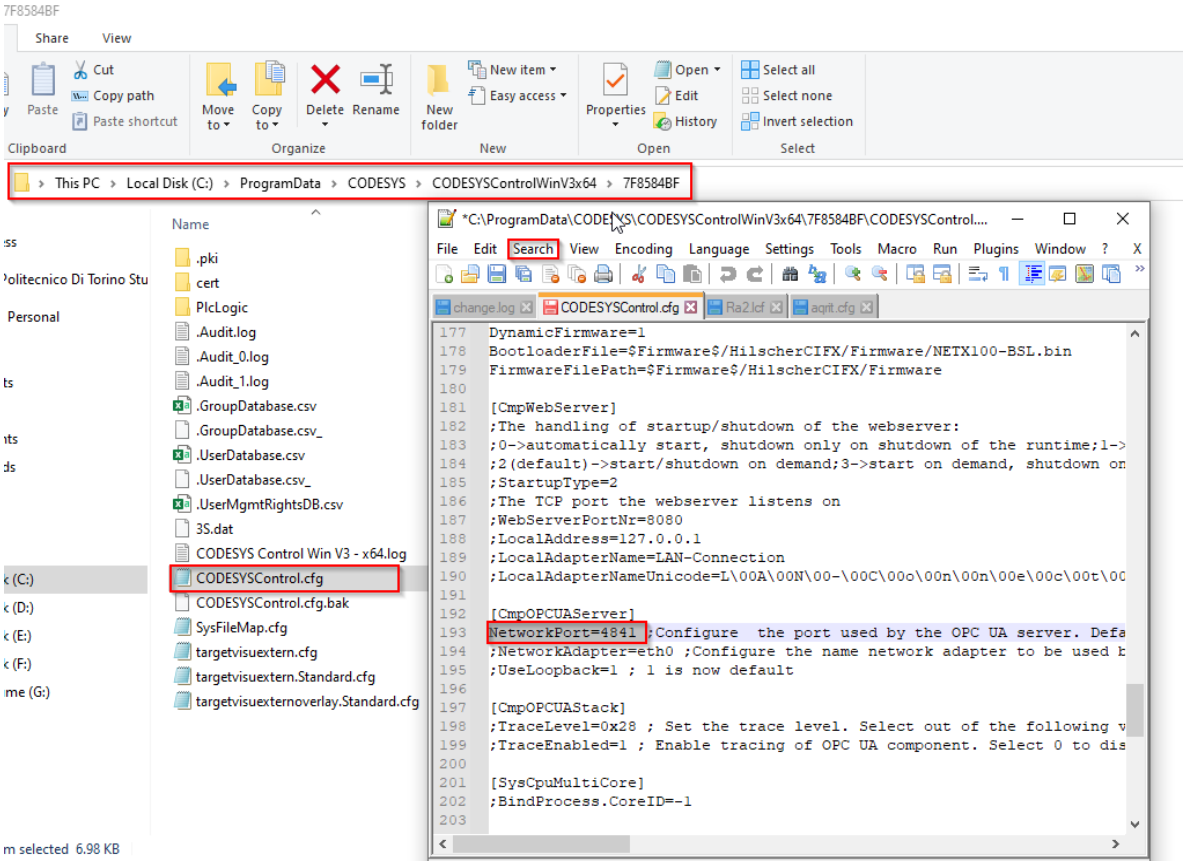

Figure 163: How to change the local discovery server in OPC UA that uses port no. 4840 to let it use port no. 4841

2. After doing these previous steps, you should open CODESYS software and begin a new project as usual, but this time we will choose a different device which is the one we have chosen in the previous step (CODESYSControlWinV3x64), if you already have a project saved, you can open it and change the device from the inside of CODESYS by making a right-click on the device you have in the "Device section", and then choose "Add Device", a new window called "Add Device" will show up, so you choose "Update Device", then you select the device we just mentioned, and press on "Update Device", and that is all as shown in the photo below:

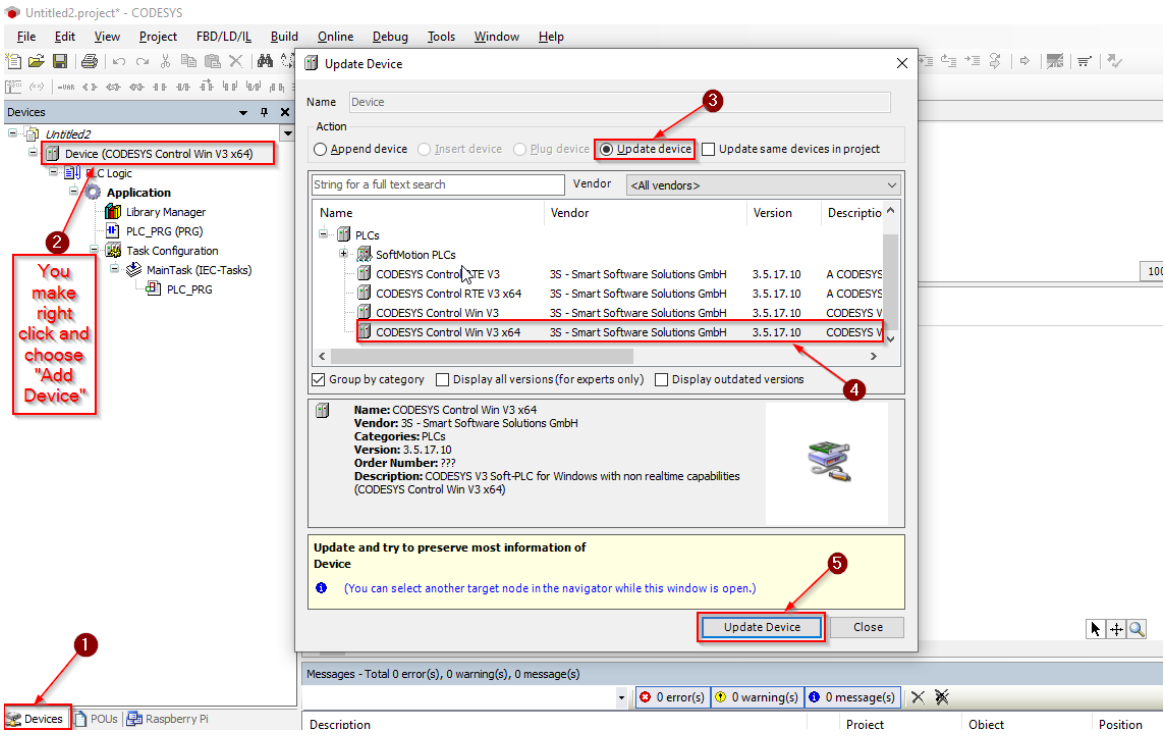

Figure 164: How to change the device of your project

Now we should start our virtual PLC, make the connection between it and our PC, the green led on the PLC will be on when the connection happens, and then download our program on it as shown in the photos below:

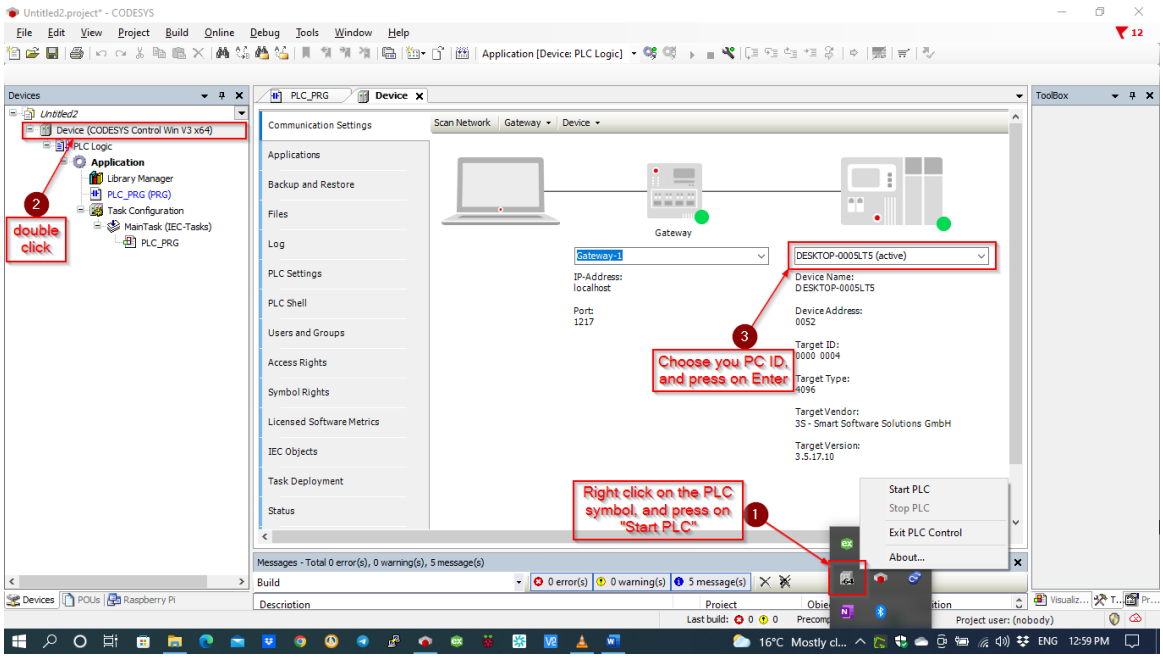

Figure 165: How to connect the PC with the virtual PLC of CODESYS

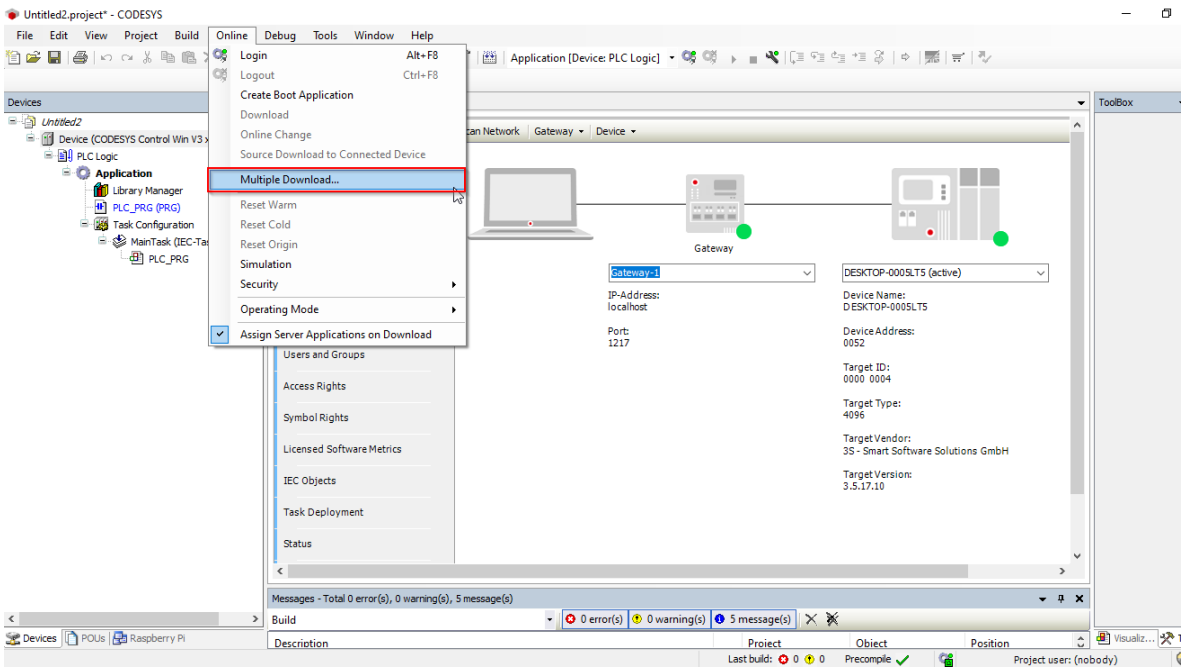

Figure 166: How to download the Ladder program on the CODESYS virtual PLC

3. Now we should go to the Automation Studio software, design our electro-pneumatic system, and then we go to “Tools”, then we open the communication manager, we need to connect the OPC of Automation Studio to CODESYS, so we should go to “OPC Client UA”, then we add the OPC server manually by write in the part related to “URL”, you write “opc.tcp://localhost: 4841”, then you connect by pressing on the true mark as shown in the photo below:

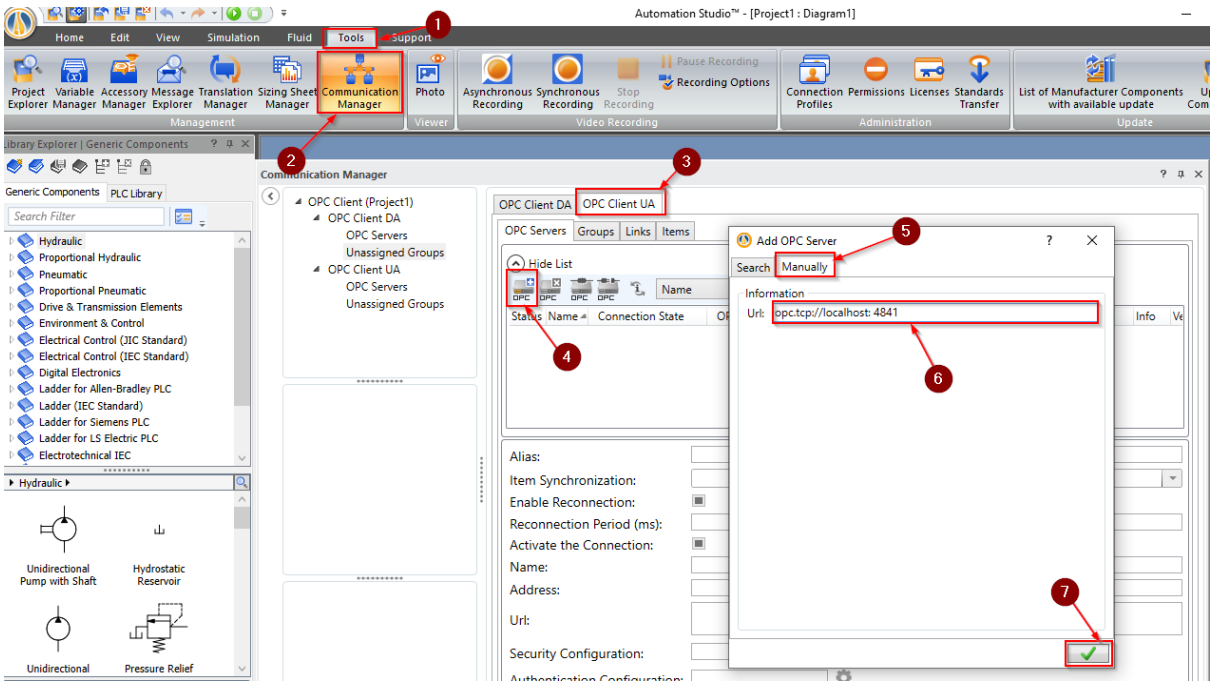

Figure 167: How to make a connection between Automation Studio Software and CODESYS software

NOTE: After the connection wasn't allowed because of some security policies, you should follow these steps and redo what we just did once again, you should download free software called "UaExpert" by following this link (https://www.unified-automation.com/products/development-tools/uaexpert.html) after you download and install the software, you open it, then you "Add server" by clicking on the plus mark in the tools bar, then you write "opc.tcp://localhost: 4841", then you press on ok, and when you open the server we just added, and then we make a connection as shown in the photos below:

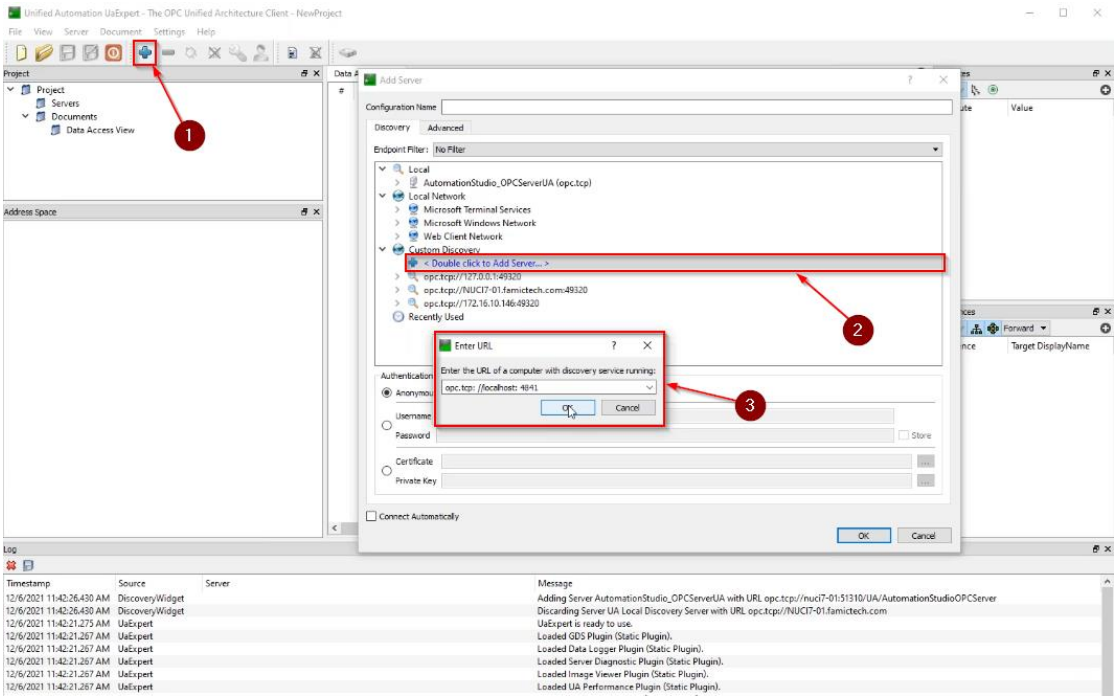

Figure 168: How to add an OPC server in UaExpert

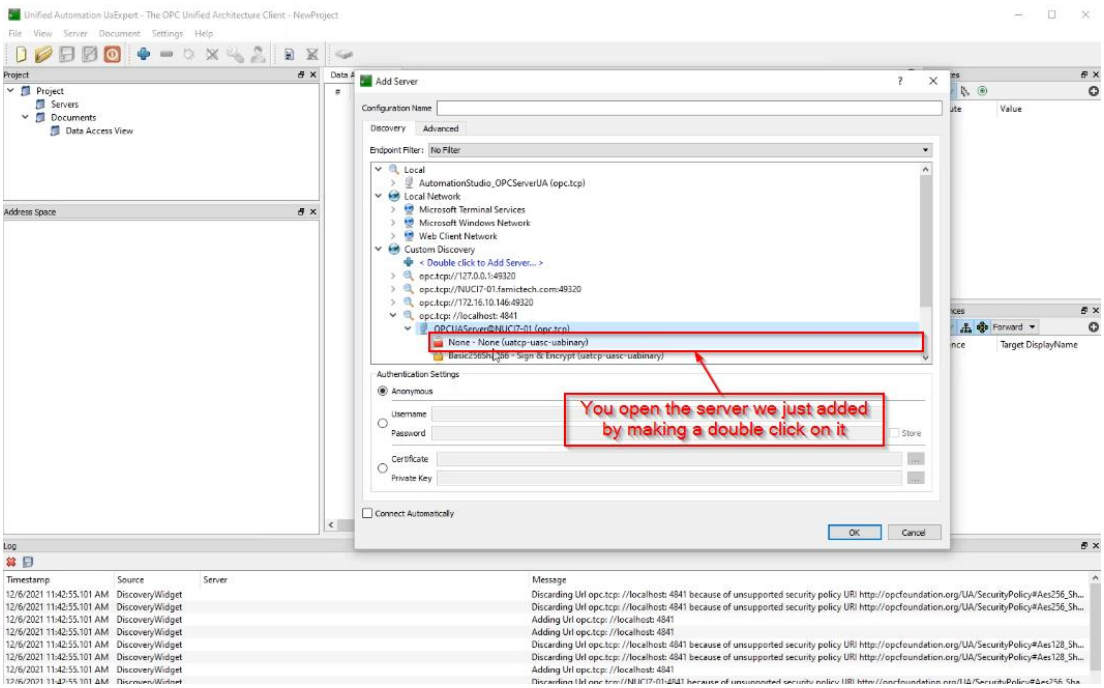

Figure 169: How to open the added server in UaExpert

When you try to make the connection, the program will tell you that the certificate is not trusted, and this is the same problem you may have in Automation Studio, so for solving the problem in UaExpert software, we just should trust the certificate and press on “Continue” as shown in the photo below:

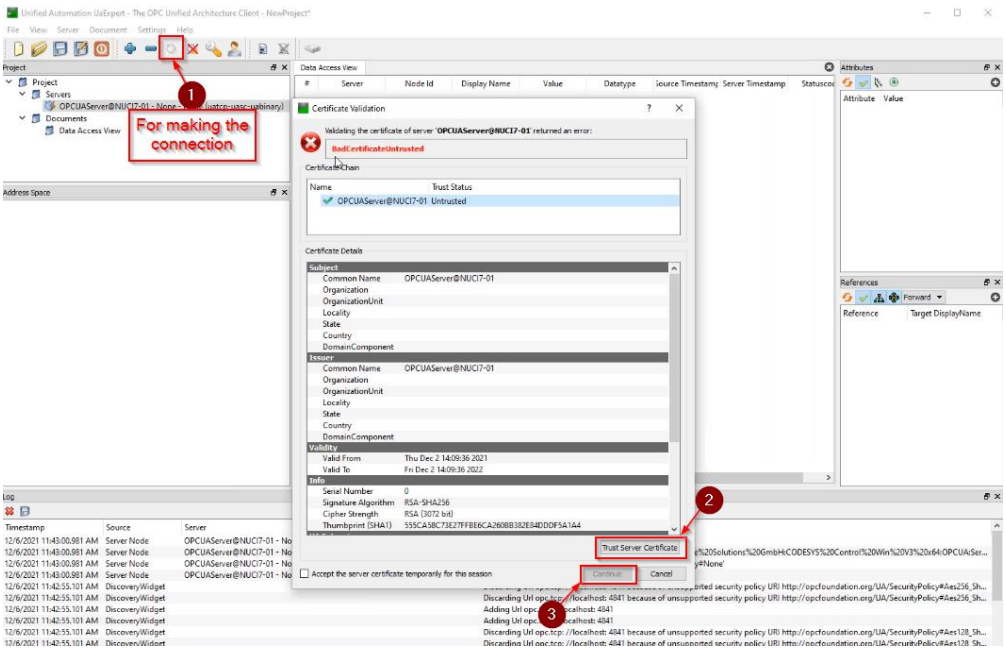

Figure 170: Trusting the certificate and making the connection

After the connection is done in UaExpert software, we must manually copy the certificate to the area where Automation Studio can recognize it, so we go to “Settings”, then “Manage Certificates”, and then “Open Certificate Location”, and then we copy the needed certificate related to OPC server, and paste it in the area shown in the photo below which is related to Automation Studio software:

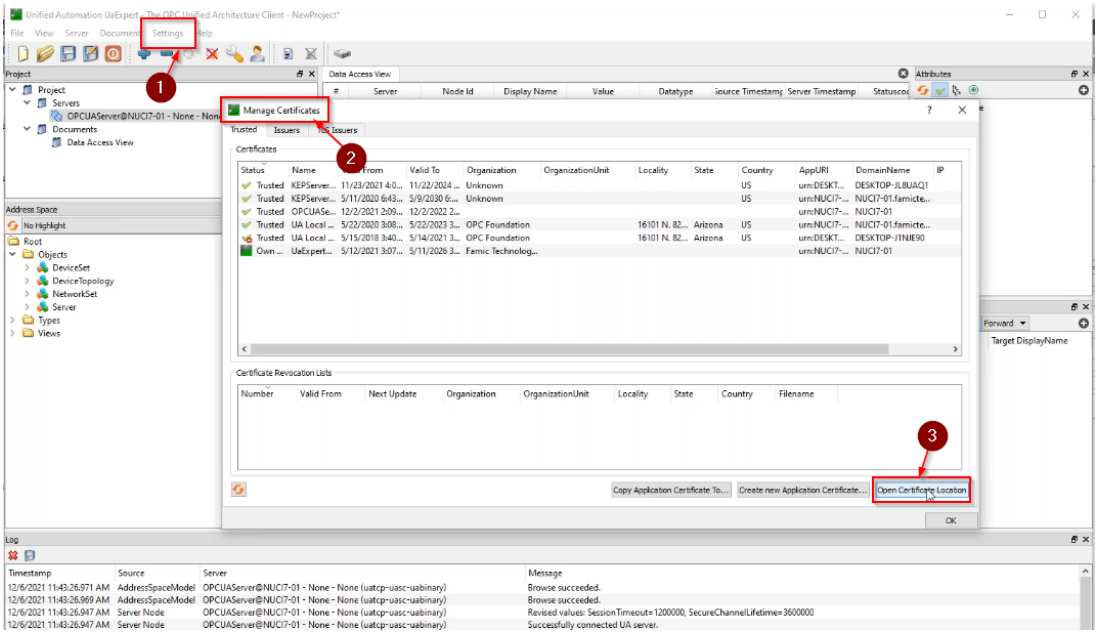

Figure 171: How to find the certificate in UaExpert software

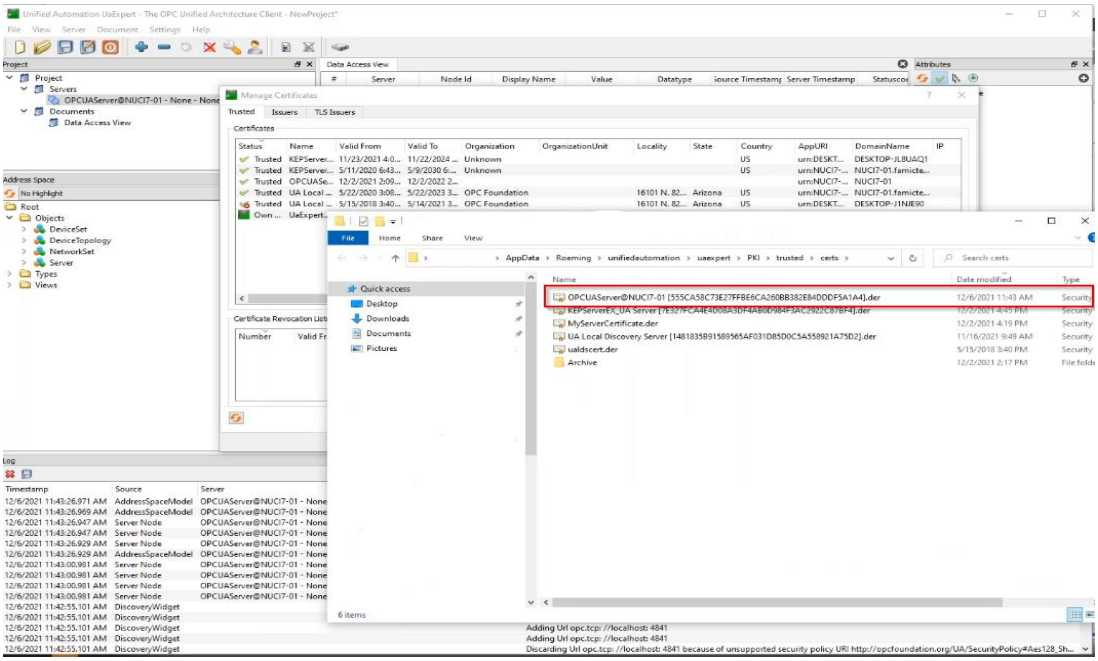

Figure 172: The certificate

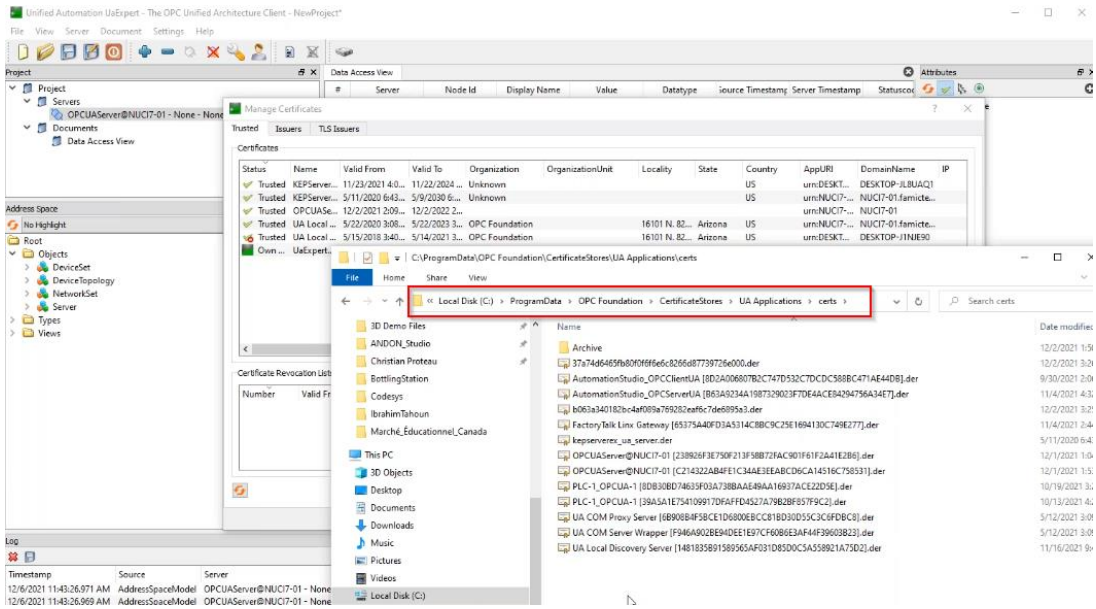

Figure 173: The Place where we can add the trusted certificate of the OPC server to Automation Studio

After doing these steps, you should go back to Automation Studio and redo the steps we mentioned before to make the communication between Automation Studio and CODESYS software, when the connection is done, you should have this green led on as shown in the photo below:

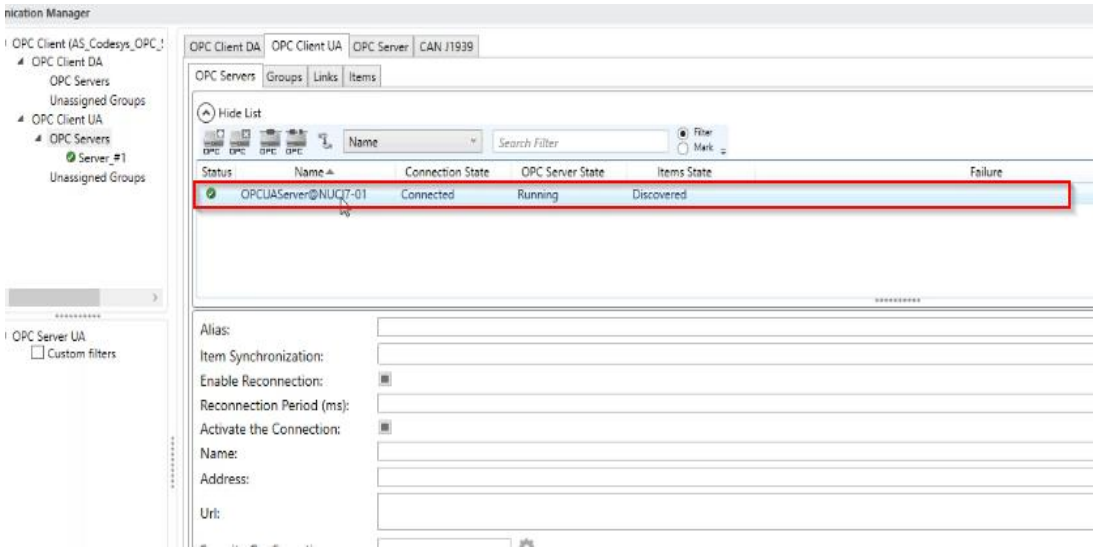

Figure 174: Indication that the connection is done in Automation Studio software

4. After the communication is done, we must “Add Group” and link it to the server we added in the previous step as you see in the photo below:

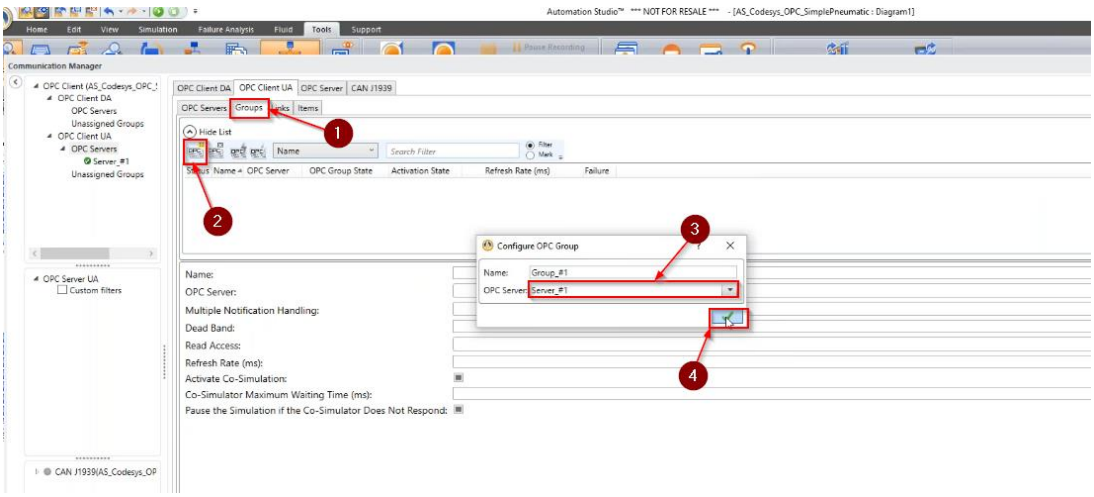

Figure 175: Add a group and link it to the server

Now we should link the variables of CODESYS and Automation Studio software together, and to do that, we should go to “Links”, then choose the group we just added, then go to our Ladder program variables in CODESYS which we will find under the “PLC_PRG” as shown in the photos below:

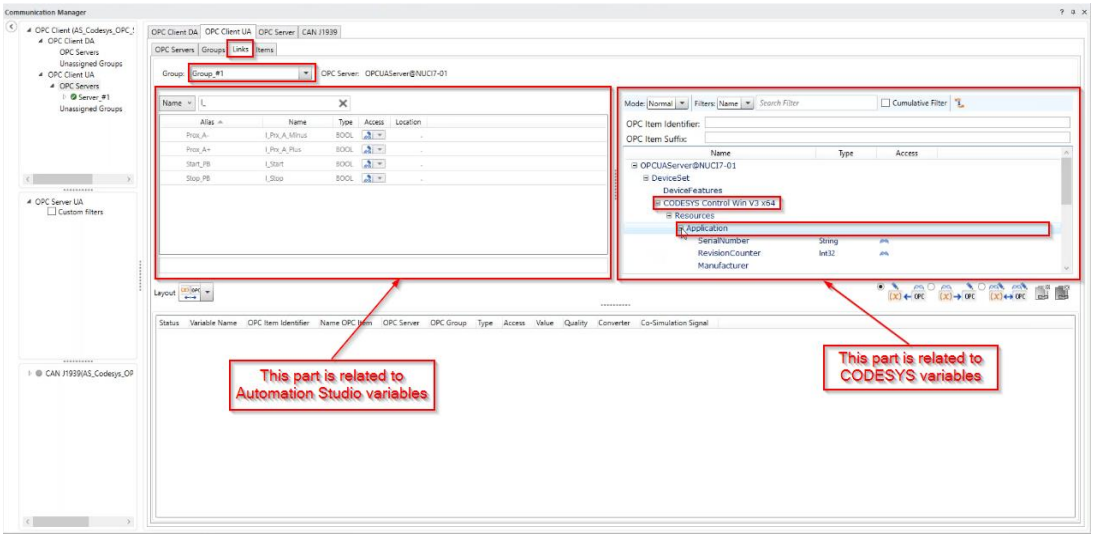

Figure 176: The part where we can start linking the variables of CODESYS and Automation Studio software

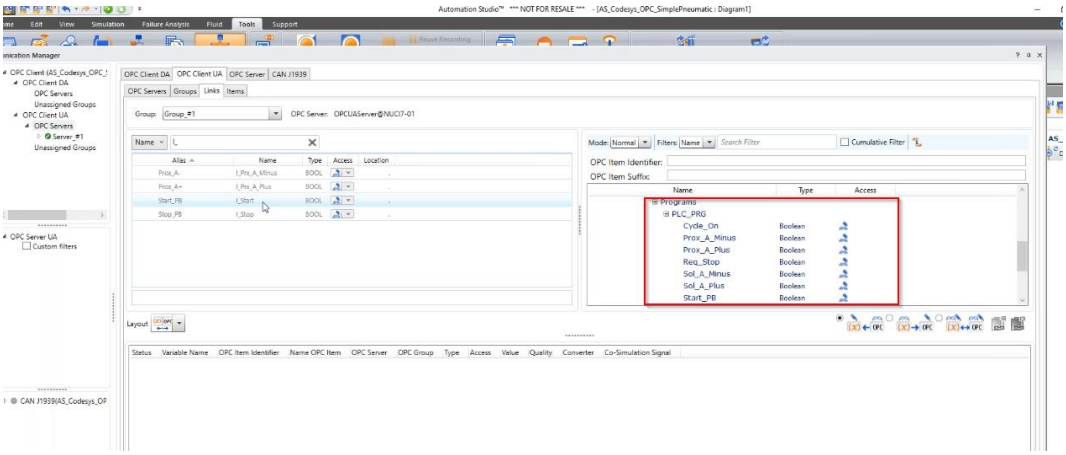

Figure 177: The place where we can find the variables of CODESYS

For example, to link a Proximity sensor in Automation Studio to CODESYS, we should select the two variables of the two software, then select the type of communication (From Automation Studio to CODESYS software because it is an input), and then make the communication as shown in the photo below:

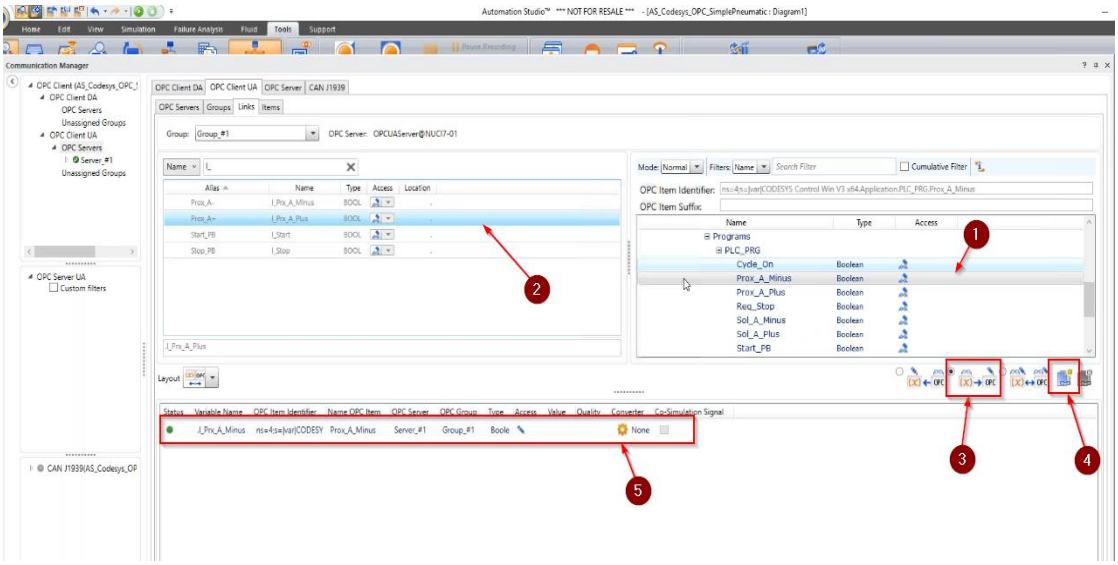

Figure 178: How to make communication between variables of Automation studio and CODESYS software

We can also make a connection from CODESYS variables to Automation Studio by choosing the other communication type. After that you must go online in both software as we already discussed, and then start the simulation, you will find that the Ladder program of CODESYS is operating the electro-pneumatic system in Automation Studio.[2]

Conclusion and future development

The test bench of electro-pneumatic cylinders for checking the leakage problems, analyzing it, and giving a complete solution and cheap one at the same time, it is a complete mechatronic project that has several fields of what we studied like designing, controlling, and manipulating data acquisition in an integrated simulation environment.

In this thesis, we started by explaining the importance of pneumatic cylinders nowadays, especially in the industry field, and how it is very important to be sure that they work very well and do their functionalities as expected. WE SUCCEEDED TO ACHIEVE the objectives set which is the designing of an electro-pneumatic system, designing a complete HMI system that allows any user to operate the project very smoothly, using the Ladder programing language for programing our PLC controller (which is used only for simulation, but our main controller in the project is the Raspberry Pi), designing a complete SCADA (Supervisory Control and Data Acquisition) system that allow any user to know everything about the project even if he/she is far away from it, and finally simulating all the project to see if everything goes as expected, all these tasks were done by using "Automation Studio" software because it has a lot of features and capabilities that allow us to do all of that, and as this software is not compatible with our project controller (Raspberry Pi), we had to search for another software so that we can program our controller, and this software is called "CODESYS", and as it doesn't have a lot of the needed features and capabilities for our project like designing and simulating a complete electro-pneumatic and HMI systems, so we searched and found a way for communicating these two software together, and let them become like a single software, and so we can have all the feature and capabilities we need for our project. In the end, we have a complete designed and simulated electro-pneumatic system (Test Bench) that can make a leakage test for one of the two cylinders we have in our project separately or even simultaneously, and we succeeded to use only the "Raspberry Pi" device for doing all of that which reduces too much the cost of the test bench.

Future developments can be by developing the system to be able also to detect the internal leakage which can happen inside the chamber of the cylinder itself because of the damaged seals or wipers located at the front end of the cylinder that must make pressure sealing to prevent leakage of compressed air from the space between the piston rod and the front end bearing, as in our system here, it is only possible to detect the external leakage which is the compressed air that goes out of the cylinder chamber. Also, in the future, it is possible to build our designed and simulated test bench to validate in real what we have done and compare the real prototype with the theoretical methods and software simulations. Also, the pressure reading values of the pneumatic pressure transducer in our project can be very helpful to allow you in the future to find a way to predict what can happen to the cylinder after some working time in terms of performance because of the expected leakage.

References

[1] https://www.famictech.com/en/ (Famic Technologies Inc., 2021/2022)

[2] Interviews with Eng. Georges Fattal from Famic Technologies company (13th of November 2021, 15th of November 2021, 27th of November 2021, 5th of December 2021, 9th of December 2021)

[3] https://pimylifeup.com/what-is-raspberry-pi (PiMyLife, 2022)

[4] https://instrumentationtools.com/ (Inst Tools, 2022)

[5] https://www.researchgate.net/ (Rg, 2022)

[6] https://support.industry.siemens.com/ (SIEMENS, 2022)

[7] http://www.plcdev.com/book/export/html/431 (Allen Bradley's PLC programming, 2022)

[8] https://www.codesys.it/ (CODESYS software, 2022)