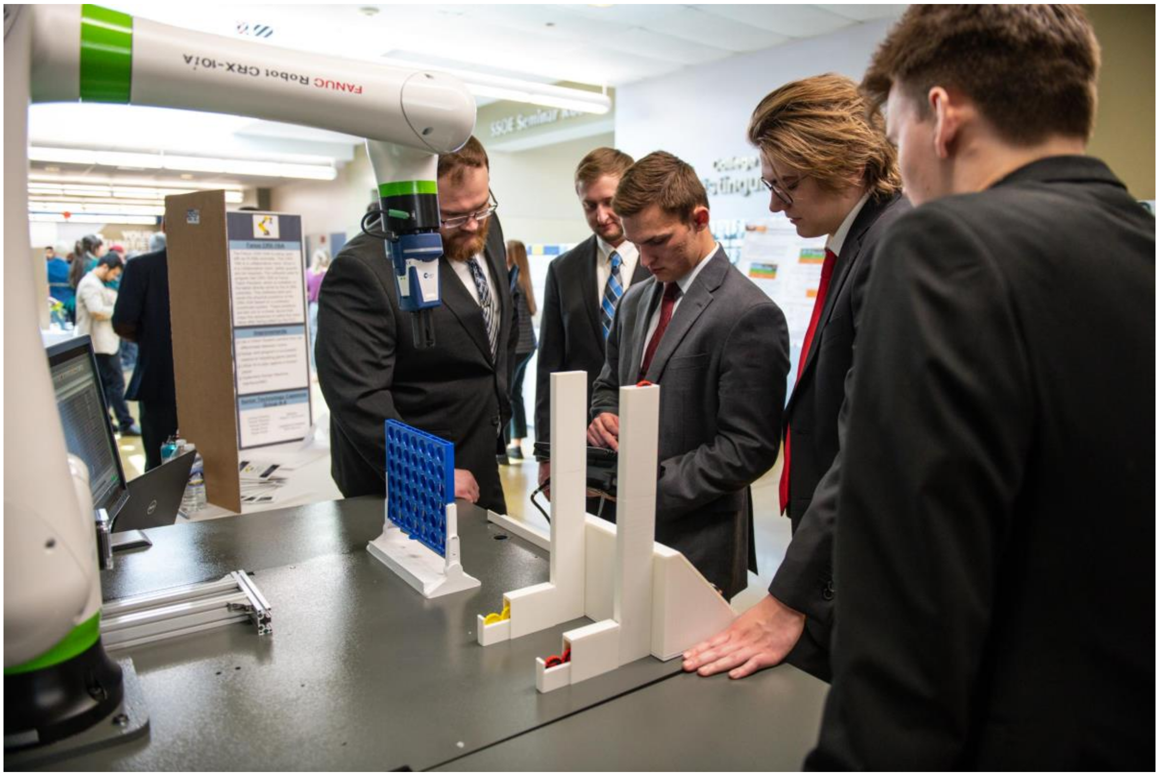
Chapter 22 ROBOTS and VISION and PLCs Oh My

Introduction



From left to right it is Josh Chaney, Sam Kahle, Wyatt Smith, Derek Rose, Tanner Durham

"How to Win Connect 4 Every Time, According to the Computer Scientist Who Solved It - [Lauren Cahn]Updated: Nov. 01, 2023

Ready to discover new expert-approved Connect 4 strategies? Learn how to win Connect 4 and conquer your opponent every time.

Since 1974, the board game Connect 4 has been delighting people of all ages. While Hasbro bills it as fun for ages 6 and up, winning Connect 4 is by no means mere child's play. Sure, school-age kids have been keeping this classic board game flying off the shelves for 50 years, but it is a mind-bender of a game. So much so that how to win Connect 4 was the topic of Dutch computer scientist Victor Allis's master's thesis. In fact, he's credited with literally solving the game in 1988.

"What is beautiful about Connect 4 is that [while it's not simple], it's not so complicated that it's difficult to learn," says Allis. "You can play many, many times and still be discovering new strategies."

Connect 4 has been a cultural phenomenon for decades, with life-size versions available for pub, office and backyard play, as well as online versions that challenge those who are "game" to attempt the near impossible—beating AI. To get the inside scoop on how to always win Connect 4 (or at least most of the time), we got tips from Allis and other leading experts on strategy games. So if you prefer to compete in (and win at) games for two people, you've come to the right place. Here's what you need to know."

Why are we introducing this chapter with a Capstone Group and a Game?

Simply because they were the first group to successfully combine a system of PLC, Vision and Robot in a single application and make it work. There had been other attempts that came close but this was the first that truly linked all three successfully.

It has taken a long time to break through to write a 22nd chapter to this text. The first twenty came pretty fast, but chapter 21 was a stretch with AI. And that was all that was ever envisioned for the topic of automation. But, along came robots and then vision and they had to be included. Today all three: the PLC, the robot and vision are used to control many processes and the inclusion of all three should be a standard for most texts.

The purchase of the equipment below has just occurred and will be part of the ongoing move to have a variety of different robotic arms, plcs and vision systems. We have has Fanuc robots for a number of years but recently added the robot shown above – the CRX collaborative Fanuc robot (white one). The purchase of a new Universal collaborative robot will add to the group of robots with its own programming style and hardware.

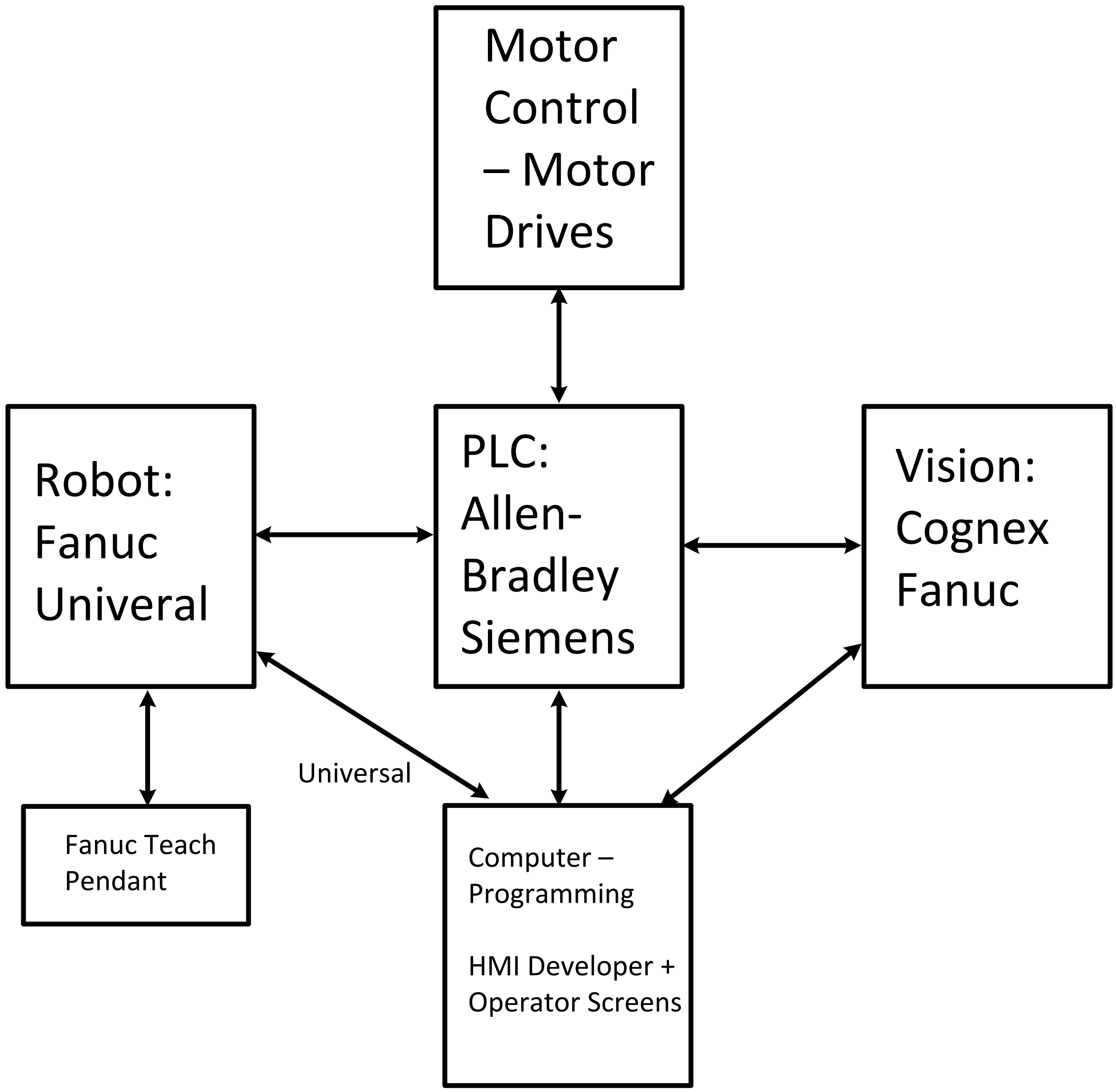
The new vision system – the Cognex 3805 – is a step toward AI with the promise of full AI down the road. It has the full features of the 7802 cameras previously purchased.

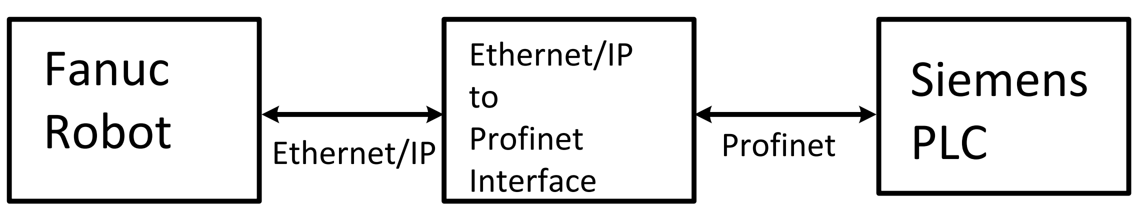
The following is a description of the purchases just made for the robotics/plc/vision course to truly take on the description of the capstone project above.

001	1.00	0.00	1.00	EA	1.0	6ES72151AG404AB1 TRAINING SIX-PACK CPU 1215C DC/DC/DC INCLUDES 6X CPU 1215C DC/DC/DC & 6X DVD STEP 7 BASIC SW & 6X PN CBL LEAD TIME: 28-30 WEEKS ARO	EA	1.0	2,020.0000	2,020.00
002	6.00	0.00	6.00	EA	1.0	6EP13321LB00 SITOP PSU100L 24V/2.5A LEAD TIME: 3-4 WEEKS ARO	EA	1.0	93.8200	562.92
003	6.00	0.00	6.00	EA	1.0	6GK50050BA001AB2 SCALANCE XB005 UNMANAGED INDUSTRIAL ETHE EMI STOCK	EA	1.0	113.0000	678.00
004	6.00	0.00	6.00	EA	1.0	1722.122MASTER HILSCHER NT 151-RE-RE/+ML LOADABLE DEVICE FOR ANY SLAVE/MASTER OR SLAVE/SLAVE PROTOCOL COMBINATION LEAD TIME: 2-3 WEEKS ARO	EA	1.0	1,198.0000	7,188.00
005	10.00	0.00	10.00	EA	1.0	30-3060-3M 8020 300MM X 60MM T-SLOT EXTRUS 3M LEAD TIME: 2-3 WEEKS ARO	EA	1.0	95.2857	952.86

Item	Product Id	Description	Qty	Net Unit Price	Net Total (Extended)
1	IS3805MP-14621-LAB	In-Sight 3805 (5MP) Mono Performance LAB Model with 16mm High Speed Liquid Lens, Multi Torch Light, & Diffused Cover	3	6,360.00	19,080.00
2	CCB-PWRIO-05	Cognex Power I/O Cable, IP67, M12 to 12 Flying Leads, 5 meters (15 feet)	3	114.00	342.00
3	CCB-84901-2001-02	Cognex X-Coded Ethernet Cable, IP67, M12 to RJ45, 2 meters (6 feet)	15	93.00	1,395.00

Item#	*Qty*	*Description*	*Part #*	*Standard List Price*	*EDU Price*	*Extended*
1	1	**Universal Robot 5e**	110305	$36,515.00	$29,062.00	$29,062.00
		Includes: Integrated Force Torque Sensor				
		Payload: 11 Pounds				
		Reach: 33.5 Inches				
		6 Rotating Joints DOF				
		Robot Weight: 45.4 Pounds				
		I/O Ports: Digital In (2) Digital Out (2) Analog In (2)				
		Control Box I/O Ports: Digital In (16) Digital Out (16) Analog In (2) Analog Out (2)				
		Robotiq Grippers				
2	1	Hand-E Gripper	HND-ES-UR-KIT	$4,725.00	$4,181.00	$4,181.00
		ALL UR Package Includes:				
		Access to the UR Online Academy- FREE				
		Access to the UR EDU Resources - FREE				


Motor Control – Motor Drives
Robot: Fanuc Univeral
PLC: Allen-Bradley Siemens
Vision: Cognex Fanuc
Universal
Fanuc Teach Pendant
Computer – Programming
HMI Developer + Operator Screens


Fanuc Robot
Ethernet/IP
Ethernet/IP to Profinet Interface
Profinet
Siemens PLC

Collaborative White Fanuc Robot

- ☐ Internet
- ☐ Ethernet I/P
- ☐ Vision Only

Three Yellow Fanuc Robots

- ☐ Internet
- ☐ Ethernet I/P

S7-1200 Communication to Fanuc Robot with EIP Interface and Starting the Robot Via the PLC - Michael Smith - MIME5450

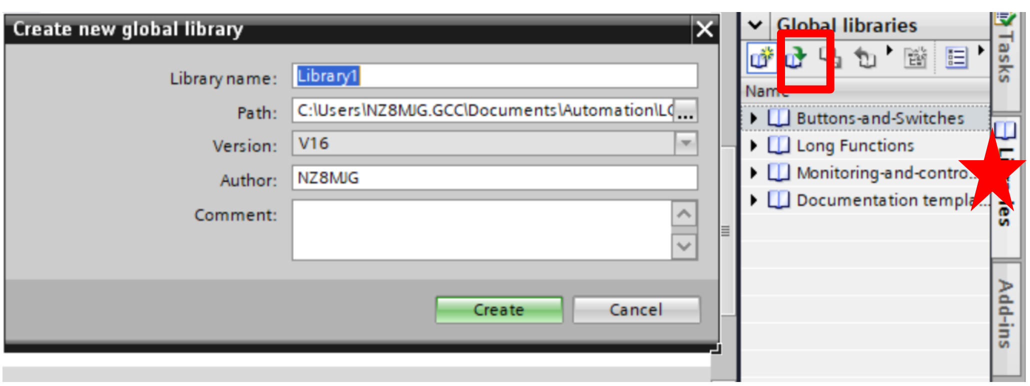
Objective
The objective of this lab is to establish communication between a Siemens S7-1215 PLC that has a ProfiNet interface to a Fanuc Robot R30-ib controller that has an Ethernet IP interface and to add logic to start the robot.

Method
The method I used to accomplish this task was to use the LCCF_EnetScanner which allows the S7-1215 to function as an ethernet IP scanner rather than using a physical device to link the PLC's ProfiNet interface to the robot controllers EIP interface. The upside of this method is that no physical device is required and the configuration is not to difficult. The downside is that this block uses a substantial amount of the PLC's memory.

LCCF Enet Scanner Library
The LCCF_EnetScanner block is not standard in TIA Portal. A library containing the block and UDTs will need to be downloaded from Siemens at the below website and installed as a library in TIA Portal. An example program already containing the library can also be downloaded from this web address.

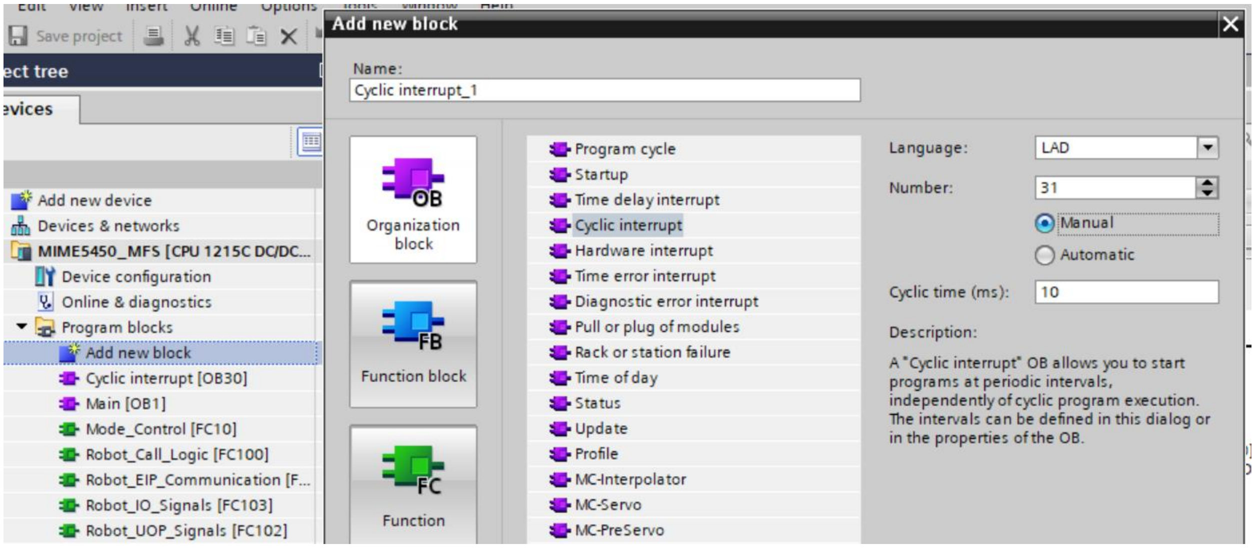
https://support.industry.siemens.com/cs/document/109782314/ethernet-ip-scanner-%E2%80%93-connecting-third-party-i-o%E2%80%99s-using-ethernet-ip?dti=0&lc=en-WW

A library can be installed in a TIA Portal by clicking on the 'Libraries' tab on the right hand side of the screen and then pressing the 'Create New Library' icon. A pop up box will be displayed were the downloaded library previously mentioned can be selected and installed.

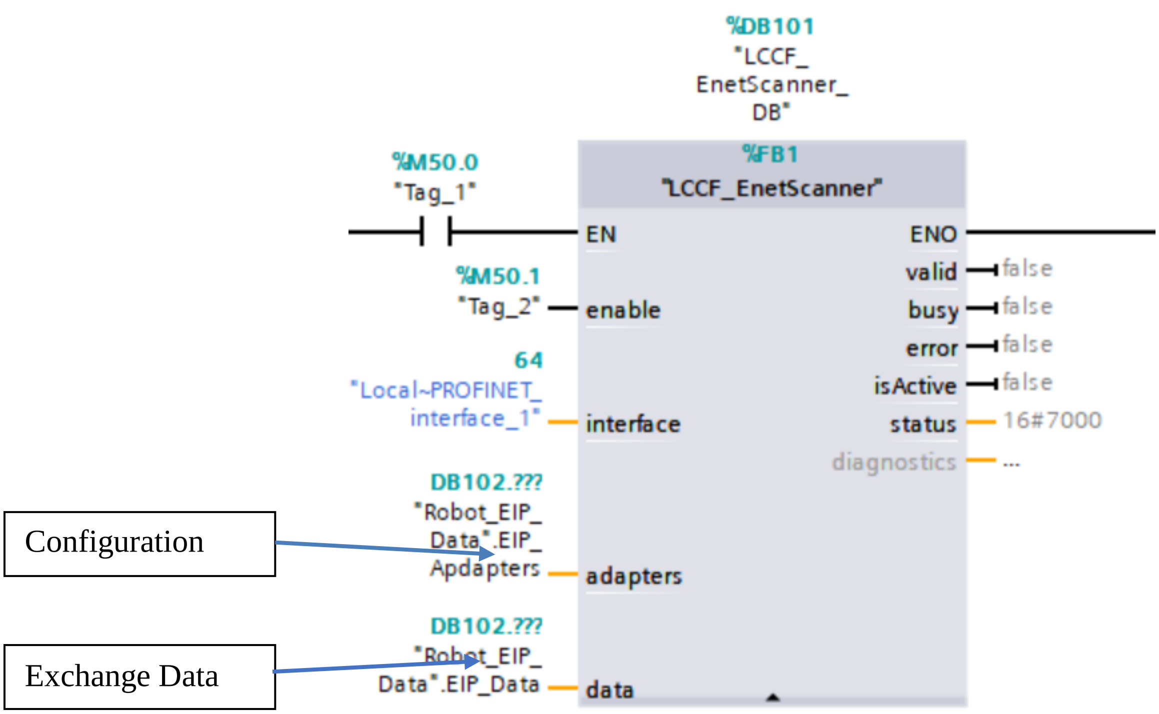


LCCF_EnetScanner Programming and Configuration

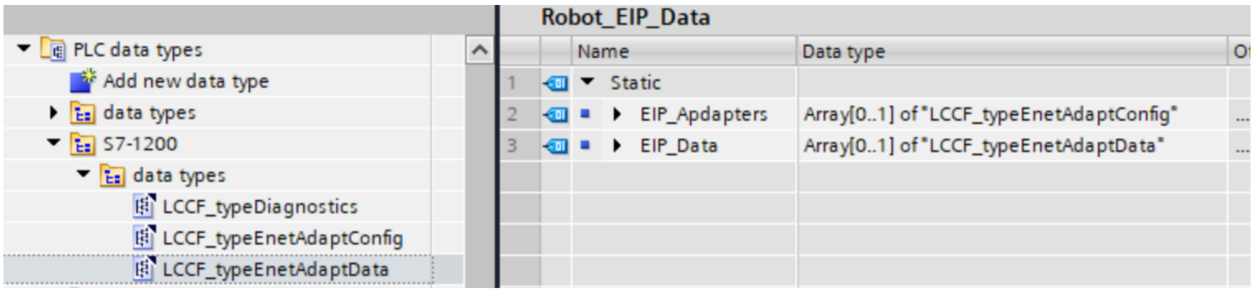
I chose to program the LCCF_EnetScanner block in a cyclic interrupt so the first thing I did was to create a cyclic interrupt block where the call for the LCCF_EnetScanner block will take place.



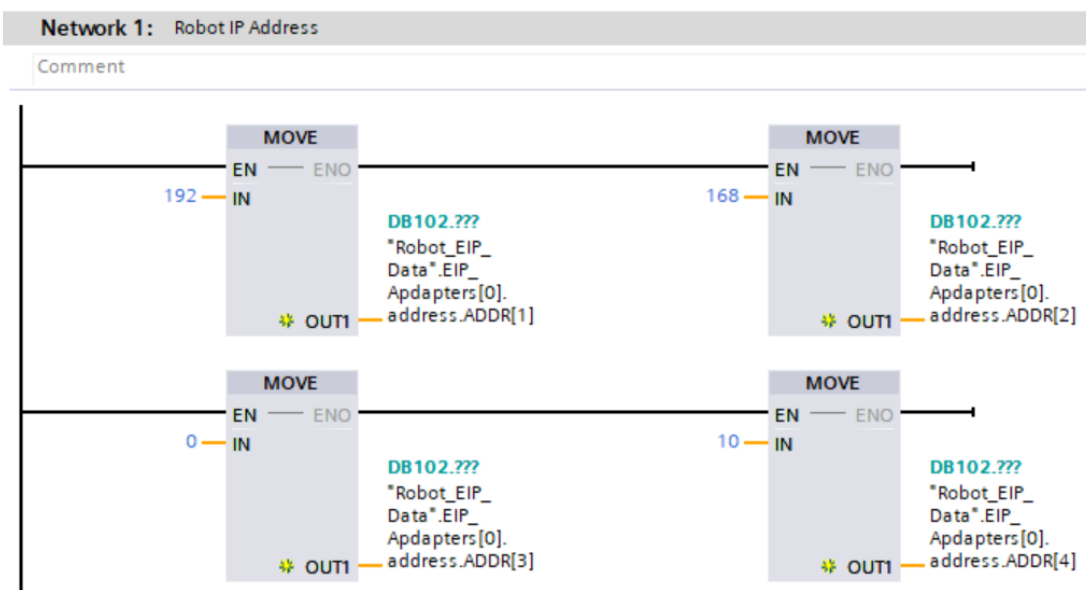
Another task that needed to be completed before any programming could be performed was to create a data block that will contain the configuration data and data that will be exchanged between the PLC and robot.



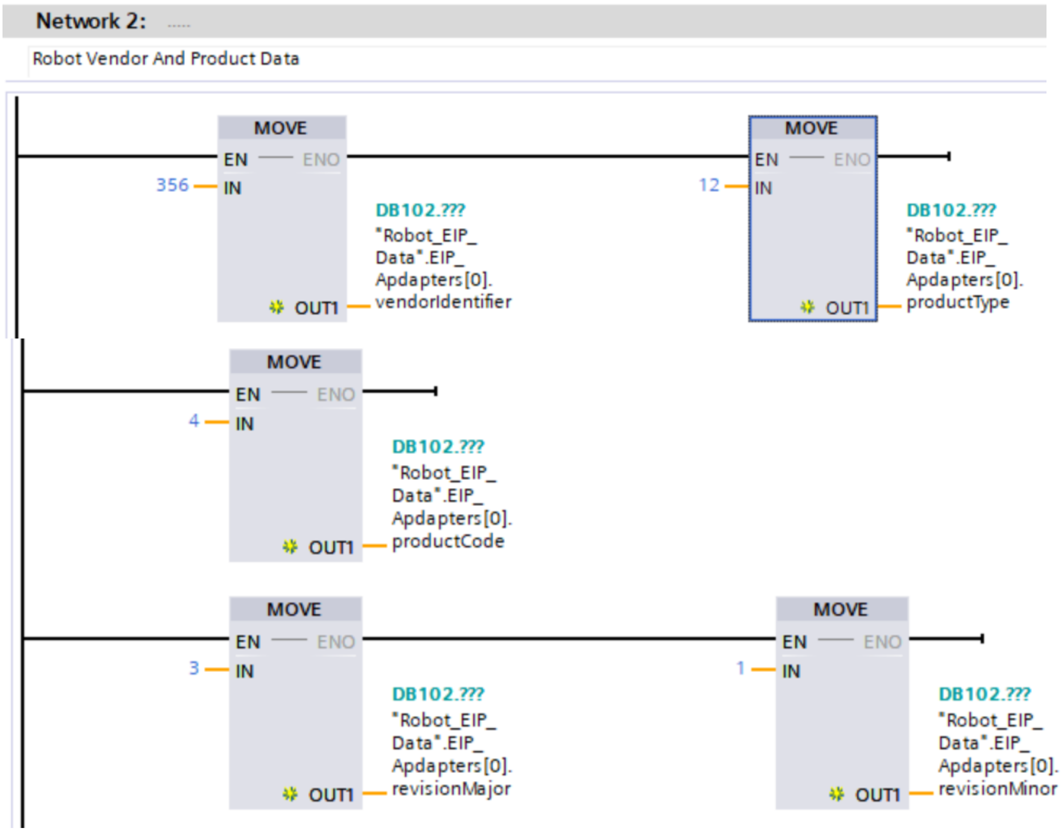
When the before mentioned library was created not only was the scanner function block installed but several UDTs or data types were also installed. The data block will need to use these data types. An array of the data type LCCF_typeEnetAdaptConfig and an array of data type LCCF_typeEnetAdaptData will need to be contained within the data block. For this lab the array only needed to contain 2 elements. I am not sure why the array is necessary when only one instance of the scanner is used but it is. For my labe I used 102 for the data block number and named the block 'Robot_EIP_Data' See the image below for the structure of the data block used in my lab.



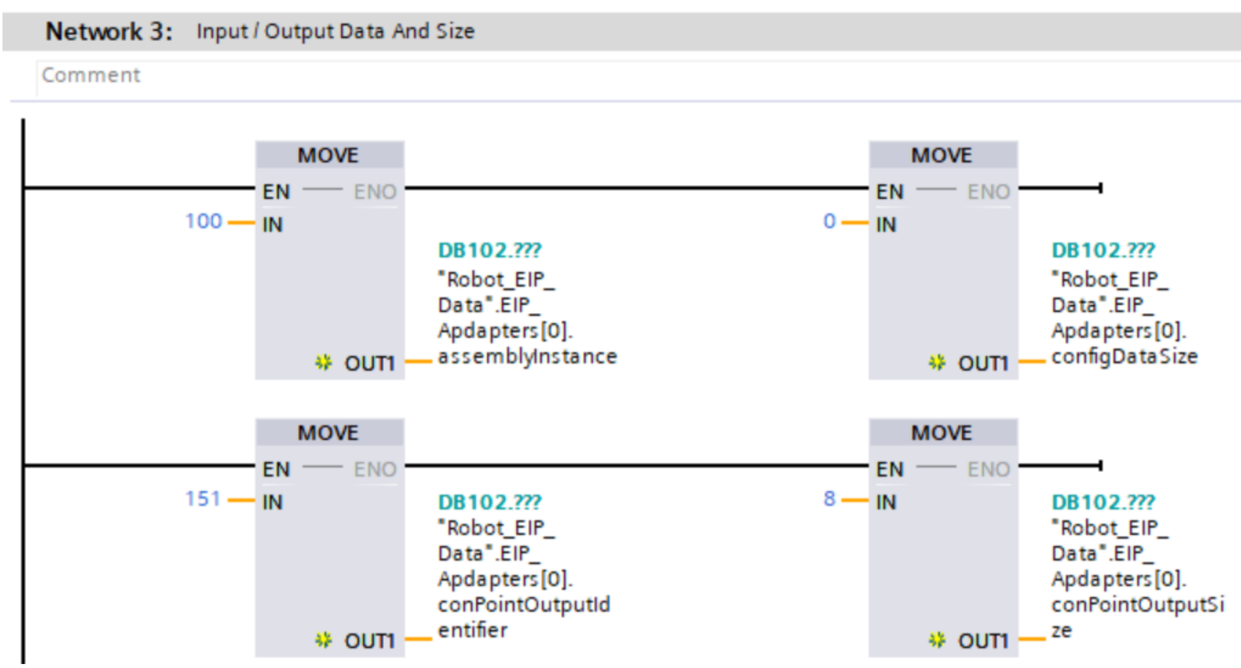
The programming in the cyclic interrupt where the LCCF_EnetScanner will be called will require additional programming that contains configuration parameters for the scanner connection. Below is the required programming to set the parameters for the scanner. In the first network I set the IP address of the robot controller as 192.168.0.10. Notice that the IP address octet values are being moved into the data block previously created.

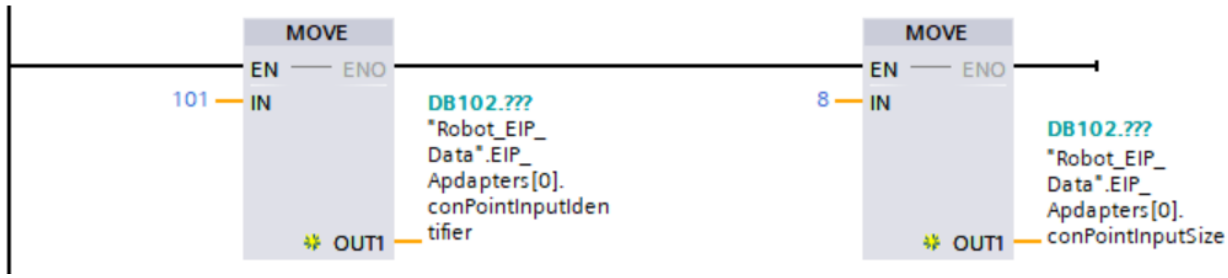


In network 2 more parameters are set pertaining specifically to the Fanuc robot such as the vendor ID and product data. Note that the LCCF_EnetScanner block can be used to connect to any ethernet IP device not just the Fanuc robot controller. The values that the following parameters are set to can found in the corresponding devices EDS file. The values shown in the image below were found in the EDS file for the Fanuc R30-ib robot controller.

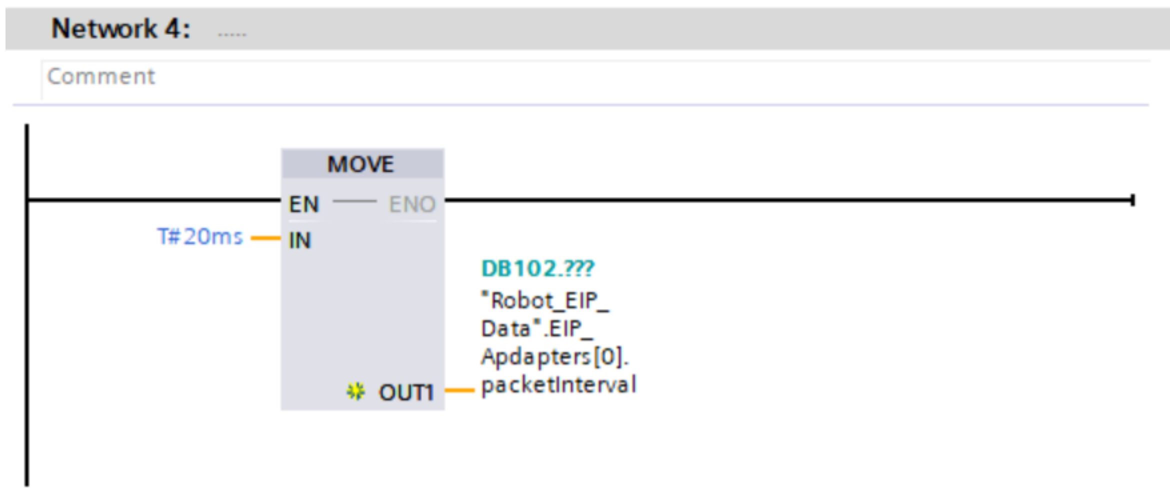


Network 3 contains parameters values pertaining to the data exchange between the PLC and robot controller. Notice in the image below that the size of the data exchange is 8 bytes for both the inputs and outputs.

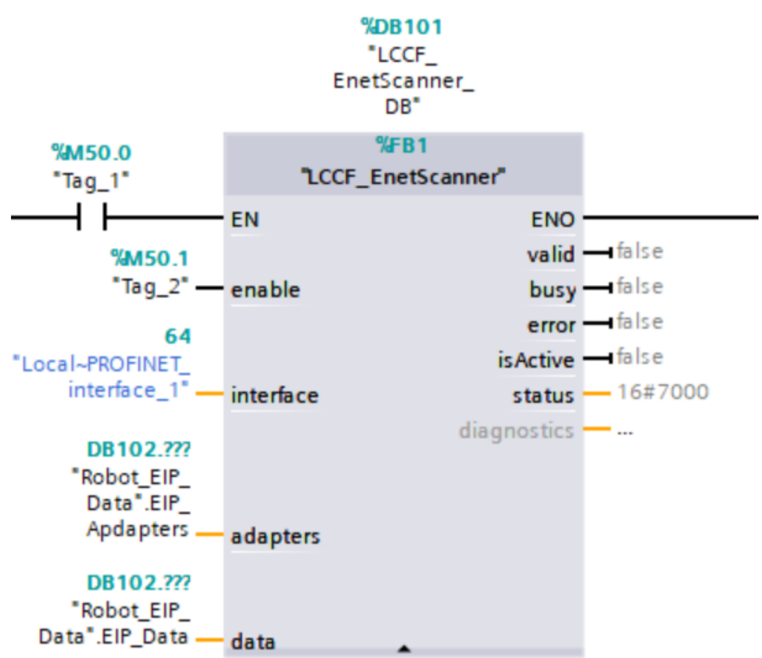




In network 4 the parameter for the rate at which the data is to be exchanged is set. Notice in the image it is set to 20ms.

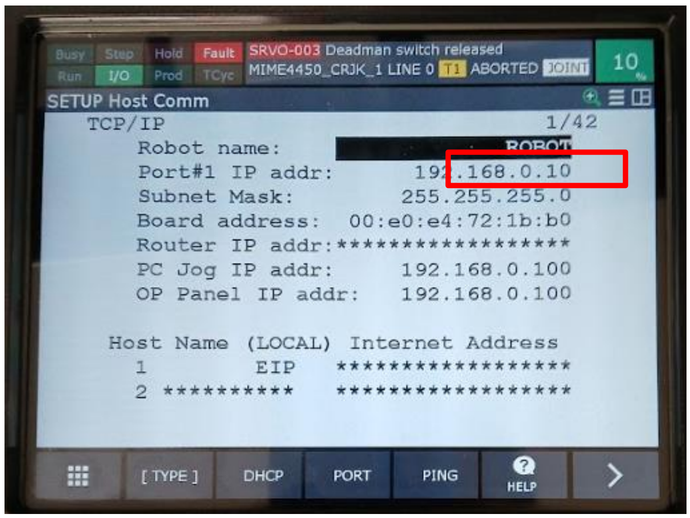


Finally in network 5 the LCCF_EnetScanner block is called. If communication is succesfully established the value at the status output will be a hex value of 7000.



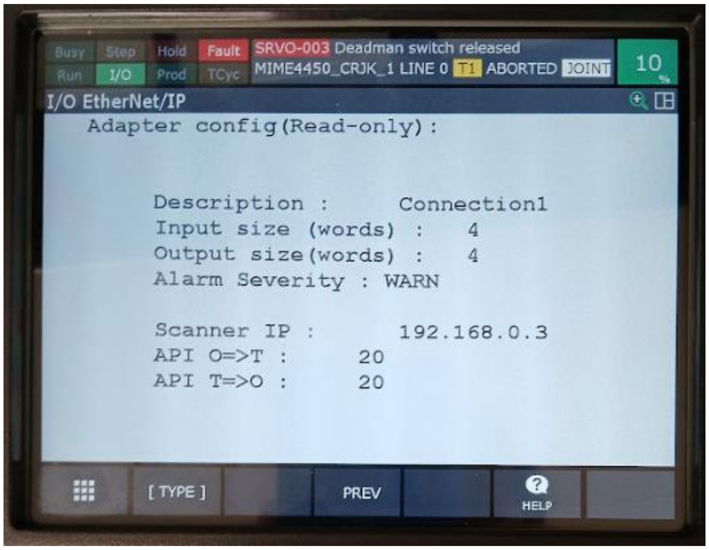
Robot Setup

The robot controller used in this lab needed several parameters changed to establish communication with the PLC. The first step was to set the IP address of the robot. This can be done from the teach pendant by selecting menu / setup / host com. Notice I set the IP address to 192.168.0.10.



Next, I enabled Ethernet IP communications from the IO screen. I enabled connection 1.

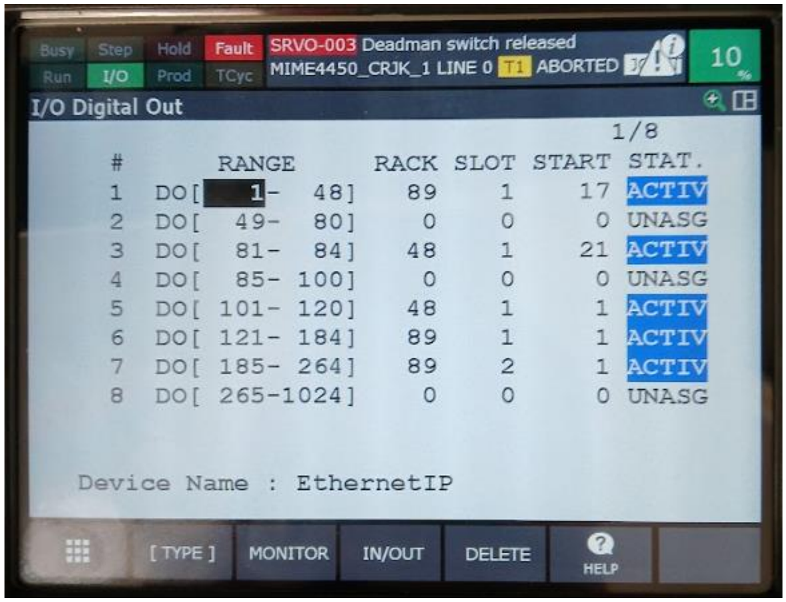
From the screen shown in the image above I selected 'config' to display the screen shown in the image. From this screen the amount of data in words that is exchanged between the PLC and robot controller can be adjusted. However, for connection 1 it appears that it is set at 4 words for both the input and output data.



Also, from the screen above if the PLC and robot controller have successfully established a connection then the IP address of the PLC will be displayed in the Scanner IP value. For my lab I set to PLC IP address to 192.168.0.3 and you can see that is visible on the screen.

Next I configured a range of inputs and outputs to signal to the PLC. In the lab I configure the input and output group 1 for rack 89, slot 1 starting at bit 17. Rack 89 is a

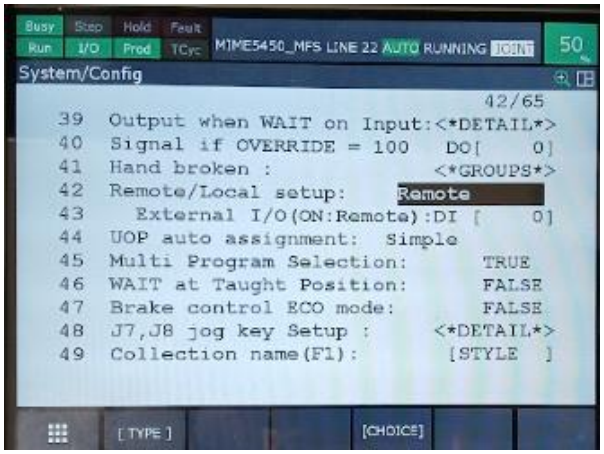
Fanuc defined code for an Ethernet IP connection and I set the start bit to 17 because the user inputs and outputs will be assigned to bits 1 through 16.



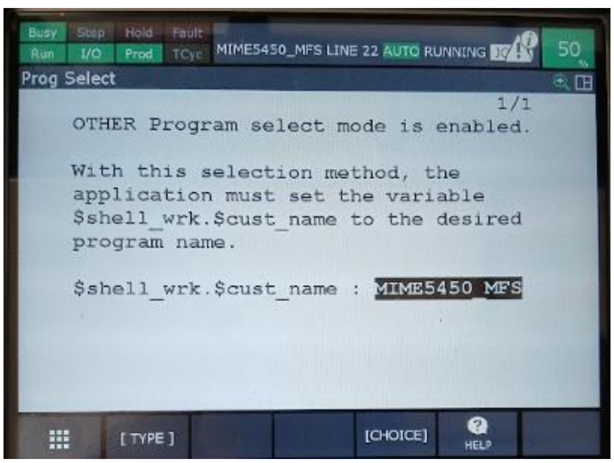
I configured the digital inputs with the same settings as the outputs. The user inputs and outputs I configured as a range of 1 -16 a rack value of 89 a slot value of 1 and a start value of 1. With these settings the first 16 bits from the PLC sent to the robot will be for user inputs and the next 48 bits will be for general digital inputs. Below is an image of the user output configuration.

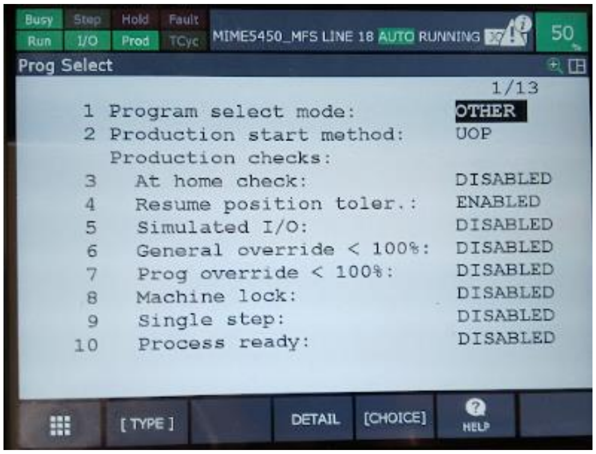
The next step was to modify a few settings from the robot pendant that will allow the robot to be started remotely via the user input signals from the PLC. The first few settings can be accessed from the system configuration screen which can be accessed on the teach pendant by pressing the Menu key then System and then Configuration.

From this screen setting 7, Enable UI Signals, needs to be set to True and setting 42 needs to be set to remote. Setting 44 I set to Simple but I am not sure if this is required.

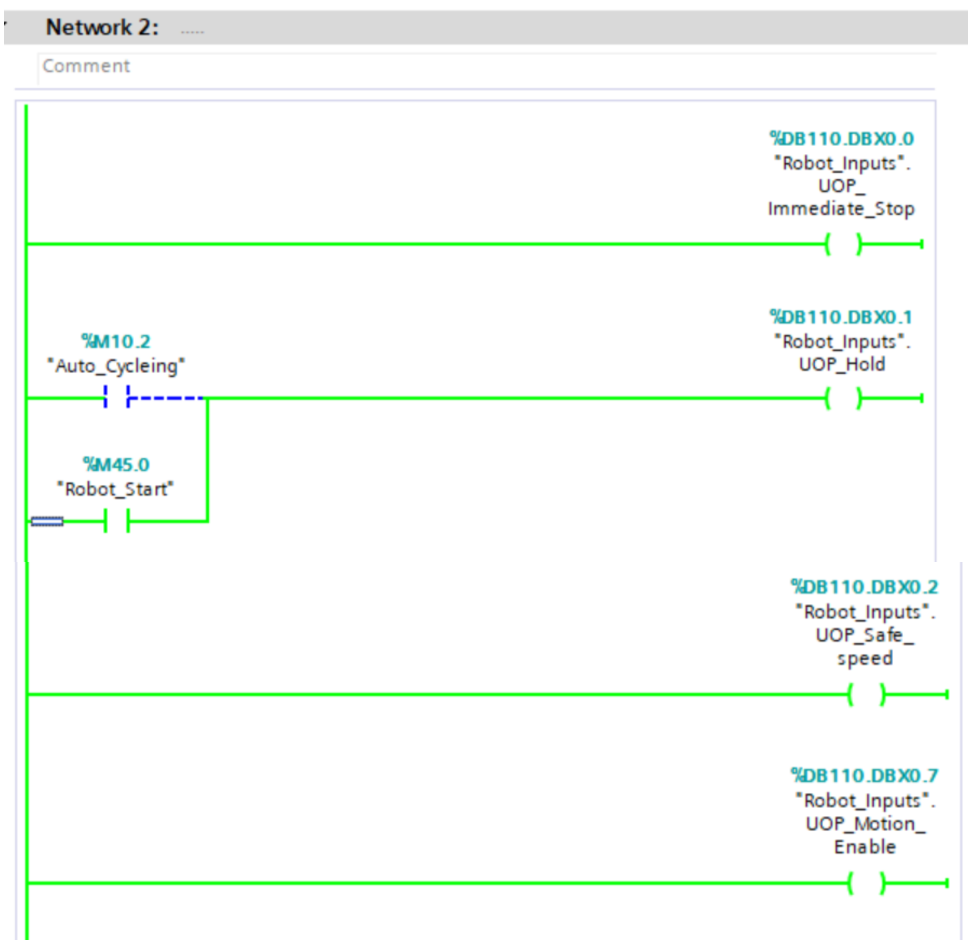


Next, I had to configure the method by which the robot is started. This can be accessed from the Program Select screen on the Teach Pendant by pressing the Menu key and then Setup followed by Program Select. From this screen set the Program Select Mode to Other and the Production Start Method to UOP. With 'OTHER' highlighted press Detail on the pendant. This will display another screen allowing for a program to be selected to run when the robot is started.

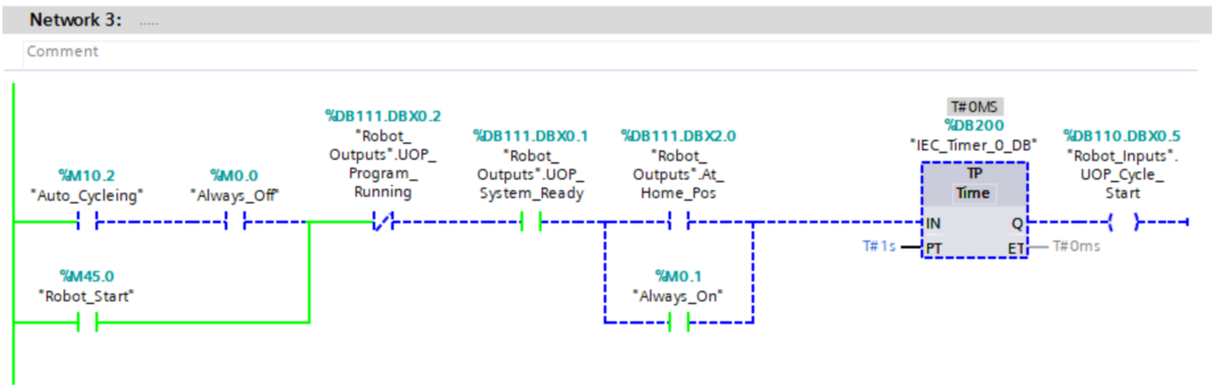




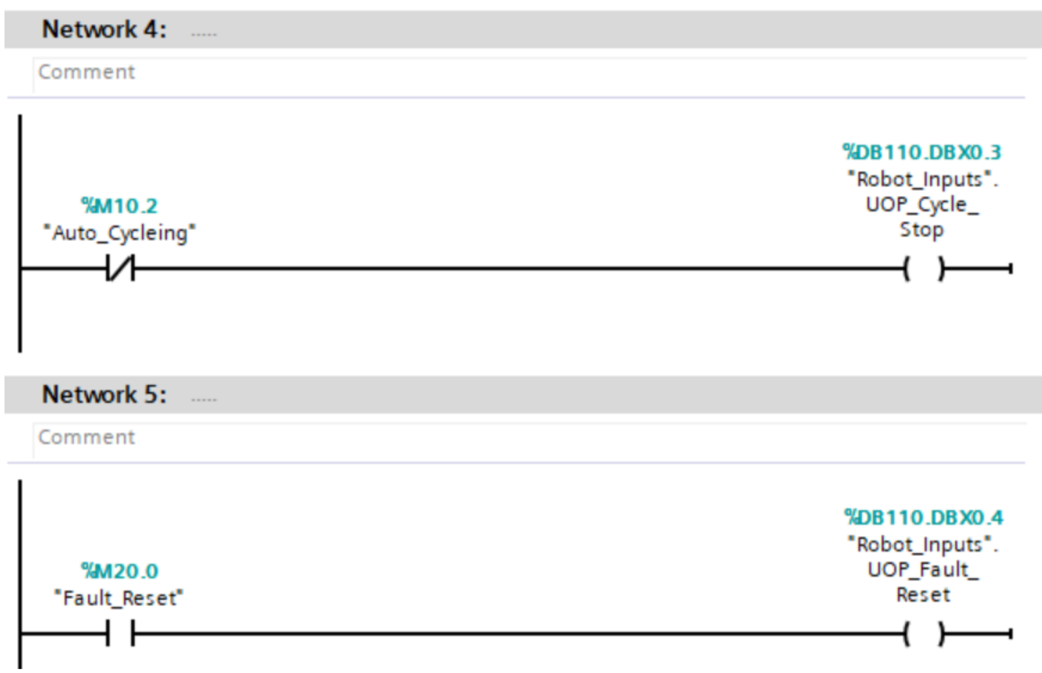
The next step was to add logic to the PLC program to start the robot via the user inputs. In order to start the robot user inputs 1, 2, 3 and 8 need to be on and user input 5 needs to be pulsed on for at least 10ms. I created a separate function call and data block in the PLC to contain this logic. I used



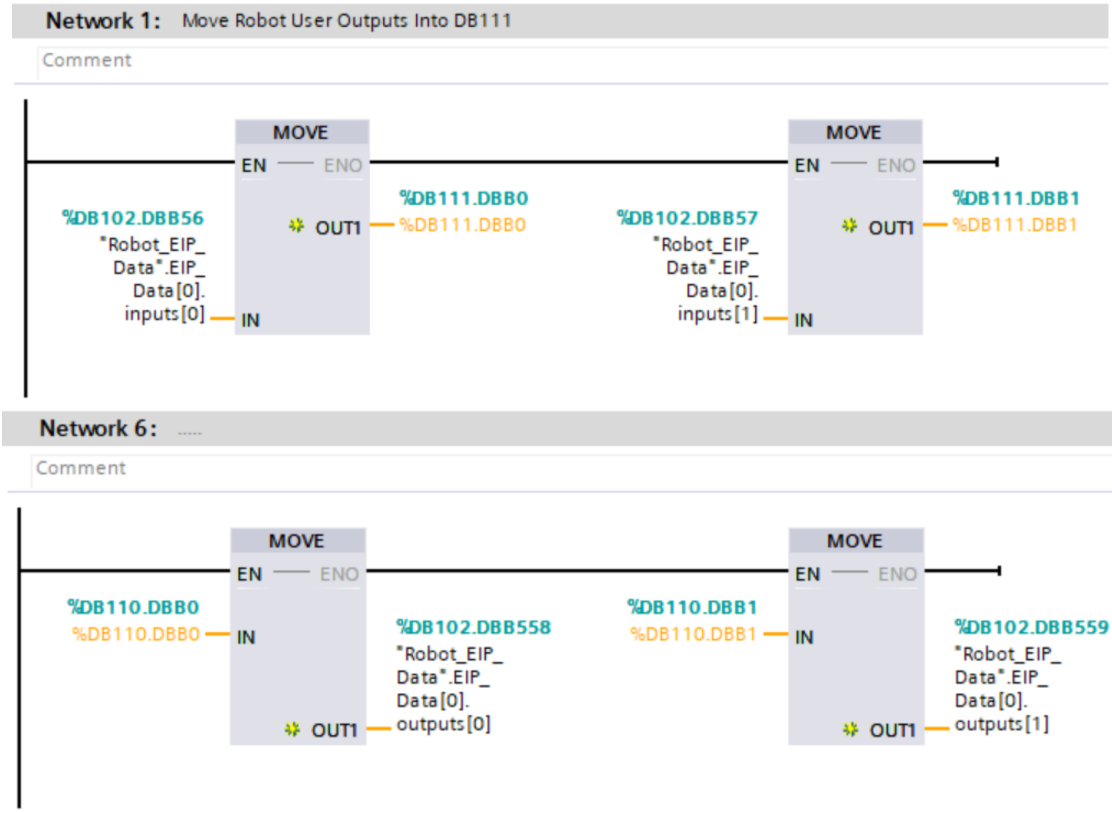
In an actual manufacturing environment the logic would not be this simple. Below is the logic I created for user input 5 which is the cycle start signal. Note that this signal can not remain on. The robot will start after this signal goes low. Notice I am using a pulse timer to keep the signal on for only a second.



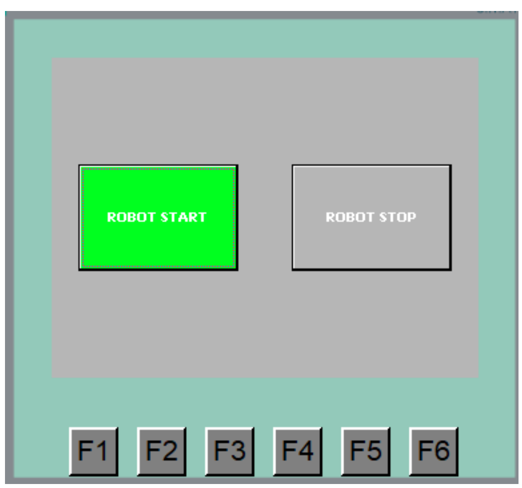
In the following networks I added logic for the cycle stop and fault reset user inputs.



In order to send this information to the robot we need to move the first 2 bytes from DB110 into the LCCF_EIPScanner block as shown in the image below.



Finally I created a simple HMI that start and stops the robot by turning on and off bit M45.0.



Rockwell to Fanuc Robot

FANUC ROBOTS INTEGRATION USING ETHERNET / IP

By AutomationTop Team Posted on September 4, 2020

The choice of communication protocol between the PLC controller and the peripheral device most often depends on the main device, i.e. usually the PLC controller. Of course, it is not always possible, then various types of converters come into play, but this is a separate topic.

In other words, we choose what we know and like, but it is not always the most optimal choice. Occasionally, certain devices offer slightly more functionality with certain network protocols. In this article, I would like to present an example of communication configuration via EtherNet / IP between a PLC and a FANUC robot. In the first part, I will present the method of establishing communication step by step. In the second part, I would like to discuss some non-obvious features related to the EtherNet / IP option in FANUC robots that give you an advantage over other methods.

STEP1 – I / O MAP

Before starting the robot-PLC communication setup, it is worth preparing a list of signals that we want to send between devices. FANUC has some predefined groups of signals that are worth, and some should be included in such a list.

The most important of these groups is UOP – User Operator Panel, here are the signals necessary for the cooperation of the robot with the PLC. With their help, we can run the robot program using the PLC controller. Cell Interface and SOP (Standard Operator Panel) groups contain a set of useful signals for integrators. Similarly, Interconnect where we can, for example, "pass" the status of internal safety signals to the controller (For monitoring purposes only).

As a rule, for a given process, the most important groups will be 3 – Digital (digital inputs / outputs) and 5 – Group (digital inputs / outputs grouped in packages). It is here that we assign signals controlling the course of a given process (process variables).

An example signal map may look like this:

	A	B	C	D	E	F	G	H	I
1	WEJŚCIA				WYJŚCIA				
2	PLC		FANUC	Komentarz	PLC		FANUC	Komentarz	BIT
3	Address	Zmienna	DI		Address	Zmienna	DO		
4	0.0		UI1	IMSTP	0.0		UO1	Cmd Enabled	1
5	0.1		UI2	HOLD	0.1		UO2	System ready	2
6	0.2		UI3	SFSPD	0.2		UO3	Prg running	3
7	0.3		UI4	Cycle Stop	0.3		UO4	Prg paused	4
8	0.4		UI5	Fault reset	0.4		UO5	Motion held	5
9	0.5		UI6	Start	0.5		UO6	Fault	6
10	0.6		UI7	Home	0.6		UO7	At perch	7
11	0.7		UI8	Enable	0.7		UO8	TP enabled	8
12	1.0		UI17	PNS strobe	1.0		UO9	Batt alarm	9
13	1.1		UI18	Prod start	1.1		UO10	Busy	10
14	1.2				1.2		UO19	SNACK	11
15	1.3				1.3				12
16	1.4				1.4				13
17	1.5				1.5				14
18	1.6				1.6				15
19	1.7				1.7				16
20	2.0		UI9	RSR1/PNS1/STYLE1	2.0		UO11	ACK1/SNO1	17
21	2.1		UI10	RSR2/PNS2/STYLE2	2.1		UO12	ACK2/SNO2	18
22	2.2		UI11	RSR3/PNS3/STYLE3	2.2		UO13	ACK3SNO3	19
23	2.3		UI12	RSR4/PNS4/STYLE4	2.3		UO14	ACK4/SNO4	20
24	2.4		UI13	RSR5/PNS5/STYLE5	2.4		UO15	ACK5/SNO5	21
25	2.5		UI14	RSR6/PNS6/STYLE6	2.5		UO16	ACK6/SNO6	22
26	2.6		UI15	RSR7/PNS7/STYLE7	2.6		UO17	ACK7/SNO7	23
27	2.7		UI16	RSR8/PNS8/STYLE8	2.7		UO18	ACK8/SNO8	24
28	3.0		DI1		3.0		DO1		25
29	3.1		DI2		3.1		DO2		26
30	3.2		DI3		3.2		DO3		27
31	3.3		DI4		3.3		DO4		28
32	3.4		DI5		3.4		DO5		29
33	3.5		DI6		3.5		DO6		30
34	3.6		DI7		3.6		DO7		31

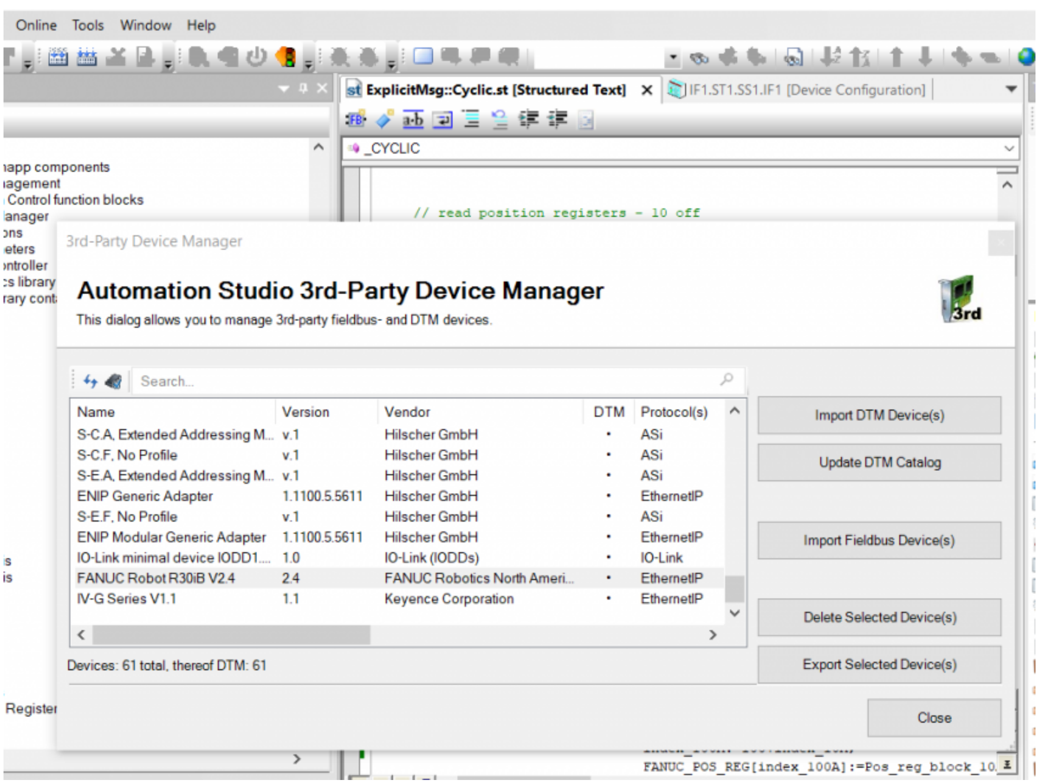
As you can see, at the beginning I put the UOP signals which are 3 bytes long. Then I put digital inputs and outputs. As a rule, between them I also put I / O used as standard in each application (cell interface), signals monitoring the robot status, etc. For demonstration purposes, however, it is not necessary, so I present here a simplified version.

From the PLC side, the number of inputs / outputs can be set practically freely, we are limited only by the size of the Ethernet / IP frame (about 1400 bytes) and the number of adapters (slaves) working with a given scanner (master). The frame is divided into all slave devices, so if we have more devices on ENIP, this should be taken into account when planning. By default, a certain number of I / O, registers and positional registers are set in the robot. However, there is nothing to prevent these values from being changed as needed. To do this, enter the MAINTENANCE mode using the Controlled Start function.

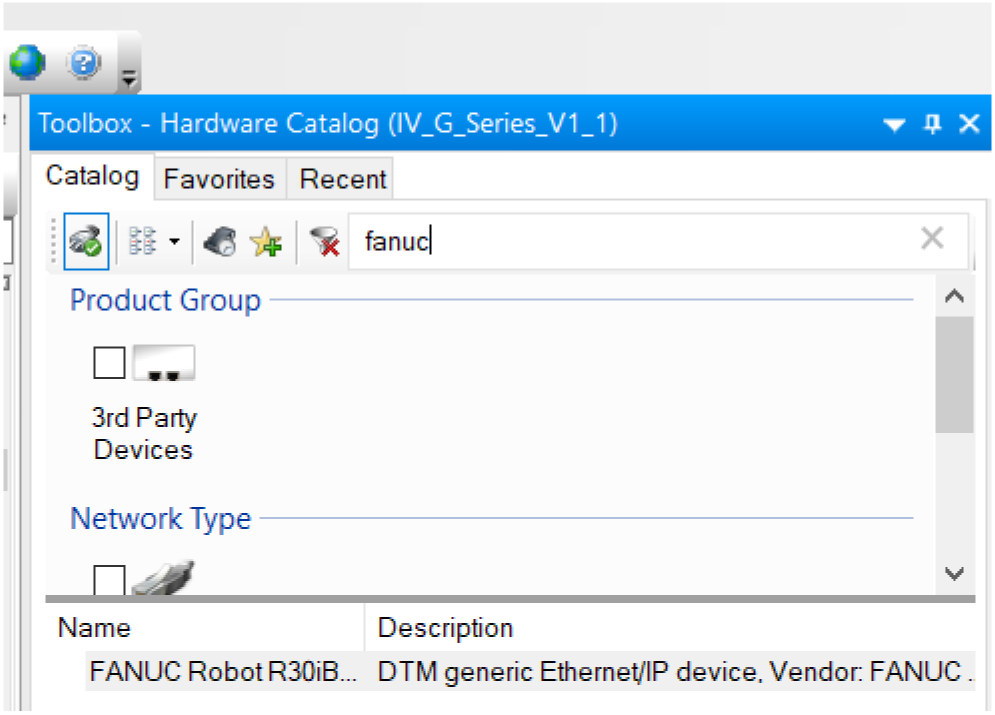
STEP2 – Connection

From the controller side, the network connection settings obviously depend on the type used. There is an AOP (add-on-profile) ready for AB controllers. I, on the other hand, use B&R controllers which, despite the need to buy an EtherNet / IP communication module, also allow very easy integration.

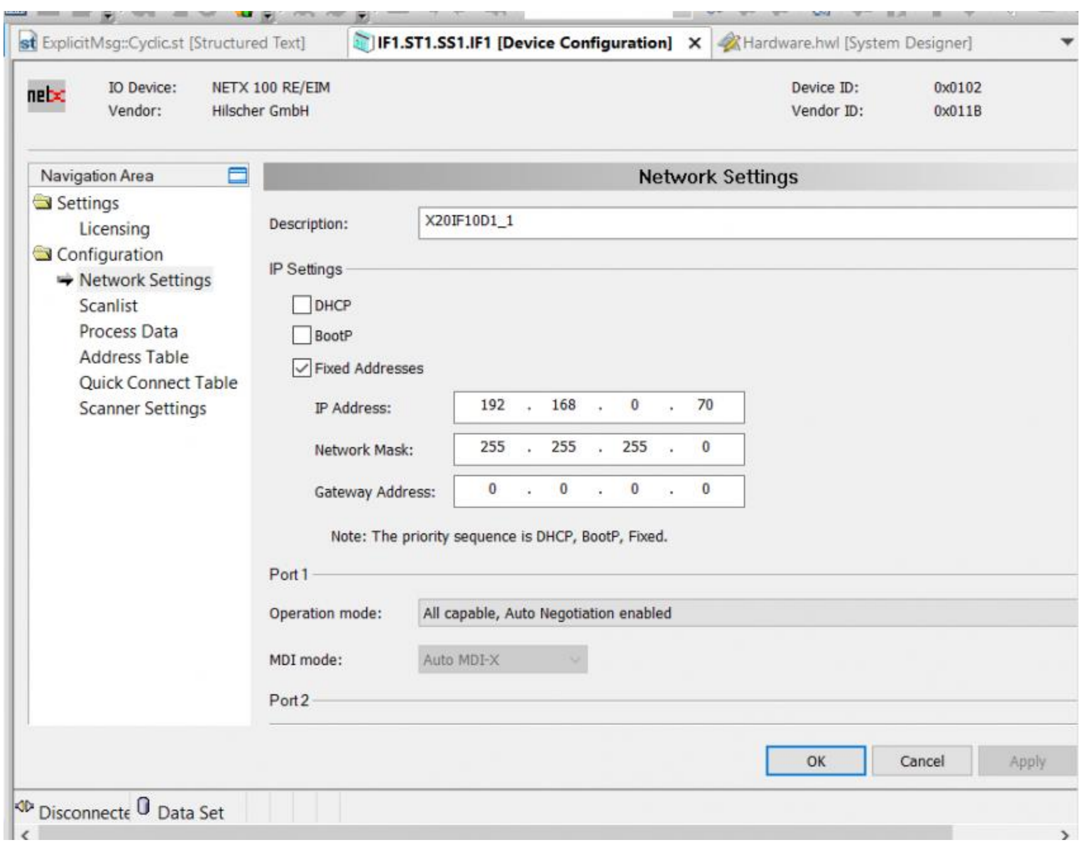
We add the Fanuc robot EDS file to Automation Studio:



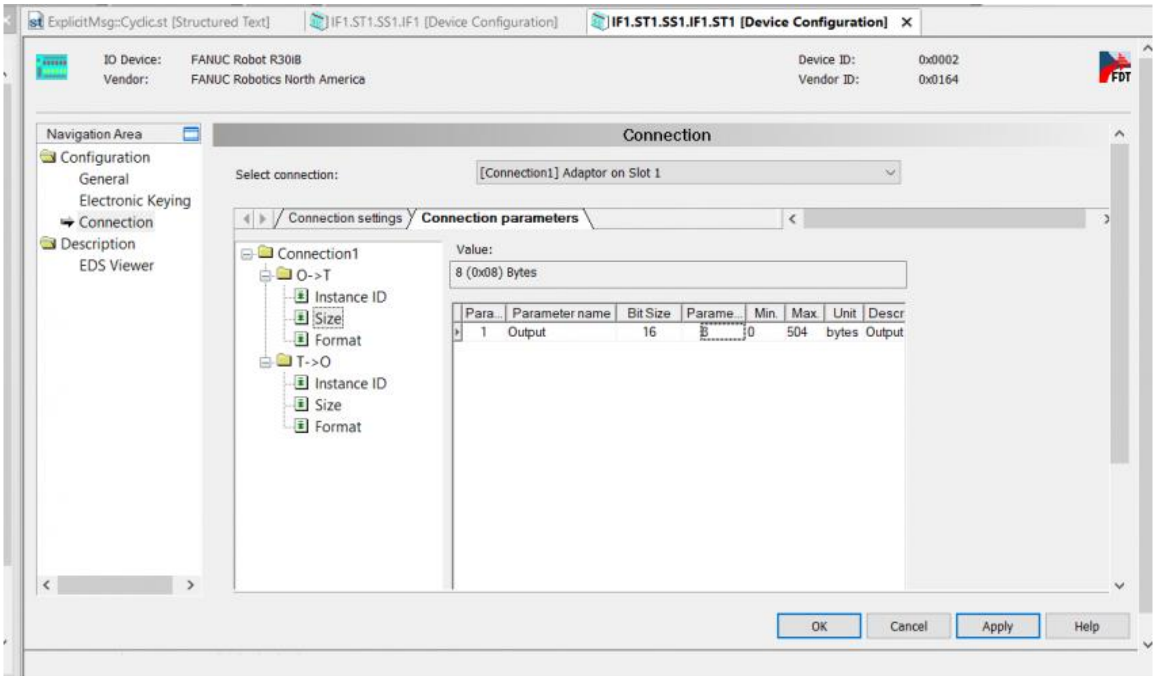
After adding the EDS file, add the robot to the device list:



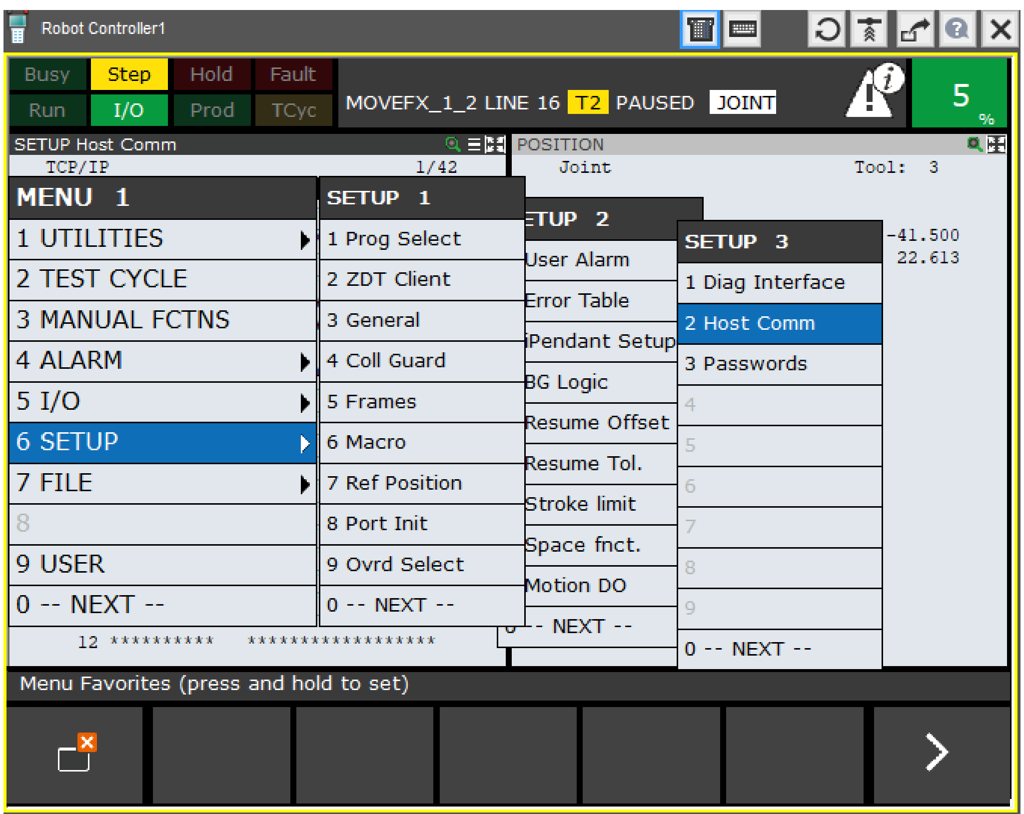
We set the scanner's IP address (master):



Frame size – in my case 64 bits in both directions:



From the robot's side, we go to the network settings on the teach pendant, where we set the robot's IP address:

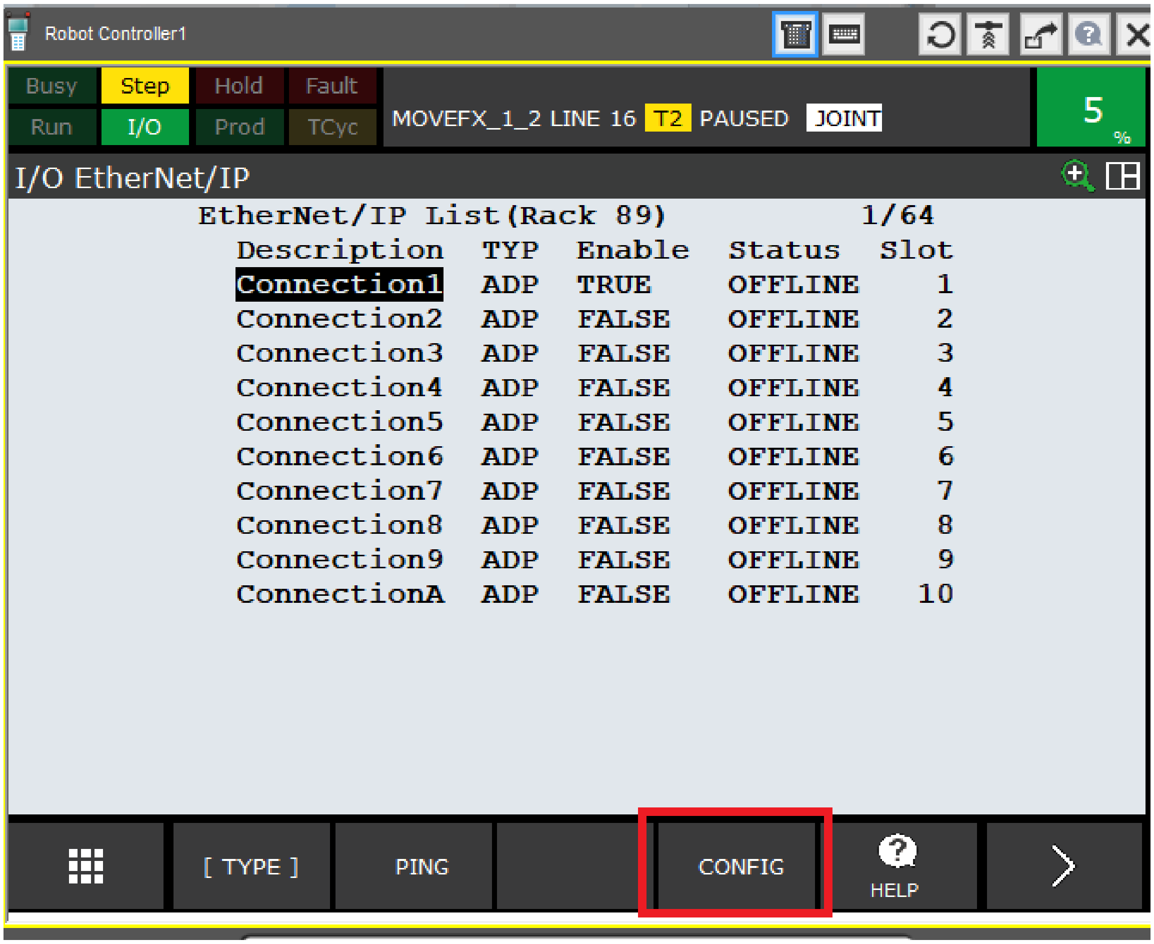


SETUP> SETUP2> Host Comm> TCP / IP

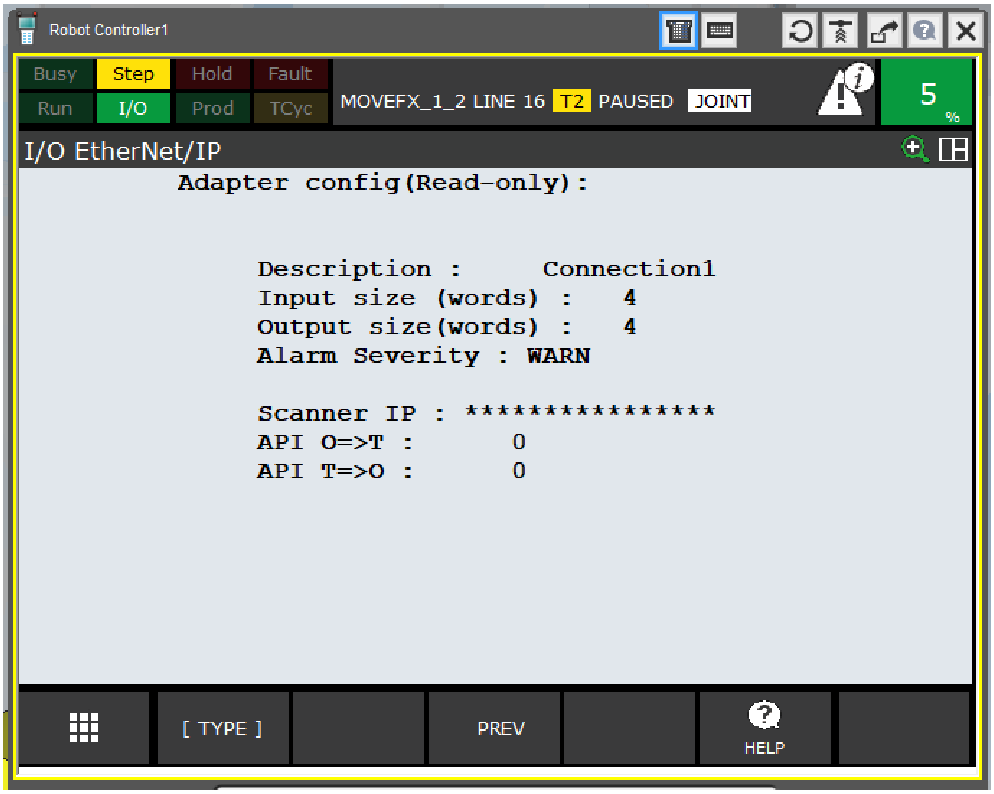
We ping and check if both devices are visible in the network.

Then go to the Ethernet IP settings. MENU> I / O>> Ethernet / IP

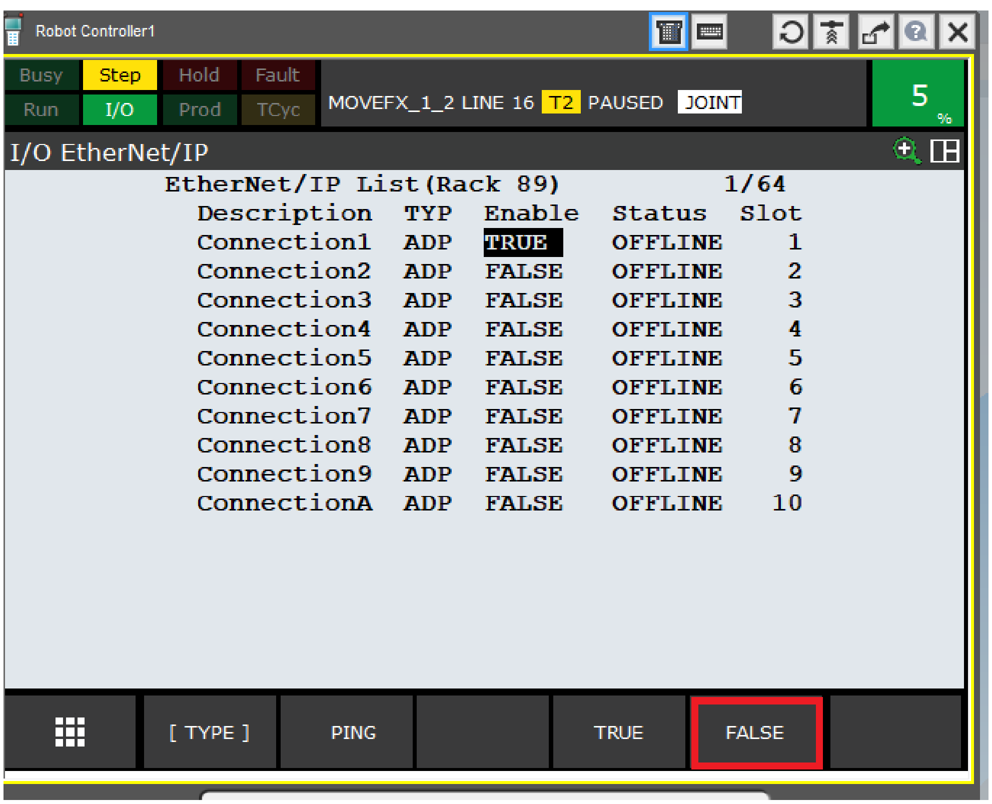
We choose the available connection that we want to use (we can have more than one ENIP adapter in the robot controller), then select F4 – Config



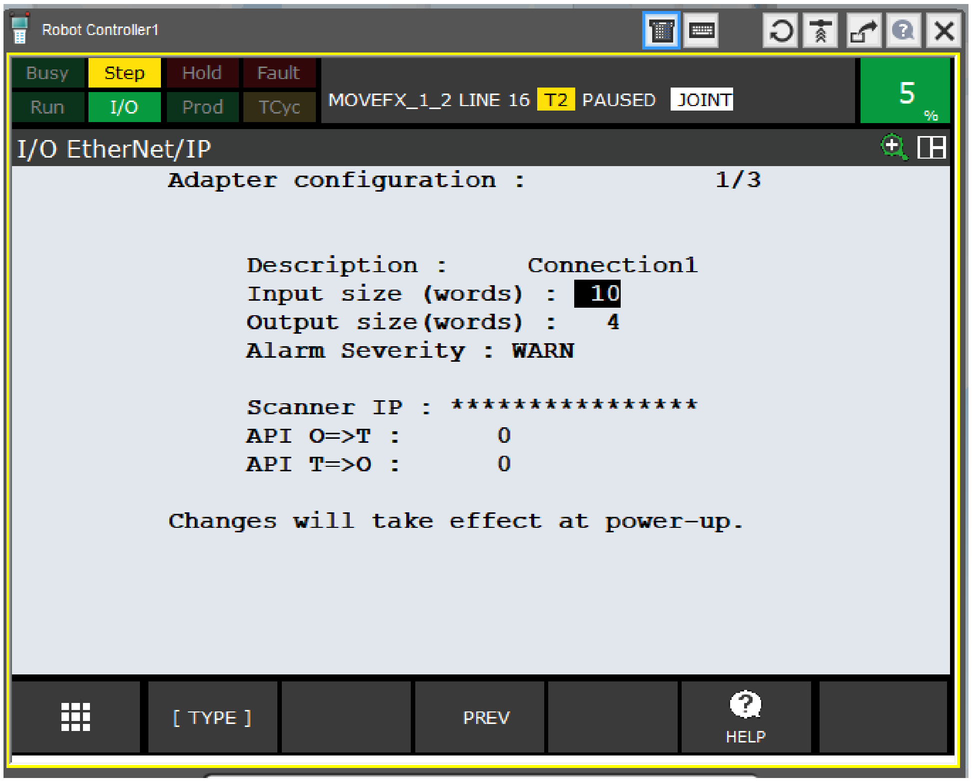
In the configuration window, we set the amount of data transferred, remember that the values are given as words, i.e. groups of 16 bits. So 4 words / 8 bytes / 64 bits, that is as set in the PLC.



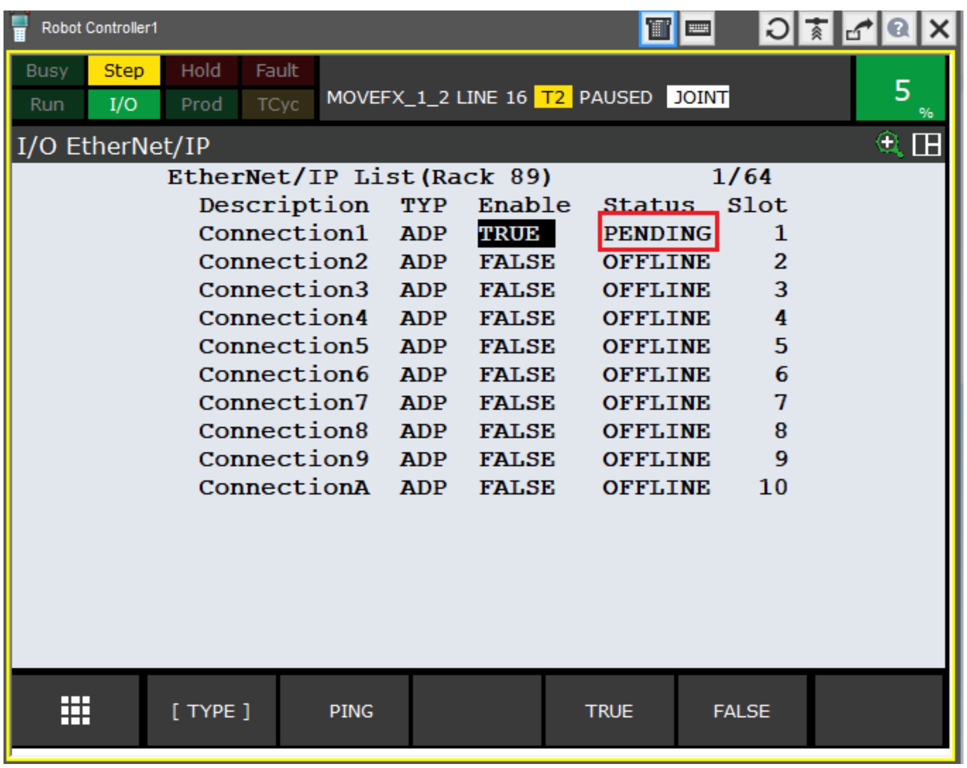
If we want to change the number of words, first we have to deactivate the connection on the previous screen.



Change the configuration:



Activate and restart the robot controller.



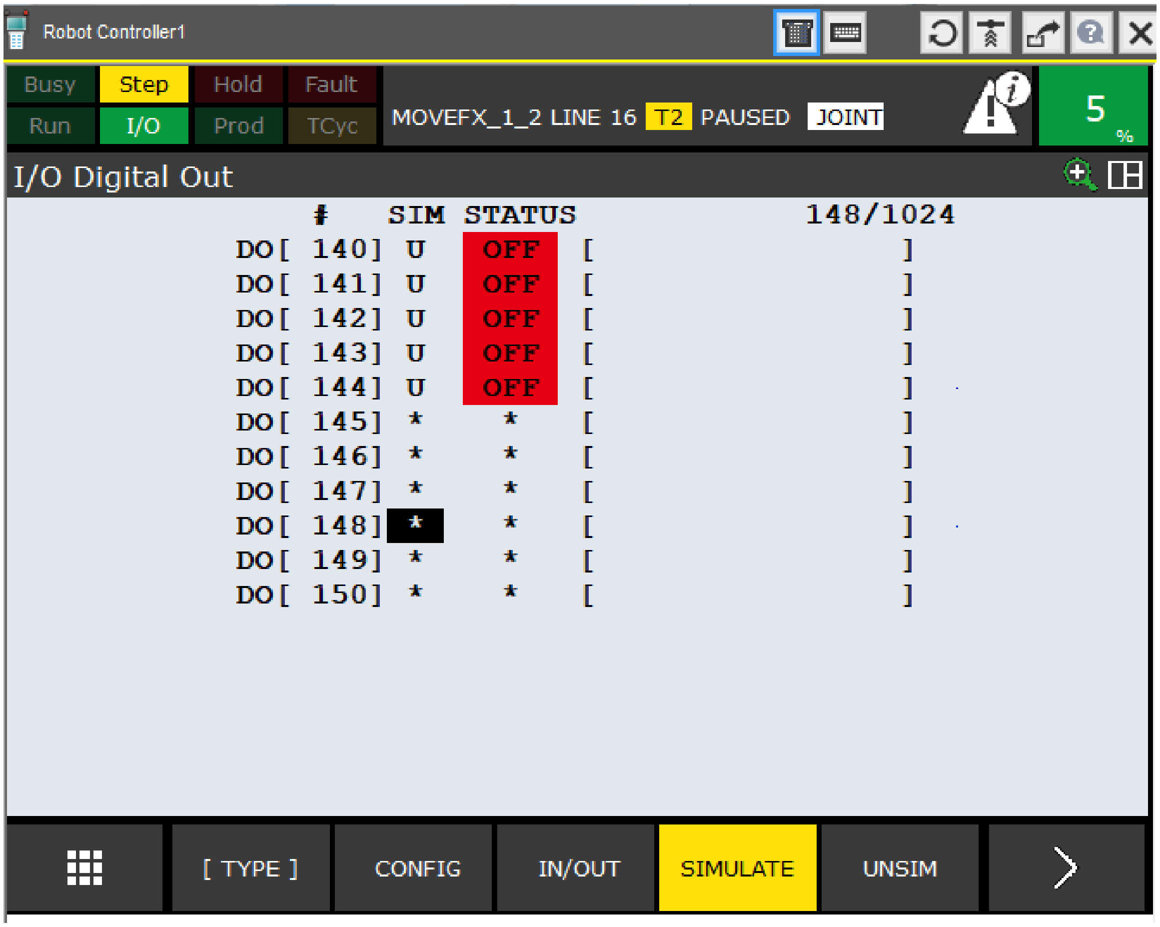
If everything is set correctly on the PLC side, after restarting the controller, we should see the status change OFFLINE -> RUNNING. I was using ROBOGUIDE offline when writing the article, so the connection is inactive.

Ok, this is how we established the connection between the controller and the robot. Unfortunately, this does not yet lead to an exchange of process data.

STEP3 – I / O configuration

To do this, go to MENU> I / O> Digital

After clicking ENTER, you should see a screen similar to the following:



Some inputs may already be assigned, while uninitialized (inactive) inputs / outputs will be marked with an asterisk *.

Press F2 (Config) and go to the configuration screen. This is where our inputs / outputs will be assigned to the appropriate places, i.e. in the case of FANUC – racks and slots.

EtherNe / IP is rack 89, we have only one robot arm plugged into the controller, so one adapter – slot 1.

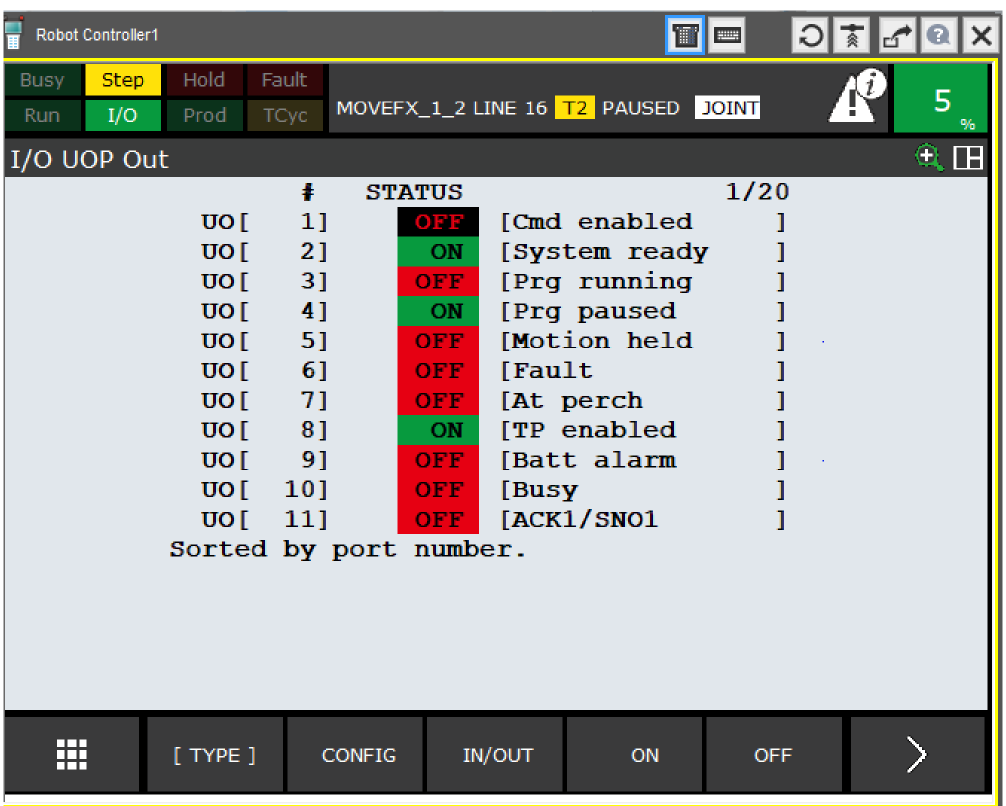
Enter the range of inputs that interest us, which we want to use, assign them to the appropriate rack and slot, give the start bit, i.e. where in the communication frame the given group of inputs / outputs will be located. To determine the start bit, we refer to the prepared table and read where the given I / O group is to be mapped.

In the example below, for inputs 1-40 it will be bit 25

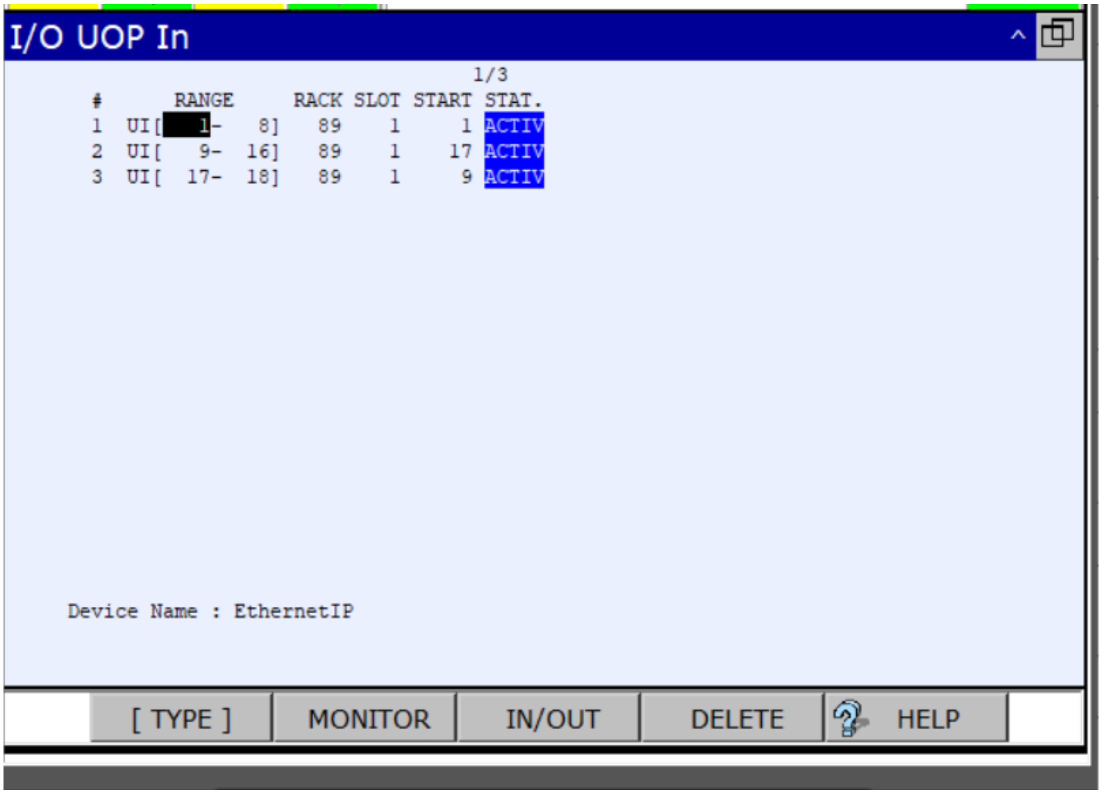
As you can see after making changes, the STATUS of the edited inputs / outputs has changed to PENDING. For the changes to take effect, restart the robot controller or select Cycle Power with FCTN.

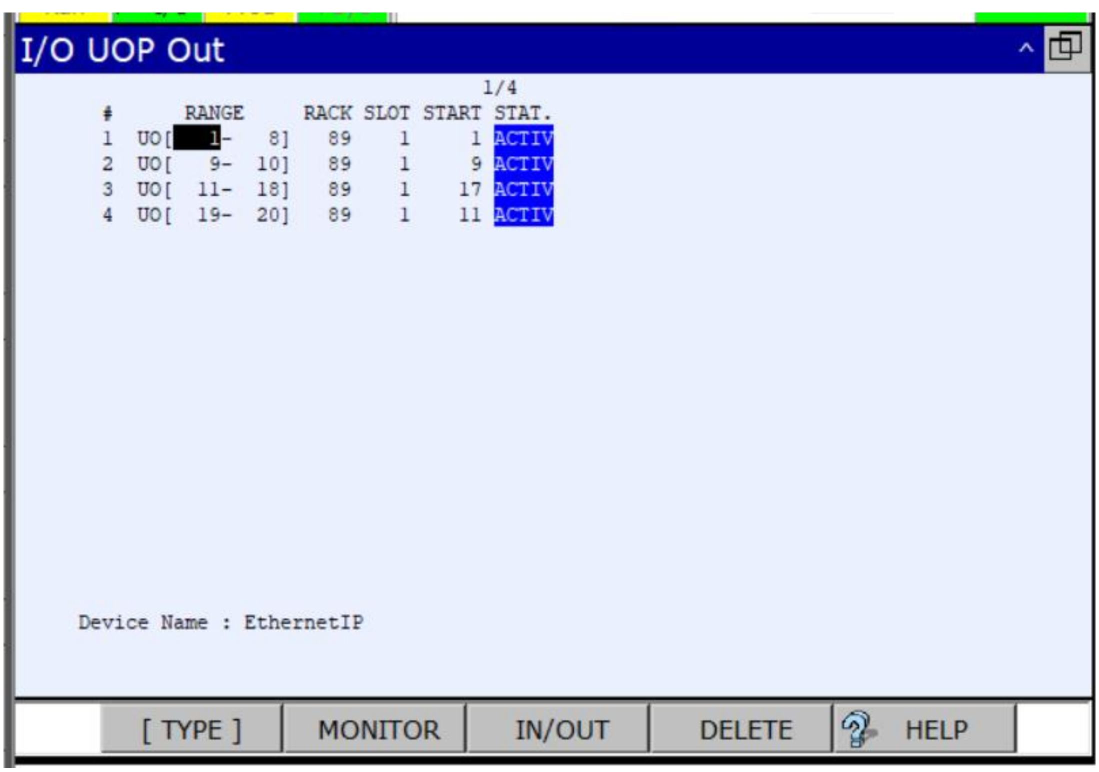
We configure group outputs depending on the needs, if we want to send values other than the bool type. We can group 2 to 16 lines that may overlap with digital I / O.

We do the same in the case of UOP signals:

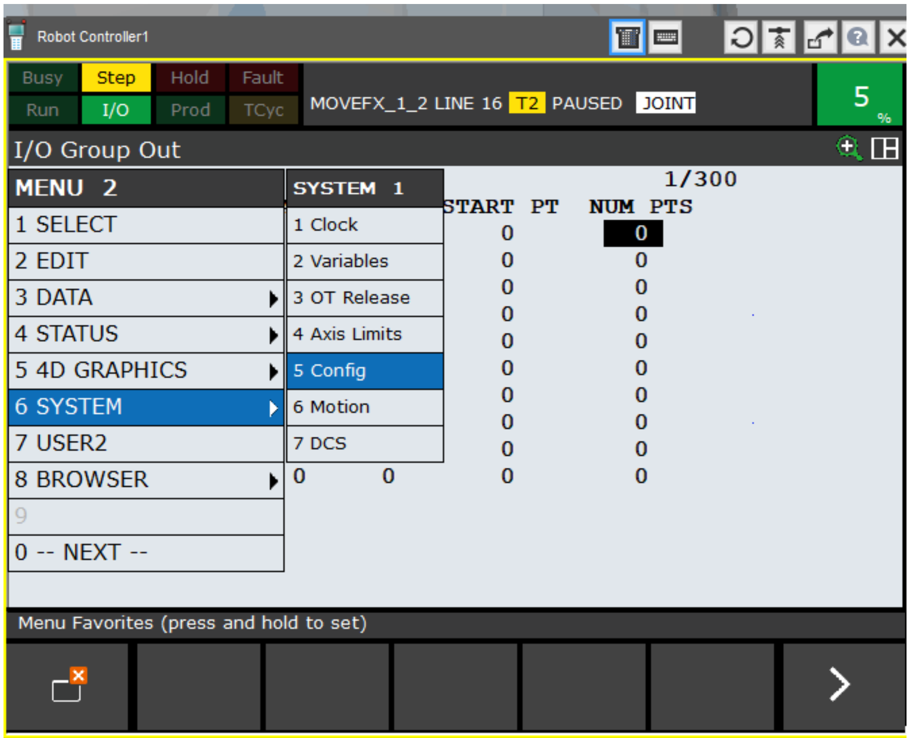


Since in step 1 I determined that the UOP signals will occupy the first three bytes, I set the configuration as follows:

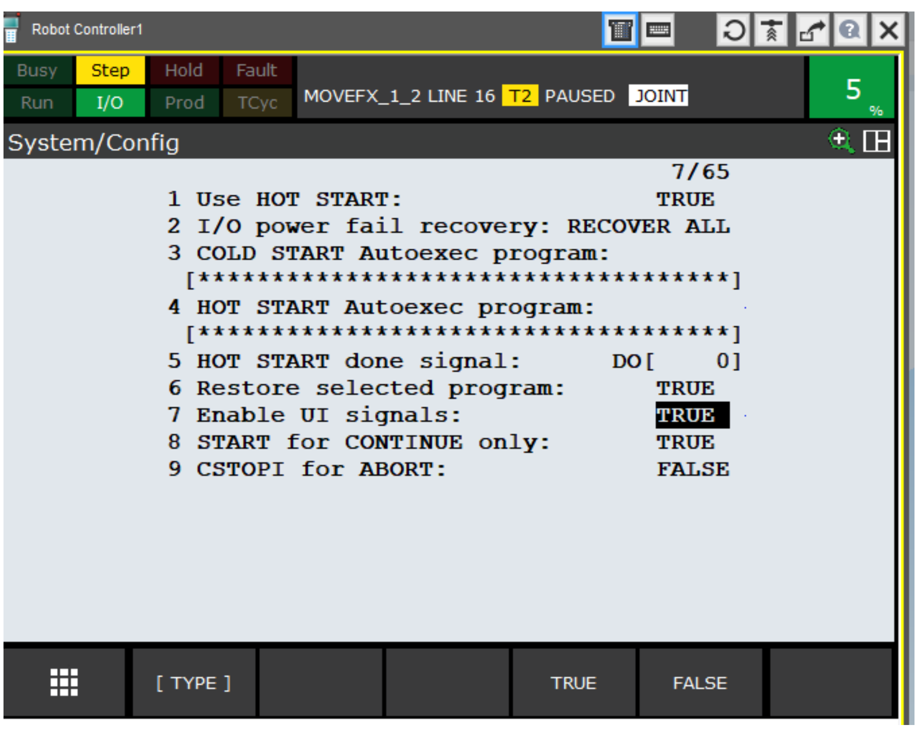




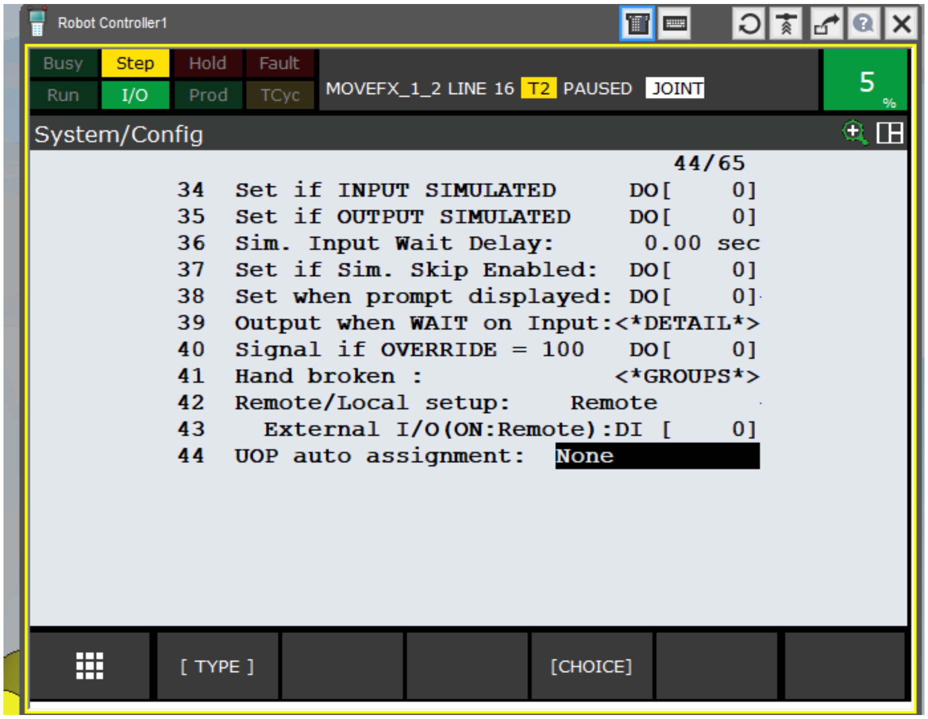
Moreover, to enable the exchange of UOP signals between the robot controller and the controllers, this option must be activated in the controller. To do this, go to the system configuration menu on the Teach Pendant



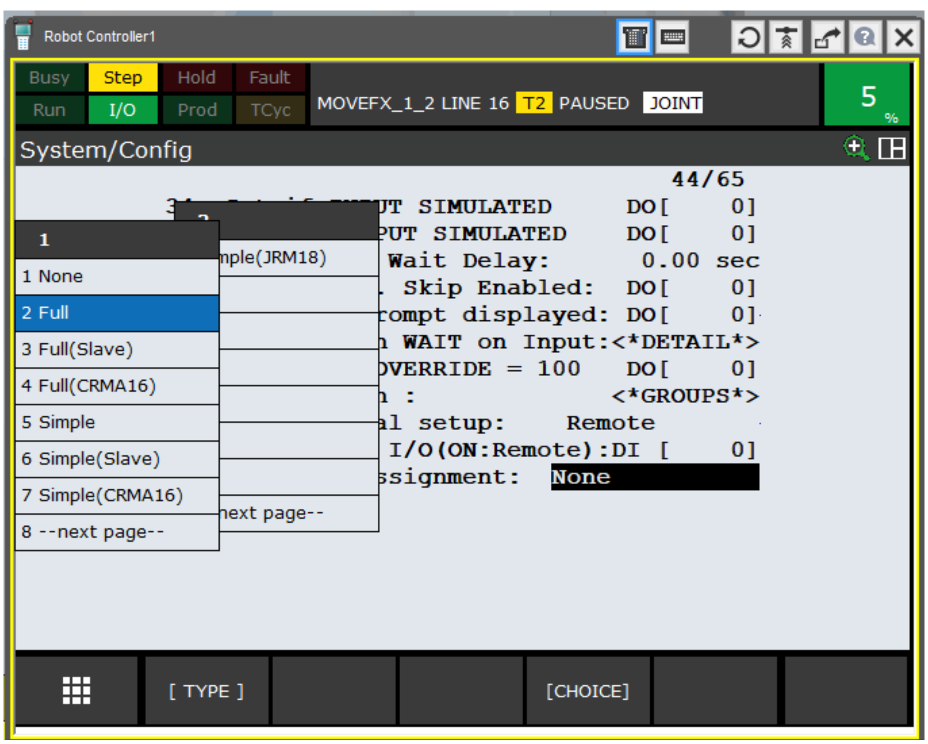
We find position number 7 – Enable UI Signals



If the value is set to FALSE, the external signals (UI [1 to 18]) are disabled. Set this value to TRUE to allow programs to be started with the PLC. In addition, you need to make sure how the UOP signals are assigned. In the same menu we find position 44:



We choose the signal allocation method (FULL).



Universal Robot Connection to Siemens PLC – from Universal Robots website:

PROFInet How-To Guide E-series

This guide provides instructions for how to start using PROFINET with Universal Robots**.**

Last modified on Sep 30, 2024

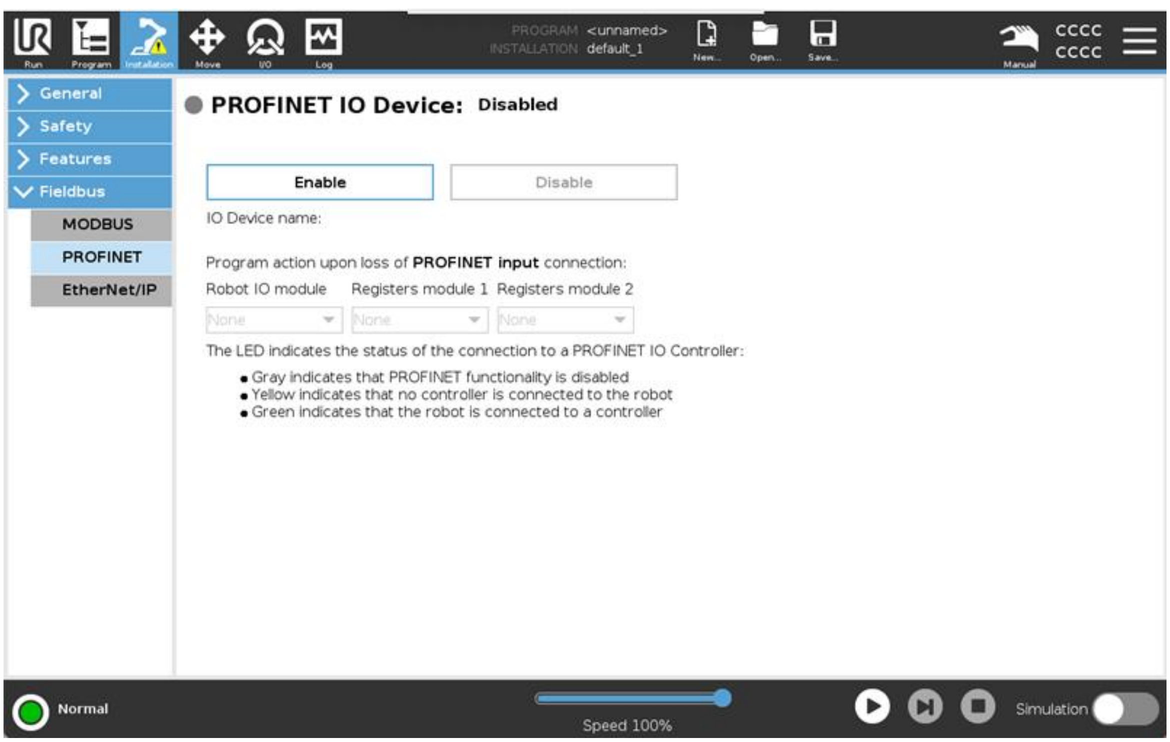
NOTE: All files are available for download at the bottom of this page.

This guide is valid for: e-Series Software version 5.12
Note: older or newer software versions may behave differently.

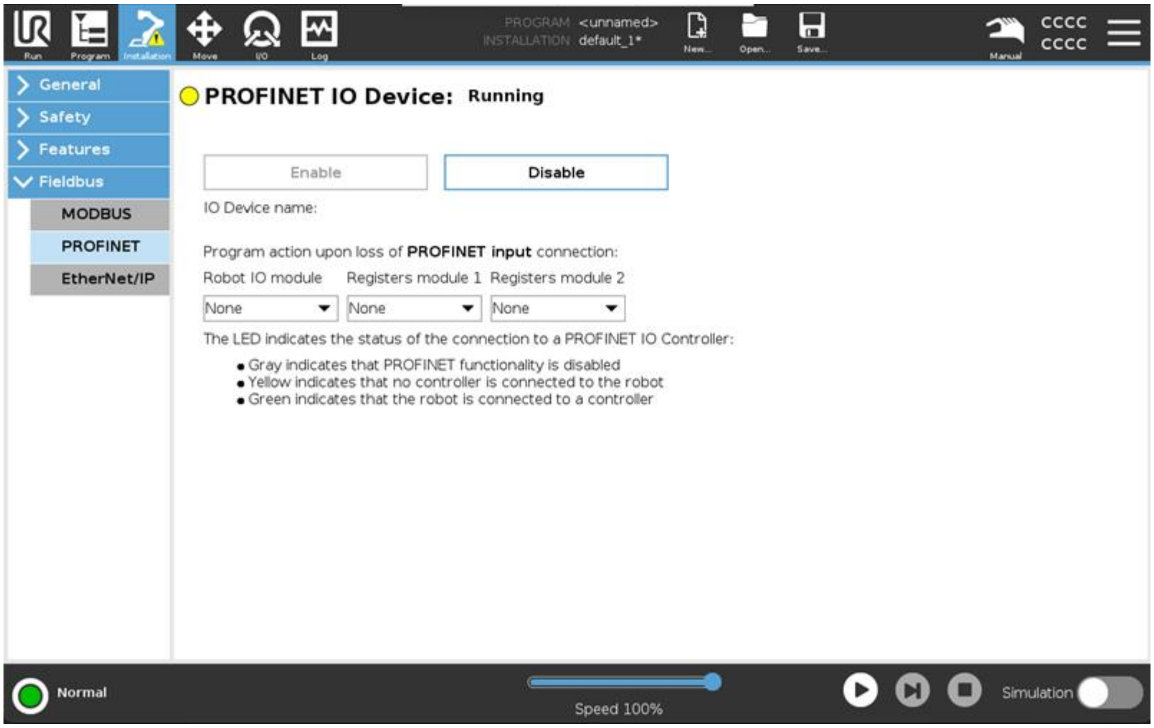
1. setup

1. **ROBOT:** Tap the Installation tab and, under Fieldbus, select **PROFINET**. Tap **Enable**.

- Save the installation for the changes to take effect the next time the installation is loaded.

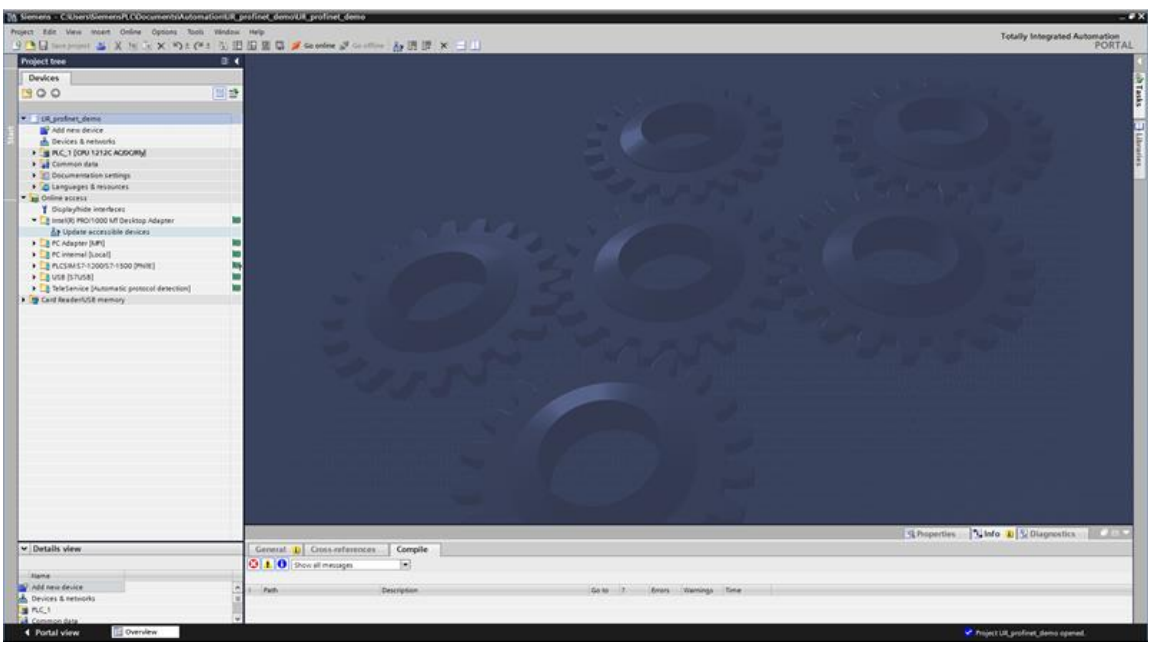


2. **ROBOT:** The yellow LED indicates PROFINET is running on the robot. No PLC/IO controller is connected to the robot.



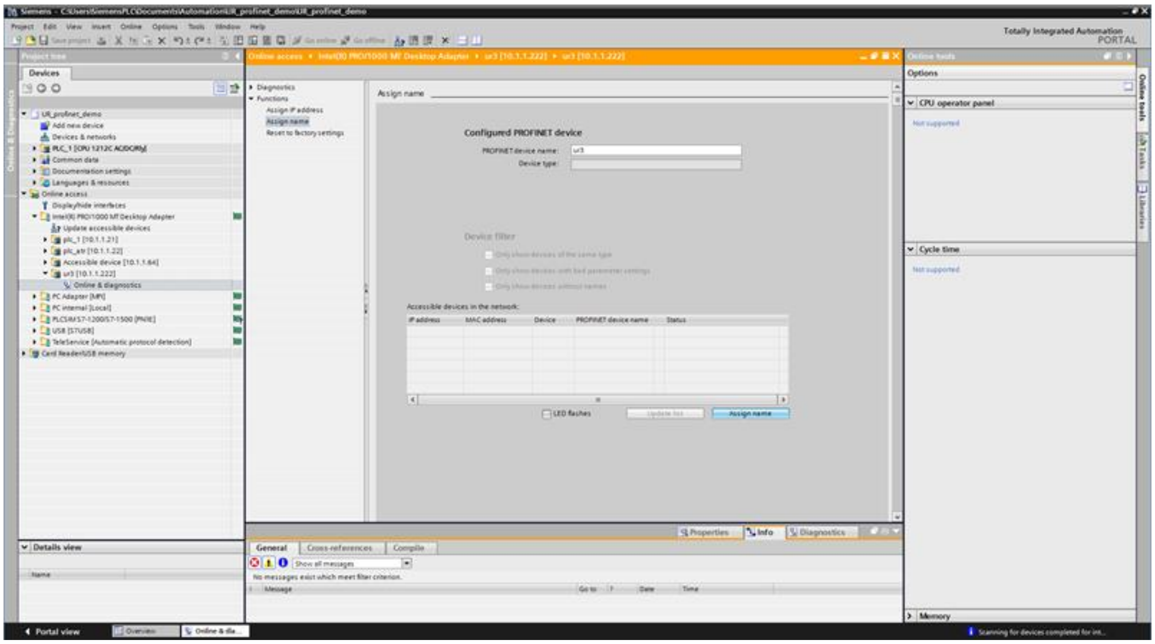
3. **PLC:** Open a project in Siemens TIA Portal.

- In the Project Tree of the Project view, navigate to: **Online access -> {your network adapter}.**
- Click **Update accessible devices**.



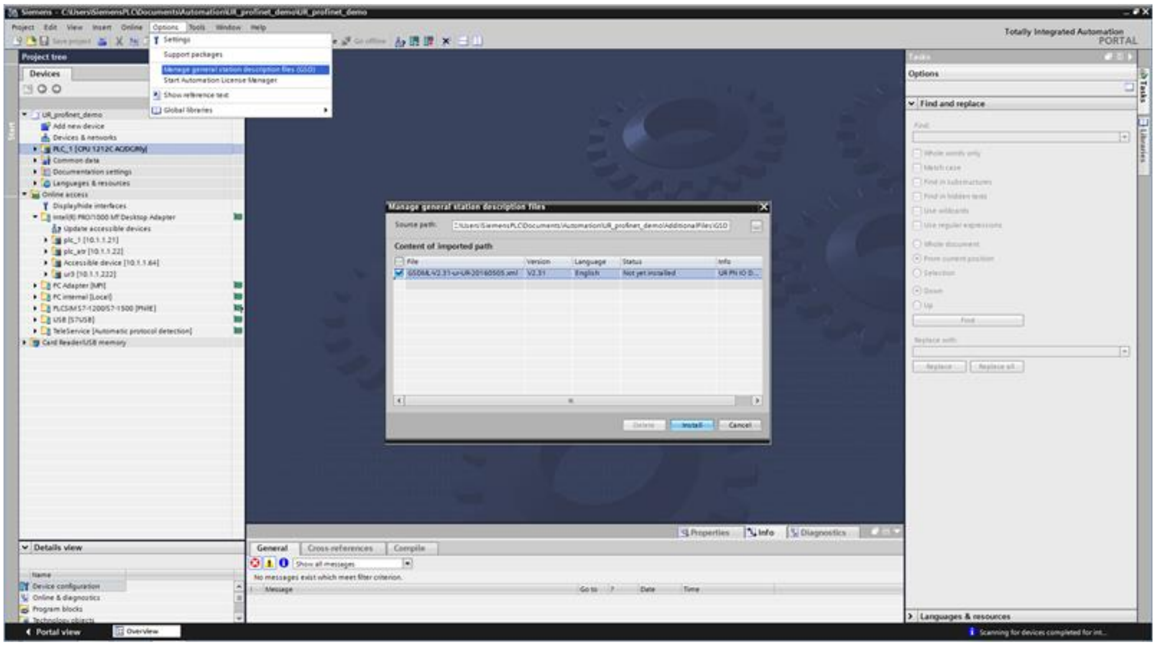
4. **PLC**: Identify the robot and the PLC you want to connect. Assign IP addresses and names to the equipment.

- Expand the desired accessible device and double-click **Online & diagnostics**.
- In the device's diagnostics page, expand Functions and click **Assign IP address**.
- Enter the desired IP address and subnet mask, and tap **Assign IP address**.
- Still in the diagnostics page click **Assign name**.
- Set the PROFINET device name and click **Assign name** .



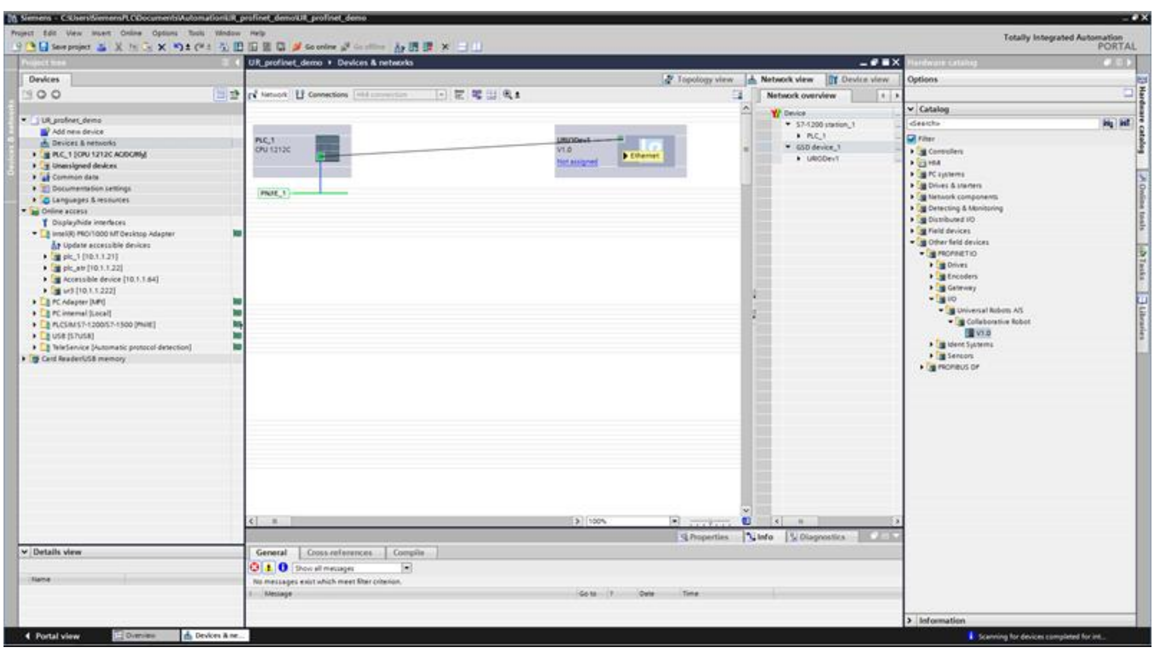
5. **PLC**: Import the Universal Robots' GSD file.

- Navigate to **Options ->Manage general station description files (GSD)**.
- Select to the right GSD and install.



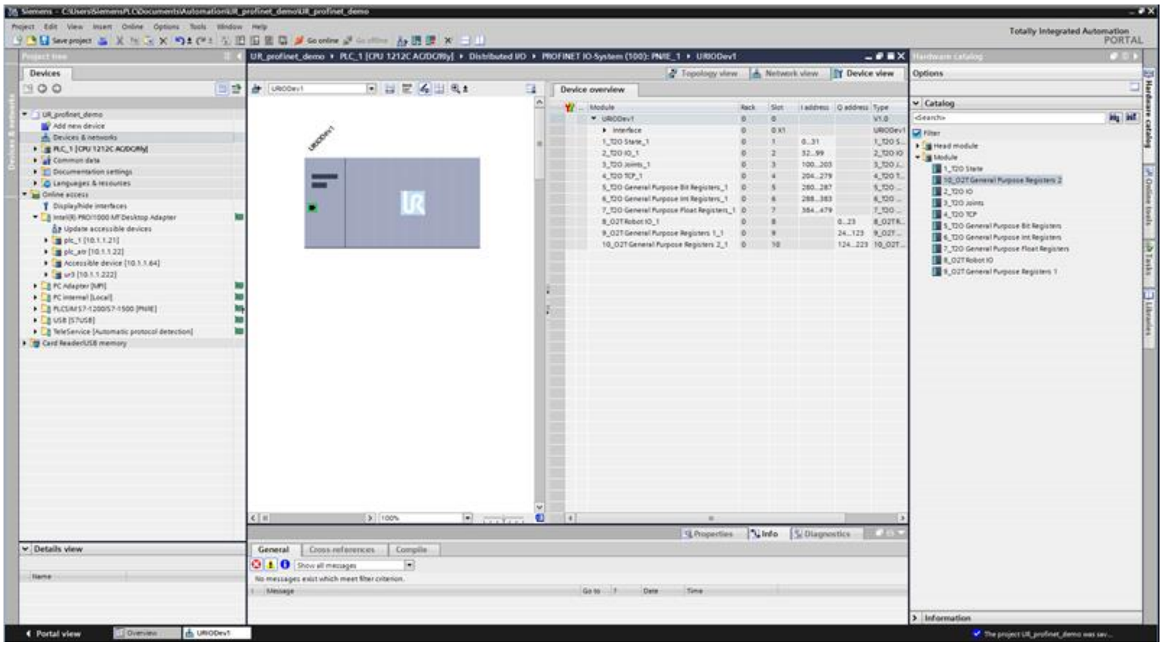
6. **PLC**: In the top of the project tree access Devices & networks.

- Locate the UR I/O device in the catalog path: **Other field devices->PROFINET IO->I/O->Universal Robots A/S->Collaborative Robot->V1.0**
- Drag-and-drop it into the network view and connect the PLC and the I/O device by dragging a line between their (green) PN/IE ports.



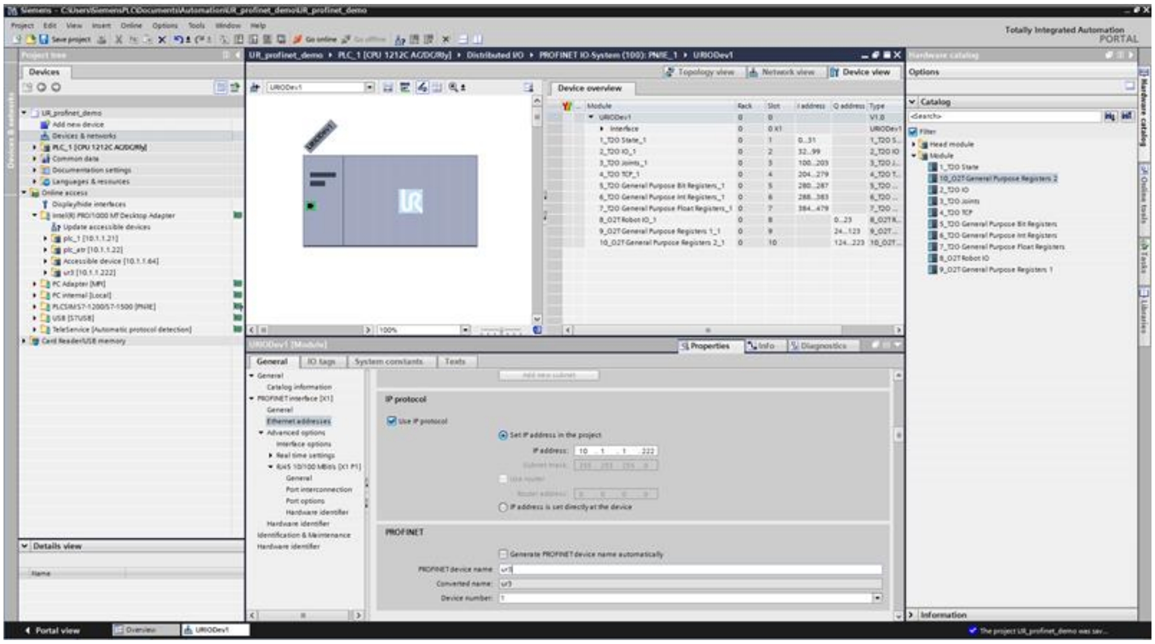
7. **PLC**: Double-click the I/O device in the Network view.

- All modules are pluggable, and are named according to their corresponding slot numbers.
- Each module can be drag-and-dropped from the **Catalog->Module** to the empty fields in the Device overview.
- In this example all modules are plugged in. Notice how the input addresses (I) and output addresses (Q) are assigned, since you will need these to access the data.



8. **PLC**: Double-click the I/O device in the Device view to get the module description.

- In the General tab select **Ethernet addresses**, and select **Use IP protocol.**
- Set the same IP address as those assigned to the online device.
- Deselect **Generate PROFINET device name automatically** and type in the same PROFINET device name as the one assigned to the online device.



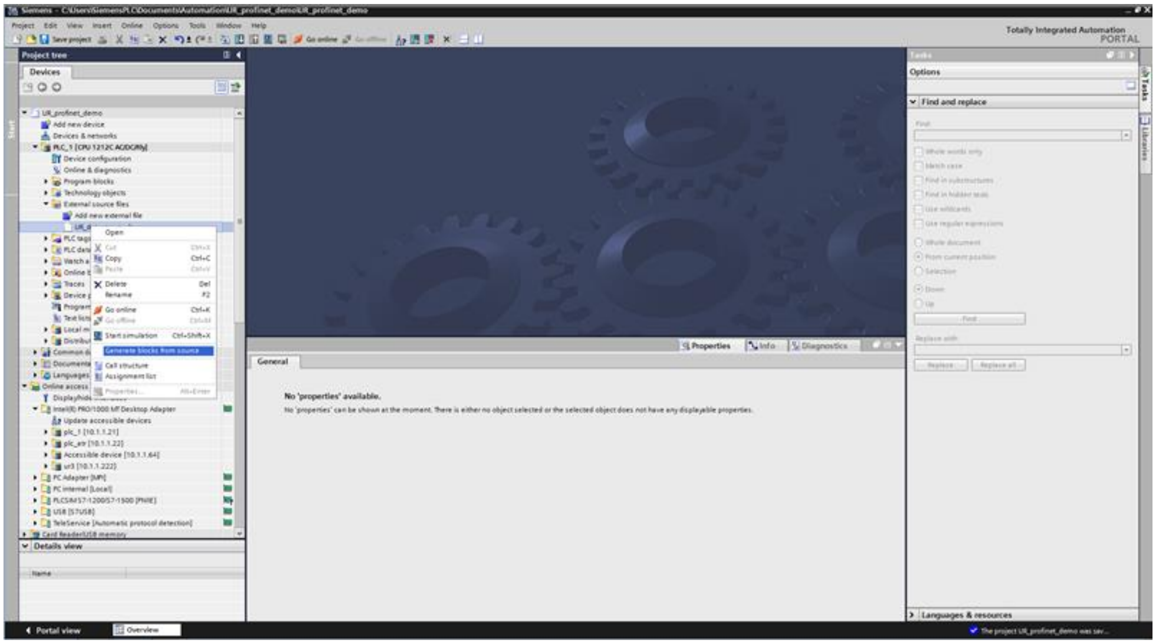
9. **PLC:**

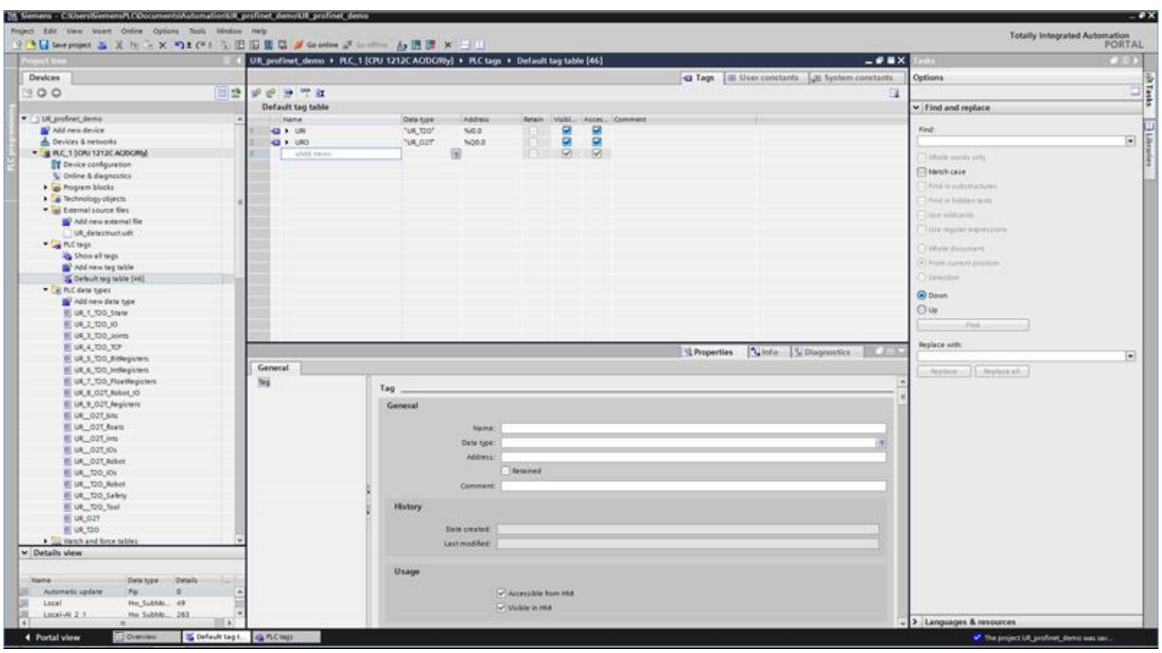
This step requires PLC firmware version 4.0 or higher.

- Navigate to **External source files**.
- In the project tree, double-click **Add new external file** to open the file **UR_datastruct.udt**.
- Import the user-defined data types by right clicking the newly added external source file and select **Generate blocks from source**.
- In the PLC tags (or your own Data Block), create tags corresponding to the modules you have enabled. Make sure the input addresses (I) and output addresses (Q) of the tags match those of the modules.

If all modules are enabled and placed in order in the IO memory, create a single PLC input tag of the data type: UR_T2O and a single PLC output tag of the data type: UR_O2T to map all data.

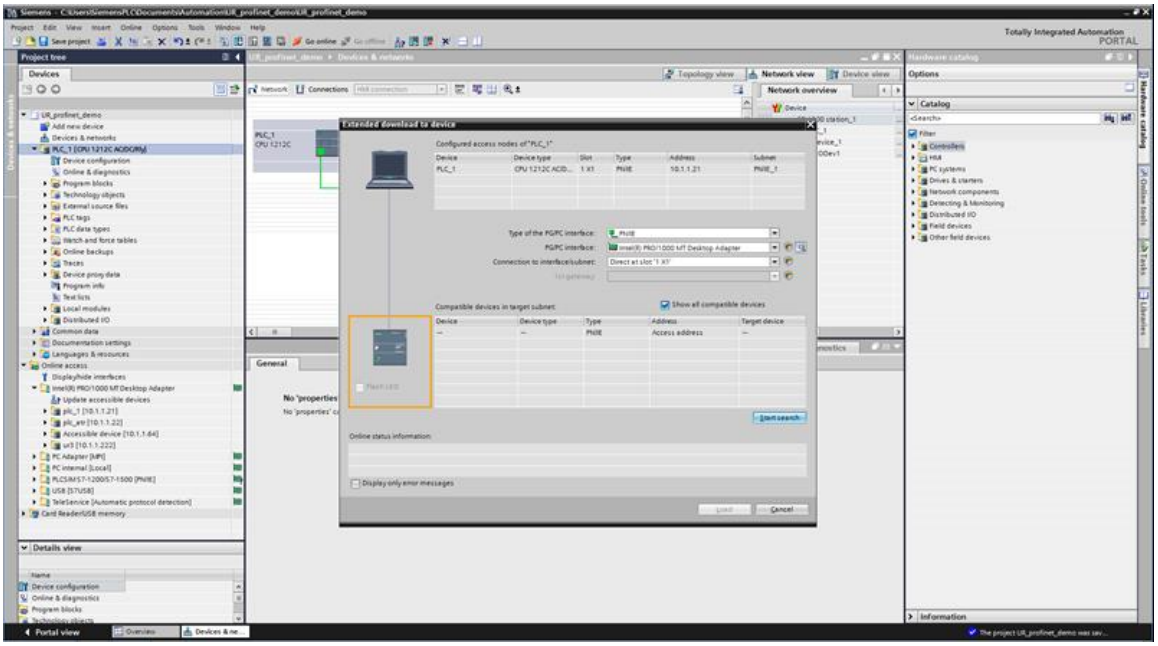
Both are visible in **PLC data types** in the project tree.



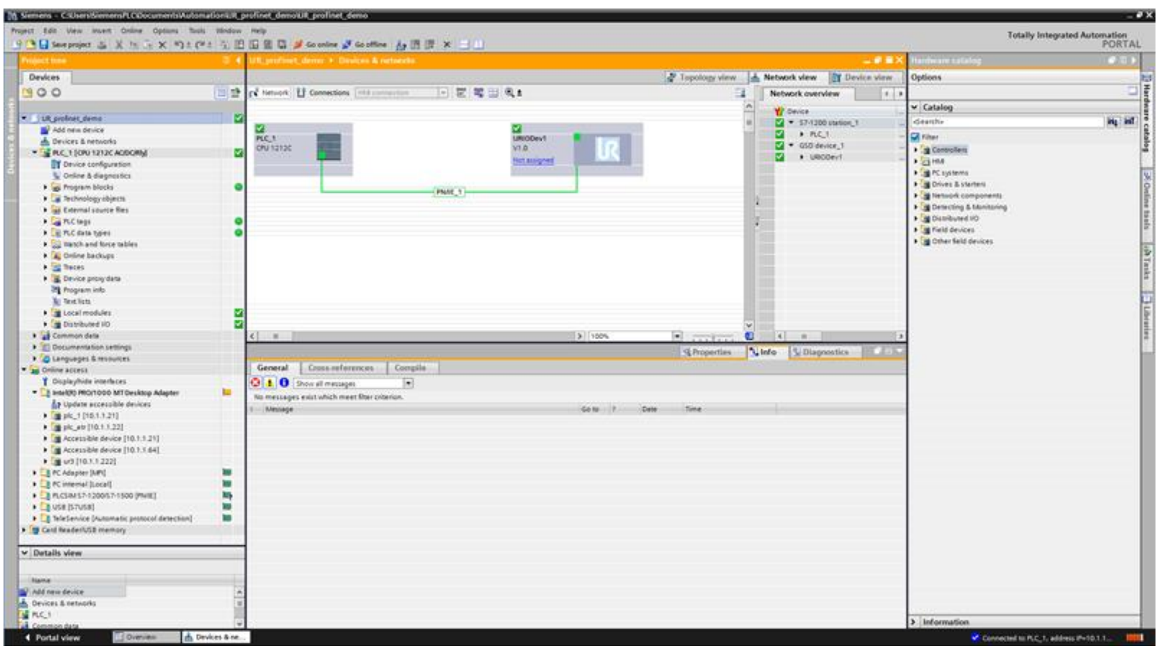


10. **PLC:** Download the program to the PLC.

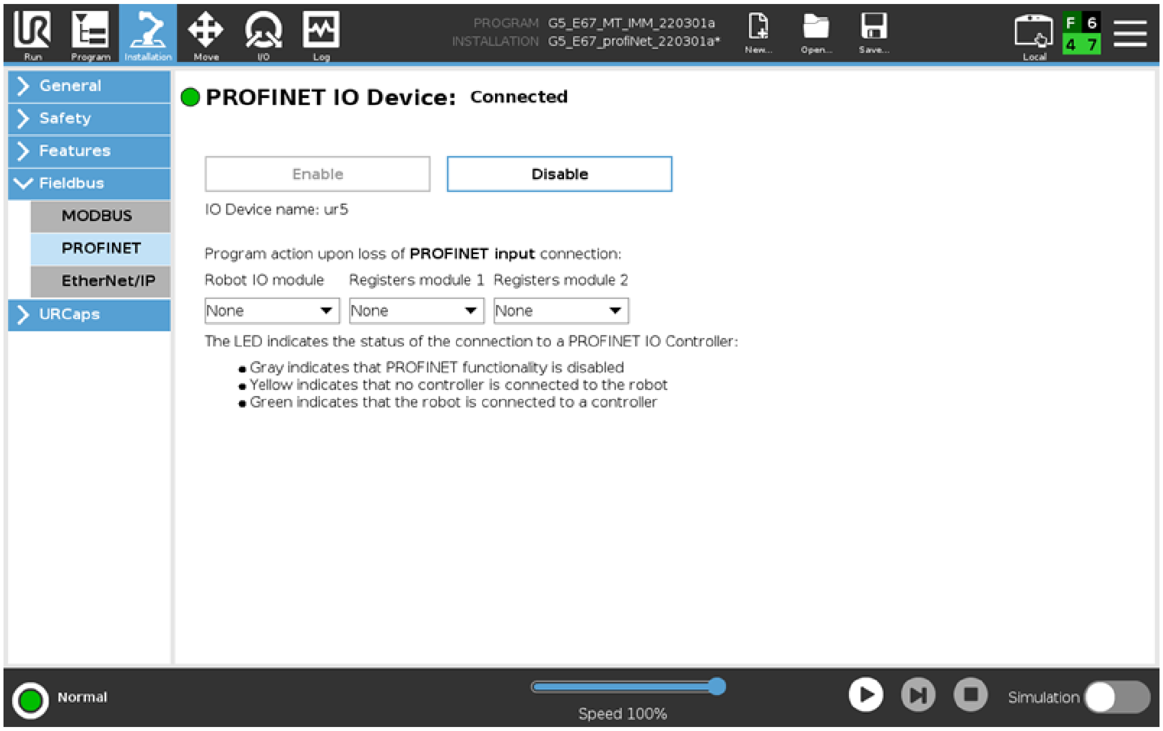
- In the menu bar click the download icon and tap **Start search** in the download dialog.
- Select the PLC and click **Load**.



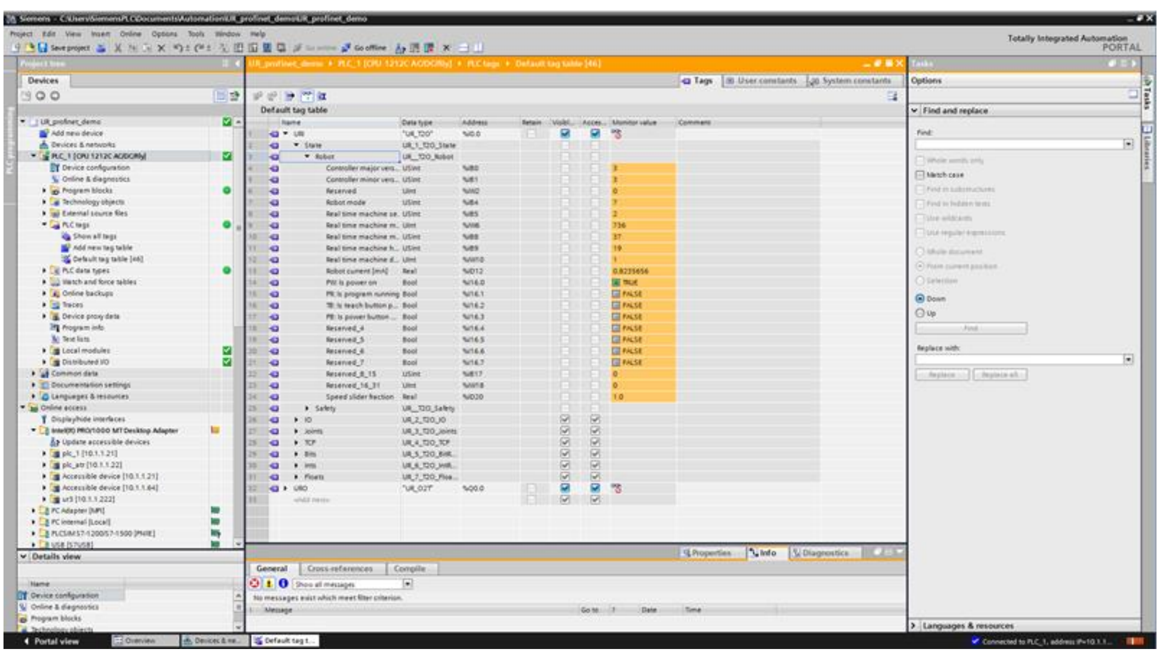
11. **PLC:** Confirm all blocks, tags and modules in the project tree are green.



12. **ROBOT:** Confirm the LED in the Installation tab is green.



13. **PLC:** In the project tree, navigate to your tag table (or Data Block) and expand your input tag to inspect values sent from the robot.



2. Specifications

The following sections provide an overview of some important information to get started.

2.0.1. Demo Sample Program

Sample files for this example are available for download at the bottom of this page:

- UR_profinet_demo.zap13: archived project file that can be loaded into the TIA Portal v13 (select **Project->Retrieve**)
- ur-pn-demo.urp: demo program for the robot

PLC (used in example)

- Siemens S7-1200, 1212C AC/DC/Rly
- Totally Integrated Automation Portal V13, STEP 7 Basic

Robot

- UR3, UR5 or UR 10 running PolyScope v. 3.3

2.0.2. I/O Message Format

For details about how the IO format is distributed between the modules, see: pn-iomessage.pdf

The robot has 10 pluggable modules:

- Seven modules containing the data that can be read from the robot
- Three modules containining data that can be set on the robot

All modules are optional, but are fixed to a specific slot according to their names: e.g. the UR_3_T2O_Joints can be plugged into slot 3, contain data from the robot (T) to the PLC (O), namely joint measurements.

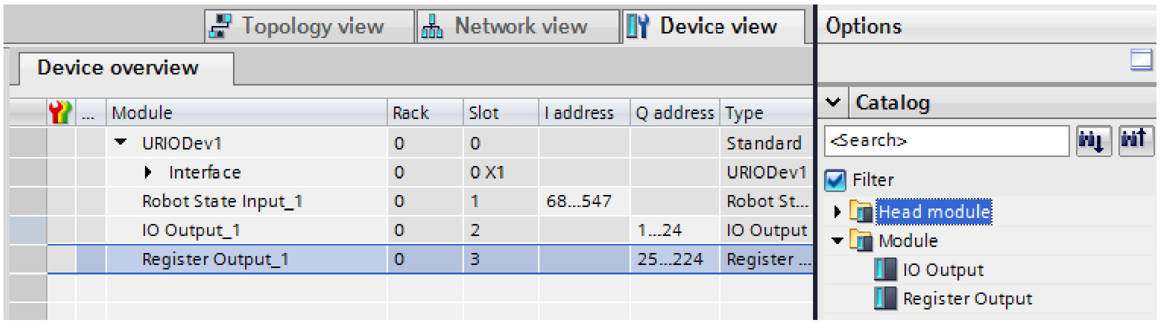
Note: When using output I/Os, it is important to set the mask accordingly.

User-defined Data Types

- For PLCs S7-1200 and S7-1500 (PLC Firmware v4.0 or higher): *UR_datastruct.udt* (can be imported into TIA Portal)
- For PLCs S7-200, S7-300 and S7-400: URI.AWL, URO.AWL (can be imported into SIMATIC and TIA Portal)

The UDT/AWL files contain user-defined data types and can be used to import the message format.

Note: If you import the UR data structure (the .udt file) and only plug in some of the modules, make sure to use the right user defined data types and match the input address (I) or output address (Q) of the module with the address of the data types in the tag table. This is also necessary if you drag the modules in "out of order", e.g. the module to slot 5 before the module to slot 2.



GSD file

GSDML-V2.42-UR-PROFIsafe-20220517.xml.zip. *(Note that the folder name is PROFIsafe but some of the files are compatible with PROFInet)*

2.0.3. Script Functions

Script Functions for synchronizing a program with a PLC:

- read_input_boolean_register(address)
- read_input_float_register(address)
- read_input_integer_register(address)
- read_output_boolean_register(address)
- read_output_float_register(address)
- read_output_integer_register(address)
- write_output_boolean_register(address, value)
- write_output_float_register(address, value)
- write_output_integer_register(address, value)

2.0.4. Notes

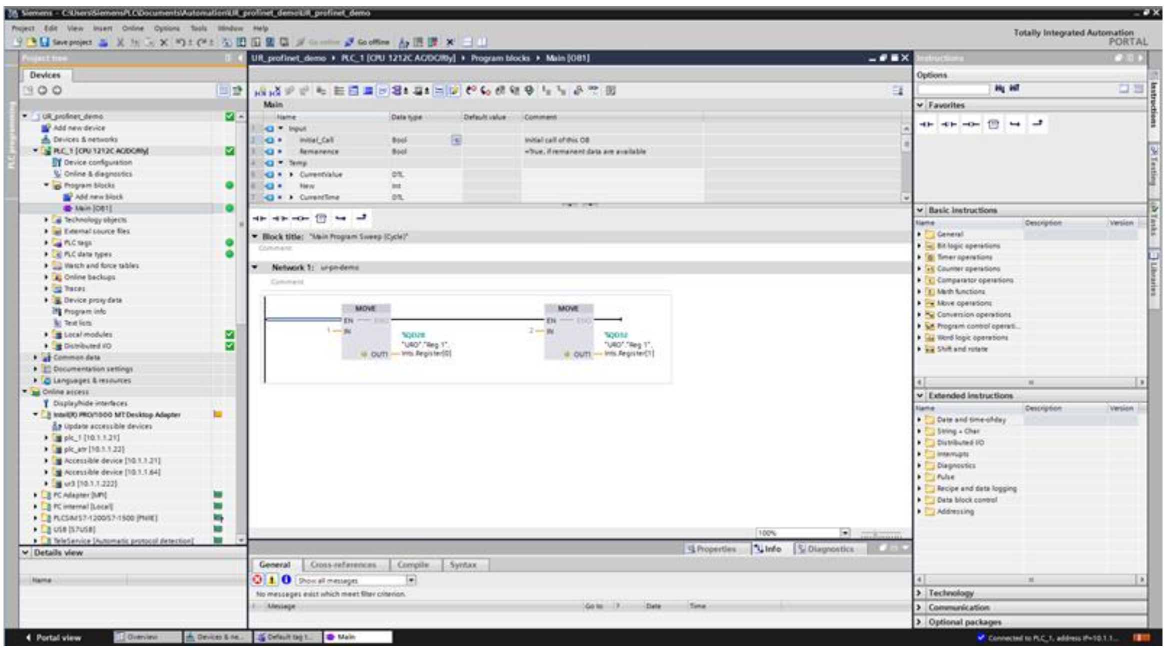
- If you change the IP of the robot in Network Settings while PROFINET is enabled, disable PROFINET and enable it again before the changes take effect.
- If the IP of the robot is changed using external PROFINET tools, e.g. through the TIA software, the changes will take effect immediately.
- The lower range (bool[0:63], int[0:23], float[0:23]) of the gp input and output registers is reserved for FieldBus/PLC interface usage. The upper range (bool[64:127], int[24:47], float[24:47]) can be used by external RTDE clients (i.e URCAPS).

3. Demo

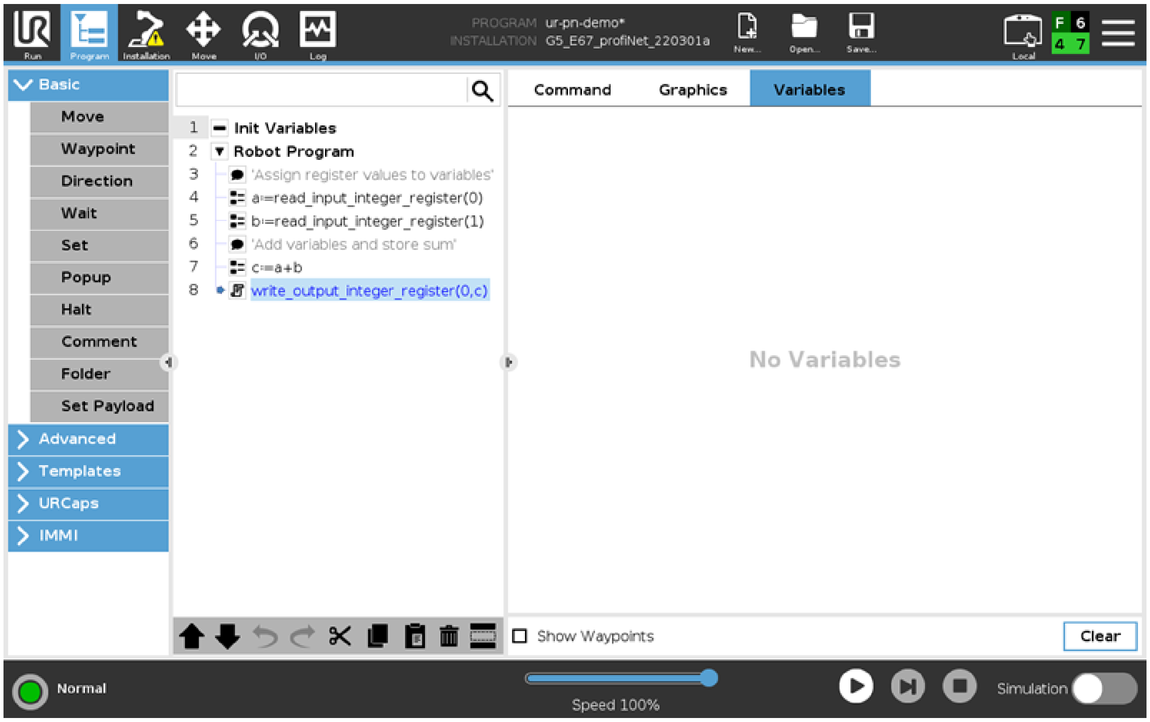
This demo is provided to show a basic example of how to communicate between the PLC and the UR robot. The demo allows you to set two numbers in PLC output registers.

The robot retrieves the numbers, calculates their sum and stores the result in a PLC input register.

1. **PLC:** Navigate to the Main Program Block and set integer registers URO."Reg 1".Ints.Register[0] and URO."Reg 1".Ints.Register[1] to some values and download the program to the PLC



2. **ROBOT:** Open ur-pn-demo.urp and run the program. Navigate to the Variables tab and confirm the numbers are received.



4. Robot Modes

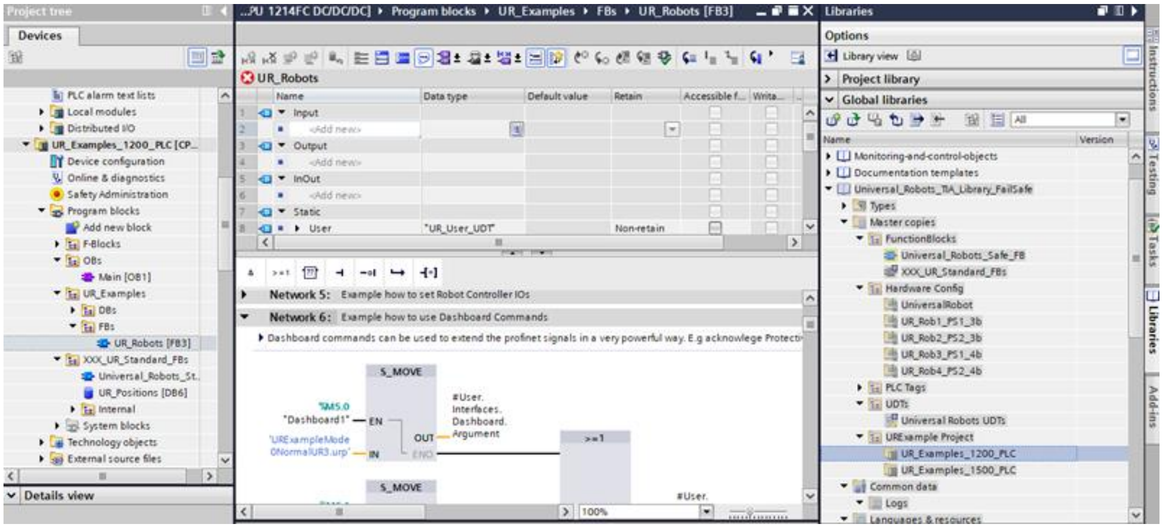
Mode	Description
-1	ROBOT_MODE_NO_CONTROLLER
0	ROBOT_MODE_DISCONNECTED
1	ROBOT_MODE_CONFIRM_SAFETY
2	ROBOT_MODE_BOOTING
3	ROBOT_MODE_POWER_OFF
4	ROBOT_MODE_POWER_ON
5	ROBOT_MODE_IDLE
6	ROBOT_MODE_BACKDRIVE
7	ROBOT_MODE_RUNNING
8	ROBOT_MODE_UPDATING_FIRMWARE

5. Video Guides

Italian Original : [Profinet connection between robot and PLC - YouTube]

English Translated : [Profinet connection between robot and PLC - YouTube]

6. Example image of FailSafe Library



Attached files

[GSDML-0361-0001-UR.bmp]

[GSDML-V2.42-UR-PROFIsafe-20220517.xml.zip]

[IMMI_Euromap67_input_output_bits.pdf]

[URI.AWL]

[URO.AWL]

[UR_datastruct.udt]

[Universal_Robots_TIA_Library_FailSafe_Robot_Programs.zip]

[Universal_Robots_TIA_Library_FailSafe_V16_20231016_1159.zal16]

[Universal_Robots_TIA_Library_FailSafe_V17_20231016_1333.zal17]

[Universal_Robots_TIA_Library_FailSafe_V18_20231016_1502.zal18]

[pn-iomessage.pdf]

[ur-pn-demo.urp]

To implement a connection between the older Universal 5 robot, use the following. Remember that the communication to the Universal Robot here is Modbus/TCPIP.

https://s3-eu-west-1.amazonaws.com/ur-support-site/66329/UR%20Modbus_bba_mods_0.1.4.pdf

Ethernet IP guide

This is an Ethernet IP guide to help you when using this with Universal Robots

Last modified on Sep 30, 2024

NOTE: All files are avalable for download at the bottom of this page.

Examples are valid for:
CB3 Software version: 3.2.18654
e-Series Software version: All versions
Note that newer software versions may behave differently.

These examples can be used on CB3/CB3.1 from software 3.2 and e-Series.

Specifications

Here is an overview of some important information to get started.

PLC *(used in example)*

- Allen-Bradley Compact Logix L16ER
    - Studio 5000 Logix Designer
      Note:
    - UR only supports Ethernet/IP Class 1 implicit or "I/O" messaging, not Ethernet/IP Class 3 explicit or "client/server" messaging.
    - Remember to change/select the correct CPU

Robot

- UR3, UR5, & UR 10 running Polyscope v. 3.2

Sample Programs:

The files listed are available for download at the bottom of this page and are used in the example (ACD project files are compatible with Logix Designer v.24):

- UR.ACD
- ur-eip-demo.urp

Conveyor Tracking example:

- EIP_conveyor_example.ACD

- UR-EIP-conveyor-tracking.urp

EDS file:

- UniversalRobot.eds

User-defined Data Types:

- UR_DataTypes.L5X

Note: The L5X file contains user-defined data types and can be used to import the message format into Rockwell Studio 5000 Logix Designer

I/O Message Format:

- eip-iomessage.pdf

Note: When using output registers, it is important to set the mask accordingly.

Script

Script Functions for synchronizing a program with a PLC:

- read_input_boolean_register(address)
- read_input_float_register(address)
- read_input_integer_register(address)
- read_output_boolean_register(address)
- read_output_float_register(address)
- read_output_integer_register(address)
- write_output_boolean_register(address, value)
- write_output_float_register(address, value)
- write_output_integer_register(address, value)

Notes

The lower range (bool[0:64], int[0:23], float[0:23]) of the gp input and output registers is reserved for FieldBus/PLC interface usage. The upper range (bool[64:127], int[23:47], float[23:47]) can be used by external RTDE clients (i.e URCAPS).

Trouble Shooting

- It is important to consider and set an RPI value that corresponds with the requirements of the installation. The valid range of RPI is 2ms - 3200ms where a smaller value means more frequent data synchronization but also requires more resources from the PLC, Robot and Network.

- If you experience connection issues between a scanner and the robot make sure that you have configured the scanner for unicast connection (point-to-point) and that you have sufficient network bandwidth between the scanner, the Robot and other devices to avoid colliding packets etc.

Setup

This guide gives step-by-step instruction on how to start using Ethernet/IP with Universal Robots or watch the video

[Video: How to connect your UR robot to an Allen Bradley PLC]

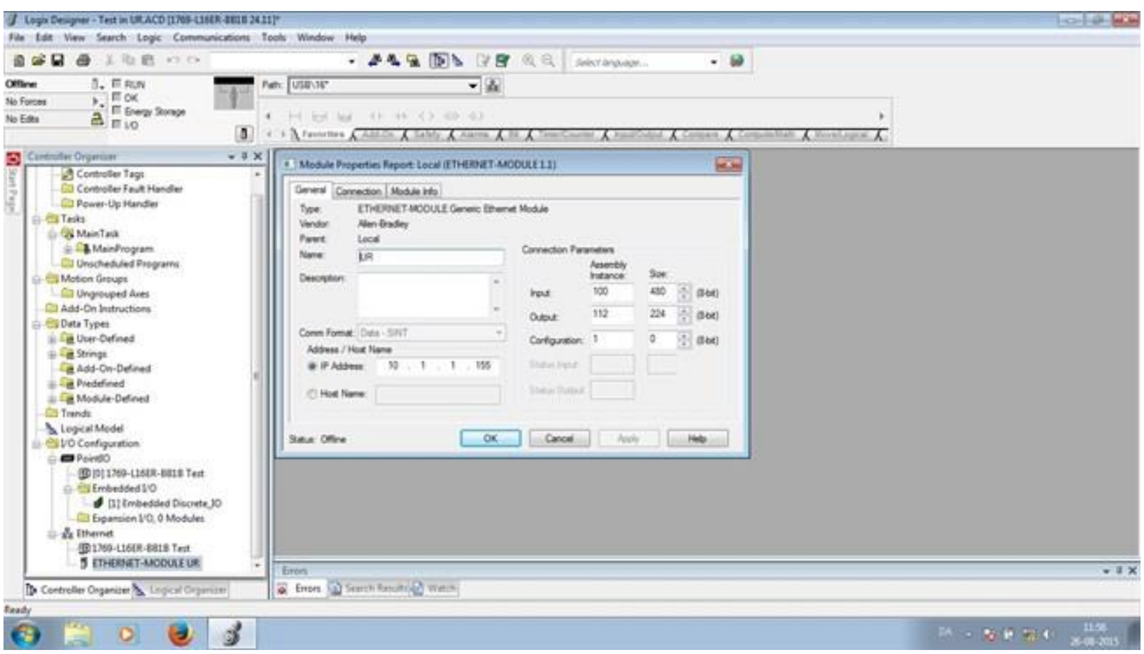
1. ROBOT: Enable Ethernet/IP in the Installation tab. Remember to save the installation afterwards for the changes to take effect the next time the installation is loaded.

2. ROBOT: The yellow LED indicates Ethernet/IP is running on the robot but no PLC/scanner is connected to the robot

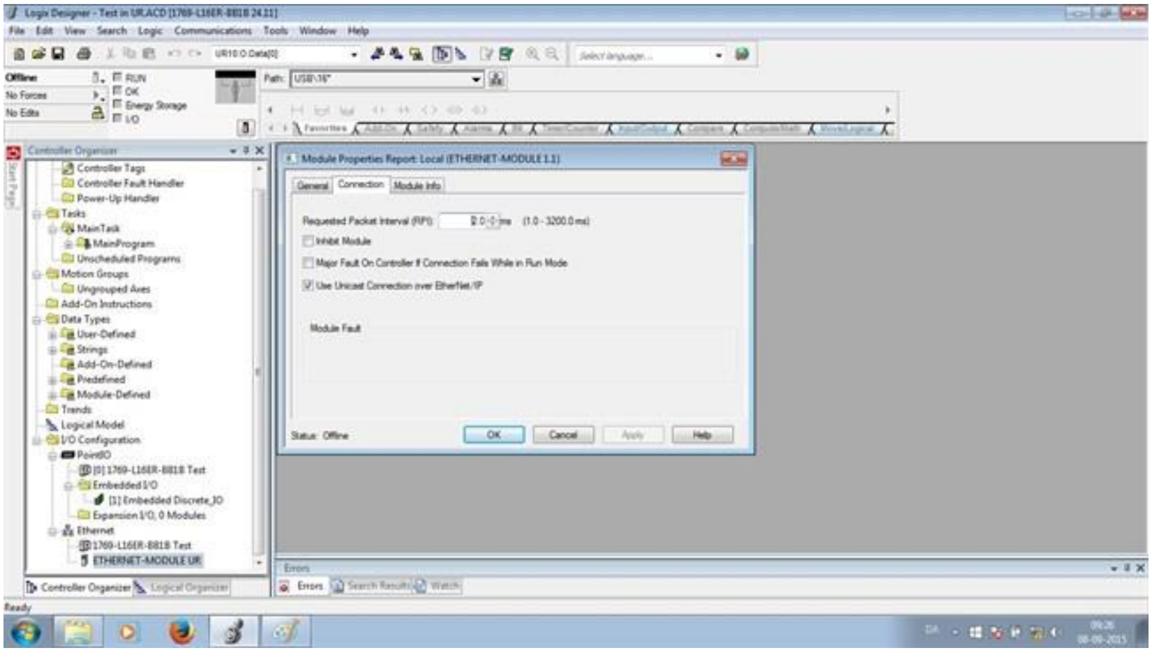
3. PLC: Open UR.ACD project in Studio 5000 Logix Designer. If your version of Logix Designer is incompatible with the project file (i.e. older than v.24), you can create a new project and import the user-defined data types from the UR_DataTypes.L5X file.

4. PLC: Navigate to I/O Configuration -> Ethernet -> Ethernet-Module UR

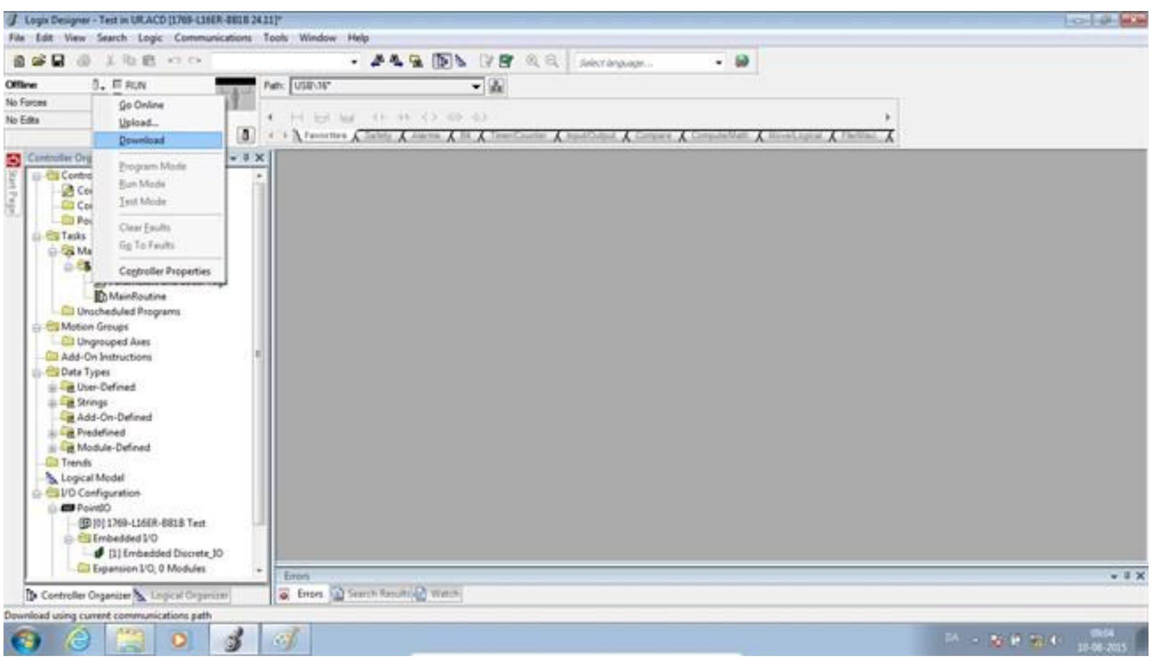
5. PLC: Set IP Address of Robot and Press "Apply"



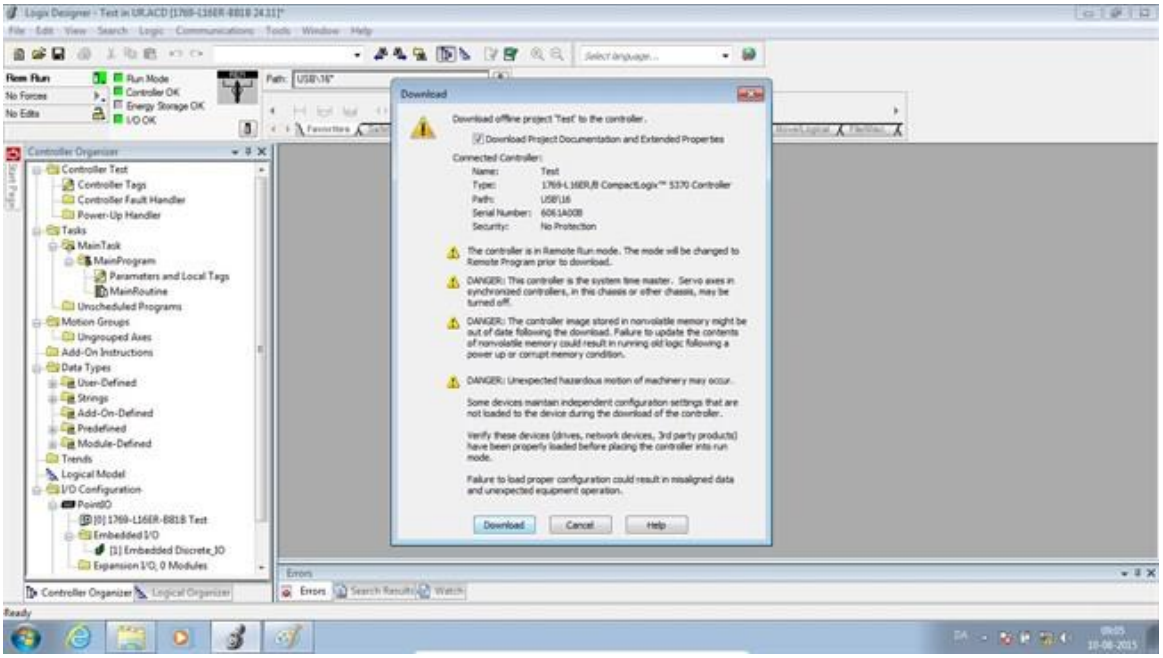
6. PLC: Navigate to "Connection" Tab and ensure that the RPI matches your requirements and that the check box next to "Use Unicast Connection over Ethernet/IP" is checked.



7. PLC: Download the program to the PLC



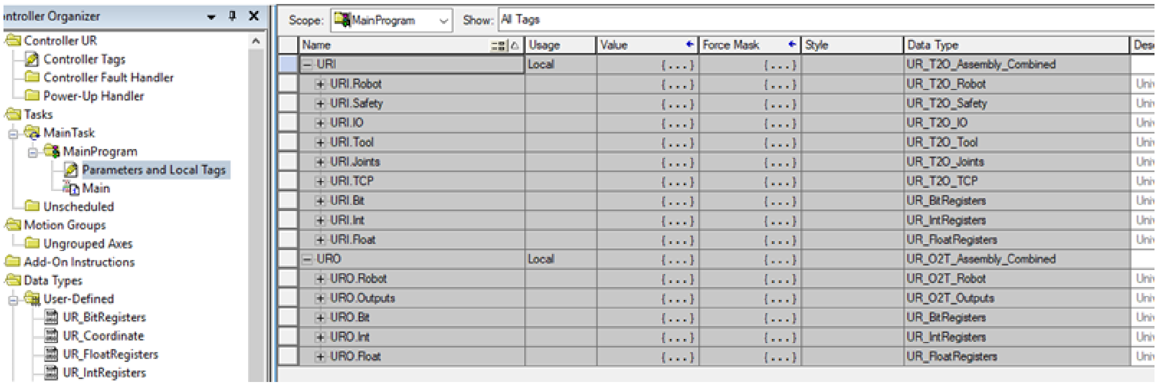
8. PLC: Confirm that "I/O OK" is solid green



9. ROBOT: Confirm that LED in Installation tab is also green

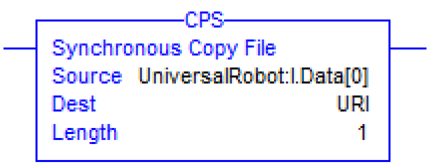
10. PLC: Navigate to Tasks -> Main Task -> Main Program -> Parameters and Local Tags

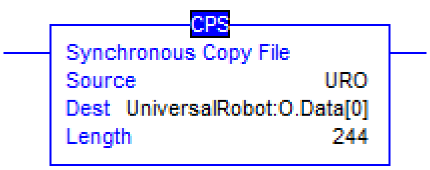
10.1. IF You did not use the demo, but will implement communication to your own code, you need to enable communication between the robot and the AB PLC. First set up URI and URO in your local or global variables, using the provided User Define Data Types:



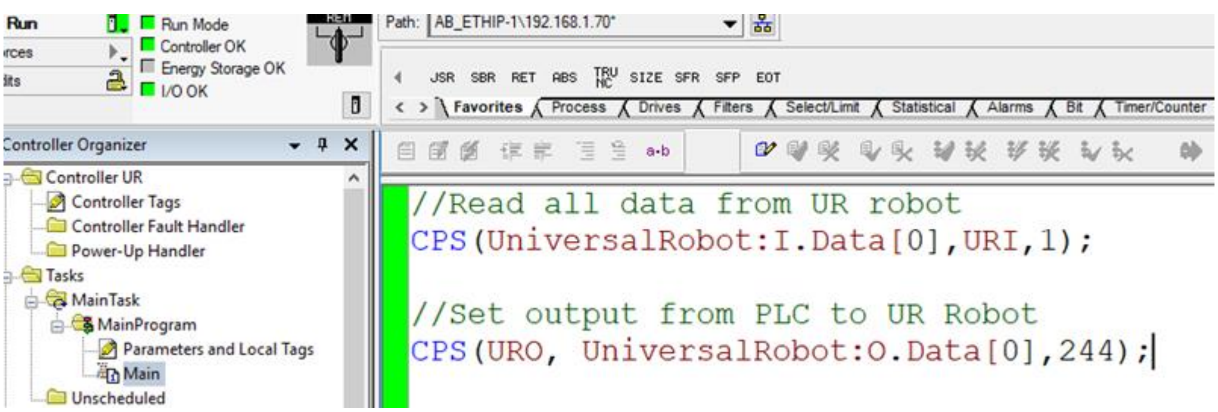
10.2. Then setup CPS between the robot and the PLC

Ladder:

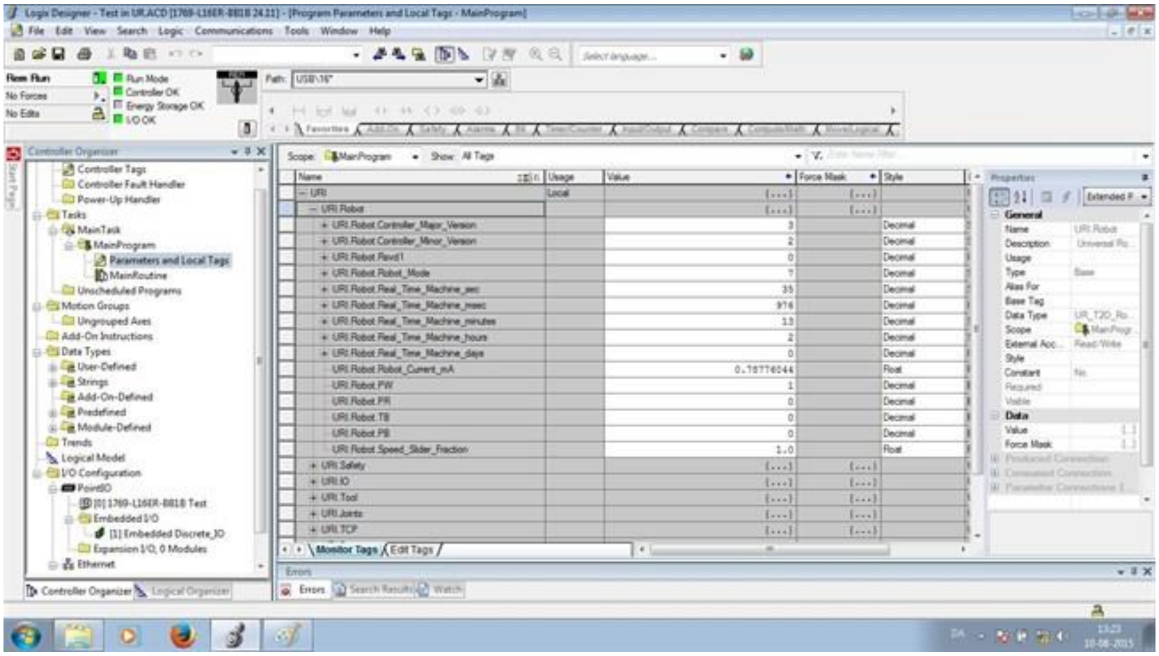




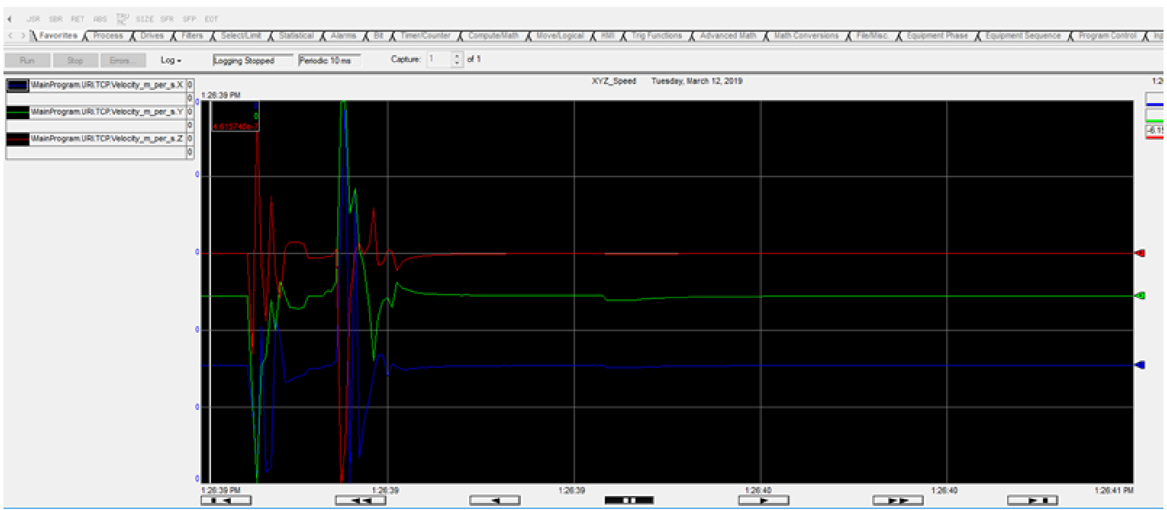
ST:



11. PLC: Navigate to URI -> URI.Robot inspect values sent from the robot



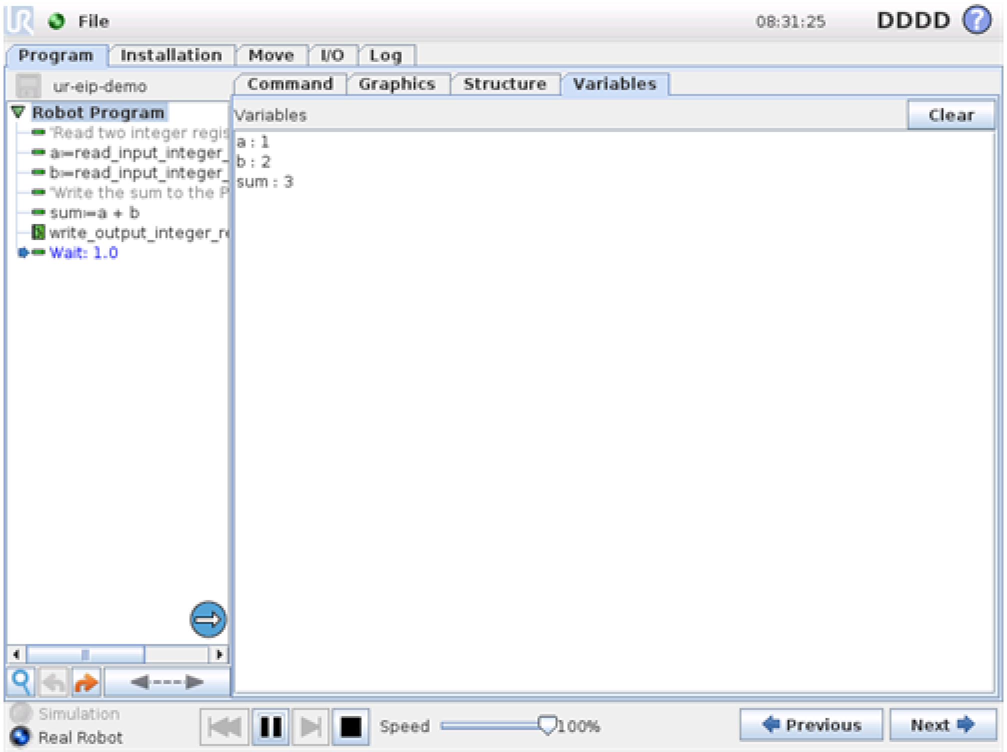
11.1. Example to test - Trend Velocity of XYZ



Demo

This demo is provided to show a basic example of how to communicate between the PLC and Robot. The demo allows the user to set two numbers in PLC output registers. The robot retrieves the numbers, calculate their sum and stores the result in a PLC input register.

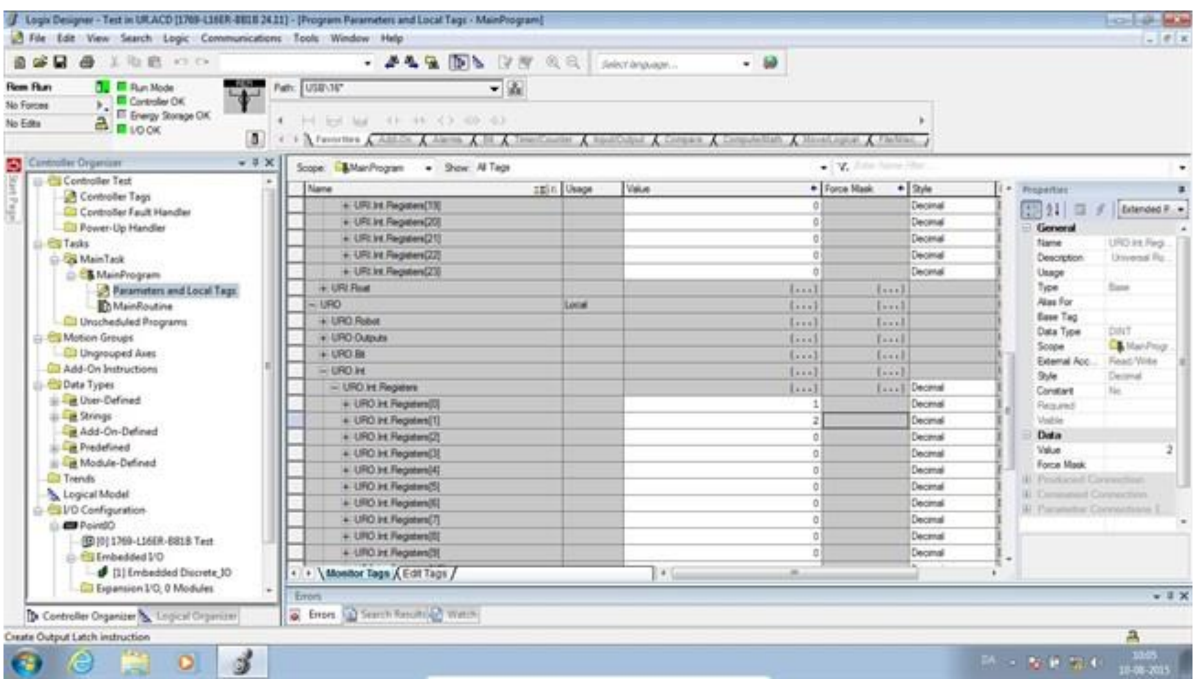
1. ROBOT: Open ur-eip-demo.urp



2. ROBOT: Run program

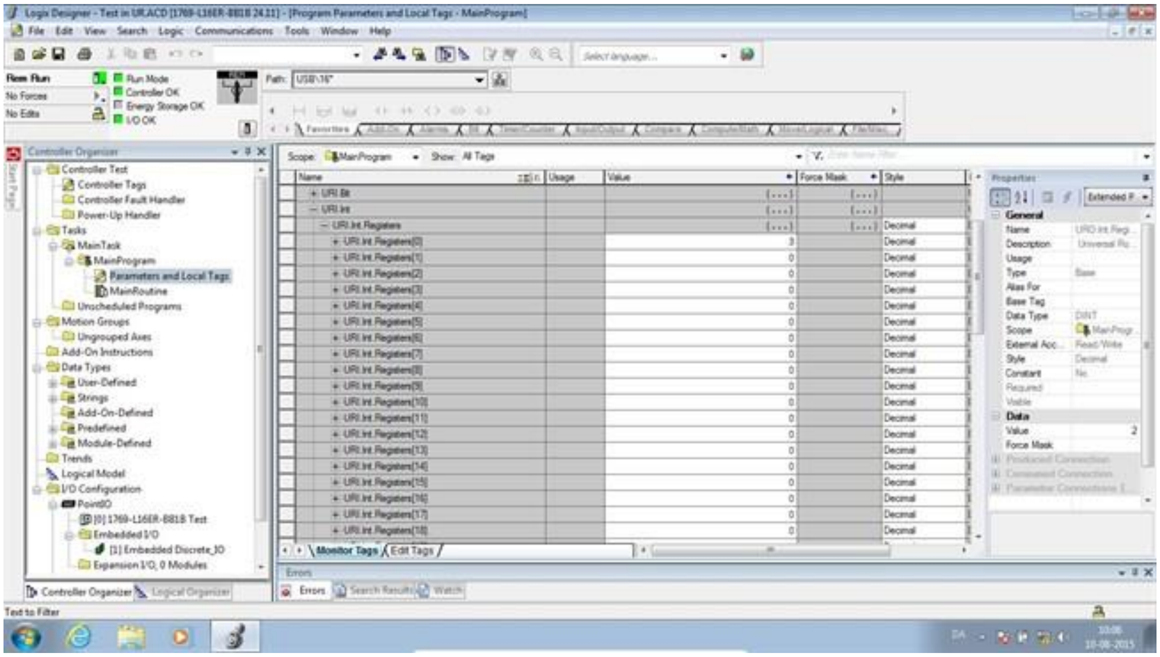
3. ROBOT: Navigate to variables tab

4. PLC: Set integer registers URO.Int.Registers[0] and URO.Int.Registers[1] to some values



5. ROBOT: Go to the Variables tab and confirm that the numbers are received

6. PLC: Validate that the URI.Int.Registers[0] contains the sum



Attached files

EIP_conveyor_example.ACD

UR.ACD

UR_DataTypes.L5X

UniversalRobot.eds

eip-iomessage.pdf

ur-eip-conveyor-tracking.urp

ur-eip-demo.urp

Fanuc CRX E Robots:

CRX E-Learning (Co-Bot) Training on the Fanuc America website.

PDF documents in CRX Folder:

02a Open and Run a Program Tutorial - Fanuc CRX 10IA Collaborative Robot: Open and Run Program

03a Creating a Program - Fanuc CRX 10IA Collaborative Robot: Creating a Program

By: Matthew Jourden - Brighton High School

Cognex Communication

Examples are at the end of Chapter 16 – See Lab 16.5.

From RealPars:

Introduction to Machine Vision for Controls Engineers

https://www.youtube.com/watch?v=emPWRQshsXQ&list=PLln3BHg93SQ9SEN8jXvhycxAeFcmkjTNs&index=155

How to Use AI in Industrial Automation: Machine Vision
https://www.youtube.com/watch?v=NGPwaExODXQ