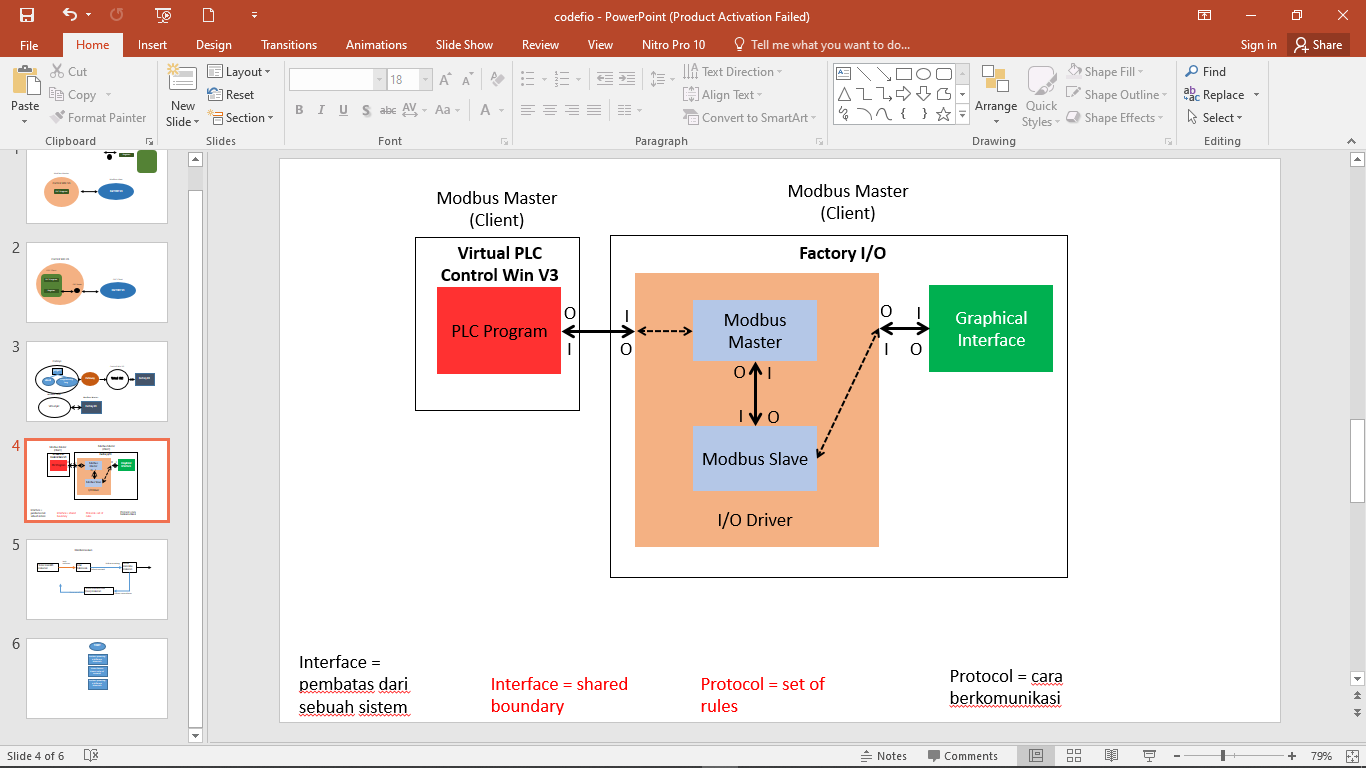
**CHAPTER 3**

**METHODOLOGY**

**3.1 Develop IoT System**

The IoT System consists of controller development system which is Codesys and Factory Simulation in Factory I/O that communicate using Modbus network protocol.



**Figure 3.1** Develop IoT System Architecture

The process of exchanging data is defined for Codesys as the Master (Client) and Factory as the Slave (Server). The virtual PLC act as controller utilizing HMI for the simulation, while the simulation program is run in the Factory I/O virtual factory.

**3.1.1 Virtual PLC Control Win V3**

The PLC used in the IoT System is the Virtual PLC Control Win V3. It has been provided by Codesys, runs inside the computer system with an IP-address and a Port number. The Codesys must connect to the Virtual PLC via the gateway; otherwise, the Codesys cannot run the PLC programs.

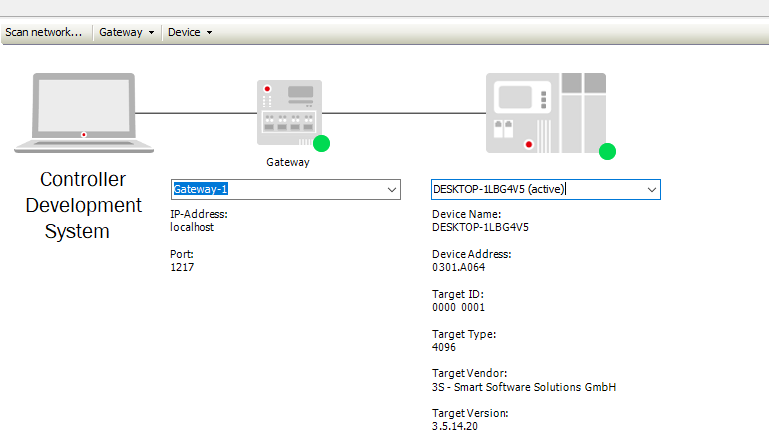


Figure 3.2 Connection Between PC and a Virtual PLC in Codesys configuration

The Virtual PLC in the IoT System uses the Modbus TCP/IP communication protocol to be able exchanging data with Factory I/O. Control Win V3 sent data to the Factory I/O using Modbus TCP/IP, Factory I/O can read or write Modbus function code. Each variable in the PLC program need to be assigned in the I/O mapping of Modbus in Codesys with a register and coils based on how the user defined each variable.

**3.1.2 Factory IO I/O Driver**

I/O Driver is a built-in feature in Factory I/O, we only need to choose what communication protocol to be used. The I/O Driver supports different communication protocol. Since the I/O Driver capable of doing various protocol and we want the Codesys become the Master and Factory IO as the Slave, “Modbus TCP/IP Server” is chosen in the I/O Driver to be used in the thesis.



**Figure 3.3** Choosing “Modbus TCP/IP Server” in I/O Driver

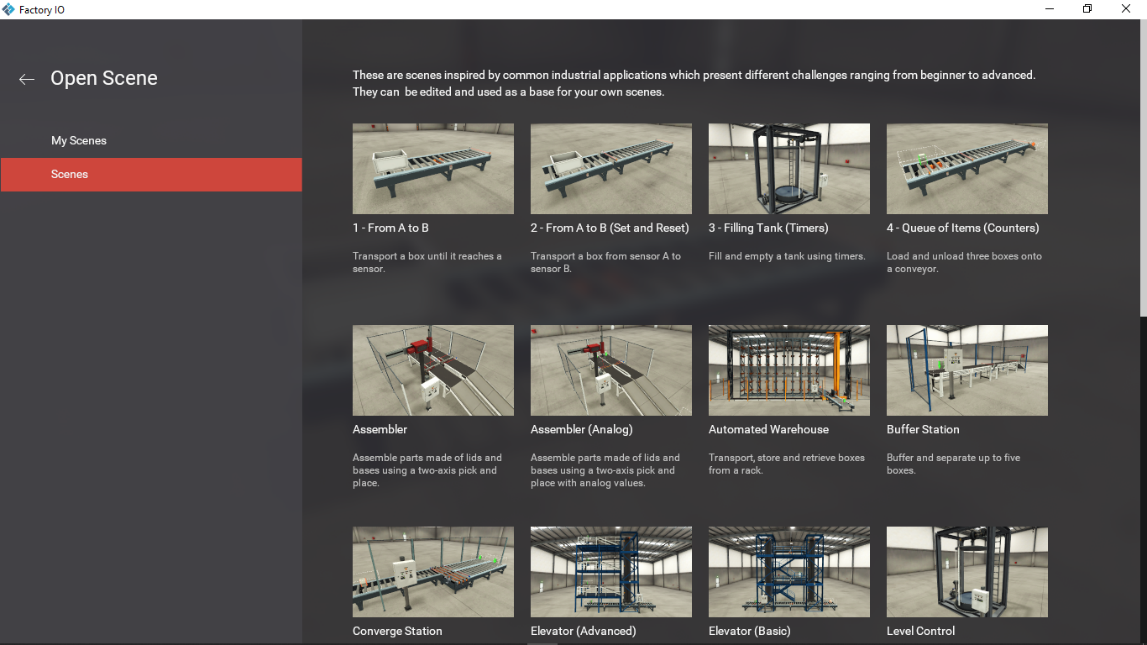
**3.2 Research Equipment Setup**

**3.2.1 Factory I/O and Codesys**

The connection between Codesys and Factory I/O was establish using the Modbus Communication protocol. Here are the steps for connecting the two devices through Modbus.

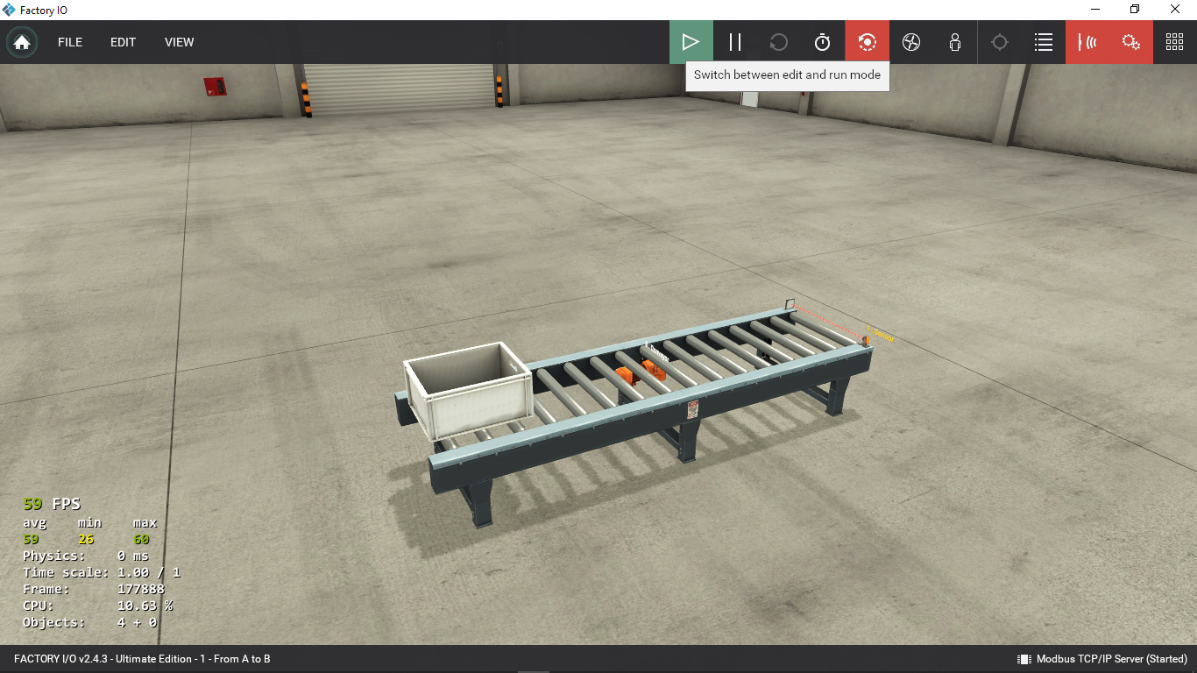
1. Create a simulation

Before we start to connect Factory I/O and Codesys, we need to define or create a factory simulation. Factory IO itself has a list of ready-to-use scenario based on common industrial applications. The scenes may be modified or customized and used as a basis for your own scenes, depending on how far you want to take the scenes to be controlled. You may also create your own scene from the start.



**Figure 3.4** Overview of the ready-to-use scene

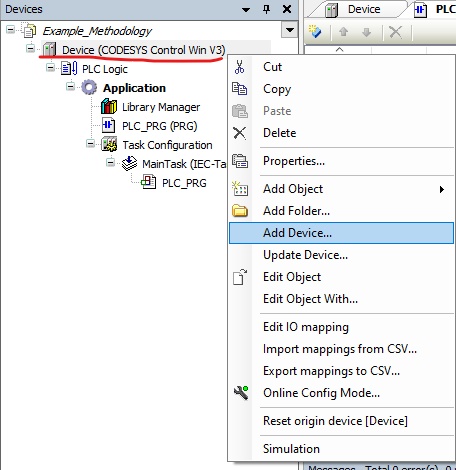
Since we want to establish the connection between Factory I/O and Codesys first, you can pick a simple scene to be the experiment.



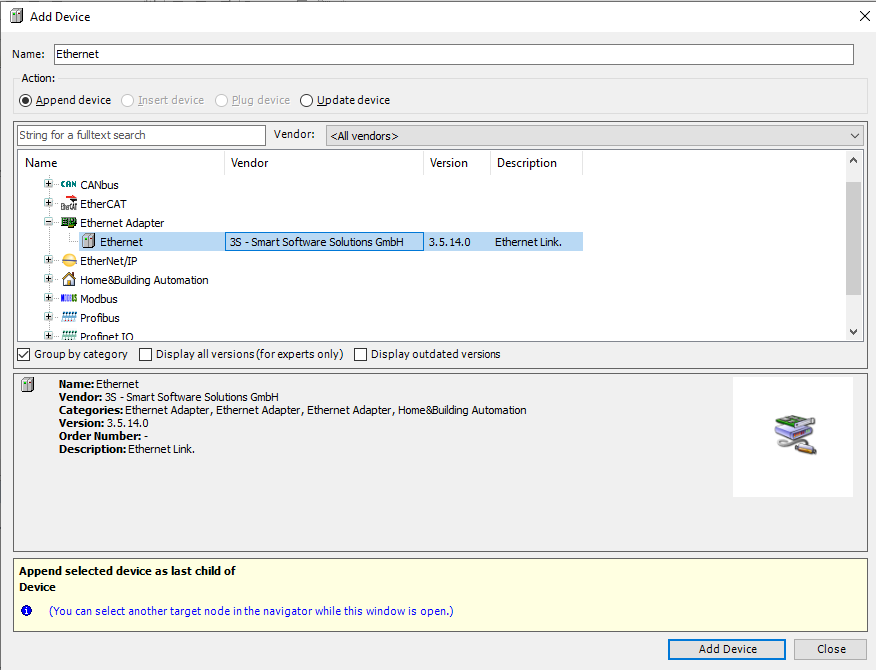
**Figure 3.5** Overview the virtual factory line scene

2. Configure the virtual PLC

The virtual PLC used in this thesis Codesys Control Win V3, the explanation regarding virtual PLC is in section **3.1.1**. Modbus communication protocol must be added to the virtual PLC's device tree in Codesys before it can be configured. This will allow the virtual PLC to make the connection between devices using Modbus. Shown in **Figures** **3.6** and **3.7** on how to enable Modbus function in the virtual PLC.

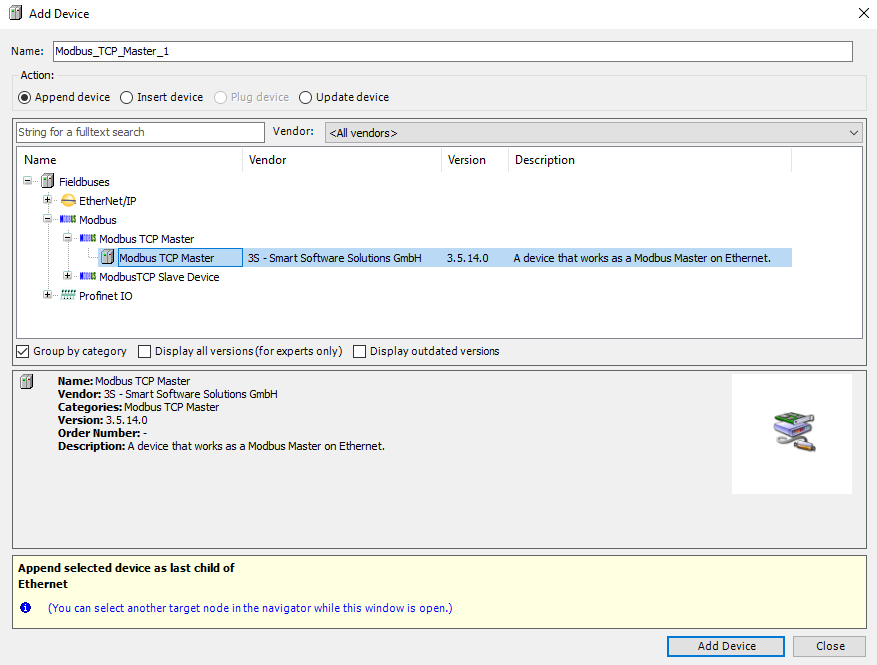


**Figure 3.6** Add Device option for the virtual PLC

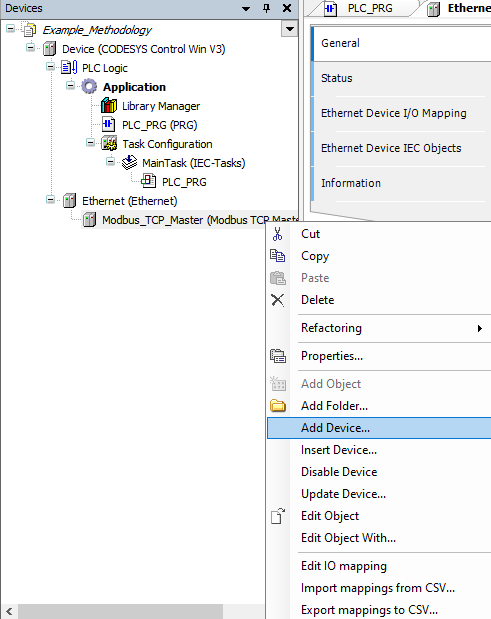


**Figure 3.7** Add Device window for the virtual PLC

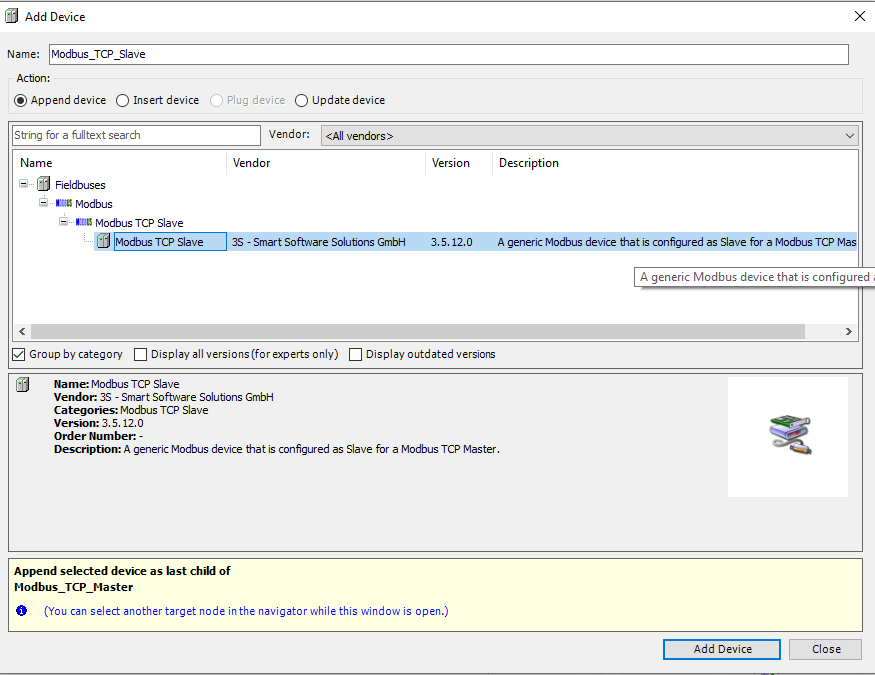
The connection happens between Factory I/O and Codesys is in localhost network. Go to the Ethernet Adapter section, there you can add the ModbusTCP and choose either as the Master or the Slave depends on how the user desired. Since we want the Codesys acts as the Master or Client, choose “Modbus TCP Master”. Then you can add the “Modbus TCP Slave” inside the Modbus TCP Master.



**Figure 3.8** Add Modbus TCP Master in Ethernet Add Device window



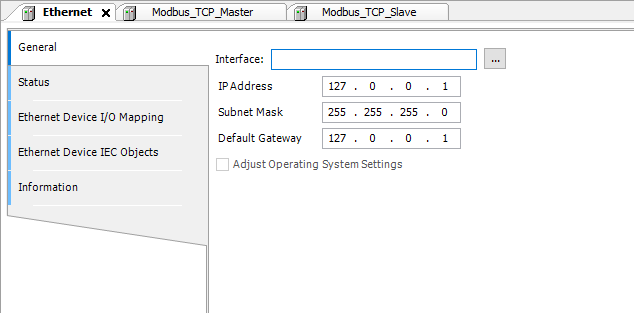
**Figure 3.9** Add Device option in Modbus TCP Master



**Figure 3.10** Add Modbus TCP Slave for become Master (Client)

3. Configure the Ethernet

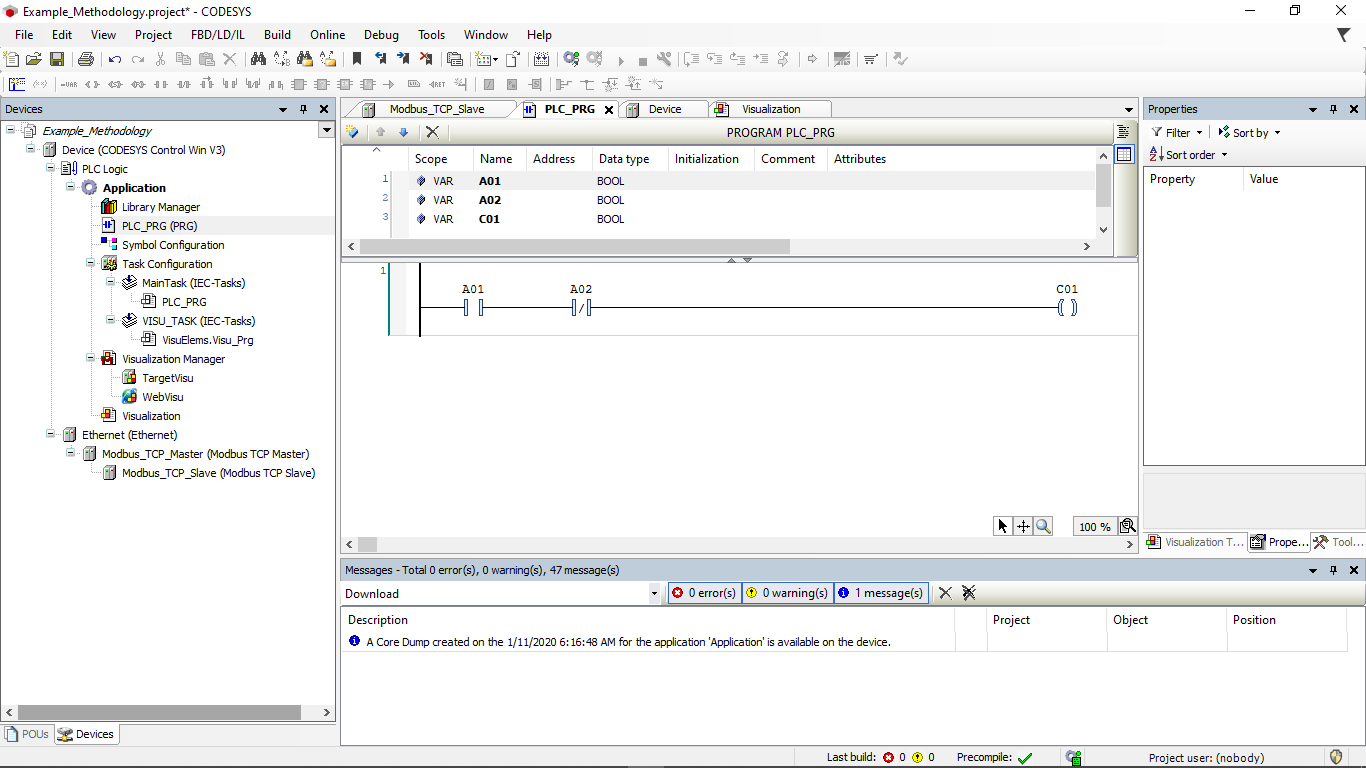
In the Ethernet section, go to the General tab and set the IP address and Default Gateway to localhost or 127.0.0.1. As shown in **Figure 3.11**.



**Figure 3.11** Ethernet’s IP address and Default Gateway set to localhost

4. Create the Ladder Diagram

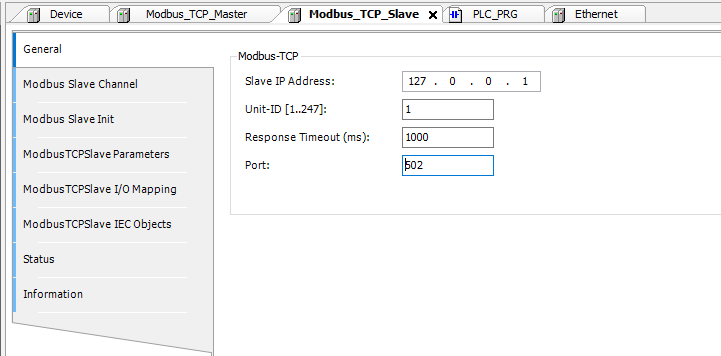
We need to create the Ladder Diagram (LD) for the scene that we want to run, you can go to “PLC\_PRG”. Codesys left the PLC\_PRG blank sheet so the user can create as much as they want. You can just click & drag the elements from the right panel into the sheet.



**Figure 3.12** A simple Ladder Logic for the scene

5. Configure the Modbus TCP Slave Device

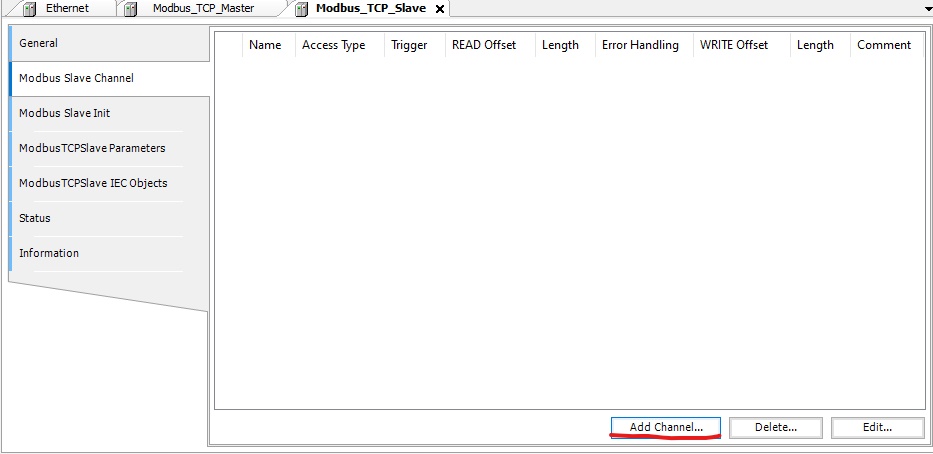
In the Modbus TCP Slave Device, go to the General section then set the slave IP address into localhost or 127.0.0.1, set the Unit-ID depends on the user and the port to default or 502. In this scene, set the Unit-ID into 1.



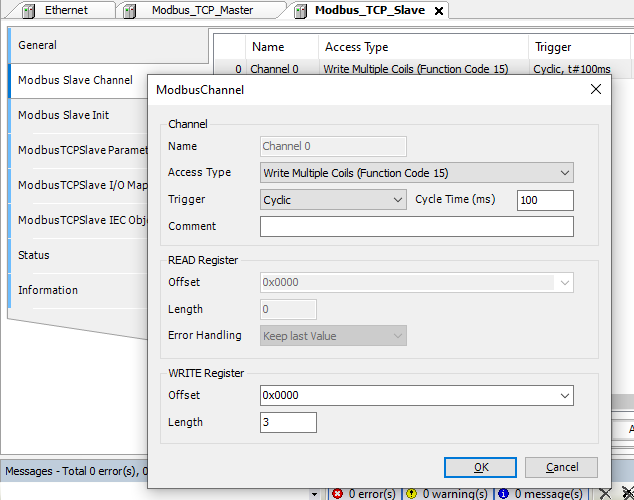
**Figure 3.13** Modbus TCP Slave device’s configuration

6. Configure the Modbus Slave Channel

The Function Code that the user desired can be define on the Modbus Slave Channel section by choose “Add Channel”, as shown in **Figure 3.14**. This allowing the user to choose what Function Code to be used and how many channel to store the variable by define the number inside the Length tab, later will be assign for each variables corresponding with what the Ladder Diagram have been made for. Shown in **Figure 3.15** for define the function code.



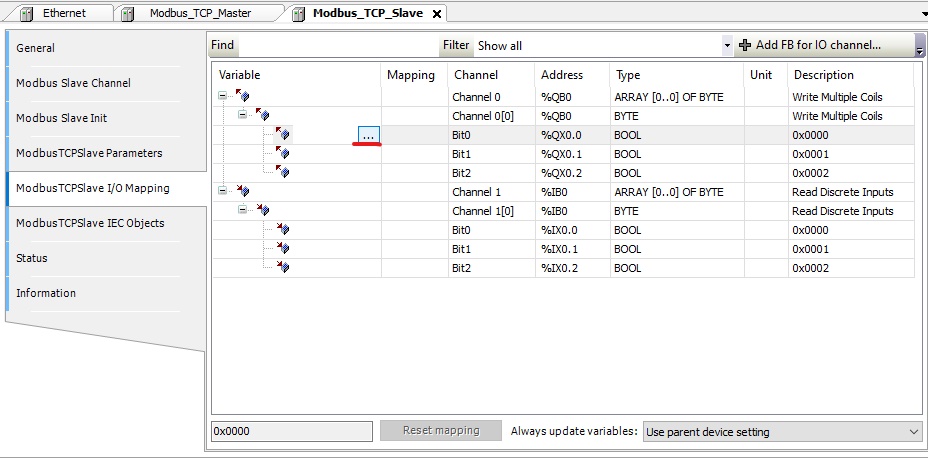
**Figure 3.14** Add channel to define Function Code



**Figure 3.15** Define what Function Code to be used

7. Set the desired variables into Modbus TCP Slave I/O Mapping

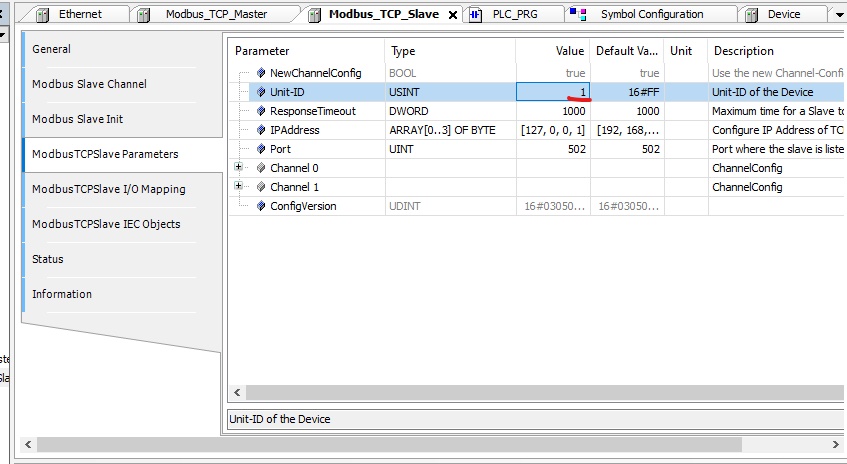
Variables in the PLC Program must to be assigned into the I/O mapping of Modbus TCP Slave, this allowing other devices connected to virtual PLC’s Modbus to recognize the address of variables in the I/O mapping. By selecting the three dots, you will be presented all of variable that can be assigned, as shown in **Figure 3.16**.



**Figure 3.16** Select the 3 dots to assign a variable

8. Configure the Unit-ID in Modbus TCP Slave Parameters

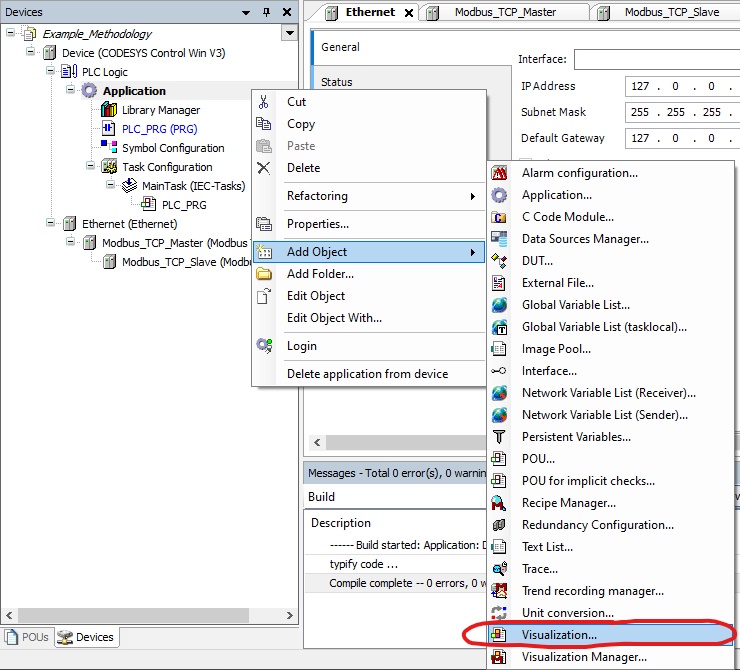
By default, the Unit-ID in Modbus TCP Slave Parameters was 255. If we define the slave Unit-ID as 1, as shown in **Figure 3.13**, the other devices that connect to Codesys will not recognize it as a Slave. It is critical to match the Unit-ID in both General and Parameters.



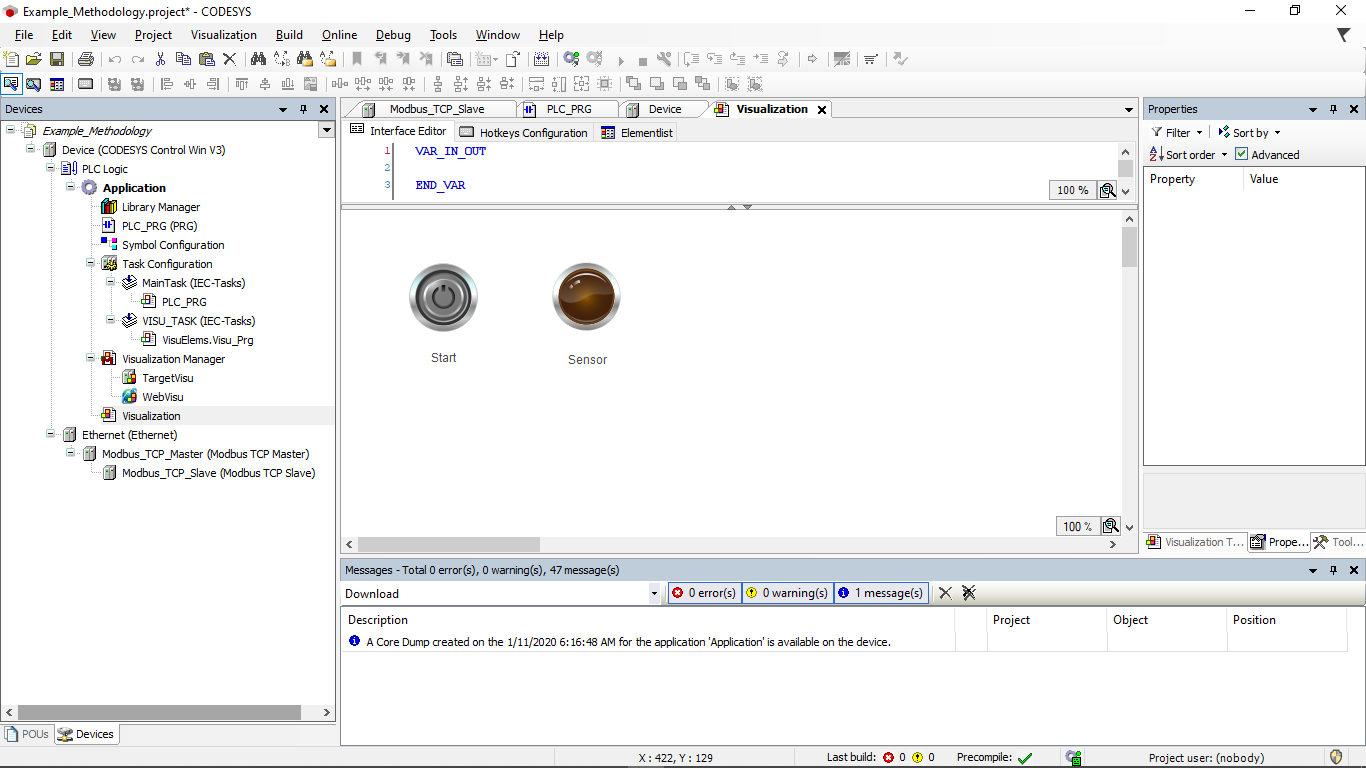
**Figure 3.17** Unit-ID in Modbus TCP Slave Parameters

9. Create the HMI in Visualization

Since the Codesys become the Master, as the designer, we wish to observe and control the factory line scene. Codesys can create a virtual control panel button with the help of the Visualization. You can click & drag the components as you want form Visualization ToolBox that available at the right panel. **Figures 3.18 and 3.19** show how to add Visualization as well as an example of a virtual control panel button.



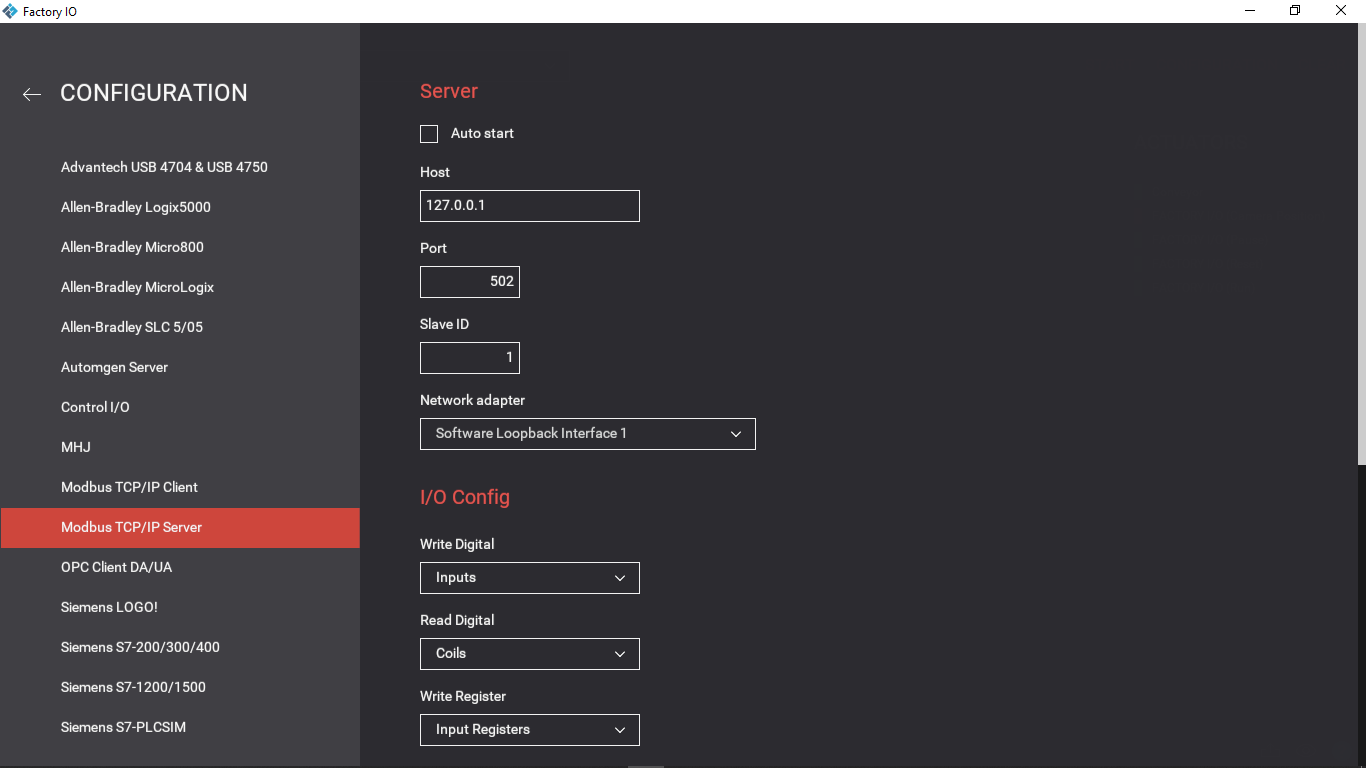
**Figure 3.18** Add Visualization option in Application



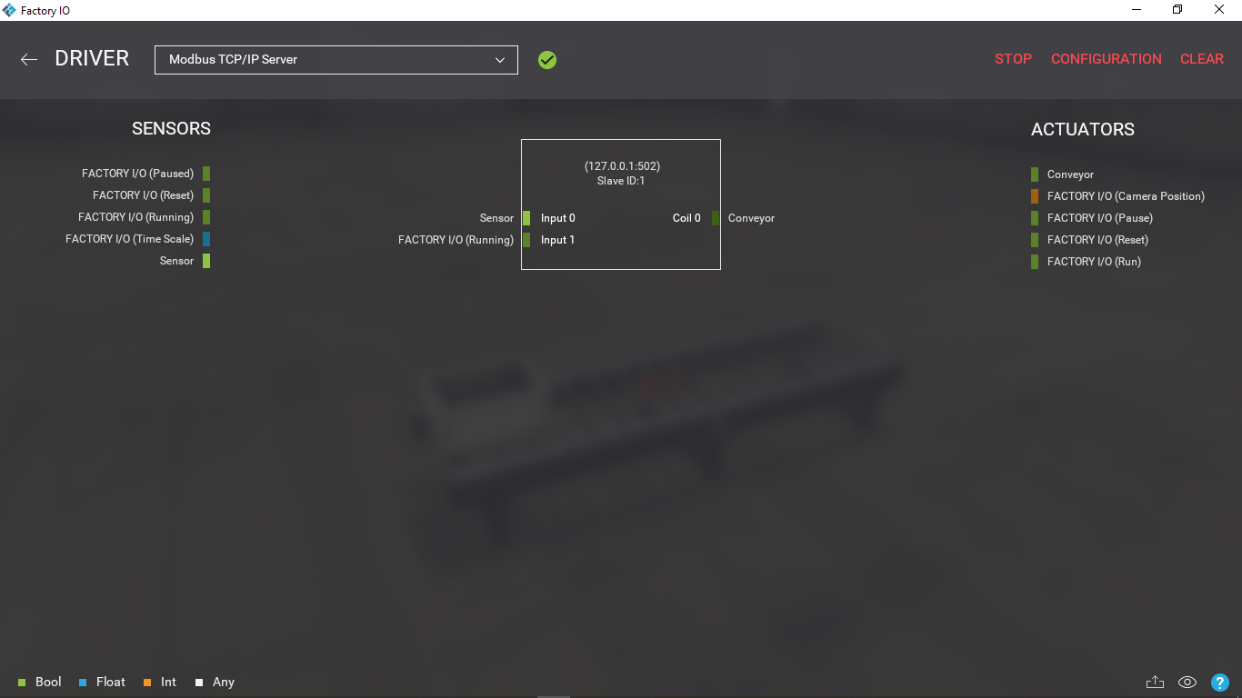
**Figure 3.19** Visualization of a virtual control panel

10. Configure the I/O Drivers

Go to the Configuration inside Drivers in the Factory I/O, then set the IP address into localhost or 127.0.0.1, the port number into default or 502, and the Slave ID into 1, as shown in **Figure 3.20.** Then select “START” to connect the Factory I/O and Codesys; if the two devices are connected, a little green check mark will appear next to I/O Drivers, as shown in **Figure 3.21.**



**Figure 3.20** Configuration in Factory I/O Driver



**Figure 3.21** A green check mark appear next to Driver selection

If the connection between Factory I/O and Codesys is successful, click “Build” to have Codesys begin the build operation for the active application. Then, to run the PLC, click "Login" and "Start."