REPUBLIC OF YEMEN

SANA’A UNIVERSITY

FACULITY OF ENGINEERING

MECHATRONICS ENGINEERING DEPARTMENT

Fourth year

Second Semester

Industrial network project

Done by:

Afnan Khaled Alashwal 202073138

Abdulrhman Afif Alaghbari 202073119

Mohammed Jalal Naji Hassan Omar 202073005

Osamah Mutea'a 202073041

Ibraheem Alkhulaidy 202073007

Supervised by:

ENG. Mohammed Abdalnasser Alzaghir

Abstract

Industrial networks form a special class of computer networks that employ specific devices, communication protocols and communication patterns. In order to study industrial networks, it is important to have an access to industrial devices and their communication. This is, however, not easy to implement in university environment. Real devices are expensive, require regular maintenance and are available to few operators. As alternative to the real industrial environment, it is possible to combine real devices with emulated environment. This study shows how it is possible to create an industrial network with Modbus protocols and real devices like PLCs and RTUs together with emulator of physical processes using I/O Factory software. In this study we show how to build a virtual factory that includes a simple assembly line and the sorting conveyor controlled by PLCs

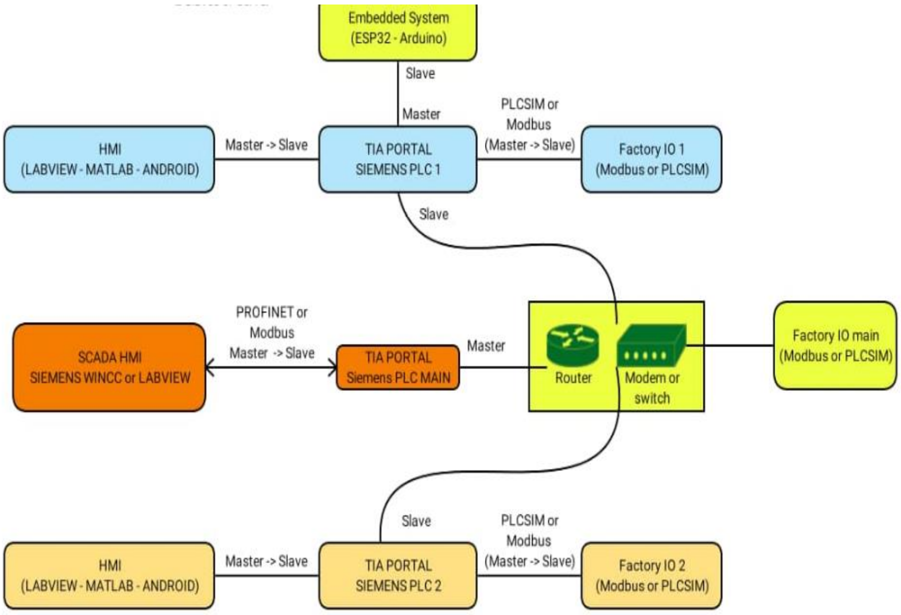

Embedded System (ESP32 - Arduino)
Slave
Master
HMI (LABVIEW - MATLAB - ANDROID)
Master -> Slave
TIA PORTAL SIEMENS PLC 1
PLCSIM or Modbus (Master -> Slave)
Factory IO 1 (Modbus or PLCSIM)
Slave
SCADA HMI SIEMENS WINCC or LABVIEW
PROFINET or Modbus Master -> Slave
TIA PORTAL Siemens PLC MAIN
Master
Router
Modem or switch
Factory IO main (Modbus or PLCSIM)
Slave
HMI (LABVIEW - MATLAB - ANDROID)
Master -> Slave
TIA PORTAL SIEMENS PLC 2
PLCSIM or Modbus (Master -> Slave)
Factory IO 2 (Modbus or PLCSIM)

TABLE OF CONTENT

Introduction

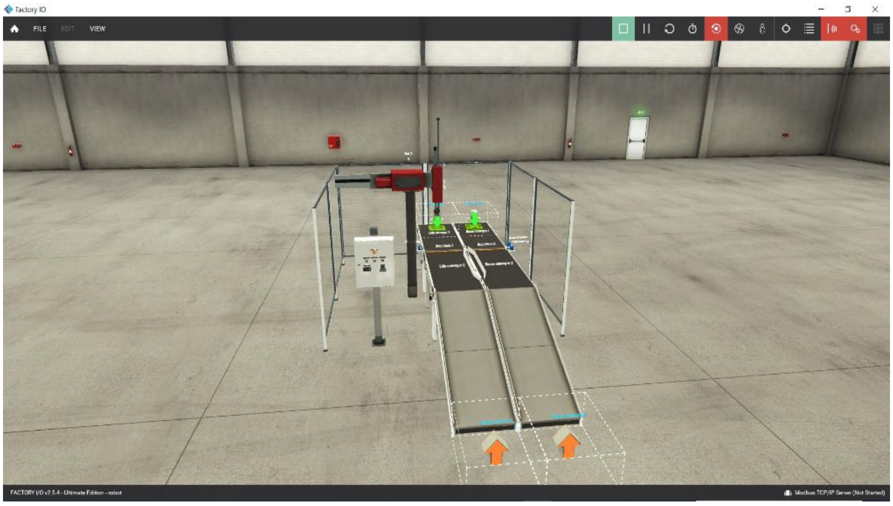
As a member of the TRACTOR1 (Traffic Analysis and security Operations for ICS/SCADA) project at Faculty of Information Technology, Brno University of Technology, our task was to create a testing environment for Modbus TCP communication protocol, which would allow testing of various types of attacks on SCADA networks.

Nowadays, great emphasis is placed on the automation of various industrial systems. However, the more it is automated, the more number of components that need to be interconnected increases. With a large number of these devices, it is impossible to communicate on the physical layer, and therefore their mutual communication had to be transferred to the IP layer. However, moving to the IP layer gives attackers new ways to break into the system, which we would like to prevent, as these systems are often a part of the critical infrastructure and their disruption could cause major damage (power plant - interruption of electricity supply to thousands of households, factory - production shutdown, etc.) [2].

Our job was to create a testing environment where Modbus TCP communication can be created, captured and analyzed. Our testing environment simulate real world production line, where single components communicate via Modbus TCP protocol. It also allows to create several types of attacks and analyses how the system would behave.

Section 1 describes creation of two types of production lines in Factory I/O simulation program. The hardware part, which includes all physical components and their interconnections, is described in Section 2. Software part is described in Section 3, where main focus is on scripts for automatic control of our lines. One line also can be controlled via HMI (Human

Machine Interface), where user can control several parts of line manually.


Factory IO
FILE
EDIT
VIEW
FACTORY I/O v2.5.4 - Ultimate Edition - robot
Modbus TCP/IP Server (Not Started)

Objective

1.Understanding Industrial Networks:

• Industrial networks are a specialized class of computer networks that employ specific devices, communication protocols, and communication patterns.

2.PLC Connectivity:

• PLCs (Programmable Logic Controllers) are widely used in industrial automation to control and monitor various processes.

3.HMI and SCADA Integration:

• HMIs provide a user interface for operators to monitor and control industrial

4.HMI, PLC, and Factory I/O Integration:

• HMIs can be connected to PLCs to provide a user interface for monitoring and controlling the industrial processes.

5.Reading Values from TIA Portal:

• TIA Portal (Totally Integrated Automation Portal) is a software suite from Siemens that provides a unified engineering environment for programming and configuring industrial automation systems.

Methodology

Assembly line it is necessary to reproduce three distinct routines, one with the pick and place robot and one with each conveyor. The conveyors routines are activated by the arrival of parts and the manufacturing cycle time of the assembly line. Using Part Emitters, the bases and lids parts are injected in the assembly line to start the work. This in an external event not specified in the DES model. The robot arm's routine is activated by the sensors "Lid at place" and "Base at place" which give the information that the parts ready and the robot will begin its work producing the necessary output signal changes until it has finished.

Result

Main project (MODBUS)

Factory io

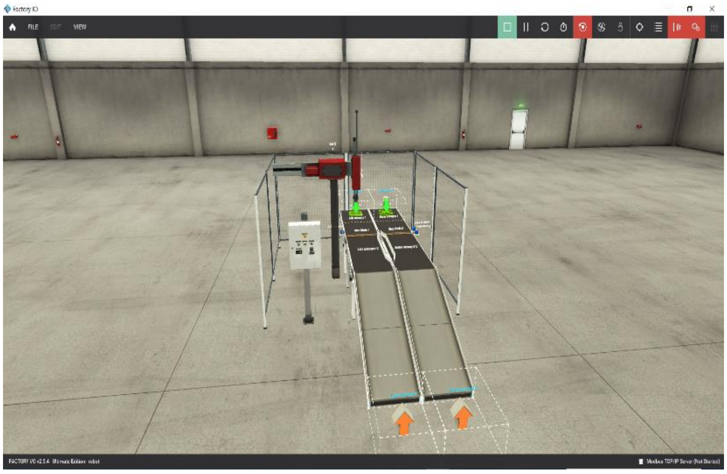
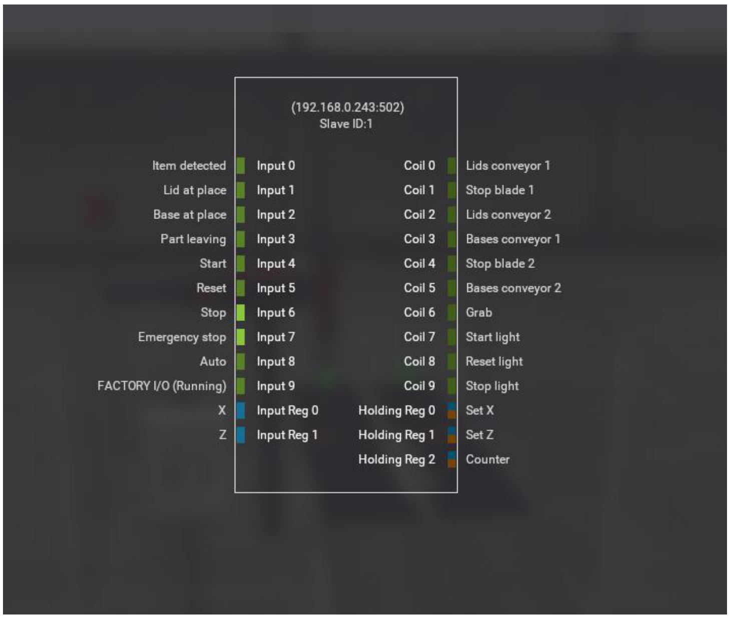

Factory IO
FILE
EDIT
VIEW
FACTORY I/O v2.5.4 Ultimate Edition robot
Modbus TCP/IP Server (Not Started)

Figure 1Factory IO

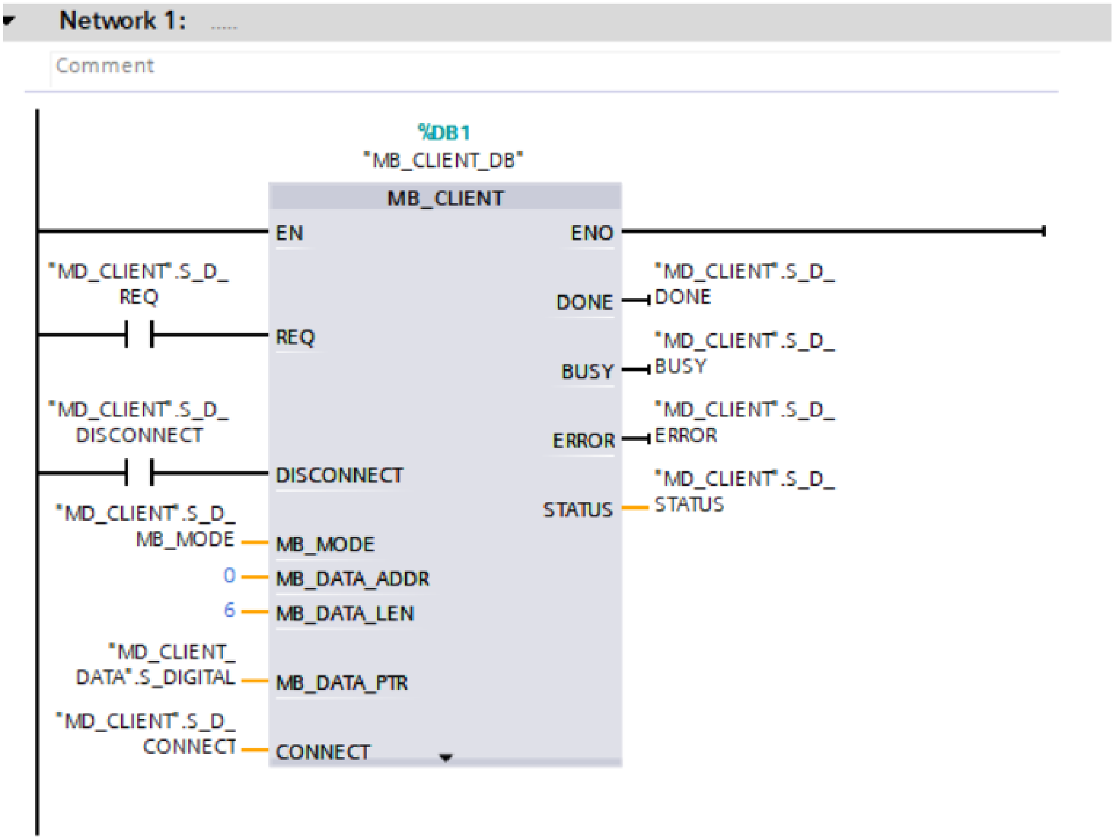


(192.168.0.243:502)
Slave ID:1

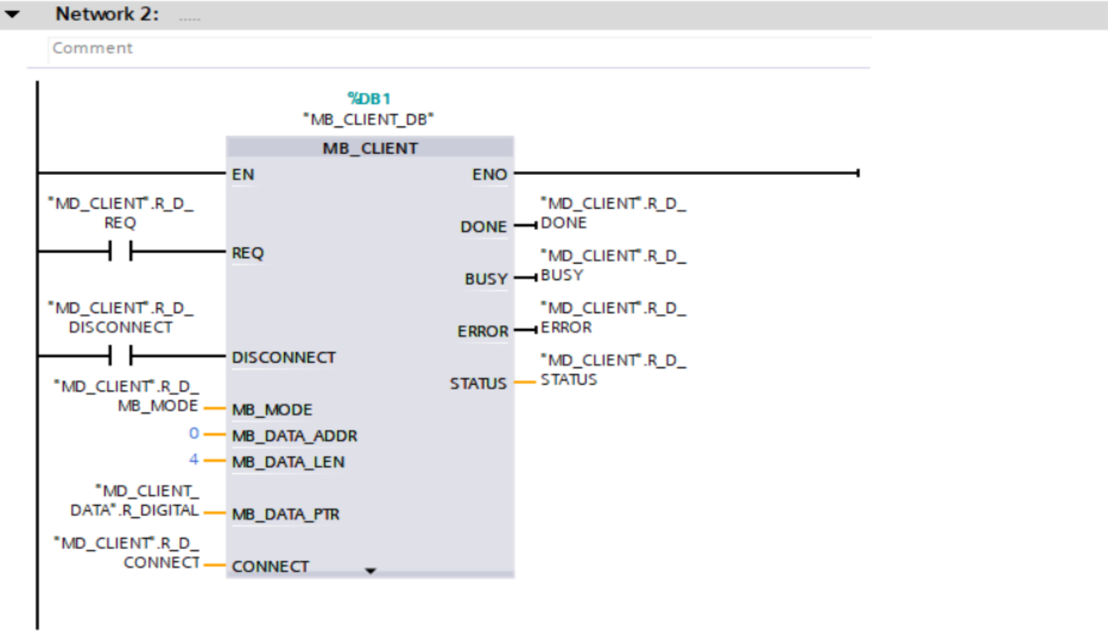
Item detected	Input 0	Coil 0	Lids conveyor 1
Lid at place	Input 1	Coil 1	Stop blade 1
Base at place	Input 2	Coil 2	Lids conveyor 2
Part leaving	Input 3	Coil 3	Bases conveyor 1
Start	Input 4	Coil 4	Stop blade 2
Reset	Input 5	Coil 5	Bases conveyor 2
Stop	Input 6	Coil 6	Grab
Emergency stop	Input 7	Coil 7	Start light
Auto	Input 8	Coil 8	Reset light
FACTORY I/O (Running)	Input 9	Coil 9	Stop light
X	Input Reg 0	Holding Reg 0	Set X
Z	Input Reg 1	Holding Reg 1	Set Z
		Holding Reg 2	Counter

Figure 2Factory IO Tags

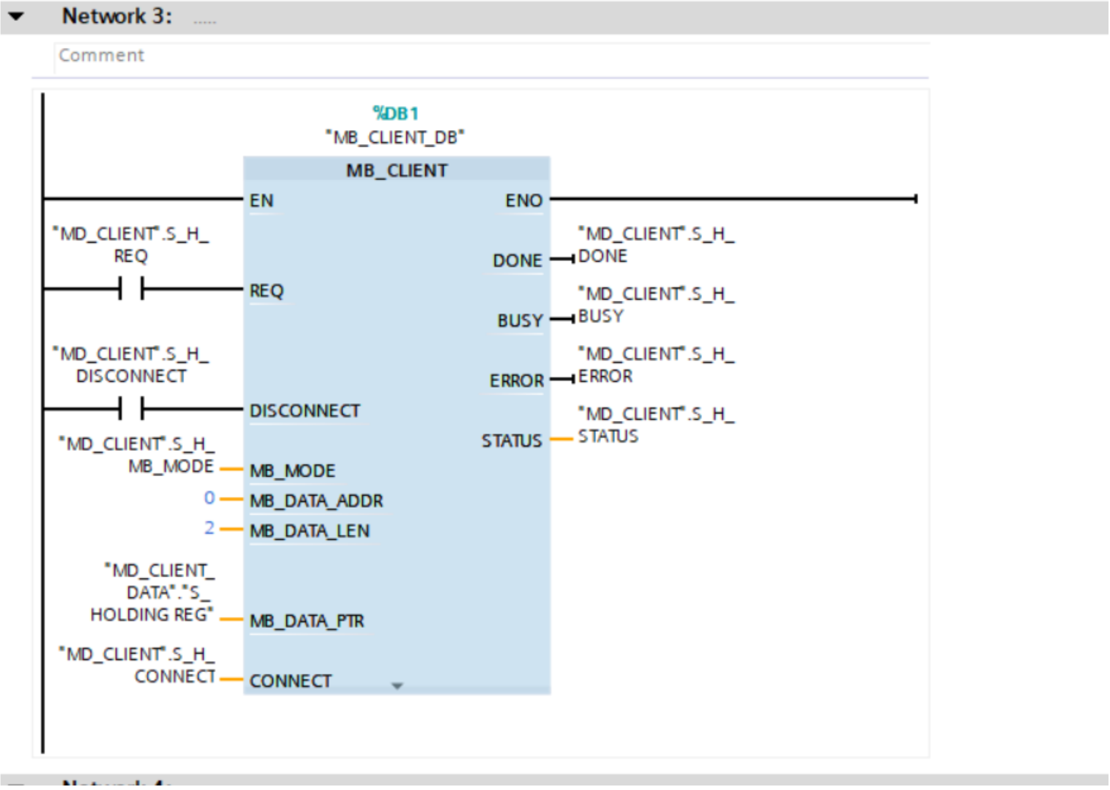
Plc tia portal



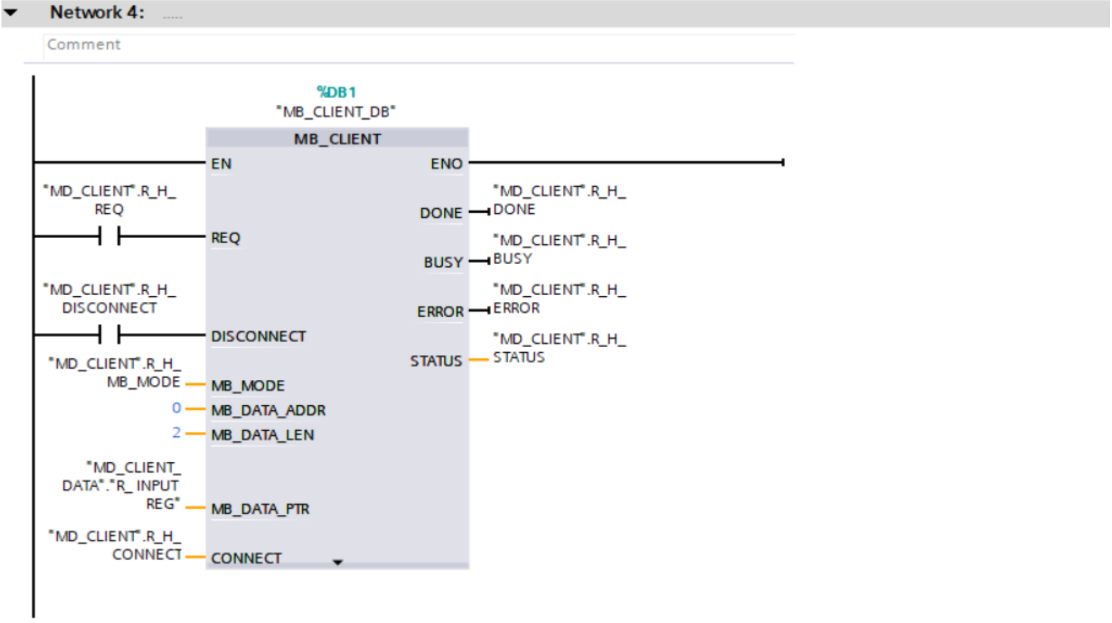
MOD n1 ٣ Figure



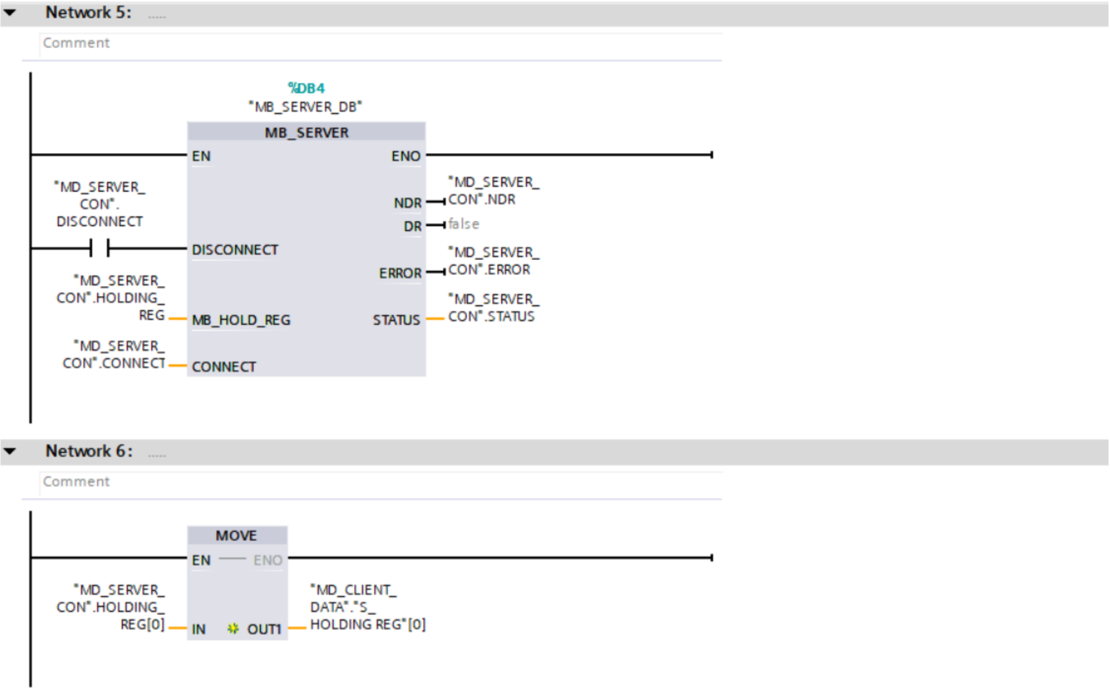
MOD n2 ٤ Figure



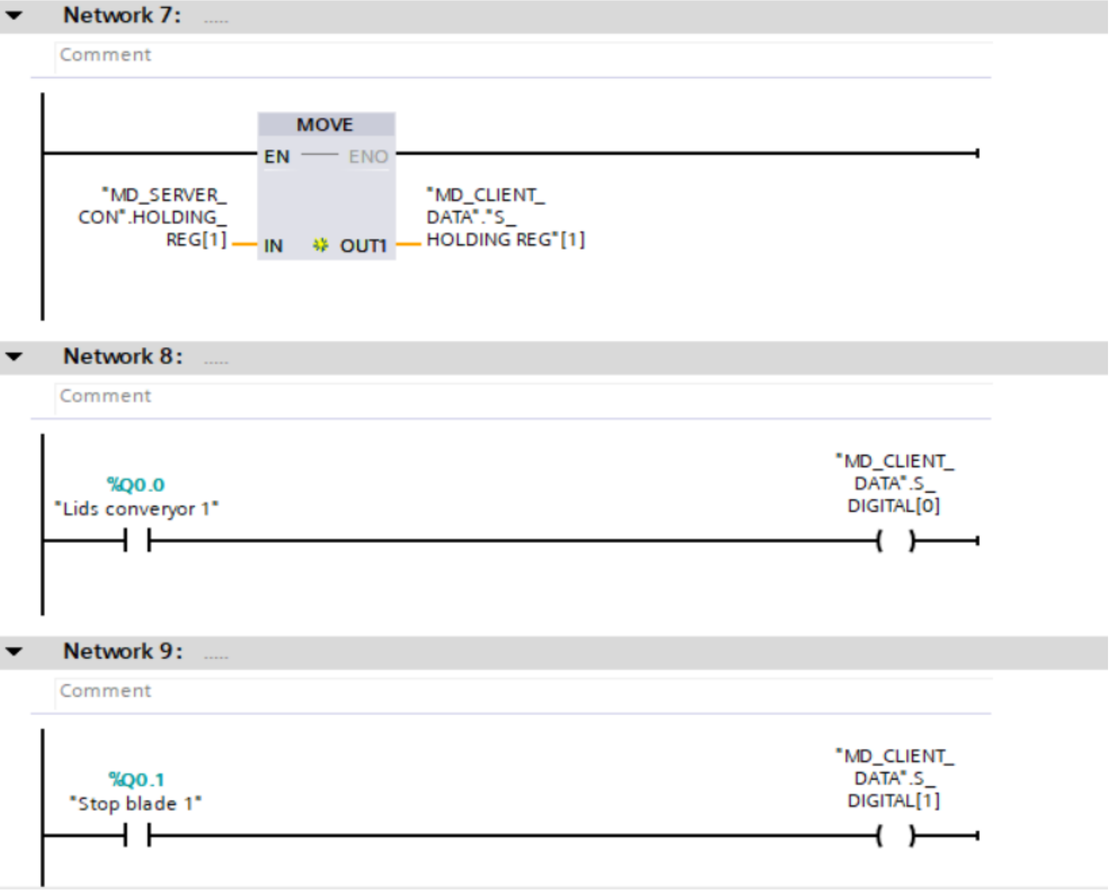
MOD n3 ٥ Figure



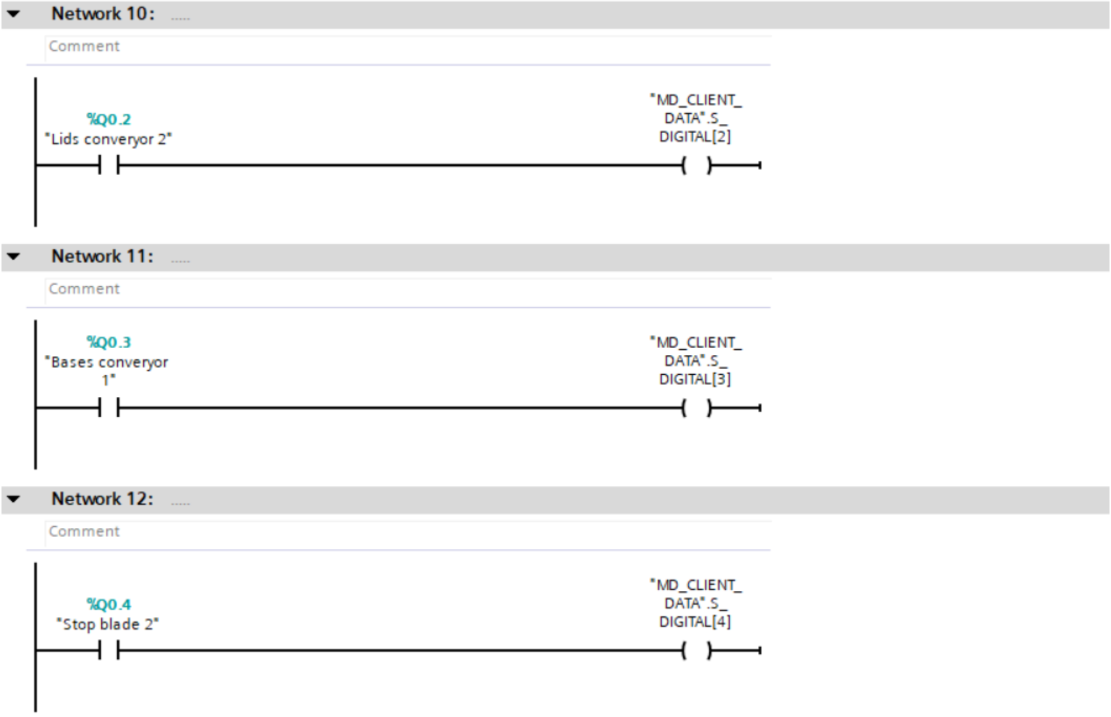
MOD n4 ٦ Figure



MOD n5,6٧ Figure



MOD7,8,9٨ Figure



Network 10: ......

Comment

%Q0.2 "Lids converyor 2" → "MD_CLIENT_DATA".S_DIGITAL[2]

Network 11: ......

Comment

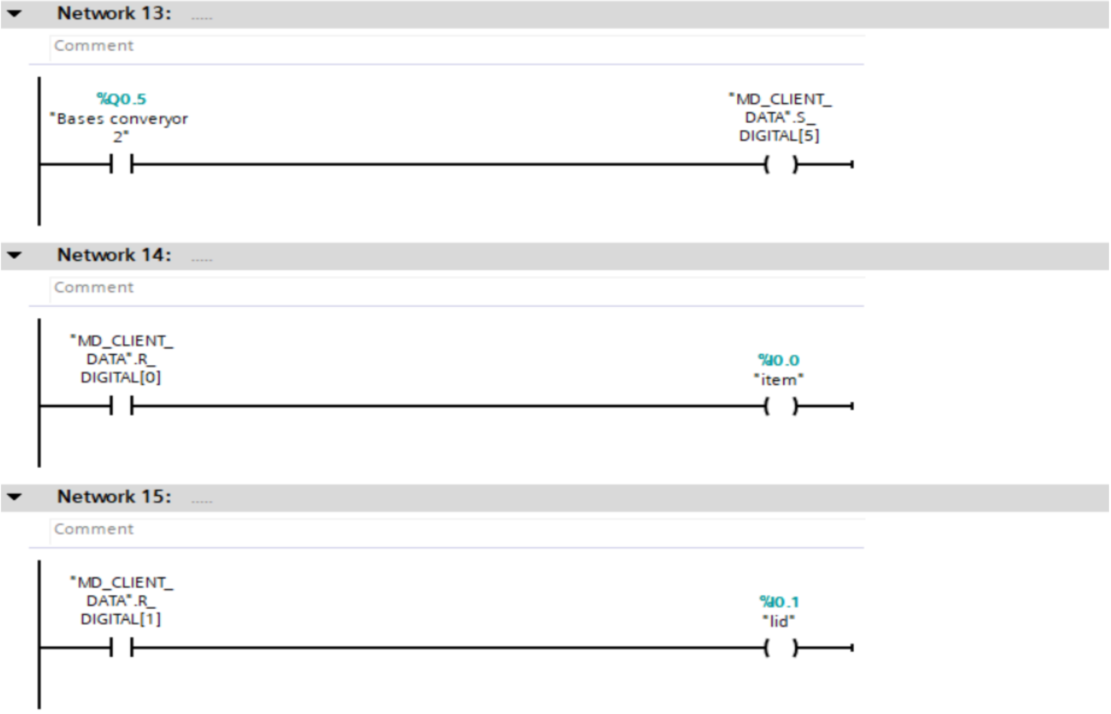
%Q0.3 "Bases converyor 1" → "MD_CLIENT_DATA".S_DIGITAL[3]

Network 12: ......

Comment

%Q0.4 "Stop blade 2" → "MD_CLIENT_DATA".S_DIGITAL[4]

MOD 10,11,12 ٩ Figure



Network 13: ......

Comment

%Q0.5 "Bases converyor 2" → "MD_CLIENT_DATA".S_DIGITAL[5]

Network 14: ......

Comment

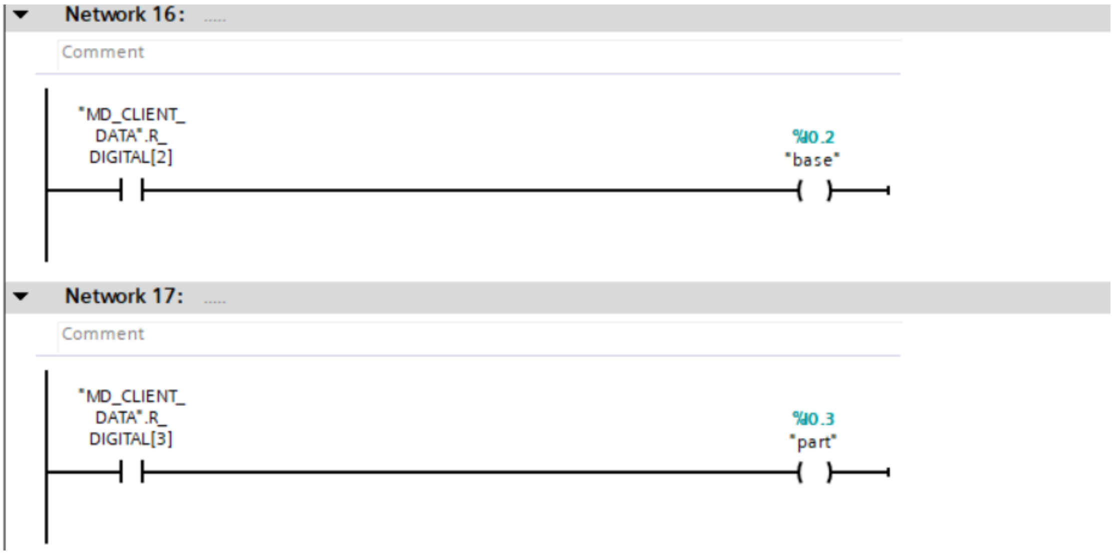
"MD_CLIENT_DATA".R_DIGITAL[0] → %I0.0 "item"

Network 15: ......

Comment

"MD_CLIENT_DATA".R_DIGITAL[1] → %I0.1 "lid"

MOD 13,14,15 ١٠ Figure



Network 16: .....

Comment

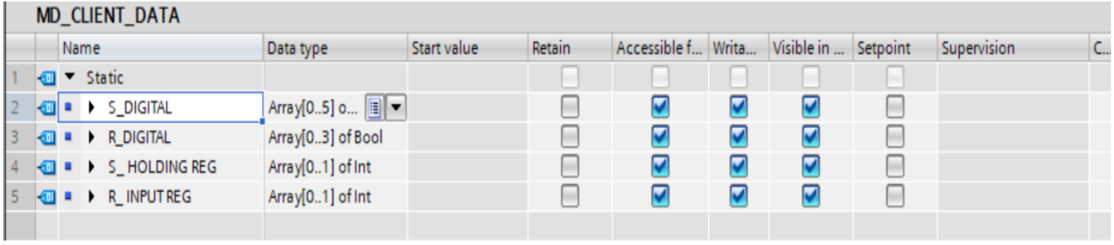
"MD_CLIENT_DATA".R_DIGITAL[2] —| |— ... —( )— %I0.2 "base"

Network 17: .....

Comment

"MD_CLIENT_DATA".R_DIGITAL[3] —| |— ... —( )— %I0.3 "part"

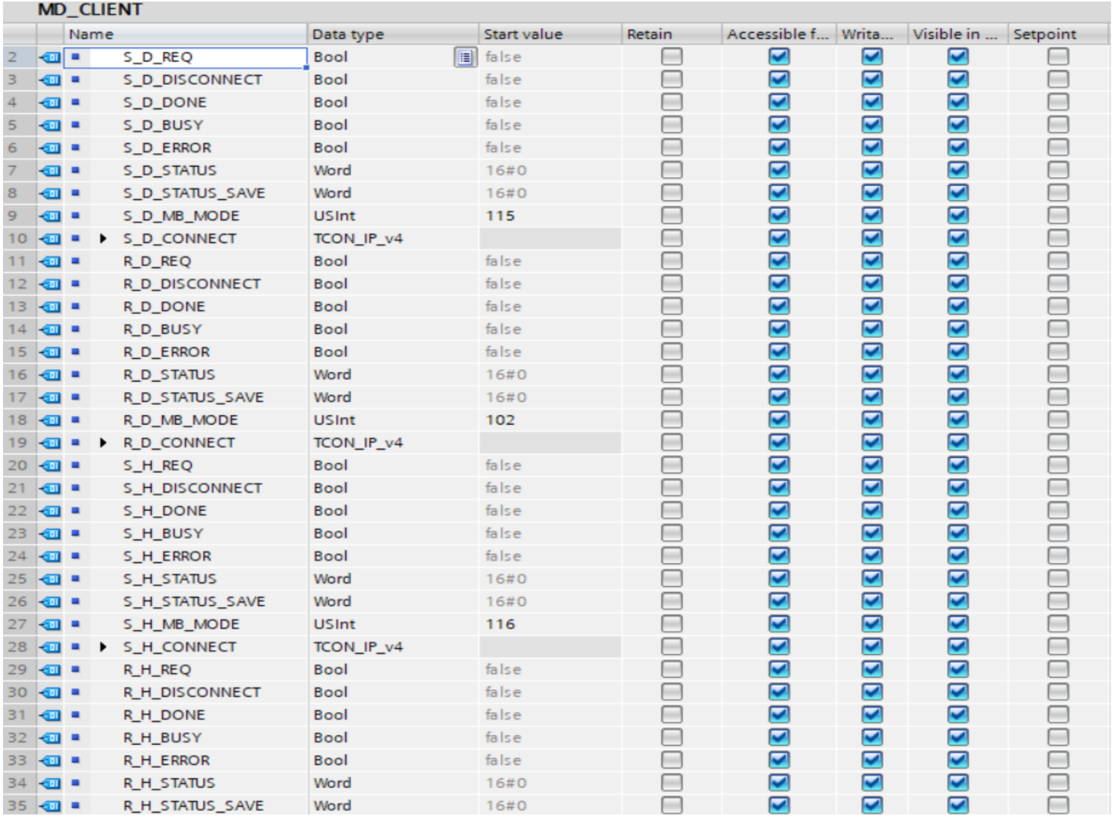
MOD 16,17 ۱۱ Figure



MD_CLIENT_DATA

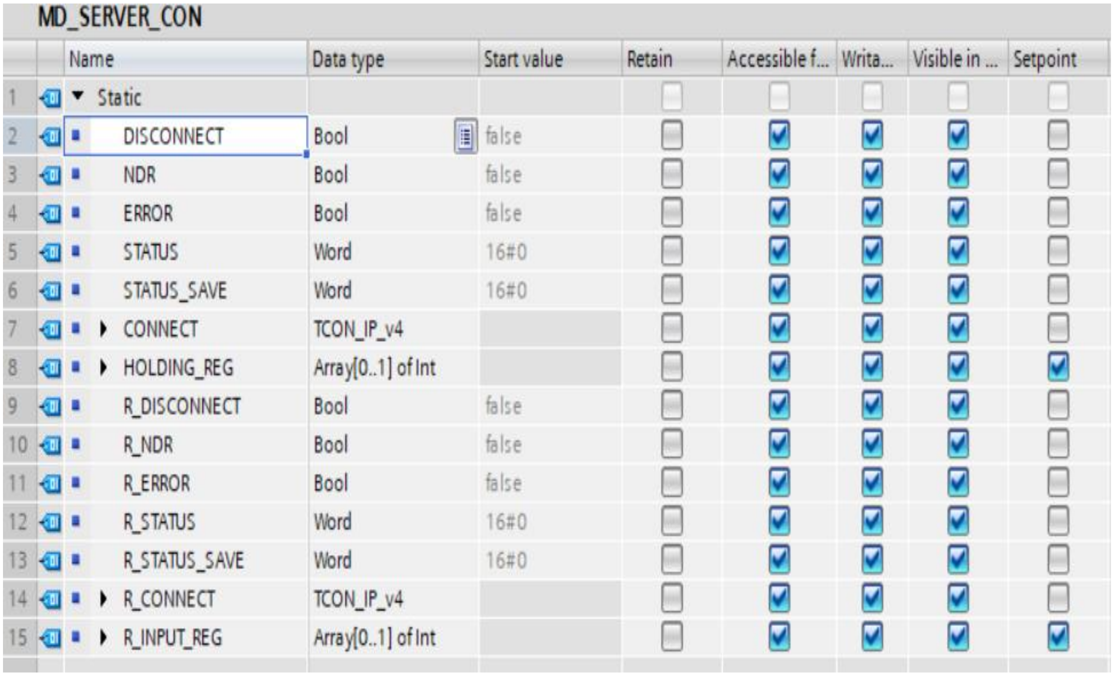
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervision	C...
1	▼ Static			□	□	□	□	□		
2	▶ S_DIGITAL	Array[0..5] o...		□	☑	☑	☑	□		
3	▶ R_DIGITAL	Array[0..3] of Bool		□	☑	☑	☑	□		
4	▶ S_HOLDING REG	Array[0..1] of Int		□	☑	☑	☑	□		
5	▶ R_INPUT REG	Array[0..1] of Int		□	☑	☑	☑	□		

Client data ۱۲ Figure



MD_CLIENT

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
2	S_D_REQ	Bool	false	□	☑	☑	☑	□
3	S_D_DISCONNECT	Bool	false	□	☑	☑	☑	□
4	S_D_DONE	Bool	false	□	☑	☑	☑	□
5	S_D_BUSY	Bool	false	□	☑	☑	☑	□
6	S_D_ERROR	Bool	false	□	☑	☑	☑	□
7	S_D_STATUS	Word	16#0	□	☑	☑	☑	□
8	S_D_STATUS_SAVE	Word	16#0	□	☑	☑	☑	□
9	S_D_MB_MODE	USInt	115	□	☑	☑	☑	□
10	▶ S_D_CONNECT	TCON_IP_v4		□	☑	☑	☑	□
11	R_D_REQ	Bool	false	□	☑	☑	☑	□
12	R_D_DISCONNECT	Bool	false	□	☑	☑	☑	□
13	R_D_DONE	Bool	false	□	☑	☑	☑	□
14	R_D_BUSY	Bool	false	□	☑	☑	☑	□
15	R_D_ERROR	Bool	false	□	☑	☑	☑	□
16	R_D_STATUS	Word	16#0	□	☑	☑	☑	□
17	R_D_STATUS_SAVE	Word	16#0	□	☑	☑	☑	□
18	R_D_MB_MODE	USInt	102	□	☑	☑	☑	□
19	▶ R_D_CONNECT	TCON_IP_v4		□	☑	☑	☑	□
20	S_H_REQ	Bool	false	□	☑	☑	☑	□
21	S_H_DISCONNECT	Bool	false	□	☑	☑	☑	□
22	S_H_DONE	Bool	false	□	☑	☑	☑	□
23	S_H_BUSY	Bool	false	□	☑	☑	☑	□
24	S_H_ERROR	Bool	false	□	☑	☑	☑	□
25	S_H_STATUS	Word	16#0	□	☑	☑	☑	□
26	S_H_STATUS_SAVE	Word	16#0	□	☑	☑	☑	□
27	S_H_MB_MODE	USInt	116	□	☑	☑	☑	□
28	▶ S_H_CONNECT	TCON_IP_v4		□	☑	☑	☑	□
29	R_H_REQ	Bool	false	□	☑	☑	☑	□
30	R_H_DISCONNECT	Bool	false	□	☑	☑	☑	□
31	R_H_DONE	Bool	false	□	☑	☑	☑	□
32	R_H_BUSY	Bool	false	□	☑	☑	☑	□
33	R_H_ERROR	Bool	false	□	☑	☑	☑	□
34	R_H_STATUS	Word	16#0	□	☑	☑	☑	□
35	R_H_STATUS_SAVE	Word	16#0	□	☑	☑	☑	□



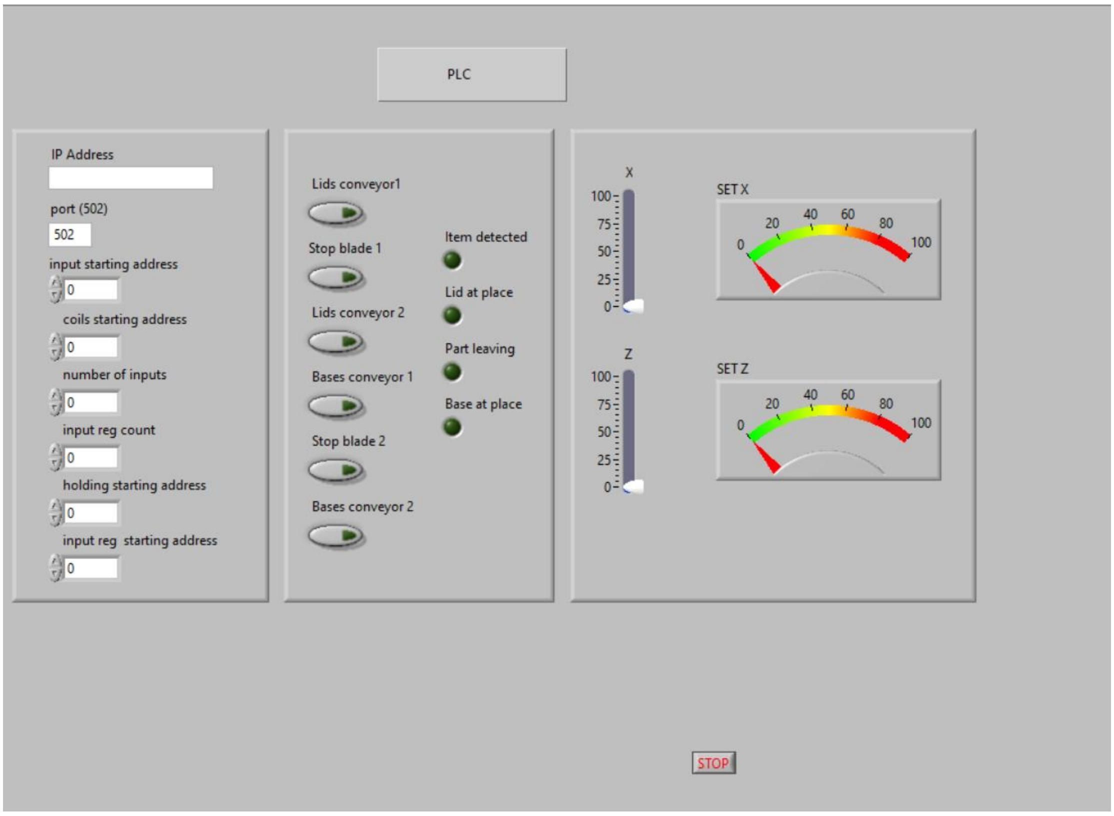
MD_SERVER_CON

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	▼ Static			□	□	□	□	□
2	DISCONNECT	Bool	false	□	☑	☑	☑	□
3	NDR	Bool	false	□	☑	☑	☑	□
4	ERROR	Bool	false	□	☑	☑	☑	□
5	STATUS	Word	16#0	□	☑	☑	☑	□
6	STATUS_SAVE	Word	16#0	□	☑	☑	☑	□
7	▶ CONNECT	TCON_IP_v4		□	☑	☑	☑	□
8	▶ HOLDING_REG	Array[0..1] of Int		□	☑	☑	☑	☑
9	R_DISCONNECT	Bool	false	□	☑	☑	☑	□
10	R_NDR	Bool	false	□	☑	☑	☑	□
11	R_ERROR	Bool	false	□	☑	☑	☑	□
12	R_STATUS	Word	16#0	□	☑	☑	☑	□
13	R_STATUS_SAVE	Word	16#0	□	☑	☑	☑	□
14	▶ R_CONNECT	TCON_IP_v4		□	☑	☑	☑	□
15	▶ R_INPUT_REG	Array[0..1] of Int		□	☑	☑	☑	☑

Dissection

1. From network 1to 4: net1and2: Digital, net 3and 4: Analog
2. Network 5 is server
3. Network 6,7: Holding register
4. From network 8 to 13: Digital data read
5. From network 14to 17: Digital data write

Lab view



IP Address

port (502)
502

input starting address
0

coils starting address
0

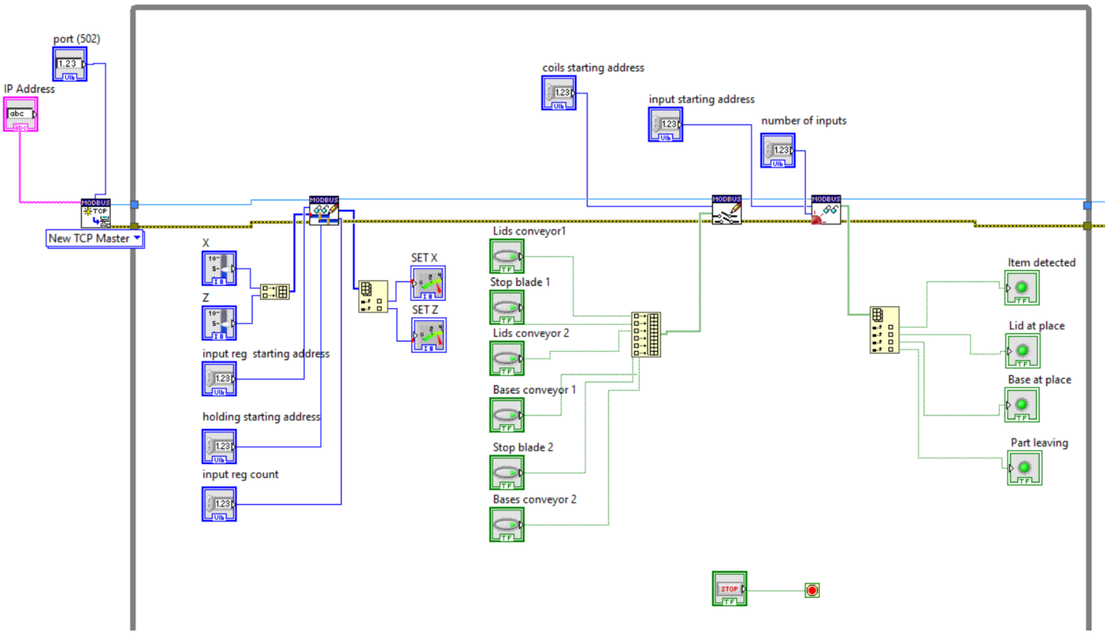
number of inputs
0

input reg count
0

holding starting address
0

input reg starting address
0

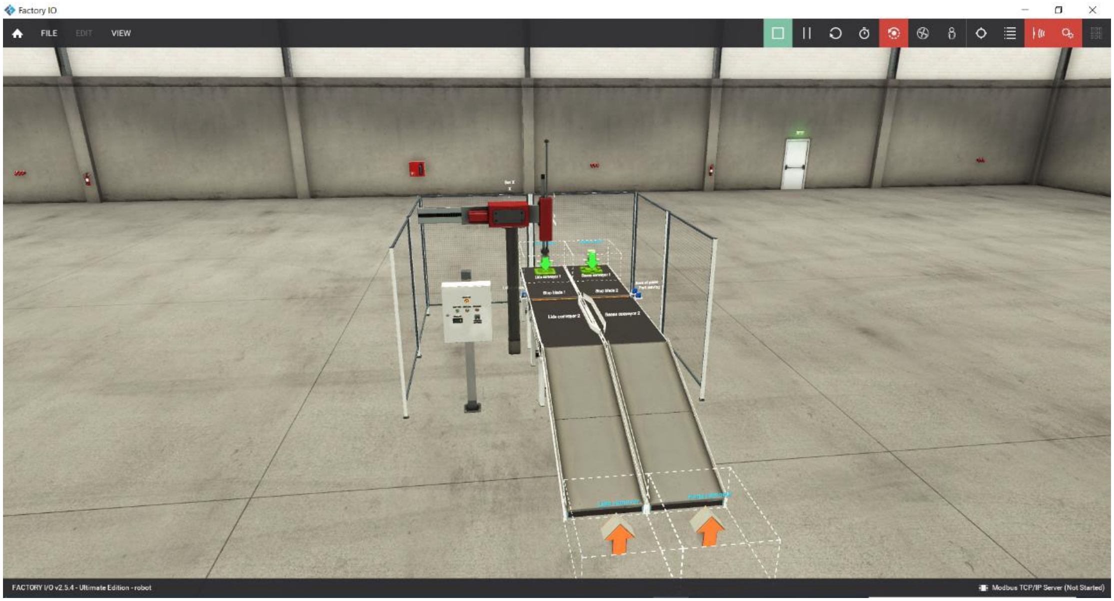
MOD LabVIEW HMI ۱۳ Figure

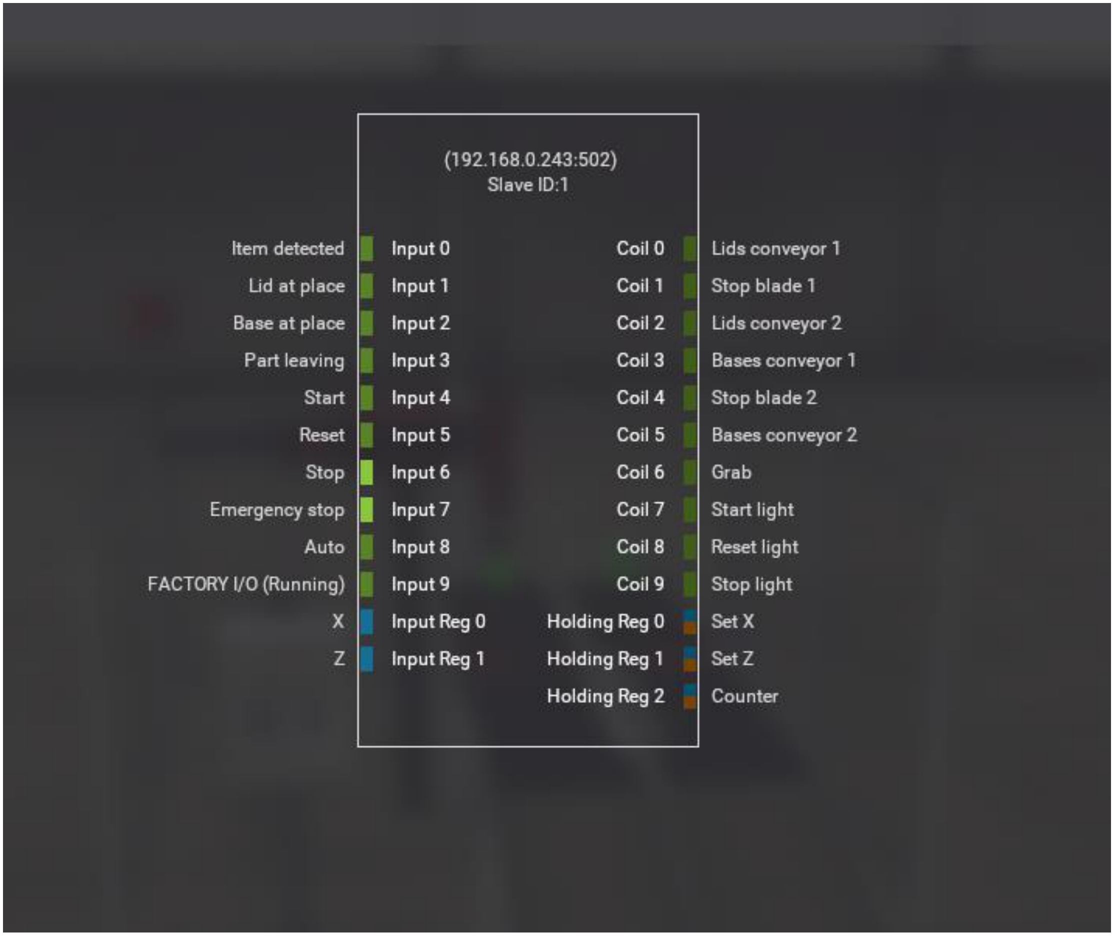


The problem we encountered

Extra project

OPC Industrial Network


Factory IO
FILE
EDIT
VIEW
FACTORY I/O v2.5.4 - Ultimate Edition - robot
Modbus TCP/IP Server (Not Started)



(192.168.0.243:502)
Slave ID:1

Item detected	Input 0	Coil 0	Lids conveyor 1
Lid at place	Input 1	Coil 1	Stop blade 1
Base at place	Input 2	Coil 2	Lids conveyor 2
Part leaving	Input 3	Coil 3	Bases conveyor 1
Start	Input 4	Coil 4	Stop blade 2
Reset	Input 5	Coil 5	Bases conveyor 2
Stop	Input 6	Coil 6	Grab
Emergency stop	Input 7	Coil 7	Start light
Auto	Input 8	Coil 8	Reset light
FACTORY I/O (Running)	Input 9	Coil 9	Stop light
X	Input Reg 0	Holding Reg 0	Set X
Z	Input Reg 1	Holding Reg 1	Set Z
		Holding Reg 2	Counter

Tia portal

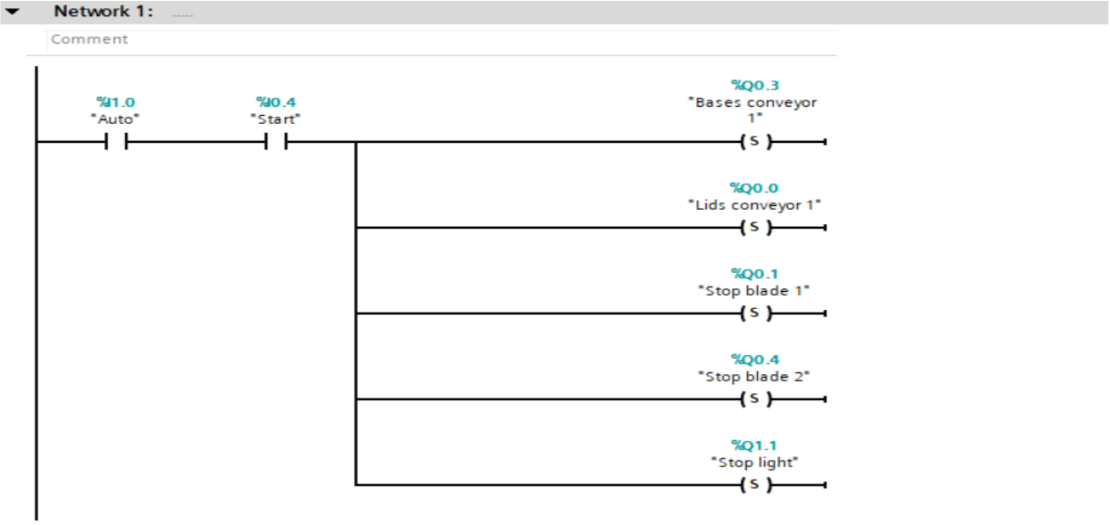


Figure 14Network1

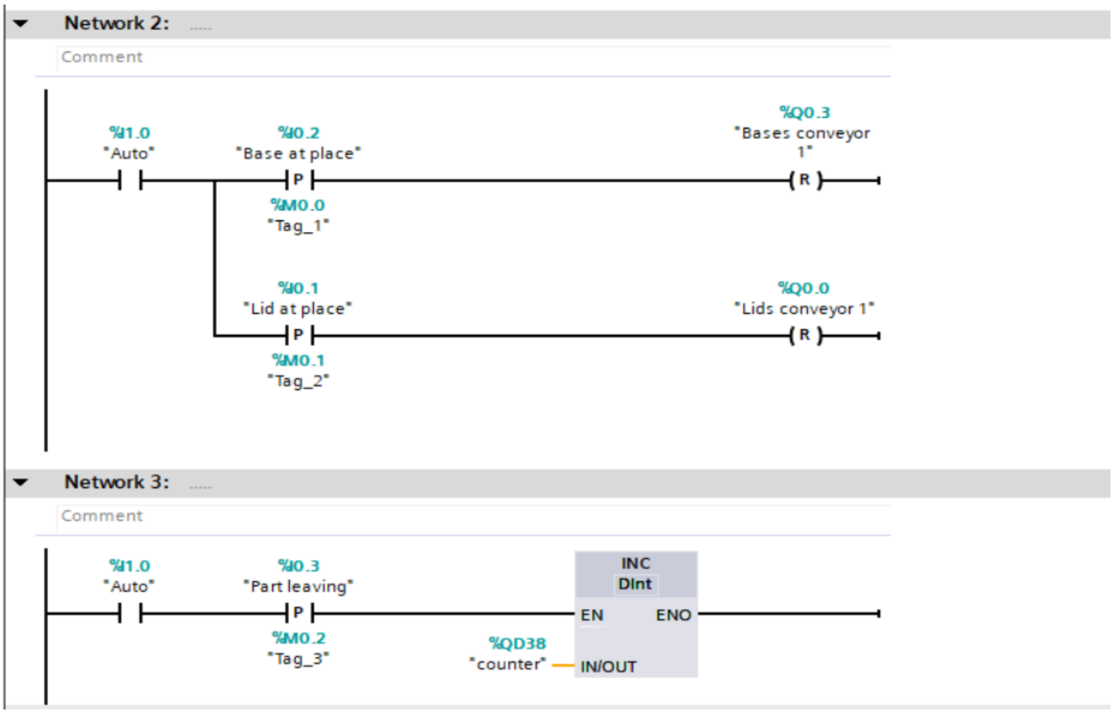
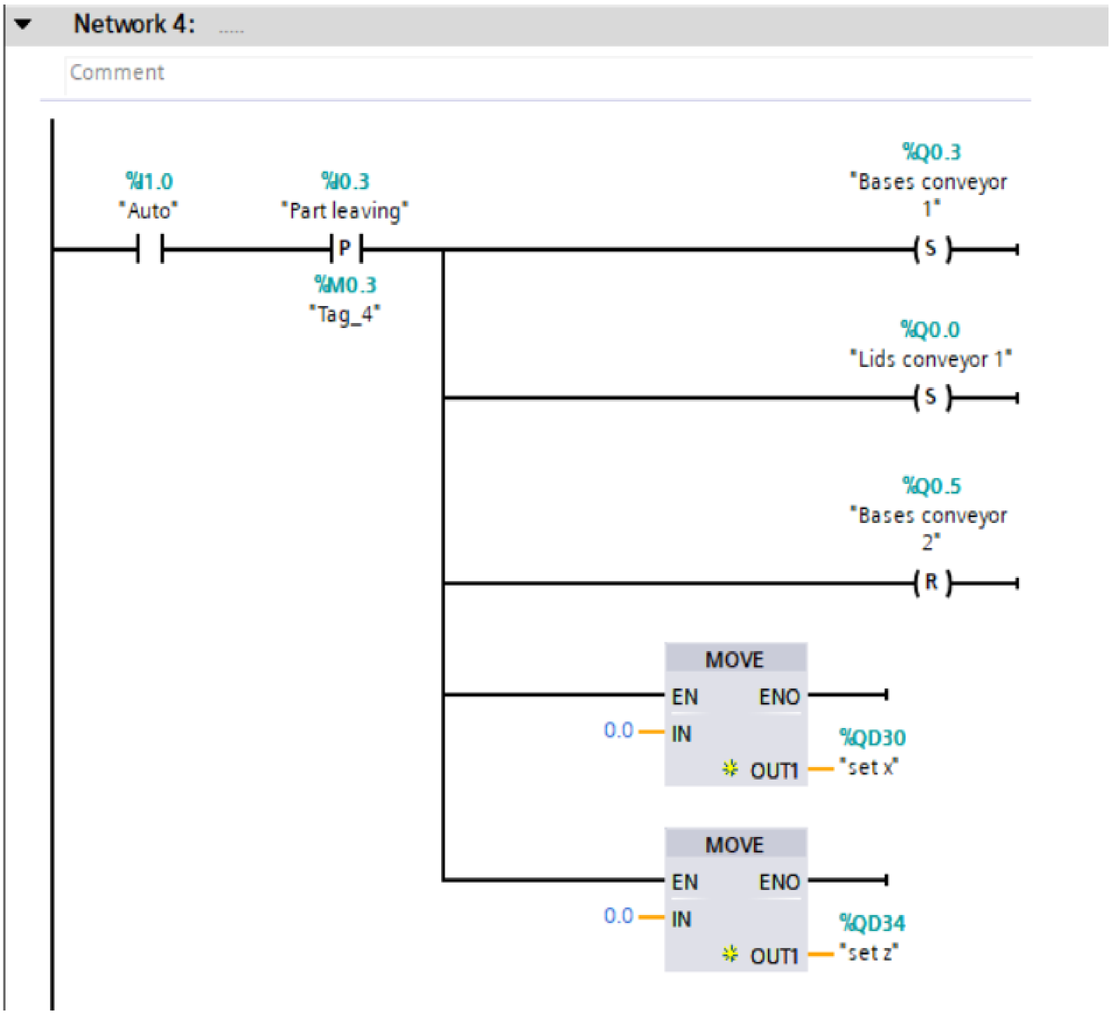
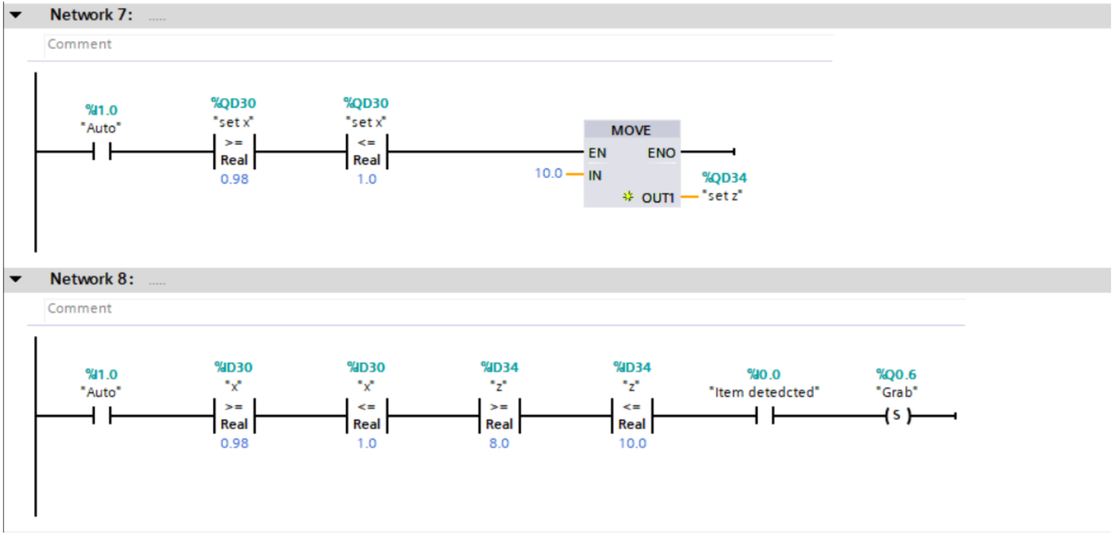


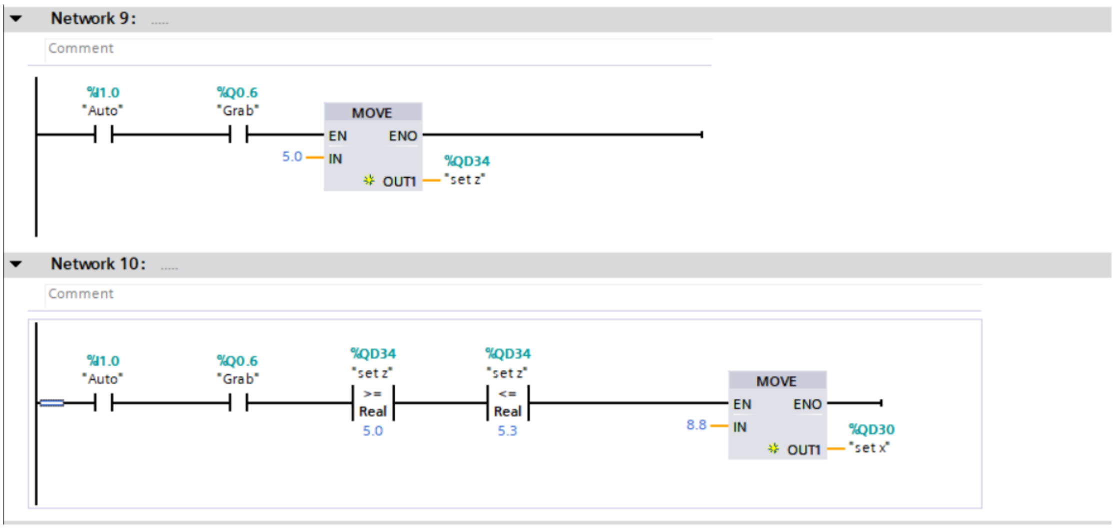
Figure 15 network2,3



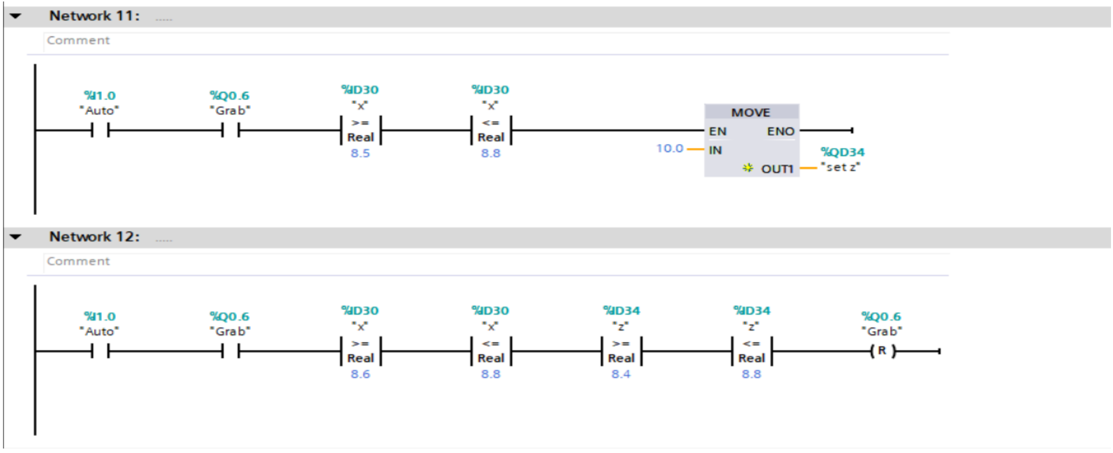
network4 ١٦ Figure



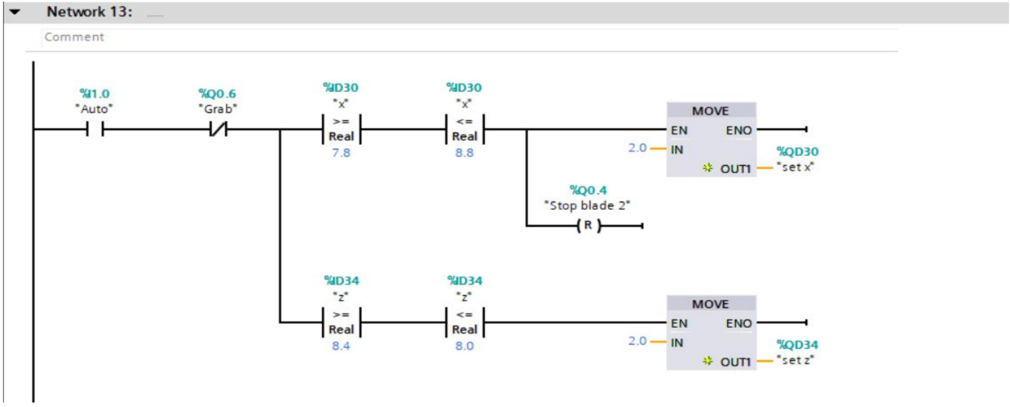
network7,8 ١٧ Figure



network9,10 ۱۸ Figure



network11,12 ۱۹ Figure



network13 ۲۰ Figure

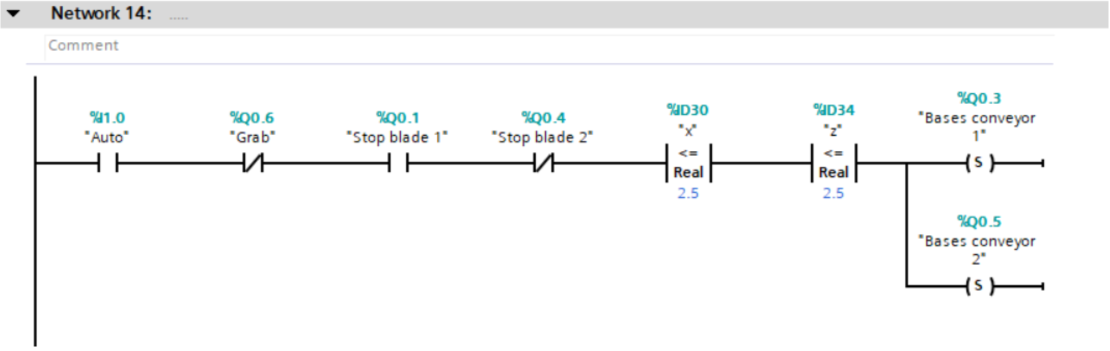
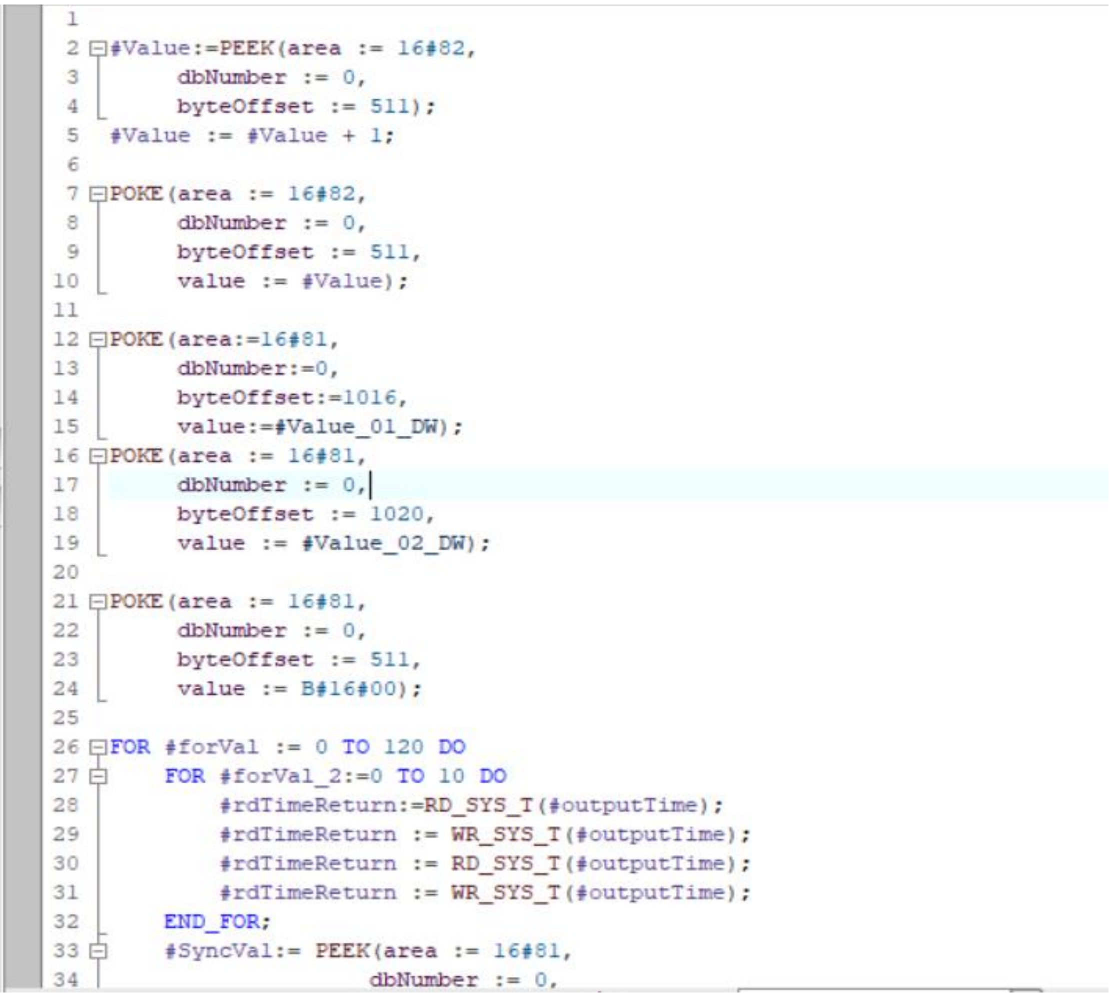
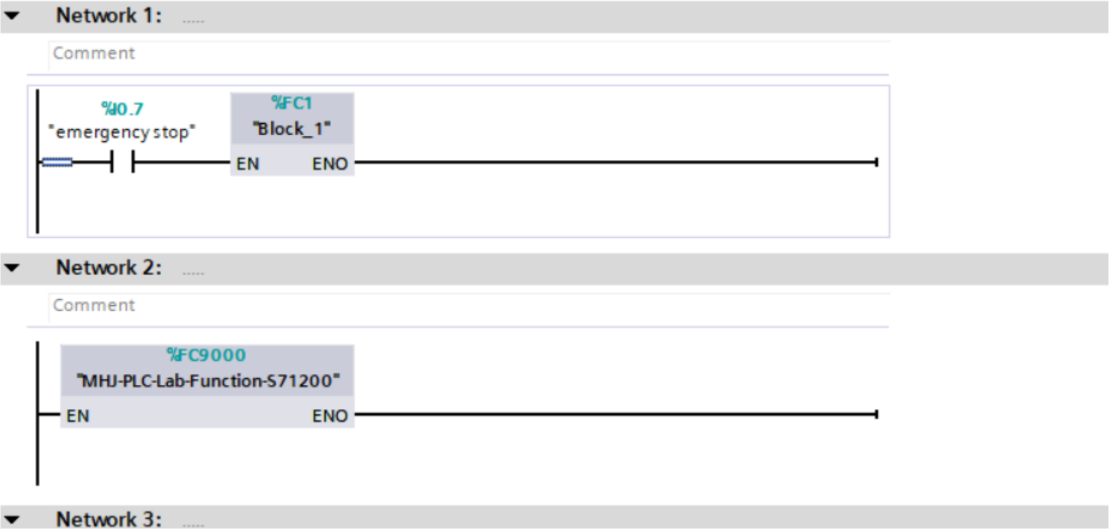


Figure 21network14

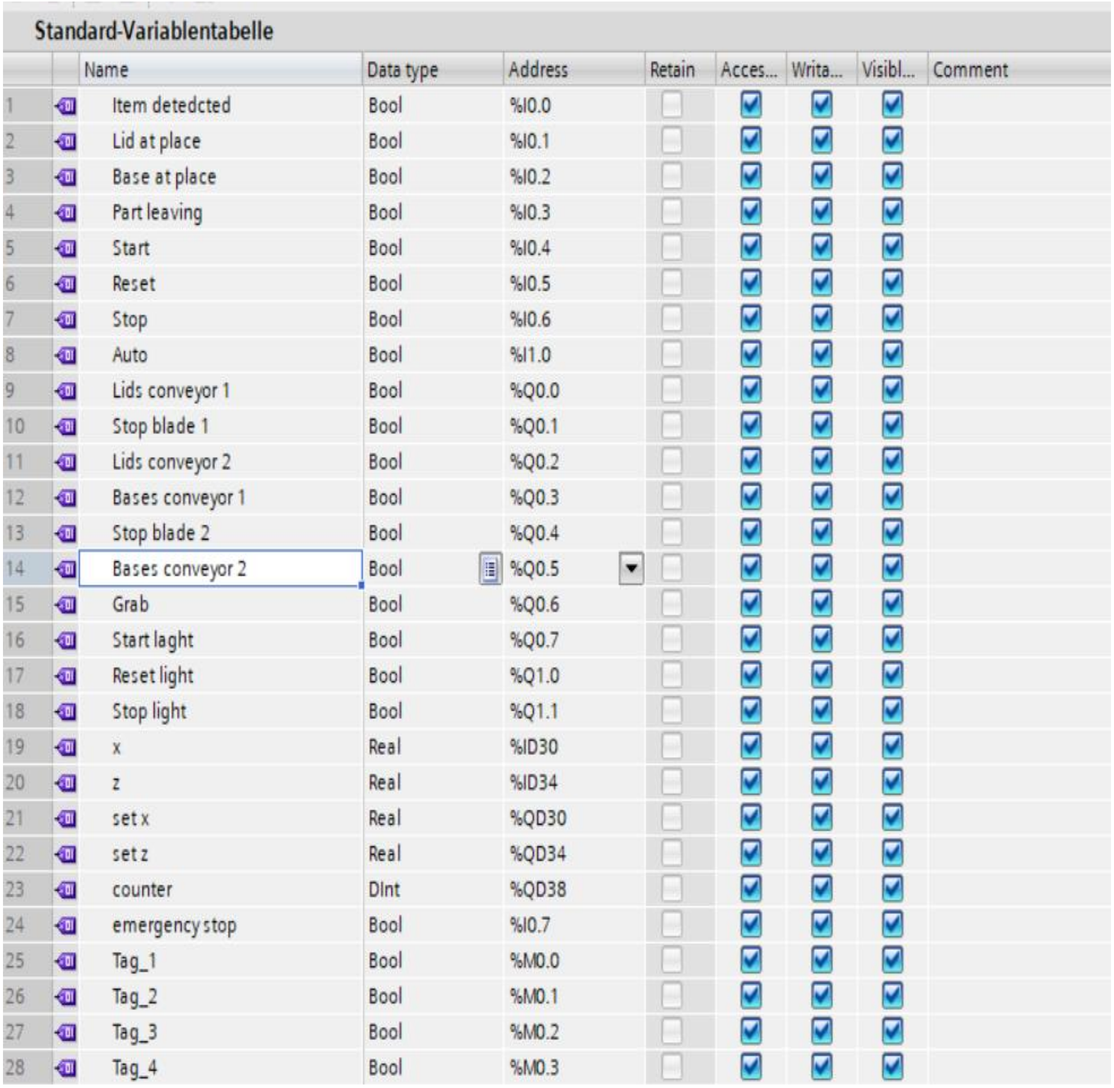


```

#Value:=PEEK(area := 16#82,
     dbNumber := 0,
     byteOffset := 511);
#Value := #Value + 1;

POKE(area := 16#82,
     dbNumber := 0,
     byteOffset := 511,
     value := #Value);

POKE(area:=16#81,
     dbNumber:=0,
     byteOffset:=1016,
     value:=#Value_01_DW);
POKE(area := 16#81,
     dbNumber := 0,
     byteOffset := 1020,
     value := #Value_02_DW);

POKE(area := 16#81,
     dbNumber := 0,
     byteOffset := 511,
     value := B#16#00);

FOR #forVal := 0 TO 120 DO
    FOR #forVal_2:=0 TO 10 DO
        #rdTimeReturn:=RD_SYS_T(#outputTime);
        #rdTimeReturn := WR_SYS_T(#outputTime);
        #rdTimeReturn := RD_SYS_T(#outputTime);
        #rdTimeReturn := WR_SYS_T(#outputTime);
    END_FOR;
    #SyncVal:= PEEK(area := 16#81,
                   dbNumber := 0,
```

Figure 22 Code for connect to plcsim



Main ۲۳ Figure


Standard-Variablentabelle

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Item detedcted	Bool	%I0.0	☐	☑	☑	☑	
2	Lid at place	Bool	%I0.1	☐	☑	☑	☑	
3	Base at place	Bool	%I0.2	☐	☑	☑	☑	
4	Part leaving	Bool	%I0.3	☐	☑	☑	☑	
5	Start	Bool	%I0.4	☐	☑	☑	☑	
6	Reset	Bool	%I0.5	☐	☑	☑	☑	
7	Stop	Bool	%I0.6	☐	☑	☑	☑	
8	Auto	Bool	%I1.0	☐	☑	☑	☑	
9	Lids conveyor 1	Bool	%Q0.0	☐	☑	☑	☑	
10	Stop blade 1	Bool	%Q0.1	☐	☑	☑	☑	
11	Lids conveyor 2	Bool	%Q0.2	☐	☑	☑	☑	
12	Bases conveyor 1	Bool	%Q0.3	☐	☑	☑	☑	
13	Stop blade 2	Bool	%Q0.4	☐	☑	☑	☑	
14	Bases conveyor 2	Bool	%Q0.5	☐	☑	☑	☑	
15	Grab	Bool	%Q0.6	☐	☑	☑	☑	
16	Start laght	Bool	%Q0.7	☐	☑	☑	☑	
17	Reset light	Bool	%Q1.0	☐	☑	☑	☑	
18	Stop light	Bool	%Q1.1	☐	☑	☑	☑	
19	x	Real	%ID30	☐	☑	☑	☑	
20	z	Real	%ID34	☐	☑	☑	☑	
21	set x	Real	%QD30	☐	☑	☑	☑	
22	set z	Real	%QD34	☐	☑	☑	☑	
23	counter	DInt	%QD38	☐	☑	☑	☑	
24	emergency stop	Bool	%I0.7	☐	☑	☑	☑	
25	Tag_1	Bool	%M0.0	☐	☑	☑	☑	
26	Tag_2	Bool	%M0.1	☐	☑	☑	☑	
27	Tag_3	Bool	%M0.2	☐	☑	☑	☑	
28	Tag_4	Bool	%M0.3	☐	☑	☑	☑	

Tags ۲٤ Figure

Lab view

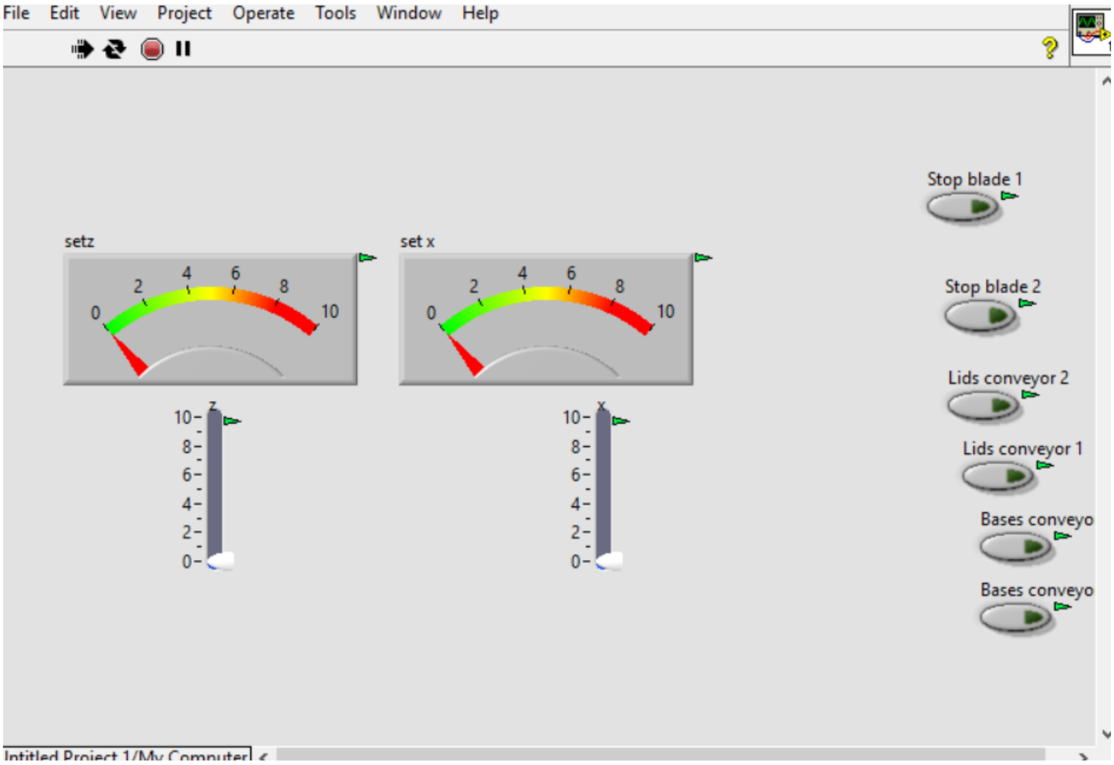
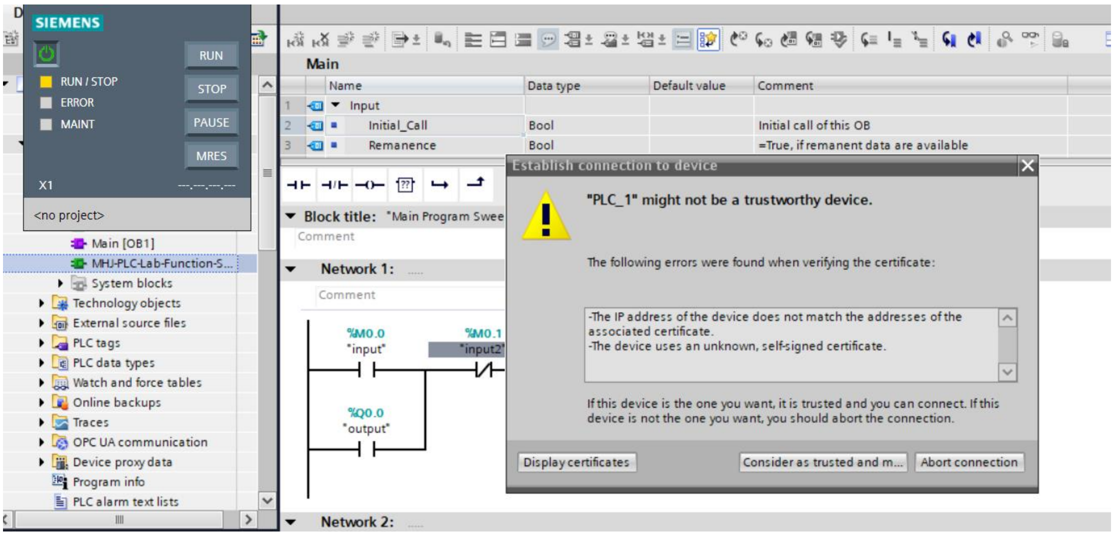


Figure 25 Labview HMI

Linking process

0



	Name	Data type	Default value	Comment
1	Input			
2	Initial_Call	Bool		Initial call of this OB
3	Remanence	Bool		=True, if remanent data are available

Establish connection to device

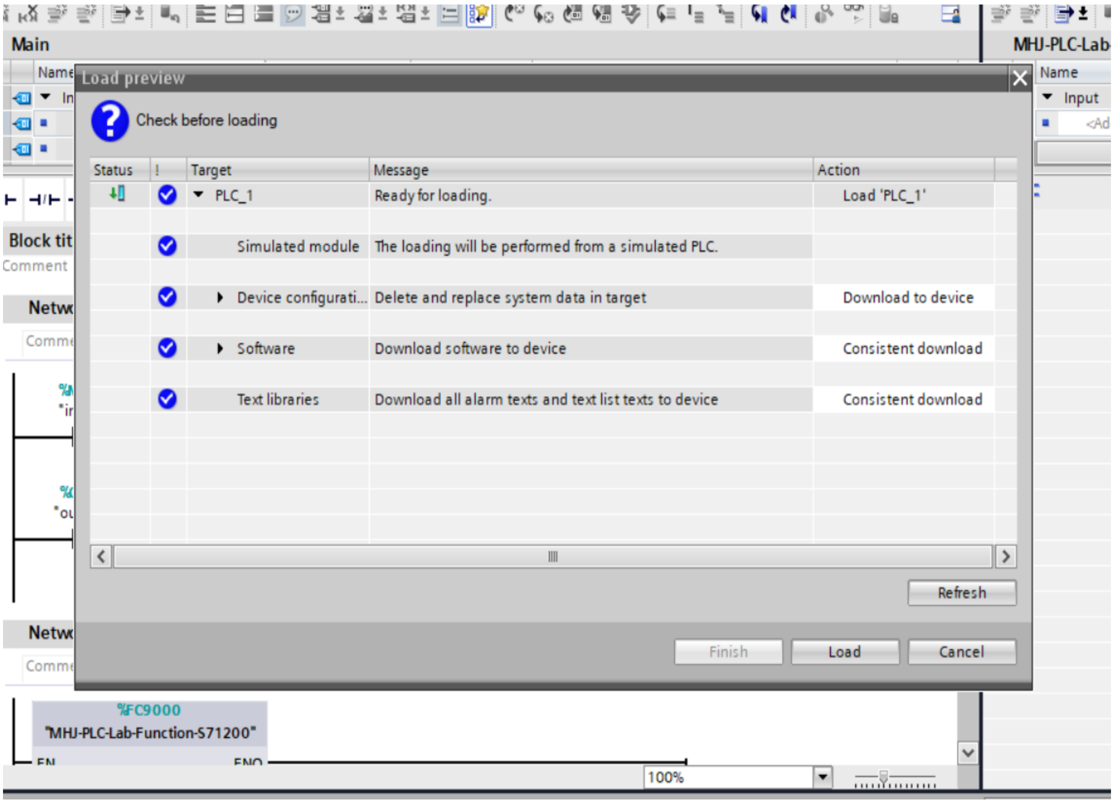
"PLC_1" might not be a trustworthy device.

The following errors were found when verifying the certificate:

-The IP address of the device does not match the addresses of the associated certificate.
-The device uses an unknown, self-signed certificate.

If this device is the one you want, it is trusted and you can connect. If this device is not the one you want, you should abort the connection.

Connect tia portal with plcsim ٢٦ Figure

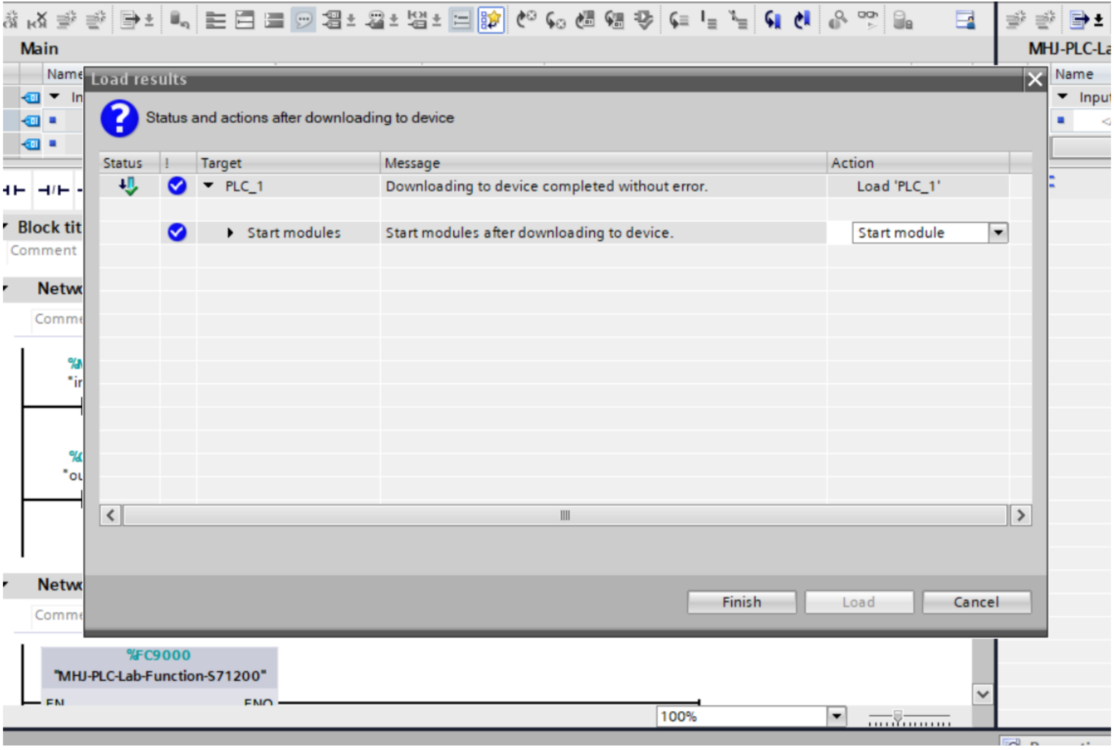


Load preview

Check before loading

Status	!	Target	Message	Action
	✓	PLC_1	Ready for loading.	Load 'PLC_1'
	✓	Simulated module	The loading will be performed from a simulated PLC.	
	✓	Device configurati...	Delete and replace system data in target	Download to device
	✓	Software	Download software to device	Consistent download
	✓	Text libraries	Download all alarm texts and text list texts to device	Consistent download

Load the code ٢٧ Figure

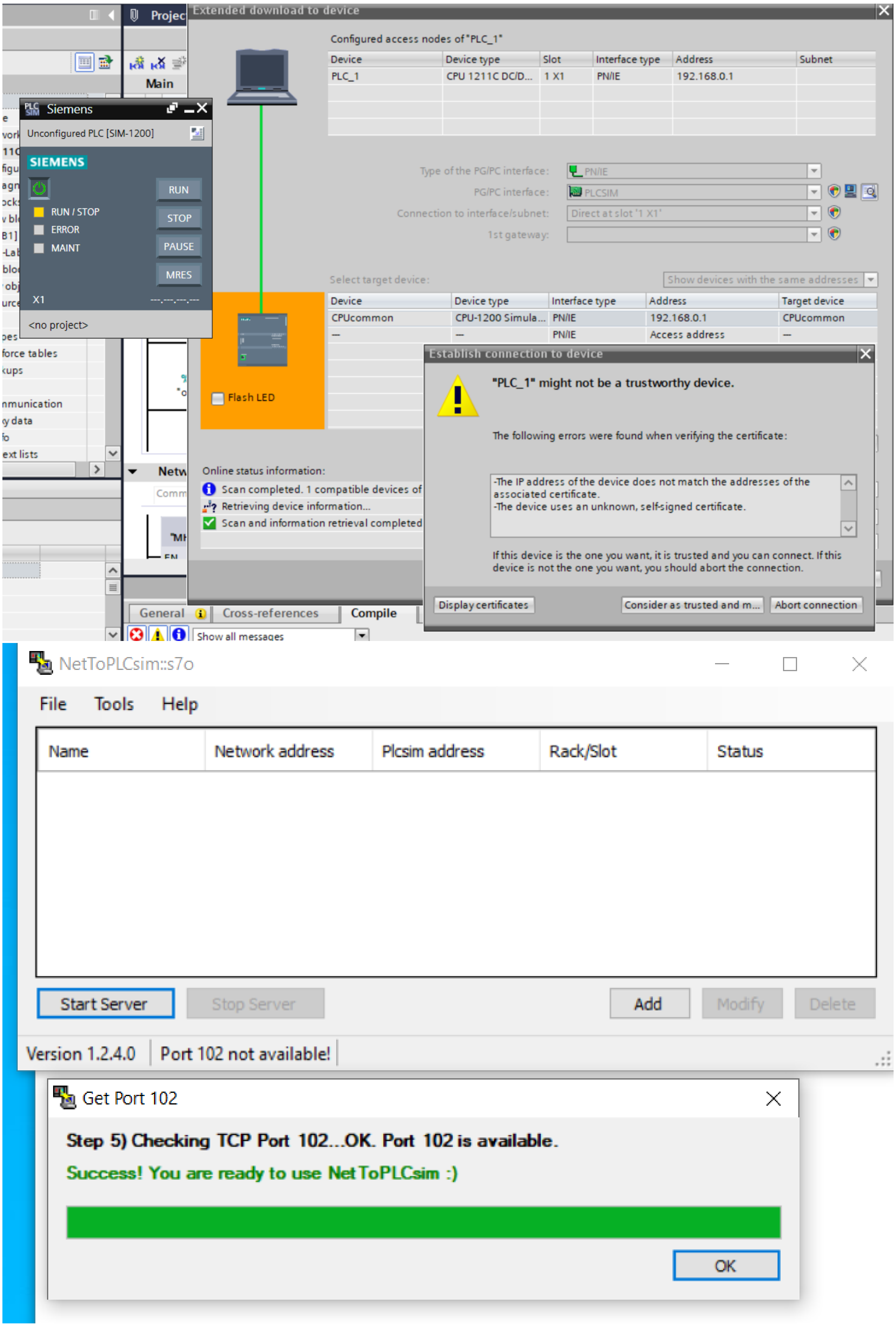


Load results

Status and actions after downloading to device

Status	!	Target	Message	Action
	✓	PLC_1	Downloading to device completed without error.	Load 'PLC_1'
	✓	Start modules	Start modules after downloading to device.	Start module

Finsh the code ٢٨ Figure



Extended download to device

Configured access nodes of "PLC_1"

Device	Device type	Slot	Interface type	Address	Subnet
PLC_1	CPU 1211C DC/D...	1 X1	PN/IE	192.168.0.1	

Type of the PG/PC interface: PN/IE

PG/PC interface: PLCSIM

Connection to interface/subnet: Direct at slot '1 X1'

1st gateway:

Select target device: Show devices with the same addresses

Device	Device type	Interface type	Address	Target device
CPUcommon	CPU-1200 Simula...	PN/IE	192.168.0.1	CPUcommon
—	—	PN/IE	Access address	—

Online status information:

Scan completed. 1 compatible devices of

Retrieving device information...

Scan and information retrieval completed

Establish connection to device

"PLC_1" might not be a trustworthy device.

The following errors were found when verifying the certificate:

-The IP address of the device does not match the addresses of the associated certificate.
-The device uses an unknown, self-signed certificate.

If this device is the one you want, it is trusted and you can connect. If this device is not the one you want, you should abort the connection.

Display certificates | Consider as trusted and m... | Abort connection

NetToPLCsim::s7o

File Tools Help

Name	Network address	Plcsim address	Rack/Slot	Status

Start Server | Stop Server | Add | Modify | Delete

Version 1.2.4.0 | Port 102 not available!

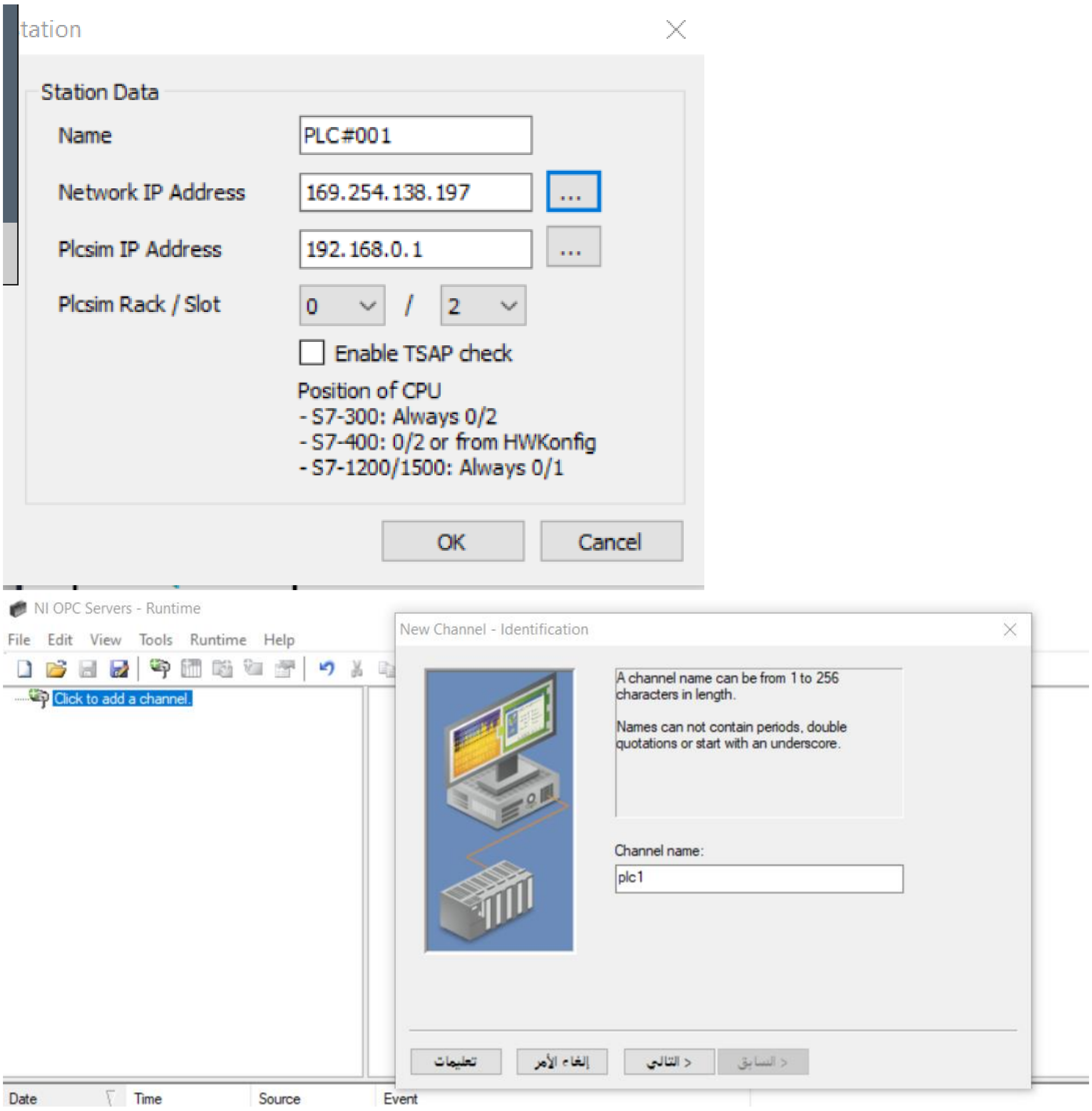
Get Port 102

Step 5) Checking TCP Port 102...OK. Port 102 is available.

Success! You are ready to use NetToPLCsim :)

OK

Connect ip of plcsim to the dives ٢٩ Figure



tation

Station Data

Name: PLC#001

Network IP Address: 169.254.138.197

Plcsim IP Address: 192.168.0.1

Plcsim Rack / Slot: 0 / 2

□ Enable TSAP check

Position of CPU
- S7-300: Always 0/2
- S7-400: 0/2 or from HWKonfig
- S7-1200/1500: Always 0/1

NI OPC Servers - Runtime

Click to add a channel.

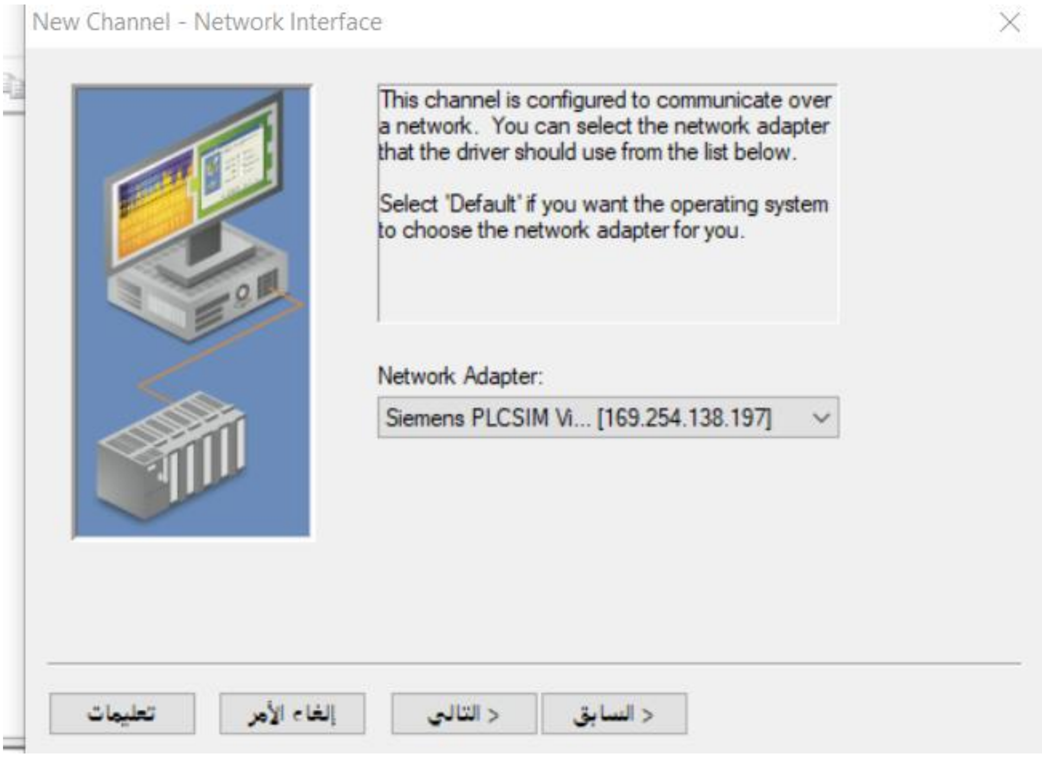
New Channel - Identification

A channel name can be from 1 to 256 characters in length.

Names can not contain periods, double quotations or start with an underscore.

Channel name: plc1

Use OPC erver to connect ۳۰ Figure



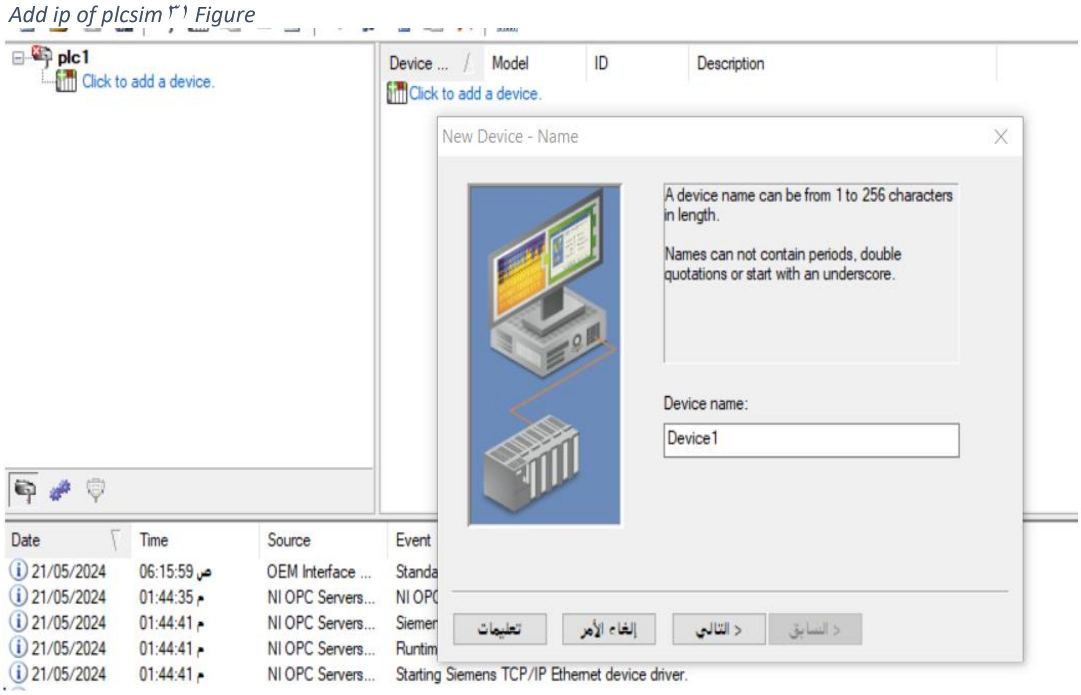
New Channel - Network Interface

This channel is configured to communicate over a network. You can select the network adapter that the driver should use from the list below.

Select 'Default' if you want the operating system to choose the network adapter for you.

Network Adapter: Siemens PLCSIM Vi... [169.254.138.197]

Add ip of plcsim ٣١ Figure



plc1
Click to add a device.

Device ...	Model	ID	Description
Click to add a device.			

New Device - Name

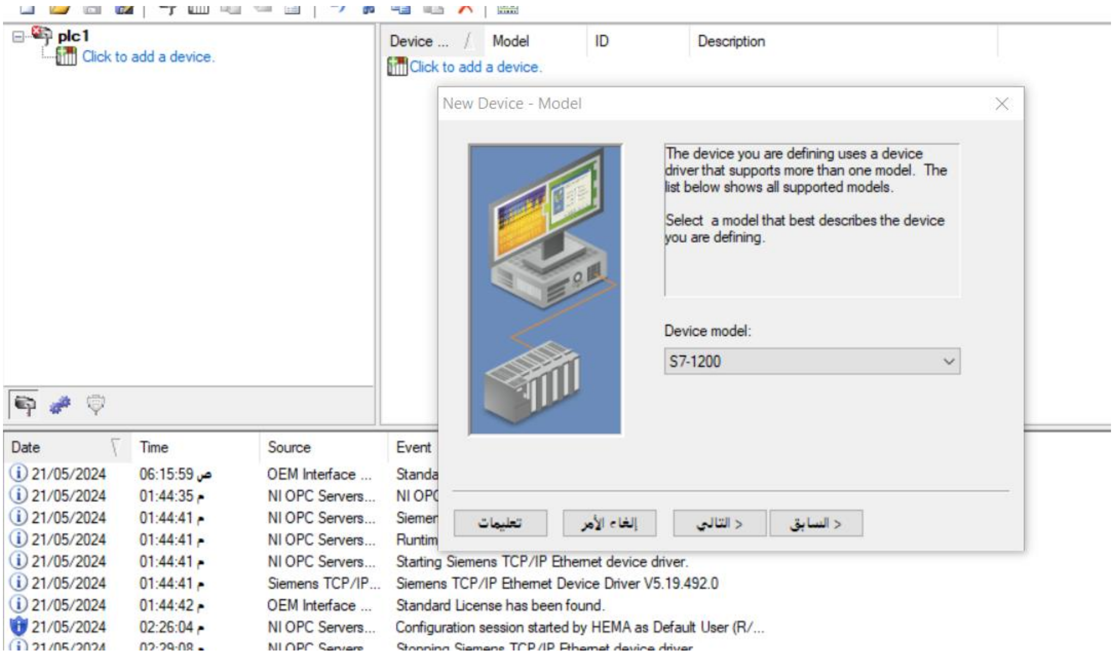
A device name can be from 1 to 256 characters in length.

Names can not contain periods, double quotations or start with an underscore.

Device name: Device1

Date	Time	Source	Event
21/05/2024	06:15:59 ص	OEM Interface ...	Standa
21/05/2024	01:44:35 م	NI OPC Servers...	NI OPC
21/05/2024	01:44:41 م	NI OPC Servers...	Siemer
21/05/2024	01:44:41 م	NI OPC Servers...	Runtim
21/05/2024	01:44:41 م	NI OPC Servers...	Starting Siemens TCP/IP Ethernet device driver.

OPC step2 ٣٢ Figure



New Device - Model

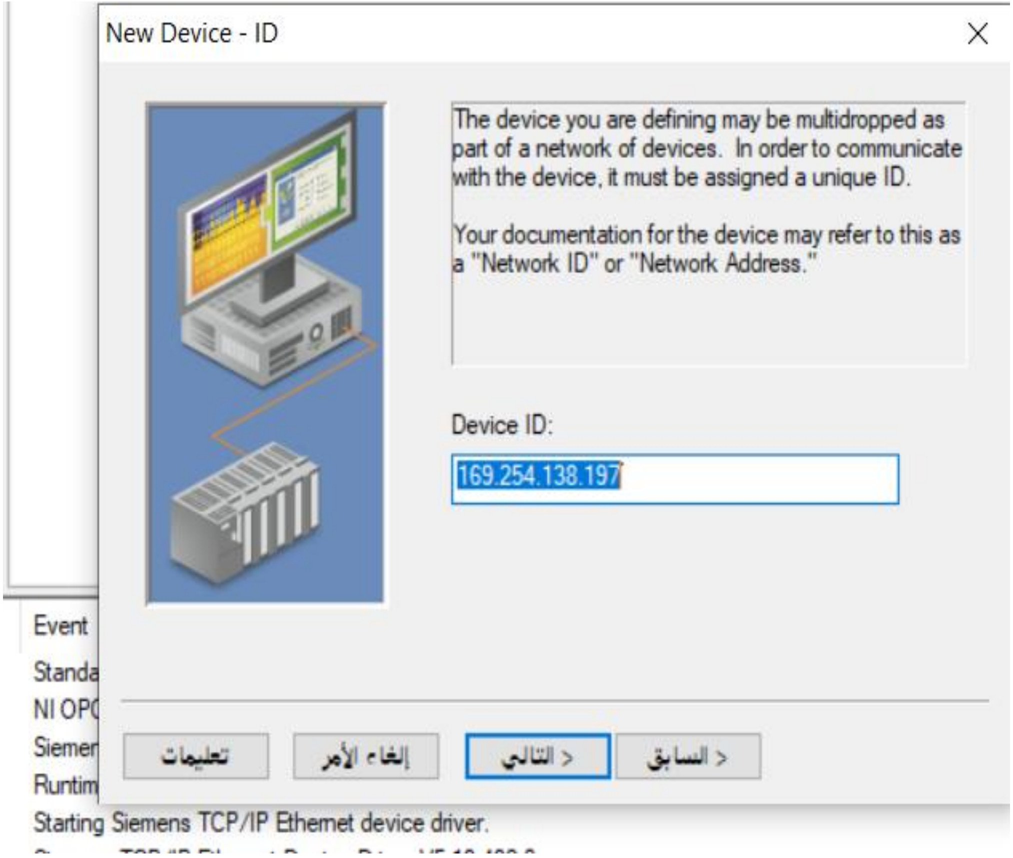
The device you are defining uses a device driver that supports more than one model. The list below shows all supported models.

Select a model that best describes the device you are defining.

Device model: S7-1200

Date	Time	Source	Event
21/05/2024	06:15:59 ص	OEM Interface ...	Standa
21/05/2024	01:44:35 م	NI OPC Servers...	NI OPC
21/05/2024	01:44:41 م	NI OPC Servers...	Siemen
21/05/2024	01:44:41 م	NI OPC Servers...	Runtim
21/05/2024	01:44:41 م	NI OPC Servers...	Starting Siemens TCP/IP Ethernet device driver.
21/05/2024	01:44:41 م	Siemens TCP/IP...	Siemens TCP/IP Ethernet Device Driver V5.19.492.0
21/05/2024	01:44:42 م	OEM Interface ...	Standard License has been found.
21/05/2024	02:26:04 م	NI OPC Servers...	Configuration session started by HEMA as Default User (R/...
21/05/2024	02:29:08 م	NI OPC Servers	Stopping Siemens TCP/IP Ethernet device driver

OPC step3 ٣٣ Figure



New Device - ID

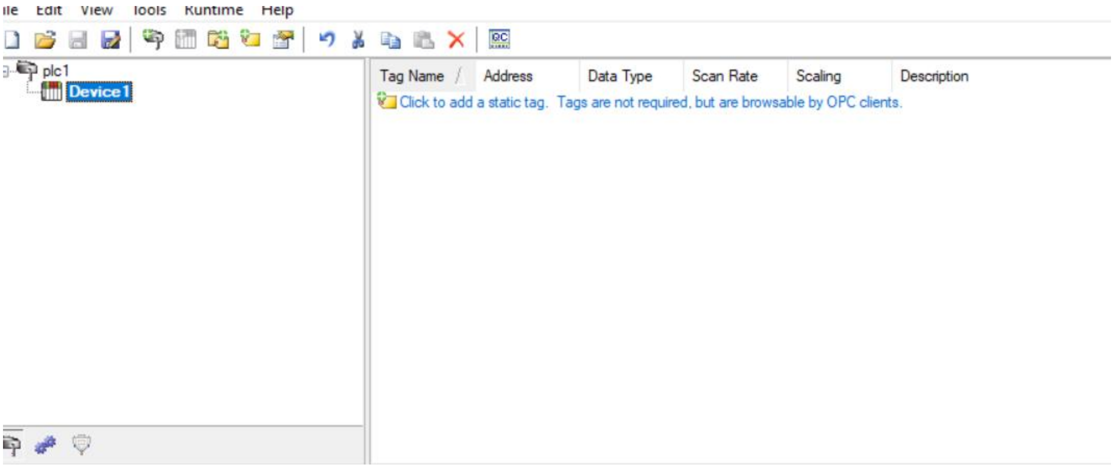
The device you are defining may be multidropped as part of a network of devices. In order to communicate with the device, it must be assigned a unique ID.

Your documentation for the device may refer to this as a "Network ID" or "Network Address."

Device ID: 169.254.138.197

Starting Siemens TCP/IP Ethernet device driver.

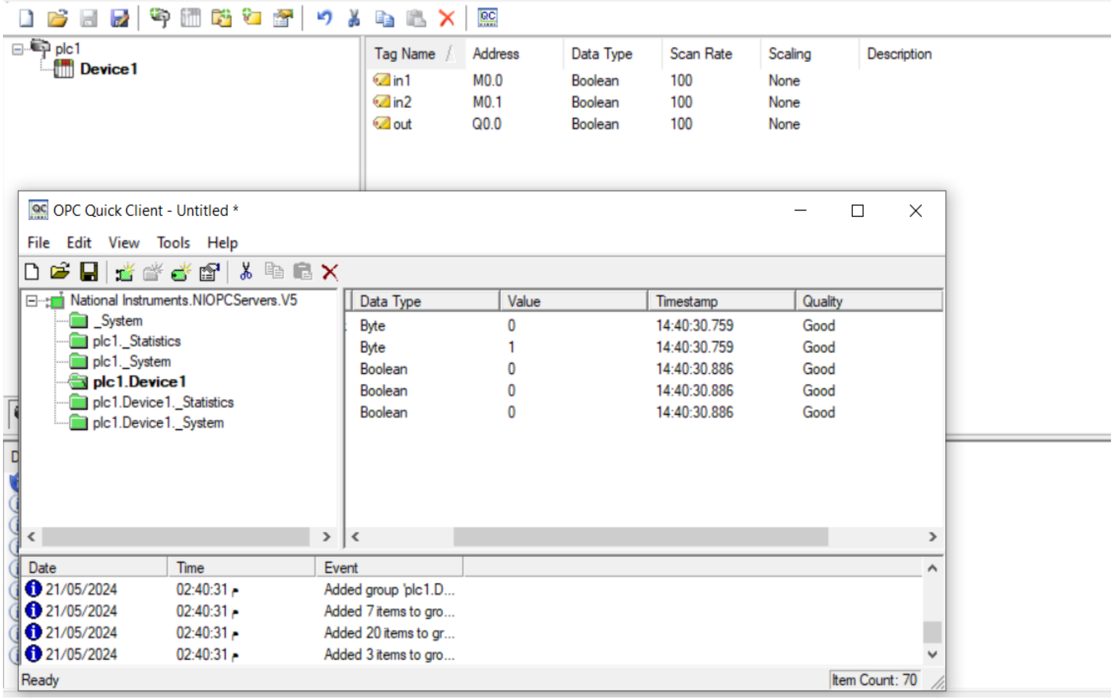
Add the ip of device at net to plcsim ٣٤ Figure



Tag Name	Address	Data Type	Scan Rate	Scaling	Description

Click to add a static tag. Tags are not required, but are browsable by OPC clients.

Add the tags in opc server ٣٥ *Figure*



Tag Name	Address	Data Type	Scan Rate	Scaling	Description
in1	M0.0	Boolean	100	None	
in2	M0.1	Boolean	100	None	
out	Q0.0	Boolean	100	None	

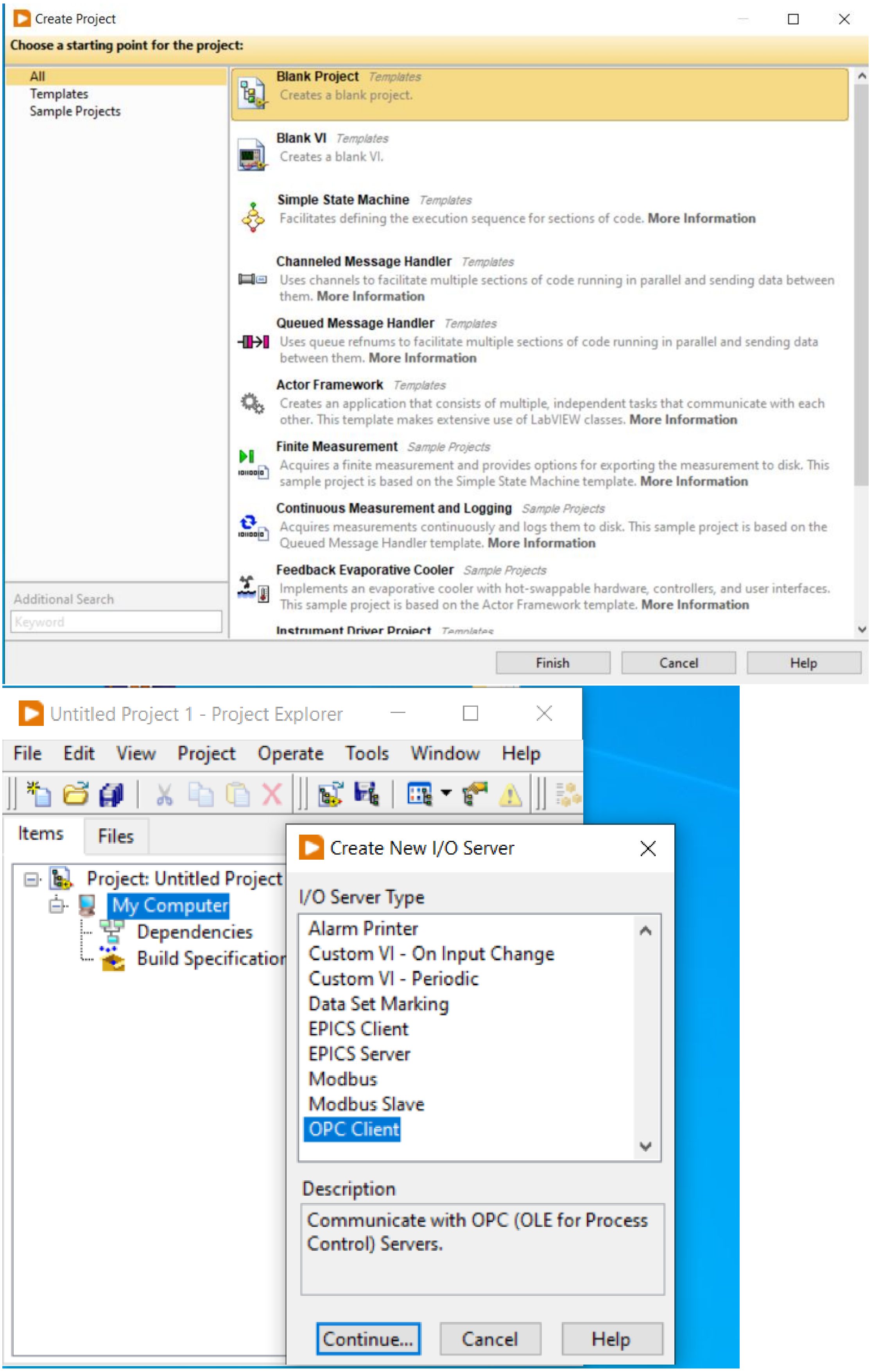
Data Type	Value	Timestamp	Quality
Byte	0	14:40:30.759	Good
Byte	1	14:40:30.759	Good
Boolean	0	14:40:30.886	Good
Boolean	0	14:40:30.886	Good
Boolean	0	14:40:30.886	Good

Date	Time	Event
21/05/2024	02:40:31 م	Added group 'plc1.D...
21/05/2024	02:40:31 م	Added 7 items to gro...
21/05/2024	02:40:31 م	Added 20 items to gr...
21/05/2024	02:40:31 م	Added 3 items to gro...

Ready

Item Count: 70

add step2 ٣٦ *Figure*



Create Project

Choose a starting point for the project:

All
Templates
Sample Projects

Additional Search

Keyword

Blank Project *Templates*
Creates a blank project.

Blank VI *Templates*
Creates a blank VI.

Simple State Machine *Templates*
Facilitates defining the execution sequence for sections of code. **More Information**

Channeled Message Handler *Templates*
Uses channels to facilitate multiple sections of code running in parallel and sending data between them. **More Information**

Queued Message Handler *Templates*
Uses queue refnums to facilitate multiple sections of code running in parallel and sending data between them. **More Information**

Actor Framework *Templates*
Creates an application that consists of multiple, independent tasks that communicate with each other. This template makes extensive use of LabVIEW classes. **More Information**

Finite Measurement *Sample Projects*
Acquires a finite measurement and provides options for exporting the measurement to disk. This sample project is based on the Simple State Machine template. **More Information**

Continuous Measurement and Logging *Sample Projects*
Acquires measurements continuously and logs them to disk. This sample project is based on the Queued Message Handler template. **More Information**

Feedback Evaporative Cooler *Sample Projects*
Implements an evaporative cooler with hot-swappable hardware, controllers, and user interfaces. This sample project is based on the Actor Framework template. **More Information**

Instrument Driver Project *Templates*

Untitled Project 1 - Project Explorer

Project: Untitled Project
My Computer
Dependencies
Build Specification

Create New I/O Server

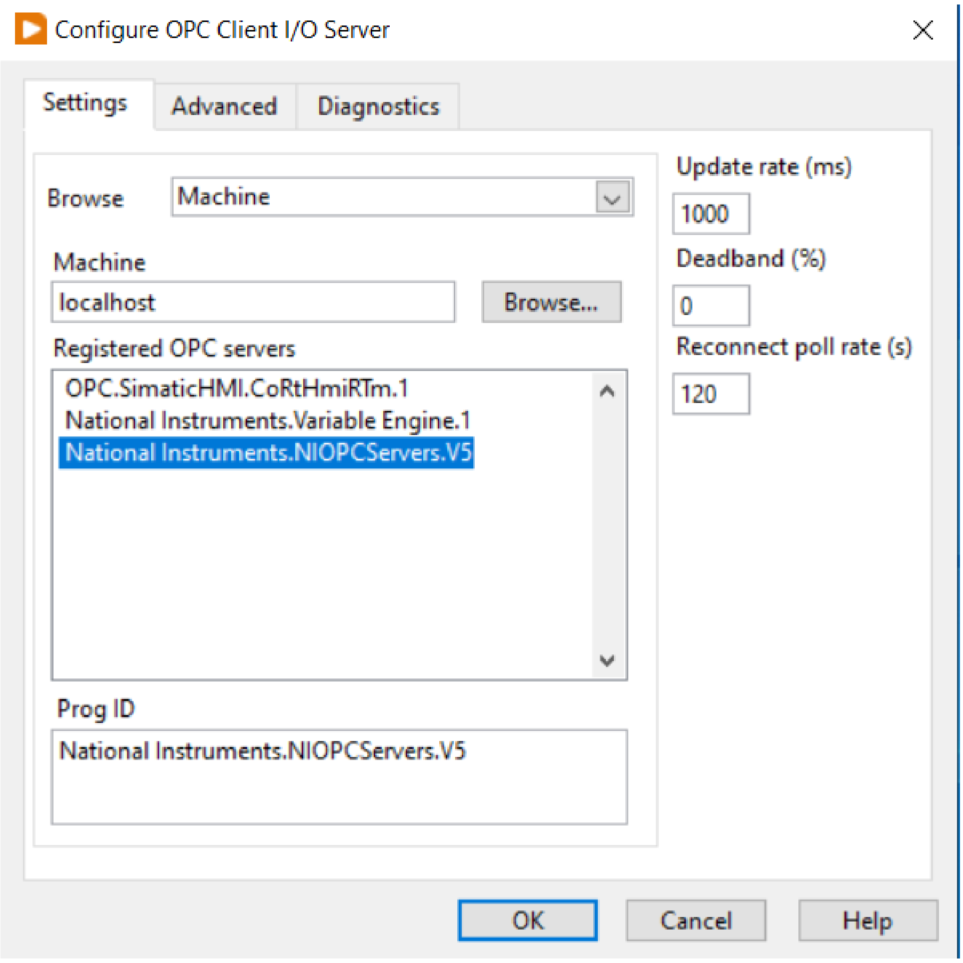
I/O Server Type

Alarm Printer
Custom VI - On Input Change
Custom VI - Periodic
Data Set Marking
EPICS Client
EPICS Server
Modbus
Modbus Slave
OPC Client

Description

Communicate with OPC (OLE for Process Control) Servers.

Create opc IO server ۳۷ Figure



Configure OPC Client I/O Server

Browse: Machine

Machine: localhost

Registered OPC servers:
OPC.SimaticHMI.CoRtHmiRTm.1
National Instruments.Variable Engine.1
National Instruments.NIOPCServers.V5

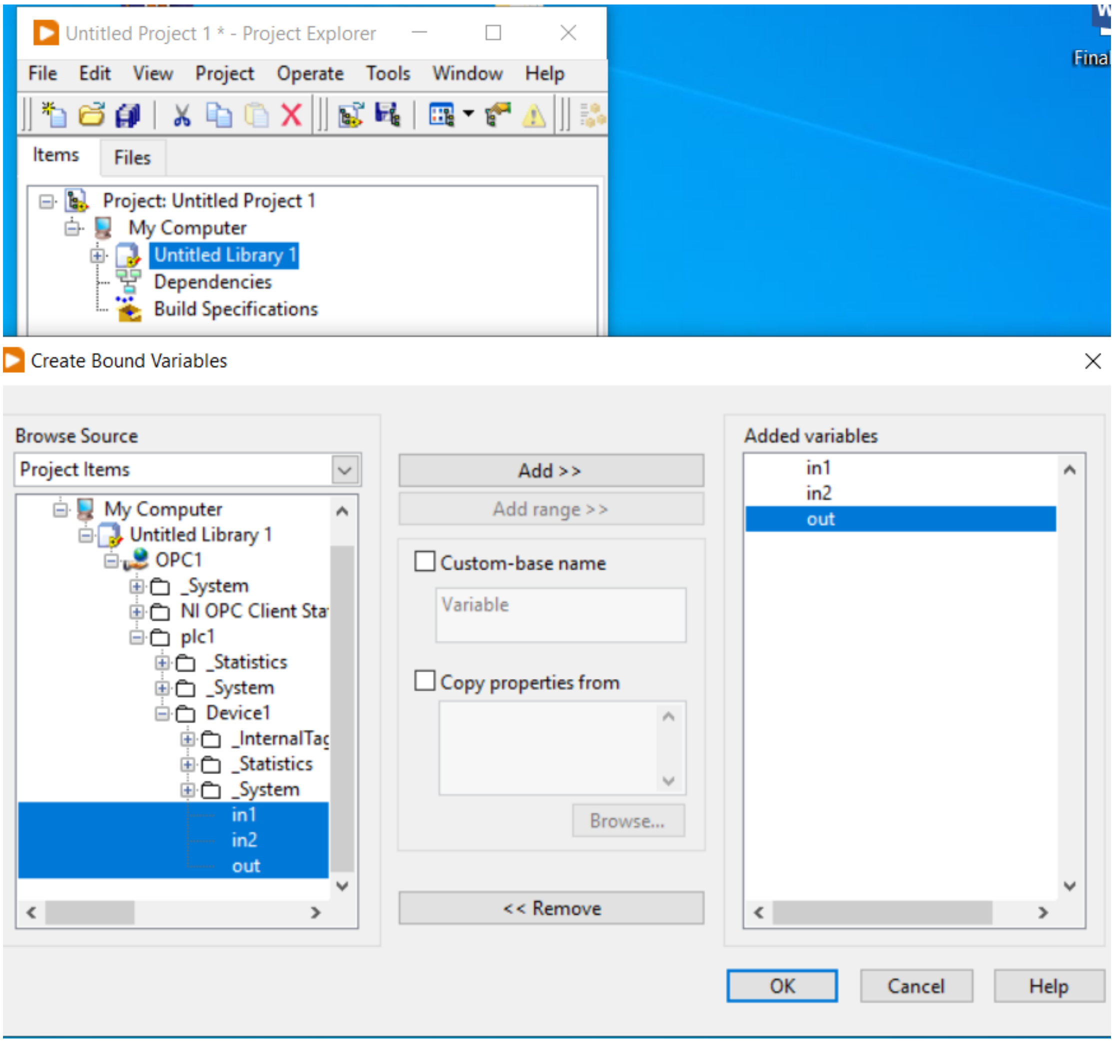
Prog ID: National Instruments.NIOPCServers.V5

Update rate (ms): 1000

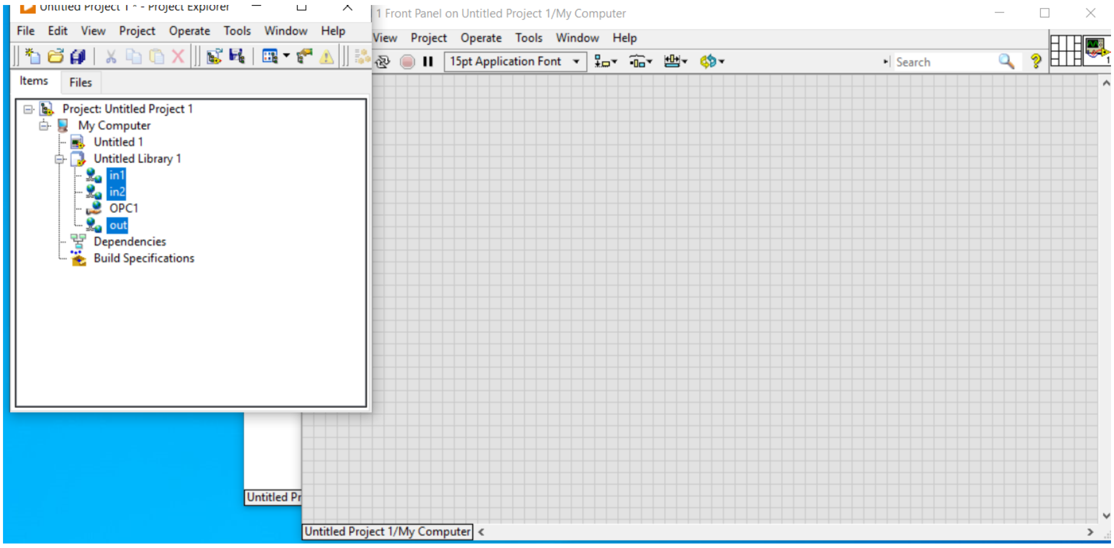
Deadband (%): 0

Reconnect poll rate (s): 120

IO step 2 ۳۸ Figure



IO step 3 ٣٩ Figure



The problem we encountered

1. Step the LabVIEW 32bit
2. Step the opc
3. Connect the ip of plcsim with factory IO
4. Connect tia portal with LabVIEW
5. Connect opc with plcsim advance

Conclusion

This paper dealt with the topics of creating a test environment and then capturing the Modbus TCP communication between the client and servers.

Within the project, a sorting line and an assembly line were created. These lines are simulated using the Factory I/O software and controlled automatically with scripts written in Python programming language. For sorting line, HMI was also created where user can control some parts of line manually through it.

Our testbed allows to create various types of attack on SCADA networks, which can be captured and analyzed. It also serve for educational purposes for students as it can be used in laboratories.

Our designed production lines are quite simple and shows only what are the options of Factory I/O software. For bigger, more realistic looking factory, more Unipi and Advantech PLCs would be needed, together with more routers to create bigger local network. Factory I/O software is really good tool for simulation of factory environment, and with proper hardware enables to create real-looking testing environment for SCADA networks.

References

[1] Product line of programmable controllers and extension modules, UniPi Neuron. User manual and technical documentation. pages 11

[2] J´an Pristaˇs. Generov´an´ı provozu IoT s´ıt´ı a detekce bezpeˇcnostn´ıch incidentů, 6 2018. pages 1

[3] Ondˇrej Ryˇsav´y and Petr Matouˇsek. Monitoring Modbus/TCP traffic using IPFIX. Technical Report FIT-TR-2020-03, Faculty of Information Technology BUT, 2020. pages 11