APPLICATION NOTE

AC500 V3 ETHERNET/IP

AC500 AS SCANNER CONNECTED TO A DRIVE

Contents

1 Introduction

1.1 Scope of the document

AC500 V3 PLCs support Ethernet/IP communication. This document describes how to configure the AC500 PLC as scanner or adapter for Ethernet/IP communication.

1.2 Compatibility

For this application the following modules and engineering system versions were used. It should also work with other versions, nevertheless some small adaptations may be necessary for future versions.

- AC500 V3 PLC (PM56xx-2ETH)
- AC500-eCo V3 PLC (PM5032-x-ETH, PM5052-x-ETH and PM5072-x-2ETH(W))
- Automation Builder 2.5.0 or newer

1.3 Overview

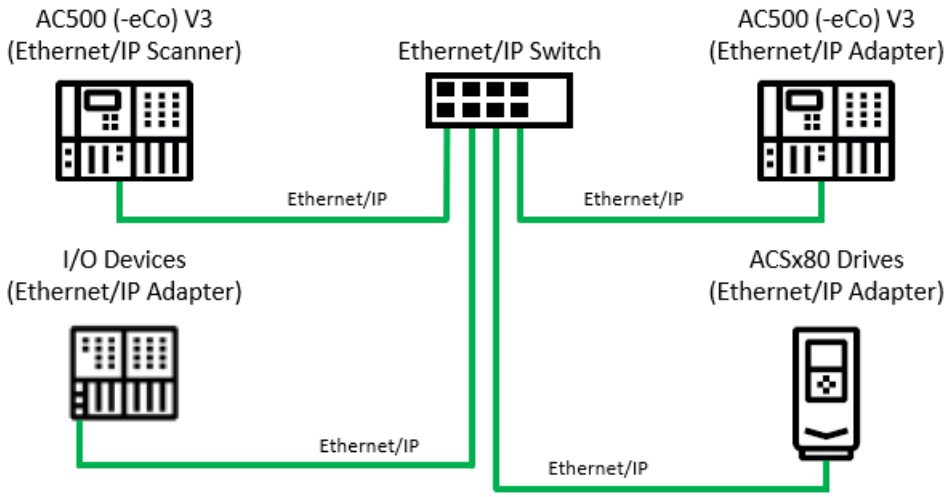
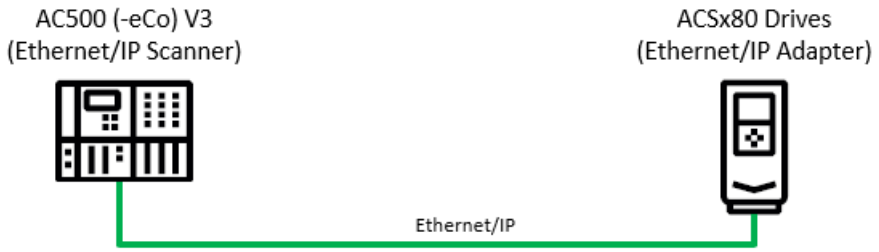


Fig. 1: AC500 V3 as Ethernet/IP scanner and Ethernet/IP adapter protocol

2 Connect to ACSx80 with FENA-x1/FEIP-xx

This chapter shows how to configure AC500(-eCo) V3 PLC as Ethernet/IP scanner and connect to the ABB ACSx80 drive with the FENA-x1 or FEIP-21 as Ethernet/IP adapter module.



2.1 ACSx80 with FENA-x1/FEIP-xx Configuration

CAUTION!

Obey the safety instructions given in this manual and the drive documentation.

After the FENA-21 adapter module has been mechanically and electrically installed according to the FENA-01/-11/-21 Ethernet adapter module user's manual, you must prepare the drive for communication with the module.

The example below shows how to configure a speed control application that uses the ODVA AC/DC drive profile, extended speed control assembly. In addition, some application-specific data is added to the communication.

The table below gives the recommended drive parameter settings on this example.

Drive parameter	Setting for ACSx80 drives	Description
50.01 FBA A enable	Enable	Enables communication between the drive and the fieldbus adapter module.
50.02 FBA A comm loss func	Fault	Enables fieldbus A communication fault monitoring.
50.03 FBA A comm loss t out	3.0 s	Defines the fieldbus A communication break supervision time.
50.04 FBA A ref1 type	Speed	Selects the fieldbus A reference 1 type and scaling.
51.01 FBA A type	ETHERNET	Displays the type of the fieldbus adapter module.
51.02 Protocol/Profile	EIP AC/DC	Selects the Ethernet/IP protocol and the ODVA AC/DC drive profile.
51.03 Commrate	Auto	Ethernet communication rate is negotiated automatically by the device.
51.04 IP configuration	Static IP	Configuration will be obtained from configuration parameters 05…13.

Drive parameter	Setting for ACSx80 drives	Description
51.05 IP address 1	192	First part of the IP address
51.06 IP address 2	168	Second part of the IP address
51.06 IP address 3	0	Third part of the IP address
51.06 IP address 4	12	last part of the IP address
51.09 Subnet CIDR	24	Sets the network mask as 255.255.255.0, allowing access only to the last subnet.
51.23 ODVA speed scale	128	Sets the scaling for the ODVA speed reference.
52.01 FBA data in1	01.07[F]	Motor current
52.03 FBA data in3	01.10[F]	Motor torque
52.05 FBA data in5	01.11[F]	DC voltage
52.07 FBA data in7	05.11[F]	Inverter temperature
52.09 FBA data in9	04.01[16]	Tripping fault
52.10 FBA data in10	04.11[16]	Latest fault
53.01 FBA data out1	23.12[F]	Acceleration time 1
53.03 FBA data out3	23.13[F]	Deceleration time 1
53.05 FBA data out5	22.26[F]	Constant speed 1
53.07 FBA data out7	22.27[F]	Constant speed 2
53.09 FBA data out9	22.28[F]	Constant speed 3
20.01 Ext1 commands	Fieldbus A	Selects the fieldbus A interface as the source of the start and stop commands for external control location 1.
22.11 Speed ref1 source	FB A ref1	Selects the fieldbus A reference 1 as the source for speed reference 1.
51.27 FBA A par refresh	Refresh	Validates any changed adapter module configuration parameter settings.
96.07 Parameter save manually	Save	Saves the valid parameter values to permanent memory. This parameter should be used to store values sent from a fieldbus, or when using an external power supply to the control board as the supply might have a very short hold-up time when powered off.

Note: The parameter group 52 “FBA Data In” and group 53 “FBA Data Out” can be mapped to other drive parameters with the format of 16bit or Float (32bit).

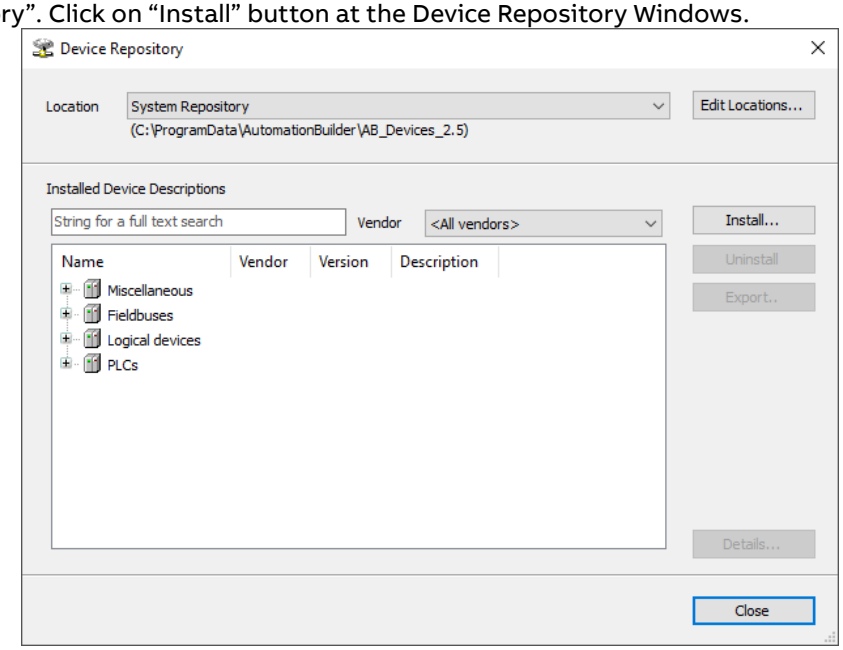
2.2 EDS installation

After the ACSx80 drive with FENA-x1/FEIP-21 setup is completed, the user needs to install the EDS into Automation Builder via the “Device Repository”.

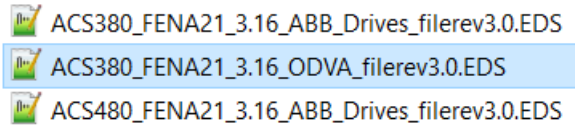
The EDS can be obtained from:

- ABB Website (https://new.abb.com/drives/connectivity/fieldbus-connectivity)
- Drive Composer Pro (Menu > Tools EDS Export)

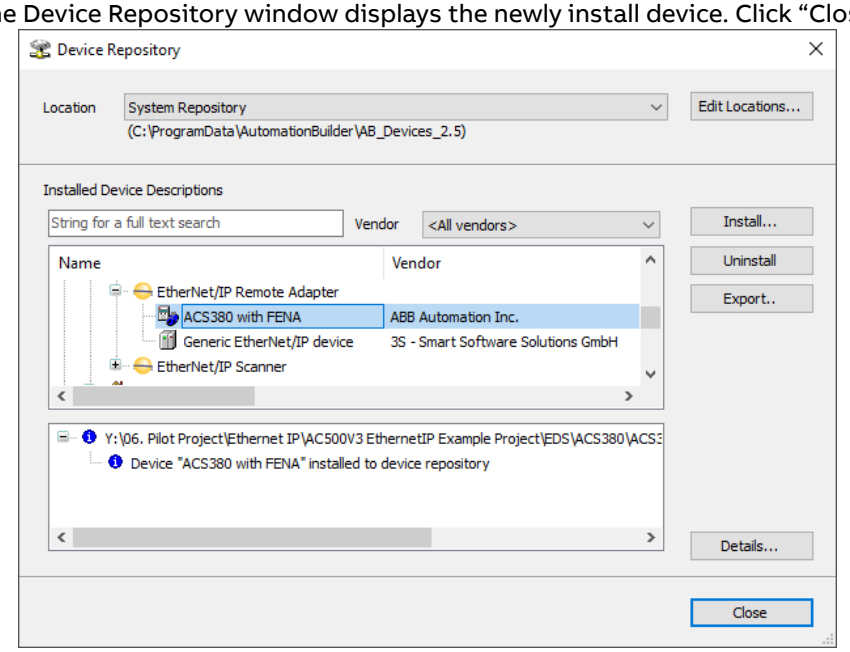
1. Extract the downloaded EDS file to a desired folder.
2. Launch Automation Builder. Go to the Automation Builder menu “Tools > Device Repository”. Click on “Install” button at the Device Repository Windows.



3. Select the EDS files based on the type of the drive and the profile.
   In chapter 2.1, the parameter “51.02 Protocol/Profile” is set to ODVA. Therefore, select the “ACS380_FENA21_3.16_ODVA_filerev3.0.EDS” and click “Open” to continue.

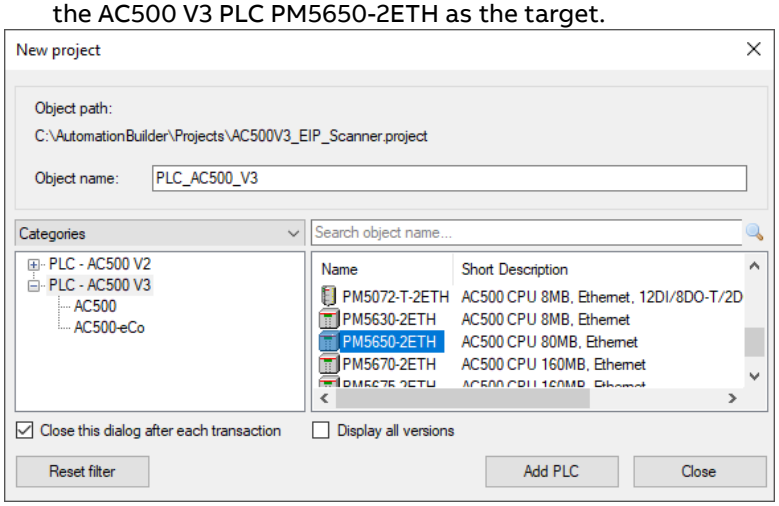


4. The Device Repository window displays the newly install device. Click “Close” to continue.

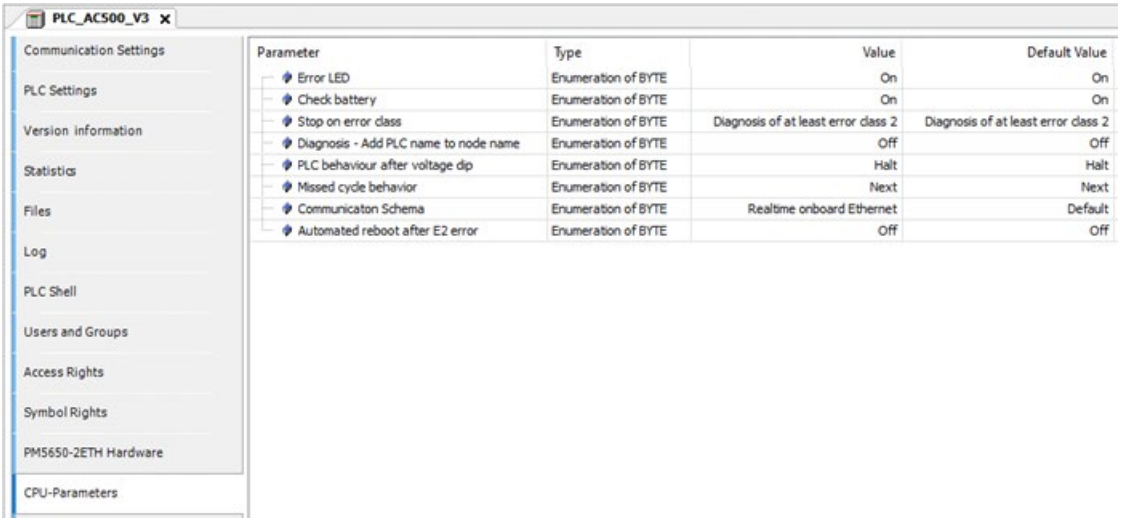


2.3 Create AC500(-eCo) V3 Ethernet/IP scanner

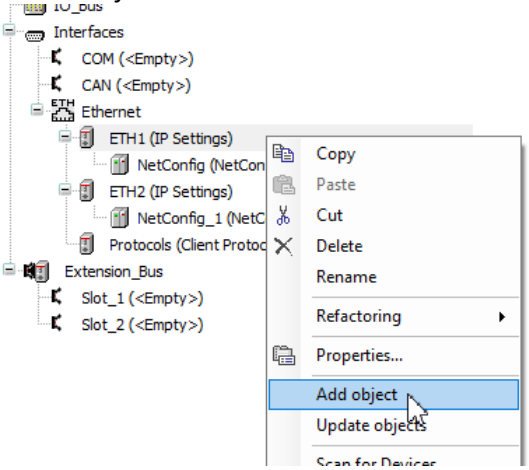
1. Launch Automation Builder and create a new project “AC500V3_EIP_Scanner.project” with the AC500 V3 PLC PM5650-2ETH as the target.



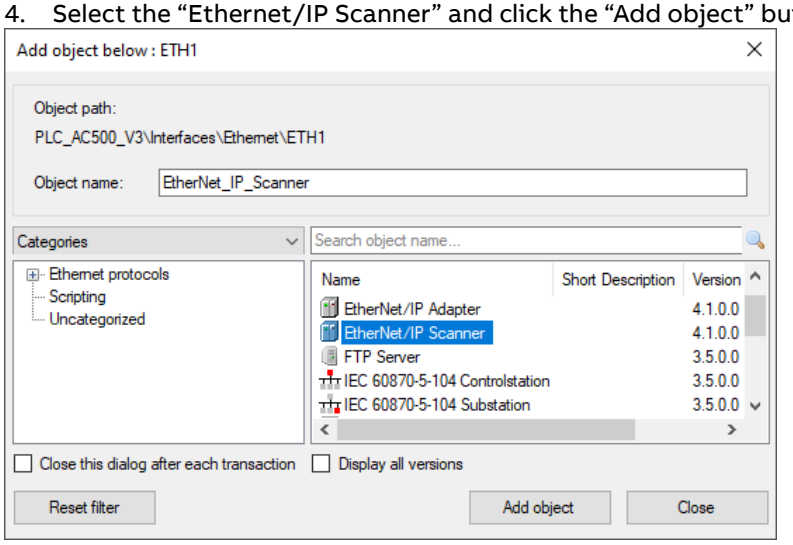
2. In the device tree, double-click on the CPU. Change to the CPU Parameters tab and change the “Communication Schema” to “Realtime onboard Ethernet”. Currently, this selection only available in AC500 V3 controller and it provides high priority to the Ethernet/IP communication.



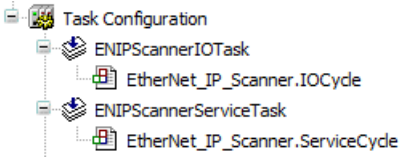
3. In the device tree, right-click on “ETH1 (IP Setting)” under “Ethernet” object and select “Add object”.



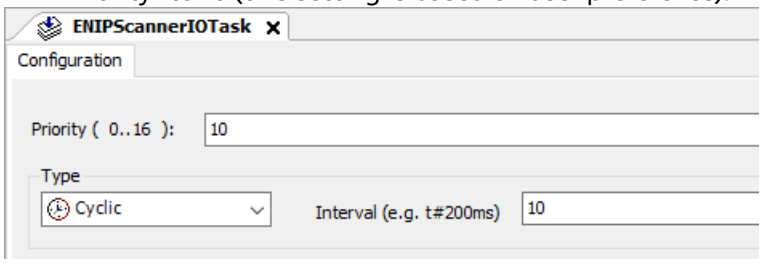
4. Select the “Ethernet/IP Scanner” and click the “Add object” button to continue.



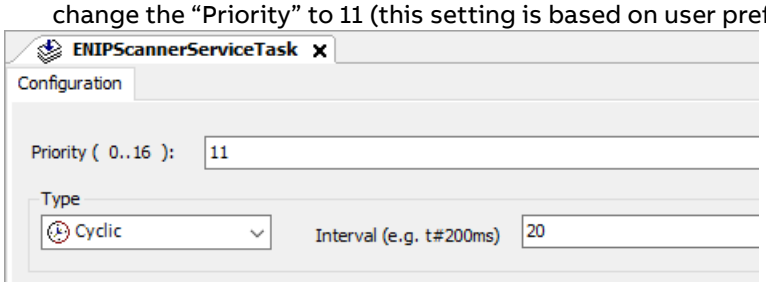
The ENIPScannerIOTask and ENIPScannerServiceTask is created automatically under “Task Configuration”.



5. Double click on the “ENIPScannerIOTask” to open the setting. In this example, change the “Priority” to 10 (this setting is based on user preference).



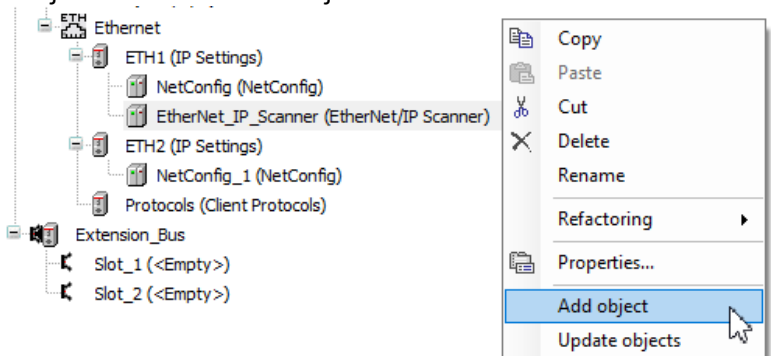
6. Double click on the “ENIPScannerServiceTask” to open the setting. In this example, change the “Priority” to 11 (this setting is based on user preference).



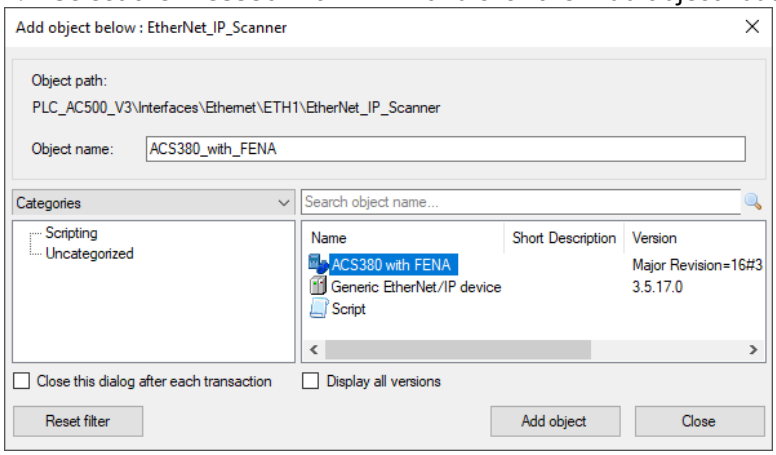
	Note: The IEC user program has 17 priorities, from 0 (highest priority) to 15 (lowest priority) runs in the real-time area. The priority 16 is the non-real-time IEC task runs in the non-real-time area.

2.4 Add ACSx80 into scanner

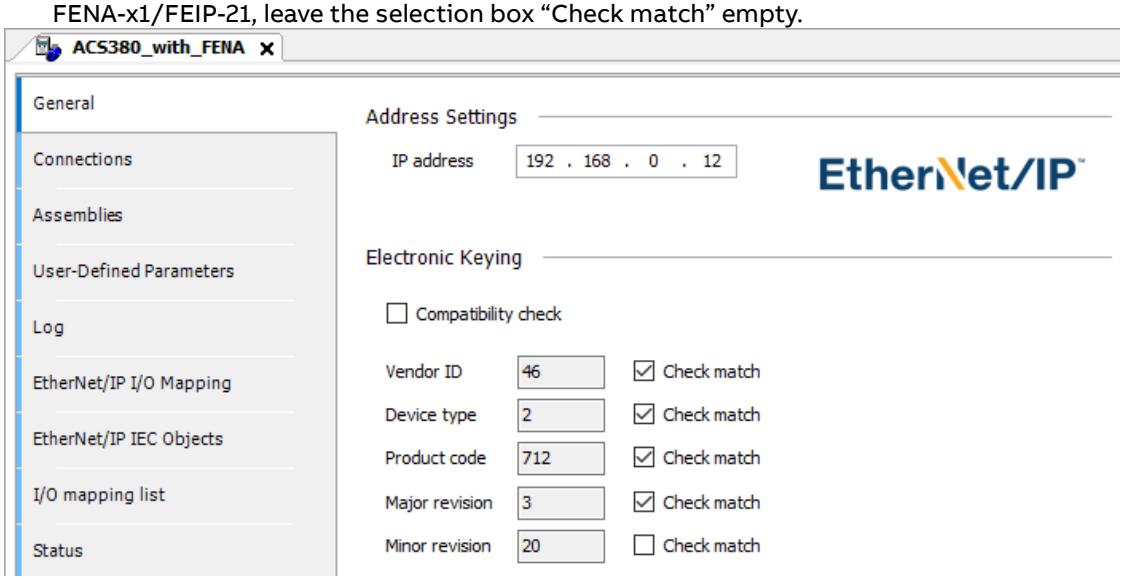
1. Right click on “EtherNet_IP_Scanner (Ethernet/IP Scanner)” under “ETH1 (IP Setting)” object and select “Add object”.



2. Select the “ACS380 with FENA” and click the “Add object” button to continue.



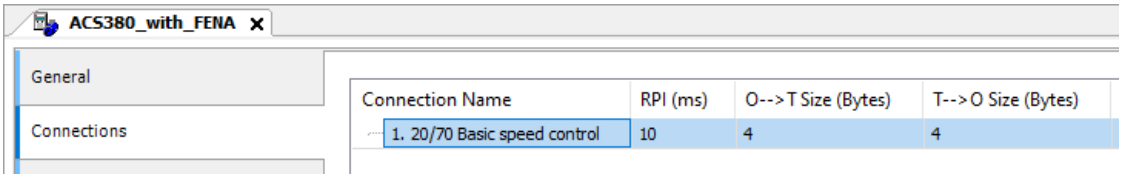
3. Double click on the “ACS380_with_FENA” to open the setting.
4. At the “General” tab, set the device IP address and cross-check the “Electronic Keying” by referencing to the FENA-x1/FEIP-21 manual. If the shown revision is different from the FENA-x1/FEIP-21, leave the selection box “Check match” empty.



2.5 Implicit messaging with ACSx80

In implicit messaging format, a control device (such as a PLC) establishes a connection with an adapter device (such as an actuator) at the outset, and the information to be exchanged is identified when the connection is established.
Implicit messages are produced and consumed at a rate - referred to as the requested packet interval (RPI) - defined by a predetermined “trigger” mechanism.

1. At the “Connections” tab, the “20/70 Basic speed control” connection is created by default.



ACSx80 Ethernet/IP implement multiple objects each with many attributes. The selection of assembly objects is limited by the choice of the communication profile. This table gives a listing of the output and input assemblies.

Name	Output instance	Input instance	Size (bytes)	Profile
Basic Speed Control	20	70	4	ODVA AC/DC drive
Enhanced Speed Control	21	71	4	ODVA AC/DC drive
Basic Speed and Torque Control	22	72	6	ODVA AC/DC drive
Enhanced Speed and Torque Control	23	73	6	ODVA AC/DC drive
Basic Speed Control plus Drive Parameters	120	170	24	ODVA AC/DC drive
Enhanced Speed Control plus Drive Parameters	121	171	24	ODVA AC/DC drive
Basic Speed and Torque Control plus Drive Parameters	122	172	26	ODVA AC/DC drive
Enhanced Speed and Torque Control plus Drive Parameters	123	173	26	ODVA AC/DC drive
ABB Drives Profile w/ Set Speed	1	51	4	ABB Drives profile
ABB Drives Profile w/ Set Speed and Set Torque	2	52	6	ABB Drives profile
ABB Drives Profile w/ Set Speed plus Drive Parameters	101	151	24	ABB Drives profile
ABB Drives Profile w/ Set Speed and Set Torque plus Drive Parameters	102	152	26	ABB Drives profile
Transparent16 w/One	11	61	4	Transparent16 profile
Transparent16 w/Two	12	62	6	Transparent16 profile
Transparent16 w/One plus Drive Parameters	111	161	24	Transparent16 profile
Transparent16 w/Two plus Drive Parameters	112	162	26	Transparent16 profile
Transparent32 w/One	21	71	8	Transparent32 profile
Transparent32 w/Two	22	72	12	Transparent32 profile
Transparent32 w/One plus Drive Parameters	121	171	28	Transparent32 profile
Transparent32 w/Two plus Drive Parameters	122	172	32	Transparent32 profile

In chapter 2.1, the ACSx80 Drive have setup the parameter group 52 FBA Data In and group 53 FBA Data Out.

Drive parameter	Setting for ACSx80 drives	Description
52.01 FBA data in1	01.07[F]	Motor current
52.03 FBA data in3	01.10[F]	Motor torque
52.05 FBA data in5	01.11[F]	DC voltage
52.07 FBA data in7	05.11[F]	Inverter temperature
52.09 FBA data in9	04.01[16]	Tripping fault
52.10 FBA data in10	04.11[16]	Latest fault
53.01 FBA data out1	23.12[F]	Acceleration time 1
53.03 FBA data out3	23.13[F]	Deceleration time 1
53.05 FBA data out5	22.26[F]	Constant speed 1
53.05 FBA data out7	22.27[F]	Constant speed 2
53.05 FBA data out9	22.28[F]	Constant speed 3

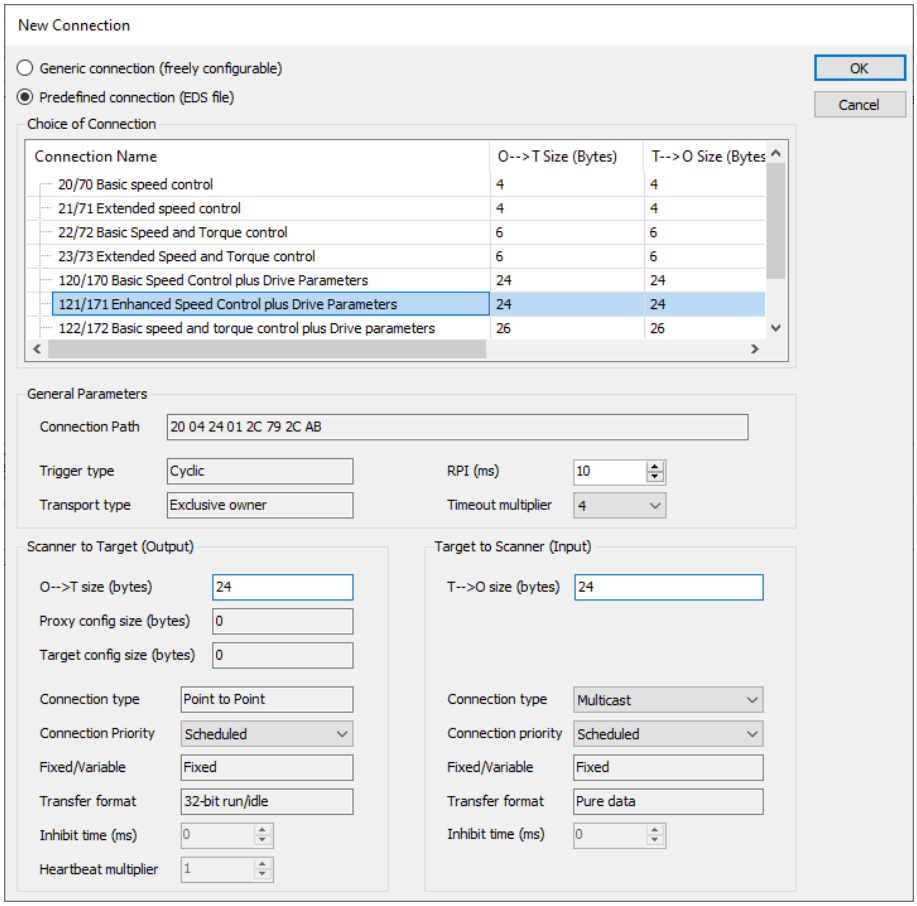
These additional parameters are used for the output instance "121" and the input instance "171". Below are the details of status word and control word for instance 121 and instance 171.

Instance 121 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0		Net Ref	Net Ctrl			Fault reset	Run Rev	Run Fwd
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	DATA OUT 1 Value (Low Byte)							
5	DATA OUT 1 Value (High Byte)							
6	DATA OUT 2 Value (Low Byte)							
7	DATA OUT 2 Value (High Byte)							
…	…							
22	DATA OUT 10 Value (Low Byte)							
23	DATA OUT 10 Value (High Byte)							

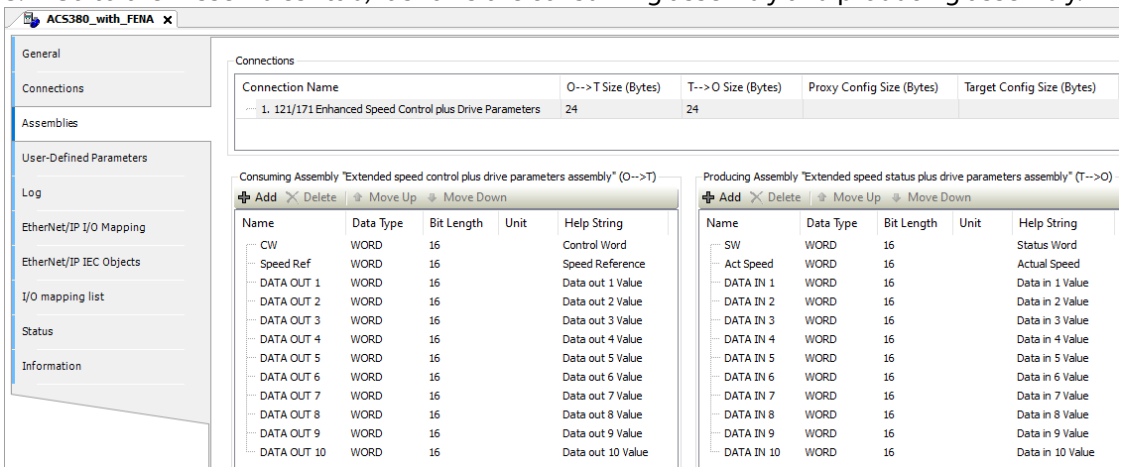
Instance 171 (ODVA AC/DC profile)								
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0	At Ref-erence	Ref From Net	Ctrl From Net	Ready	Run-ning2 (Rev)	Run-ning1 (Fwd)	Warn-ing	Faulted
1	Drive State							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	DATA IN 1 Value (Low Byte)							
5	DATA IN 1 Value (High Byte)							
6	DATA IN 2 Value (Low Byte)							
7	DATA IN 2 Value (High Byte)							
…	…							
22	DATA IN 10 Value (Low Byte)							
23	DATA IN 10 Value (High Byte)							

2. Delete the existing “20/70 Basic speed control” connection and add the “121/171 Enhanced Speed Control plus Drive Parameters” connection.

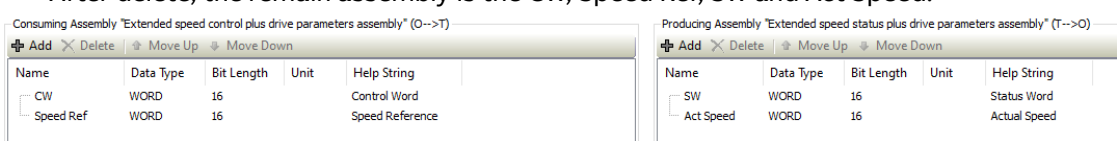
The RPI(ms) can be set based on the applications requirement.
The fastest RPI(ms) can be set is depend on the interval of the “ENIPScannerIOTask”, in this example is 10ms.



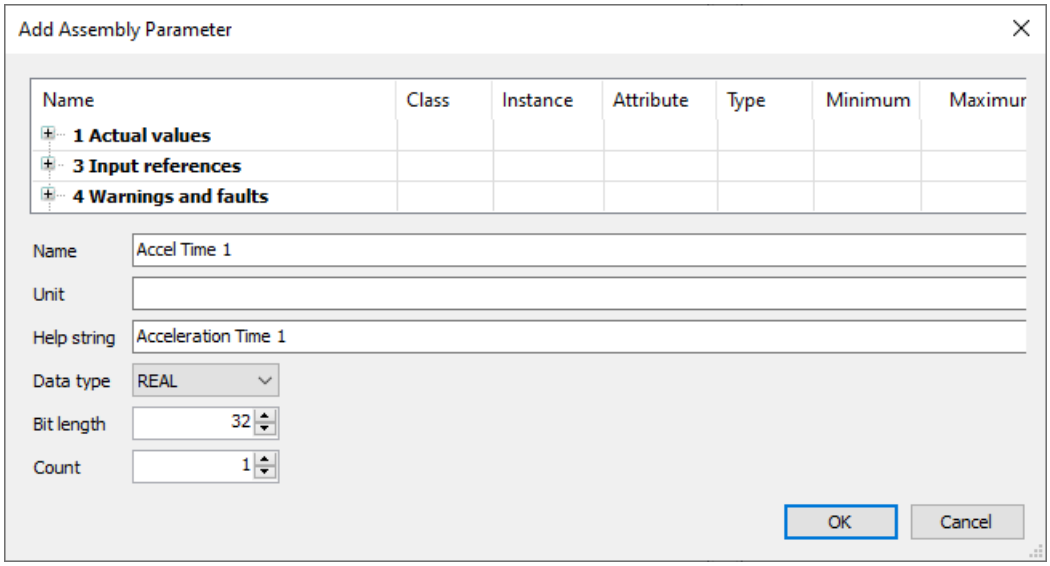
3. Go to the “Assembles” tab, it shows the consuming assembly and producing assembly.



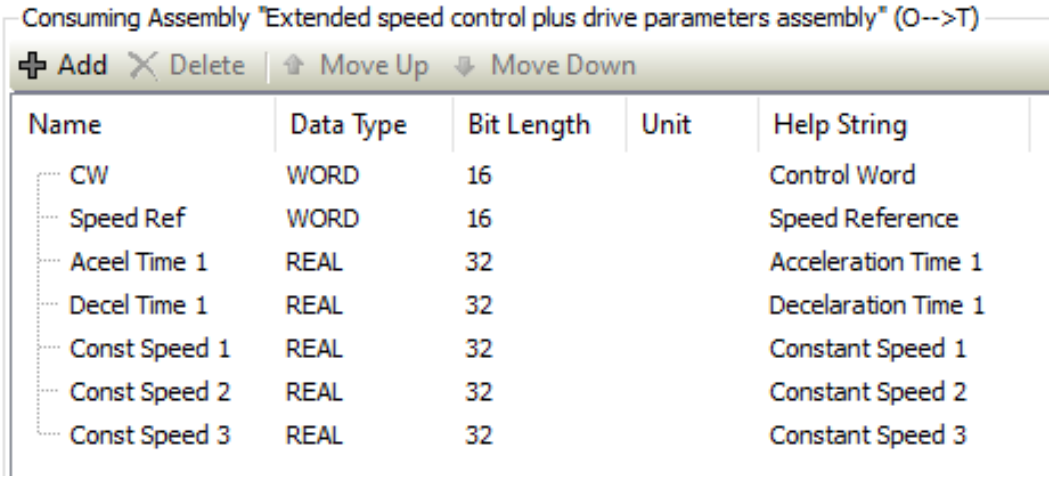
4. The default consuming and producing assembly have different data type setting compared to the configuration in the drive parameters group 52 and group 53.
   To match the drive parameters group 52 and 53 data type setting in the assemblies, delete all the “DATA OUT x” and “DATA IN x” from the assembly.
   After delete, the remain assembly is the CW, Speed Ref, SW and Act Speed.



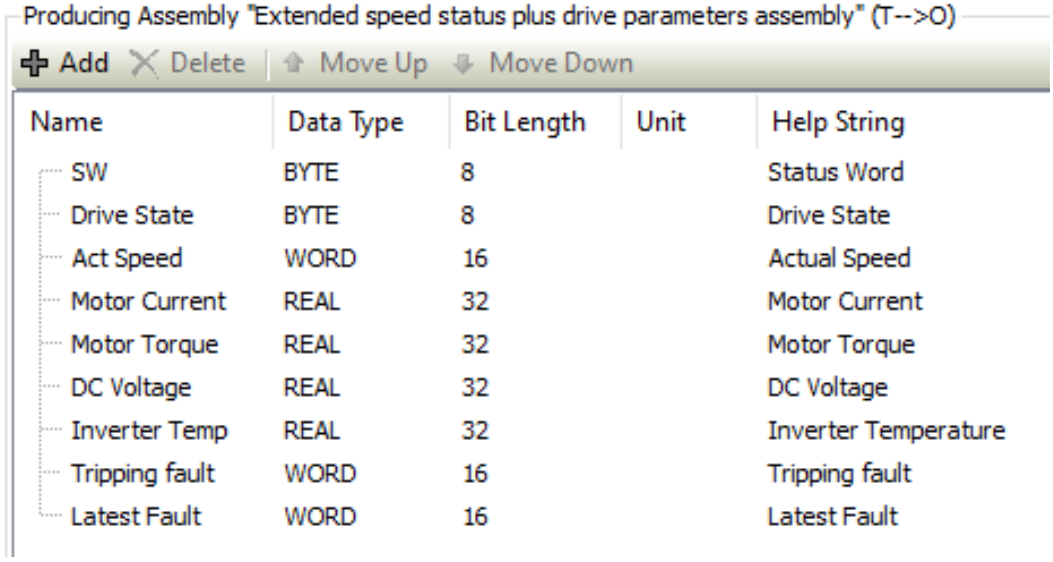
5. Click on the “Add” button at the Consuming Assembly and enter the data based on the setting in the ACSx80 drive.



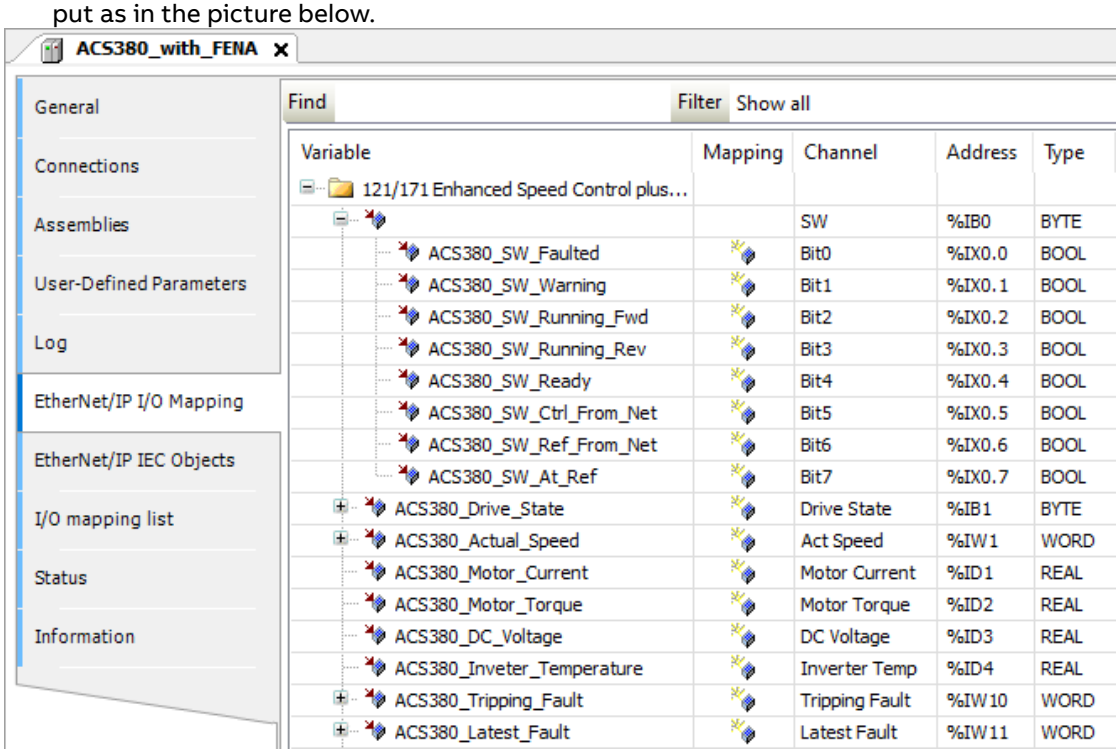
6. Repeat the previous step and add all the other data according to the picture below.



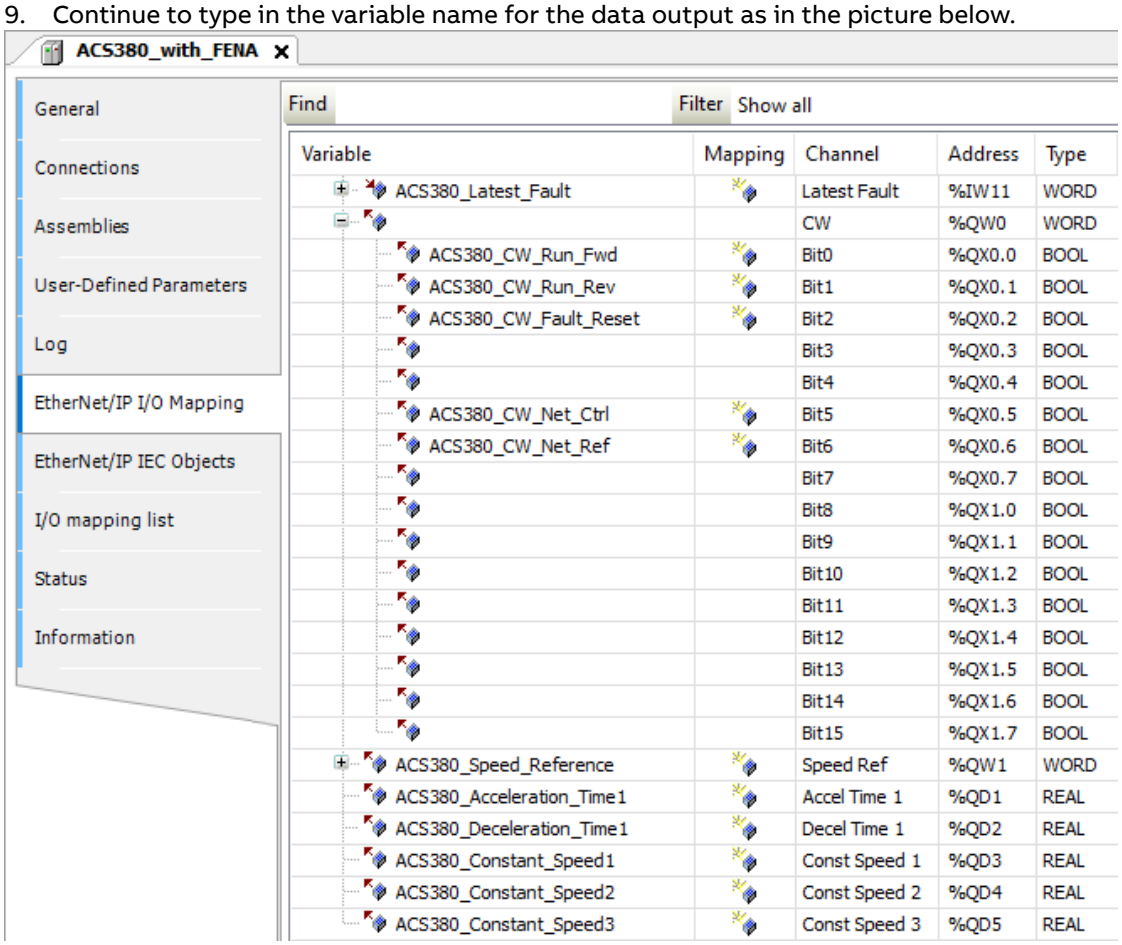
7. After that, at the Producing Assembly change the SW to BYTE and add the Drive State (BYTE) and other data as in the picture below.



8. Move on to the “Ethernet/IP I/O Mapping” tab, type in the variable name for the data input as in the picture below.



9. Continue to type in the variable name for the data output as in the picture below.



The input “Drive State” indicates the current state of the ACSx80 Drive.

State	Description
0	Vendor-specific
1	Startup
2	Not ready
3	Ready
4	Enabled
5	Stopping
6	Fault stop
7	Faulted

To assert run and stop commands of the ACSx380 with the ODVA AC/DC drive profile, please refer to the following Run/Stop event matrix.

RunFwd (Run1)	RunRev (Run2)	Trigger event	Run type
0	0	Stop	N/A
0 -> 1	0	Run	Run Forward
0	0 -> 1	Run	Run Reverse
0 -> 1	0 -> 1	No action	N/A
1	1	No action	N/A
0 -> 1	1	Run	Run Reverse
1	1 -> 0	Run	Run Forward

Note: For multiple ACSx80 drives, the I/O mapping process can be simplified by exporting the I/O mapping list to a CSV file for editing when the configuration of assembly is done.

– To export the CSV file, go to the menu “Project > Export > I/O mapping (CSV)…” and save the CSV file in the desired folder.

– After the CSV file editing is completed, go to the menu “Project > Import > I/O mapping (CSV)…”

2.6 Explicit messaging with ACSx80

Explicit messaging treats each communication between devices as a separate query and response.
Explicit messages are transmitted by TCP (Transmission Control Protocol), and because every message includes destination, source, and connection information, explicit messaging is less efficient than implicit messaging, but it offers a high degree of flexibility.

AC500(-eCo) V3 provide the following function blocks to query/set the attribute of a certain instance of a Common Industrial Protocol (CIP) object.

- Get_Attribute_Single
- Get_Attribute_All
- Set_Attribute_Single
- Set_Attribute_All

To access the ACSx80 Drive Parameter Object, the Class is always 144 (90h). Instance and attribute correspond to the drive parameter group and index in the following way:

- Instance = Parameter group (0…99) (ACx880/580: 0...255)
- Attribute = Parameter index (01…99) (ACx880/580: 0...255)

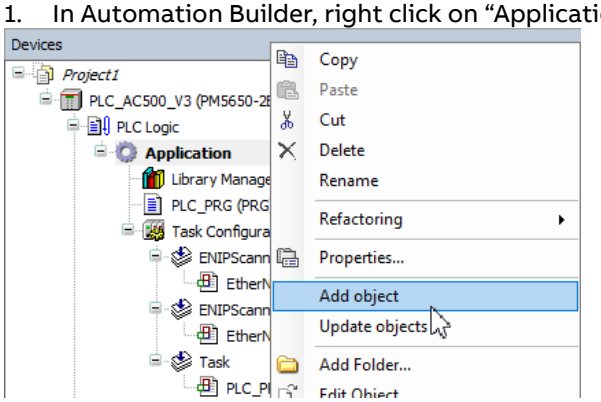
In this example, we will read the drive parameter 05.3 (Hours Run).

- Class = 144 = 16#90
- Instance = 05 = 16#05
- Attribute = 3 = 16#03

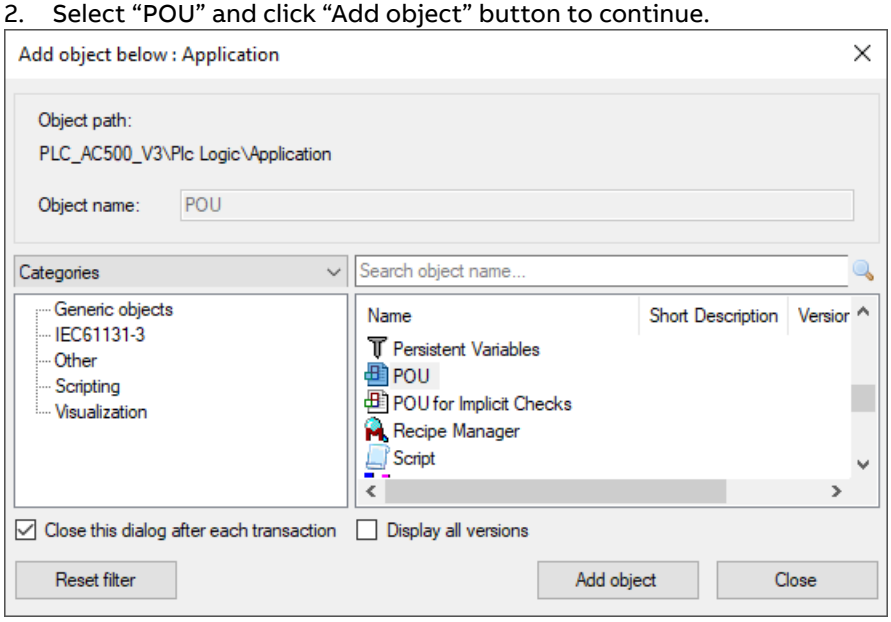
Then write to the drive parameter 22.29 (Constant Speed 4).

- Class = 144 = 16#90
- Instance = 22 = 16#16
- Attribute = 29 = 16#1D

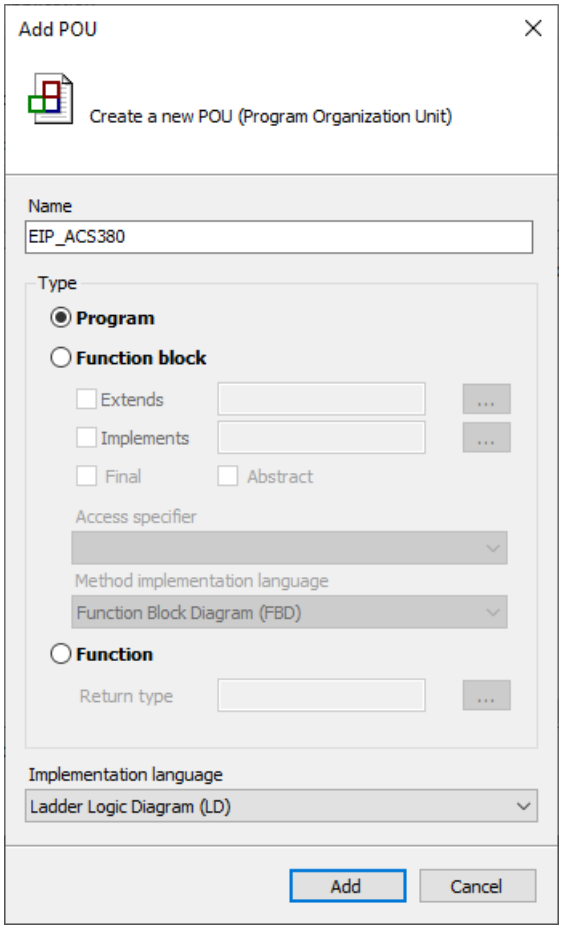
1. In Automation Builder, right click on “Application” object and select “Add object”.



2. Select “POU” and click “Add object” button to continue.

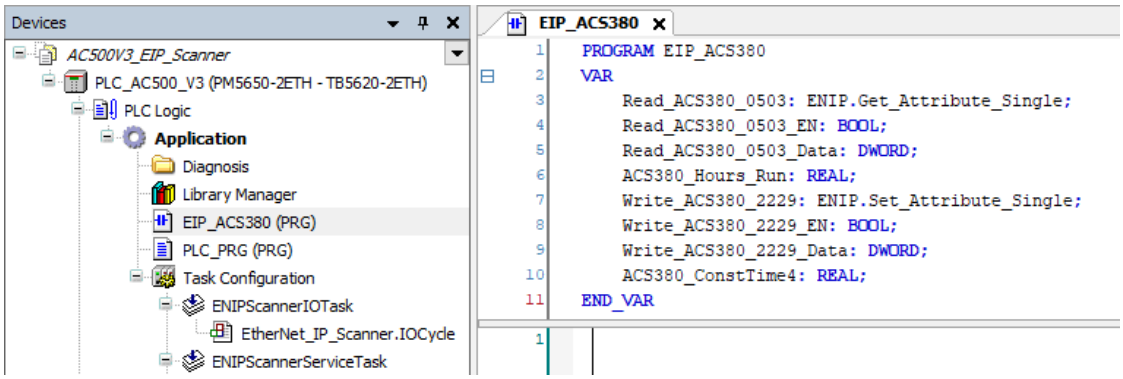


3. Name the POU as “EIP_ACS380”, select the type as “Program” and implementation language as “Ladder Logic Diagram (LD).

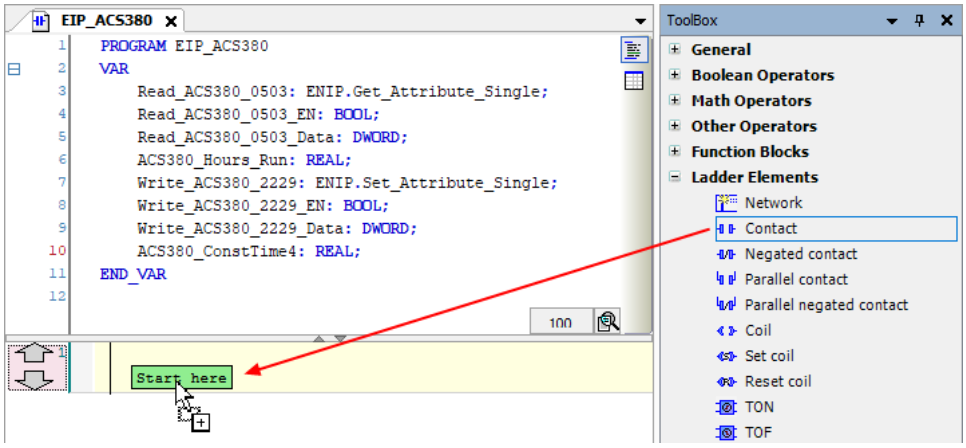


4. In the POU “EIP_ACS380”, insert the variable list as below.

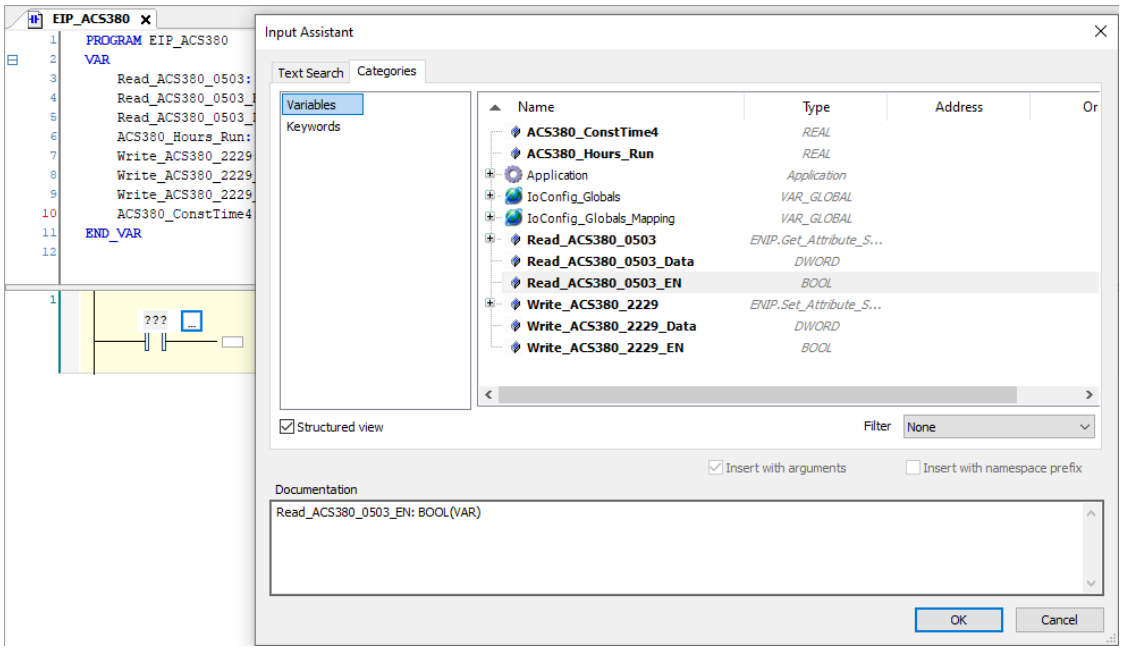
```
PROGRAM EIP_ACS380
VAR
        Read_ACS380_0503        : ENIP.Get_Attribute_Single;
        Read_ACS380_0503_EN     : BOOL;
        Read_ACS380_0503_Data   : DWORD;
        ACS380_Hours_Run        : REAL;
        Write_ACS380_2229       : ENIP.Set_Attribute_Single;
        Write_ACS380_2229_EN    : BOOL;
        Write_ACS380_2229_Data  : DWORD;
        ACS380_ConstTime4       : REAL;
END_VAR
```



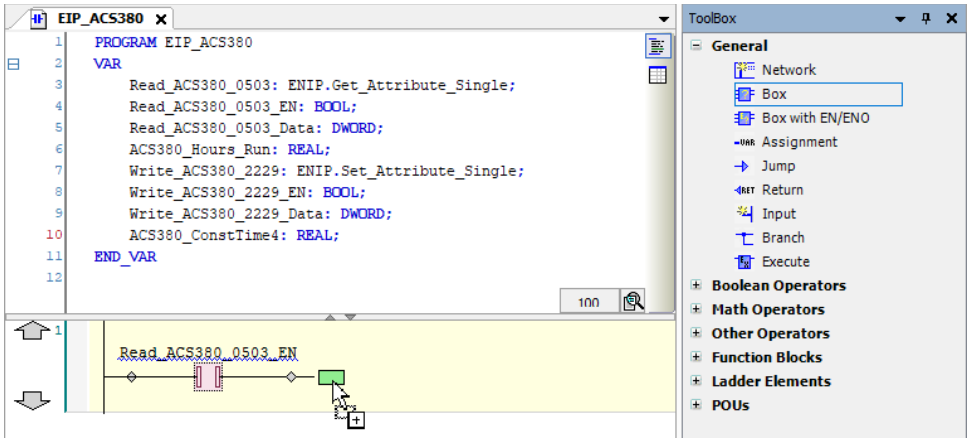
5. In the ToolBox, select “Contact” in the Ladder Elements. Drag and drop “Contact” into the "Start here" field in network "1".



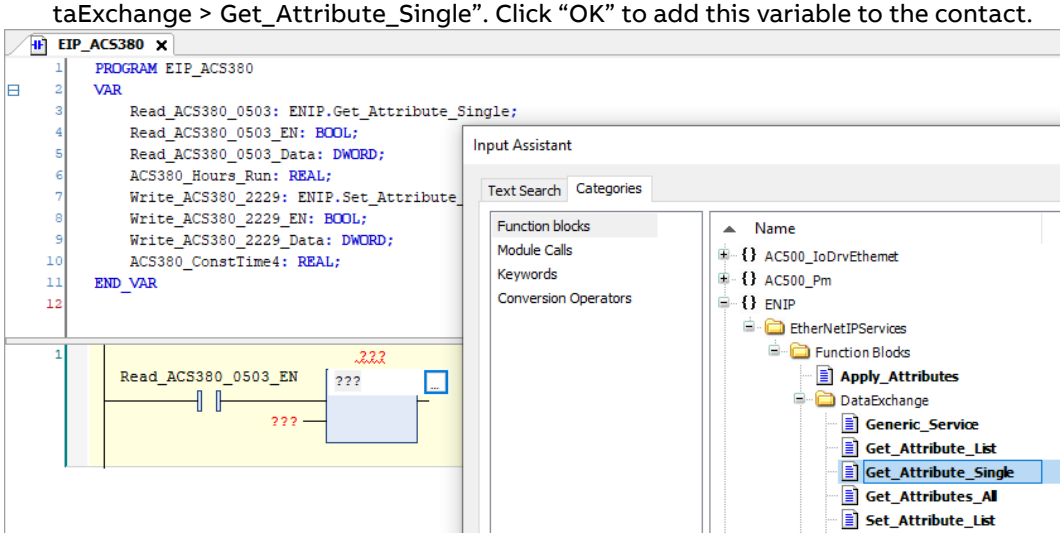
6. Select “???” on the contact, then click on “...”.
7. In “Variables” list, select “Read_ACS380_0503_EN”. Click “OK” to add this variable to the contact.



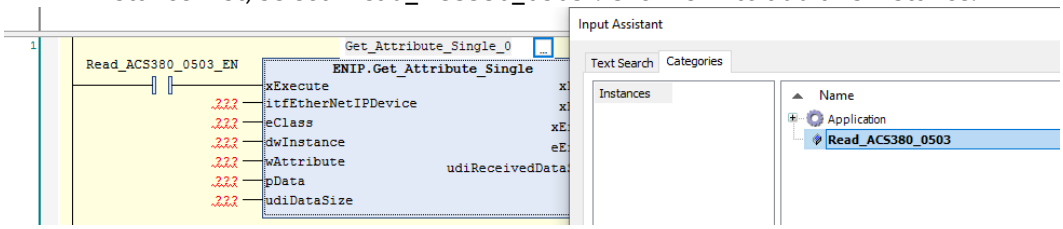
8. In the ToolBox, select “Box” under “General”. Drag and drop “Box” into the green area next to the contact.



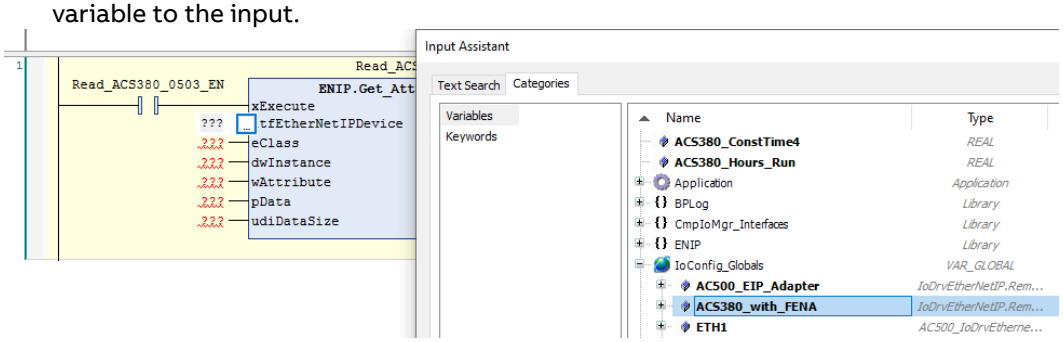
9. Select “???” inside the box, then click on “...”.
10. In “Function blocks” list, select “ENIP > EthernetIPServices > Function Blocks > DataExchange > Get_Attribute_Single”. Click “OK” to add this variable to the contact.



11. The instance of the box was automatically created, but we will not use this instance. Click on “...” beside the instance.
In “Instance” list, select “Read_ACS380_0503”. Click “OK” to add this instance.



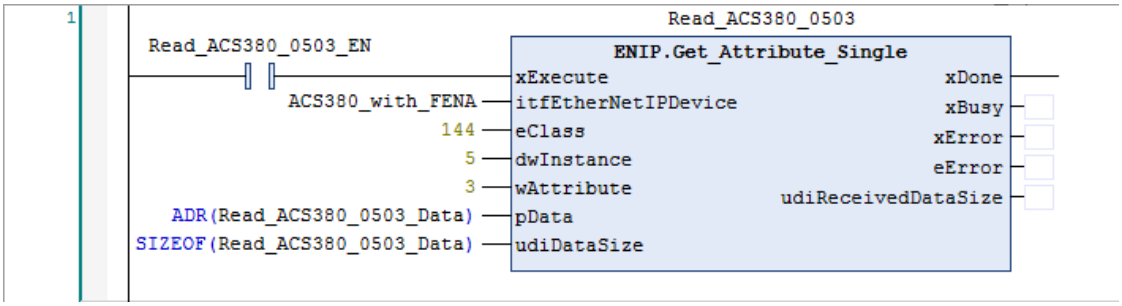
12. At the Box input “itfEthernetIPDevice”, select “???”, then click on “...”.
In “Variables” list, select “IoConfig_Globals > ACS380_with_FENA”. Click “OK” to add this variable to the input.



13. Next is to assign the eClass = 144, dwInstance = 5 and wAttribute = 3.

The Box input “pData” is the pointer to byte, the function “ADR” needs to be used to get the first byte memory location of the buffer variable assign.

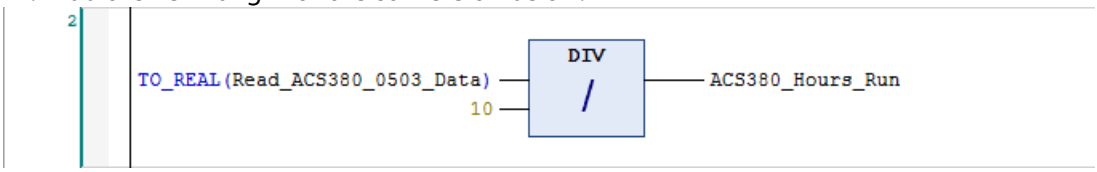
The “udiDataSize” is the size of the data received in byte, the function “SIZEOF” give size of the variable in byte.



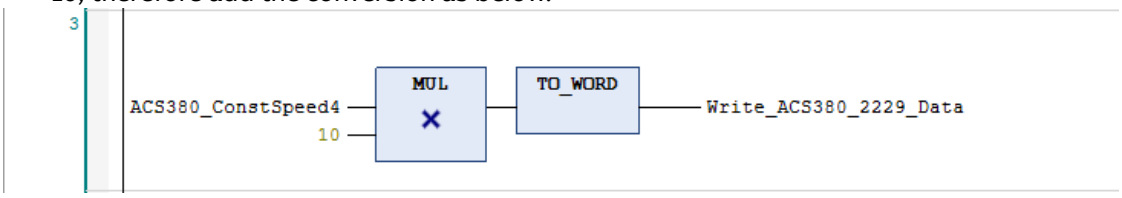
The drive parameter 05.03 Hours Run is with 1 decimal point and multiplier of 10.

05.03	*Hours run*	Corresponding parameter to *05.02 Run-time counter* in hours, that is, 24 * *05.02* value + fractional part of a day.	-
	0.0… 429496729.5 h	Hours.	10 = 1 h

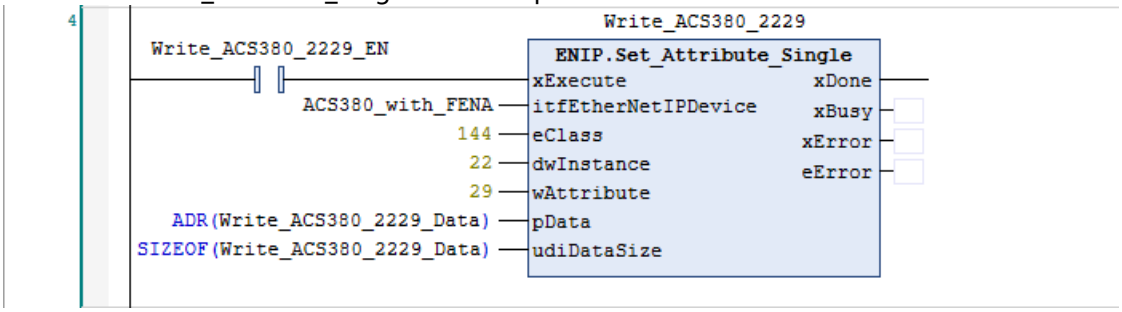
14. Add the new rung with the conversion below.



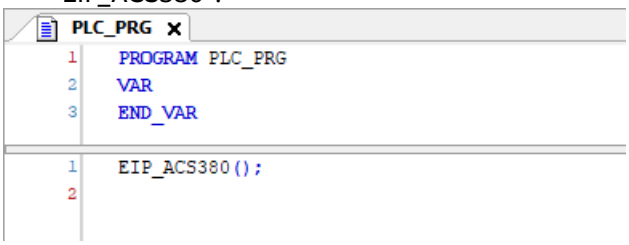
15. For the drive parameter 22.29 Constant Speed 4 is with 1 decimal point and multiplier of 10, therefore add the conversion as below.



16. Add the “Set_Attribute_Single” for drive parameter 22.29.



17. After that, open the POU “PLC_PRG” and insert the command to call the POU “EIP_ACS380”.

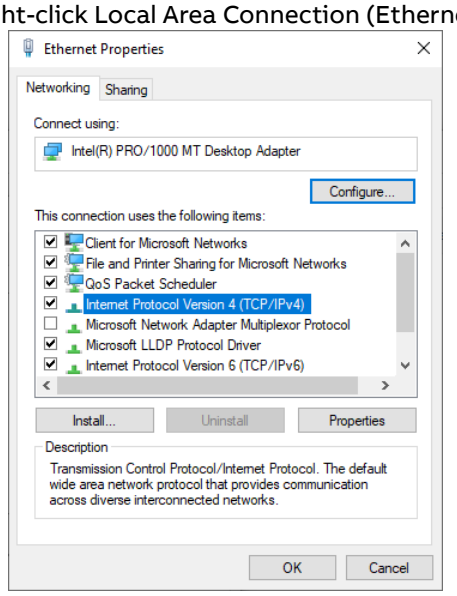


18. The setup is completed for the ACS380 with FENA-21 under the scanner. Save the project.

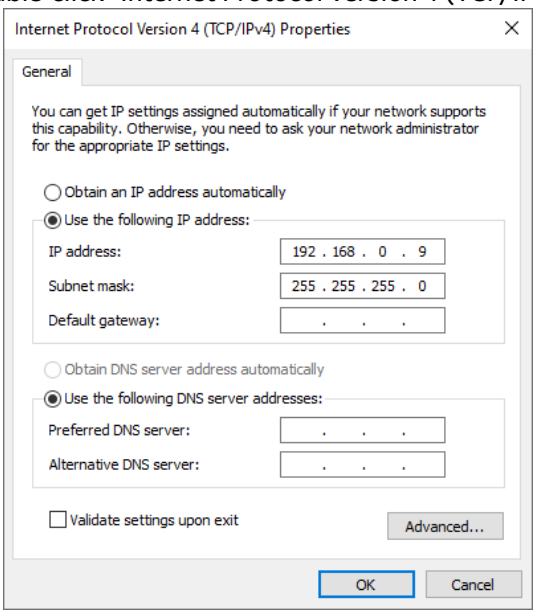
2.7 Download the project

To set-up the communication between the PC and the PLC, e.g., for downloading the compiled program, you have to set-up the communication parameters.
The IP address of your PC must be in the same class as the IP address of the CPU.
The factory setting of the IP address of the CPU is 192.168.0.10.
The IP address of your PC should be 192.168.0.X. Avoid X = 10 in order to prevent an IP conflict with the CPU.
Subnet mask should be 255.255.255.0.

1. Follow the steps below to change the PC IP address.
   a) Open Windows Control Panel. Click “Network and Internet > Network and Sharing Center”.
   b) Click Change adapter settings.
   c) Right-click Local Area Connection (Ethernet) and select Properties.



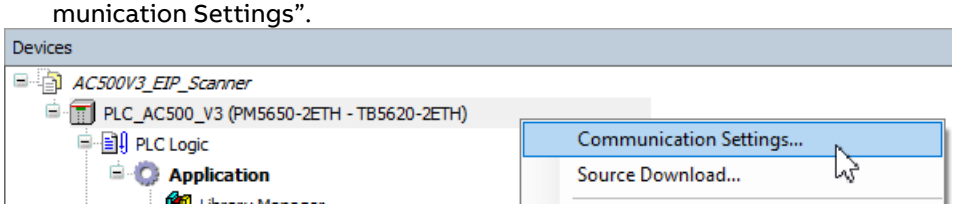
   d) Double-click “Internet Protocol Version 4 (TCP/IPv4)”.

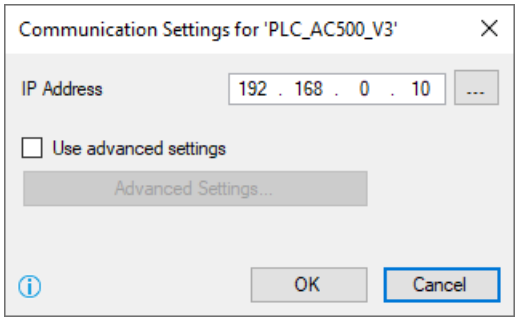


   e) Enter your desired IP address and subnet mask.
   f) Click “OK” to continue.

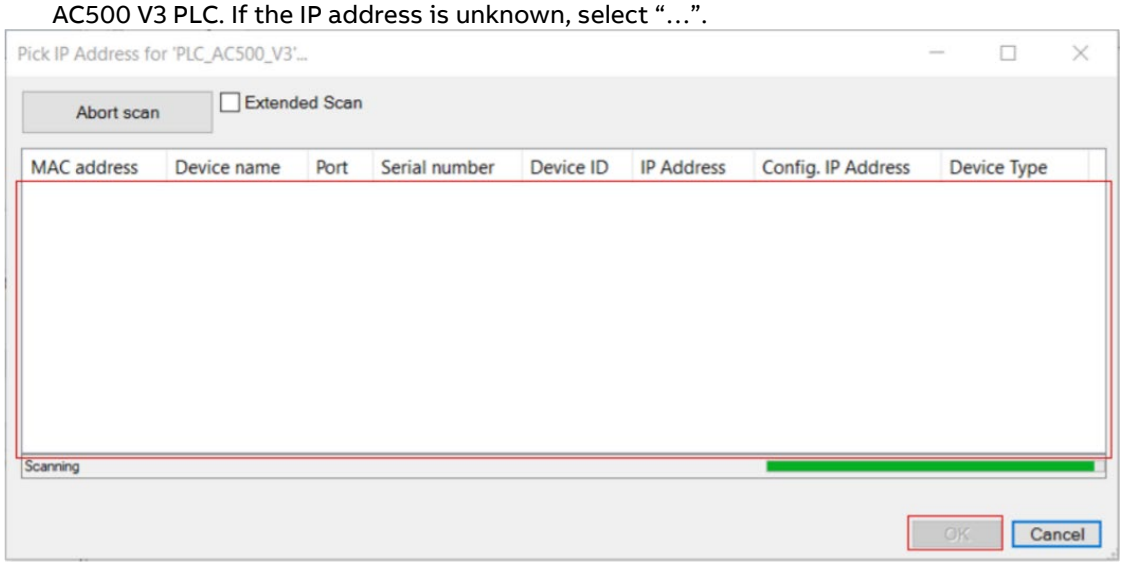
	Note: If VPN is connected, it might influence the connection to the PLC. We recommend to disconnect from VPN before connecting to the PLC locally.

5. In the Automation Builder device tree right-click “PLC_AC500_V3” and select the “Communication Settings”.





6. Keep the default value in the IP address of the CPU or type in the current IP address of the AC500 V3 PLC. If the IP address is unknown, select “…”.

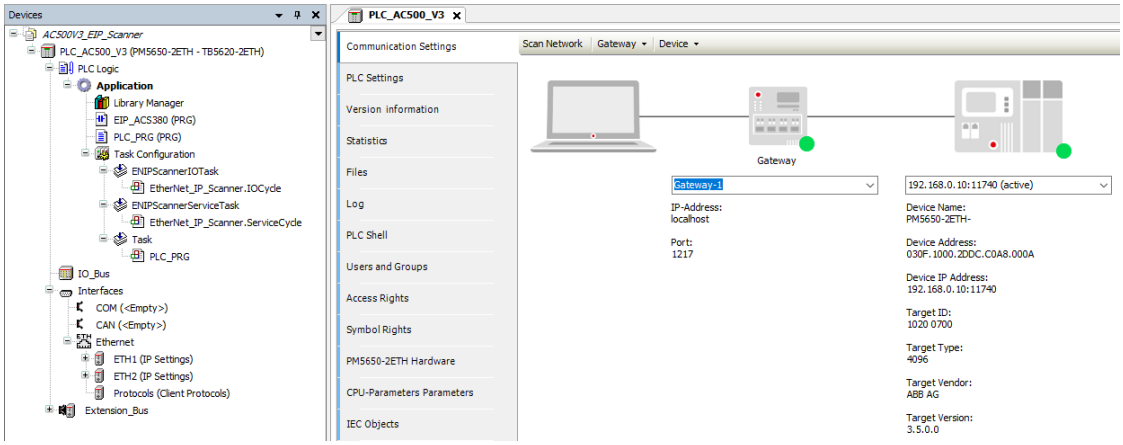


7. The automatic scan will run and the results will appear in this field.

Select the CPU in the field and select “OK” to implement the needed communications gateway.

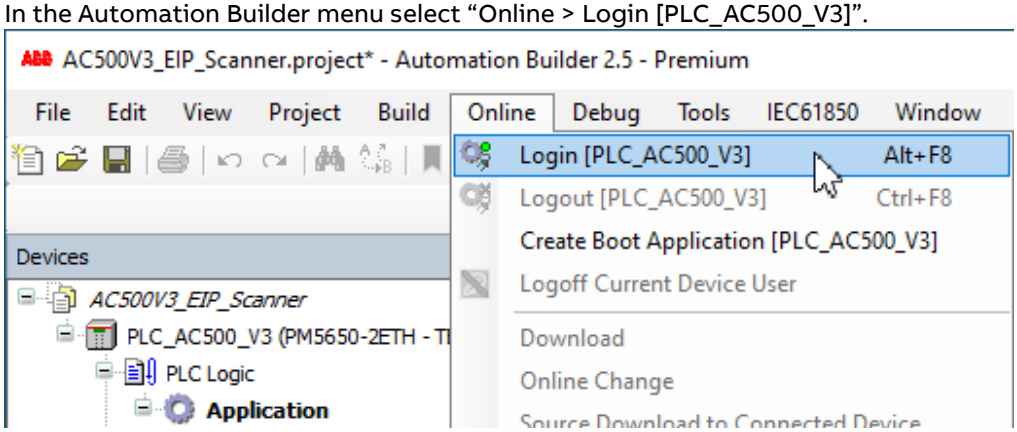
8. After that, double-click “PLC_AC500_V3” in the device tree. Select “Communication Settings”.

The selected IP address is shown. To test the connection and/or to see the CPU information press [Enter] or click on the black dot next to the PLC picture.
The black dot will turn green when communication is established.

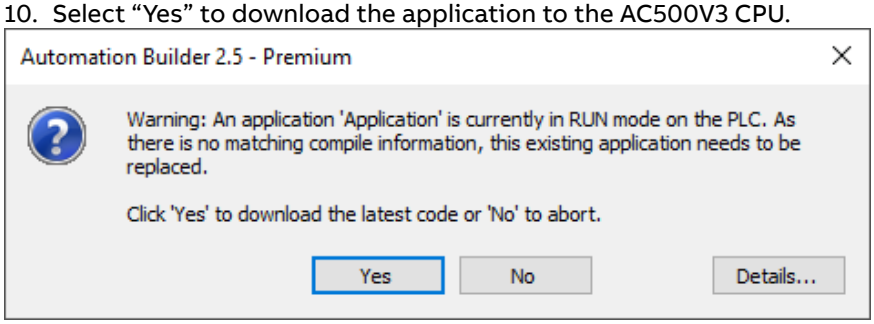


9. Logging-in to the CPU will load the project into the AC500 V3 CPU. The first log-in will also load the hardware set-up.

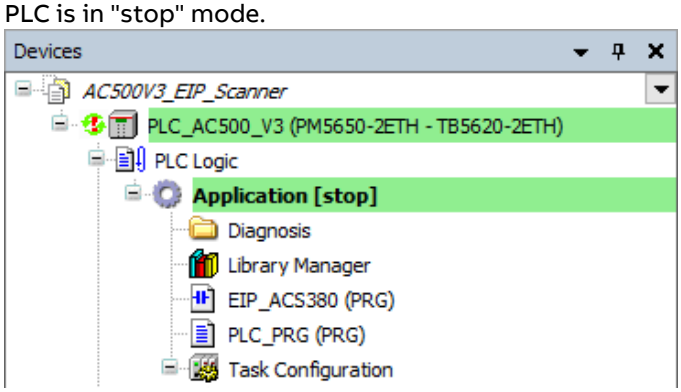
In the Automation Builder menu select “Online > Login [PLC_AC500_V3]”.



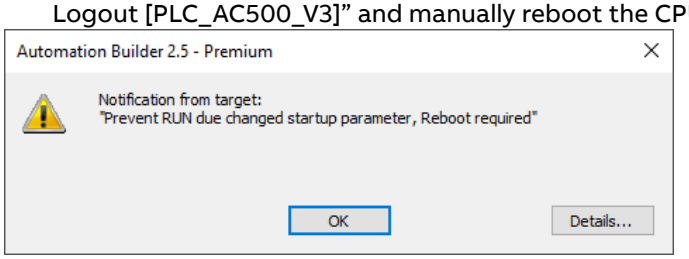
10. Select “Yes” to download the application to the AC500V3 CPU.



PLC is in "stop" mode.



11. Select menu “Debug > Start [PLC_AC500_V3]”. Alternatively, select the "start" icon in the tool bar. A pop-up appears, click “OK” to continue. After that, go to menu select “Online > Logout [PLC_AC500_V3]” and manually reboot the CPU.



12. When CPU reboot is completed, go to menu “Online > Login [PLC_AC500_V3]”.

The project is downloaded and the CPU is in RUN mode. Now you can test the application.

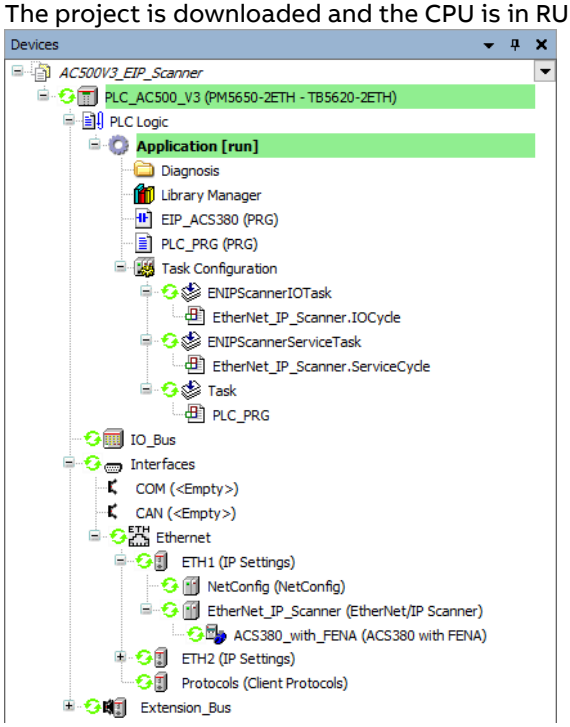


ABB AG
Eppelheimer Straße 82
69123 Heidelberg, Germany
Phone: +49 62 21 701 1444
Fax: +49 62 21 701 1382
E-Mail: plc.support@de.abb.com
www.abb.com/plc