

CMMT servo drive commissioning with MITSUBISHI IQ-R PLC through Ethernet/IP

This Application Note gives detailed procedure to integrate CMMT Servo Drive with Mitsubishi IQ-R Controller in EtherNet/IP Communication Network. Also described to control the servo motor in different control operational modes and also to Read/Write the servo motor parameter.

CMMT-AS
CMMT-ST

TitleCMMT servo drive commissioning with MITSUBISHI IQ-R PLC by ETH/IP
Version 1.4
Document no. 100346
Originalen
AuthorFesto

Last saved 12.08.2024

Copyright Notice

This documentation is the intellectual property of Festo SE & Co. KG, which also has the exclusive copyright. Any modification of the content, duplication or reprinting of this documentation as well as distribution to third parties can only be made with the express consent of Festo SE & Co. KG.

Festo SE & Co KG reserves the right to make modifications to this document in whole or in part. All brand and product names are trademarks or registered trademarks of their respective owners.

Legal Notice

Hardware, software, operating systems and drivers may only be used for the applications described and only in conjunction with components recommended by Festo SE & Co. KG.

Festo SE & Co. KG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Defects resulting from the improper handling of devices and modules are excluded from the warranty.

The data and information specified in this document should not be used for the implementation of safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect. In other respects, the regulations with regard to liability from the terms and conditions of delivery, payment and use of software of Festo SE & Co. KG, which can be found at www.festo.com and can be supplied on request, shall apply.

All data contained in this document do not represent guaranteed specifications, particularly with regard to functionality, condition or quality, in the legal sense.

The information in this document serves only as basic information for the implementation of a specific, hypothetical application and is in no way intended as a substitute for the operating instructions of the respective manufacturers and the design and testing of the respective application by the user.

The operating instructions for Festo products can be found at www.festo.com.

Users of this document (application note) must verify that all functions described here also work correctly in the application. By reading this document and adhering to the specifications contained therein, users are also solely responsible for their own application.

Table of contents

1	Introduction	6
1.1	Hardware & Software requirements	7
1.2	Topology	8
1.3	Hardware Over View of CMMT Servo Drive	9
2	Hardware Configuration for Position Control Mode	10
2.1	IP Address Configuration of CMMT Controller Parameterization Port	10
2.1.1	Configuration of CMMT Drive Parameterization Port X18 using Festo Field Device Tool	10
2.1.2	Configuration of CMMT Drive Parameterization Port X18 using Festo Automation Suite	11
2.2	CMMT-AS Servo Drive Fundamental Configuration.....	13
2.2.1	Servo Drive Selection.....	13
2.2.2	Linear Axis Configuration	16
2.2.3	Rotary Axis Configuration	28
2.2.4	CMMT Drive Parameterization for Integrate with PLC.....	30
2.3	CMMT-AS Servo Drive Fieldbus Configuration	34
2.3.1	Switch Setting for Bus Protocol	35
3	Starting a Project.....	36
3.1	Create New Project File & Configurations of IQ-R Series PLC in Gx Works 3	36
3.2	Adding Ethernet Module in Project file	38
3.3	Ethernet port parameters	42
3.3.1	IP address of PLC	42
3.3.2	IP address of Ethernet Module.....	42
3.4	Configuring RJ71EIP91 using EtherNet/IP Configuration Tool (EIP – CT) by Adding EDS File in EtherNet/IP Devices.....	43
3.5	Configuring RJ71EIP91 using EtherNet/IP Configuration Tool (EIP – CT) by Adding Generic EDS.	53
3.6	Adding Library to the project	59
3.6.1	Import Festo_CMMT_Mitsubishi_IQR_EIP_Lib in FB/Fun under program	59
4	Library Configuration.....	64
4.1	M_FBLib – RJ71EIP91 Library Configuration	65
4.1.1	Configure “M+RJ71EIP91_Class1GetInputData_00A and M+RJ71EIP91_Class1SetOutputData_00A” Input & Output Interface Tags.....	66
4.2	Festo_CMMT_Mitsubishi_IQR_EIP_Lib Library Configuration.....	70
4.2.1	Add “PTP_Drives_Festo_EIP” in PLC Program	70
	Linking Tags to Function Block74	
4.2.2	Add “PTP_Drives_Parameter_EIP” in PLC Program	82
4.2.3	Add “Tel001_Velocity_Drives_Festo_EIP” in PLC Program.....	83
4.2.4	Add “Tel102_Velocity_Drives_Festo_EIP” in PLC Program.....	84
5	Elements in the Festo_CMMT_Mitsubishi_IQR_EIP_Lib Library	85
5.1	PTP_Drives_Festo_EIP	85
5.1.1	Input and Output Interface.....	86
5.1.2	ConfigEPos Bit Description	88
5.1.3	Epos ZSW1 Bit Details.....	89
5.1.4	Epos ZSW2 Bit Details.....	89
5.1.5	Diagnostic Messages	90
5.2	PTP_Drives_Parameter_EIP	91
5.2.1	Input and Output Interface.....	91
5.2.2	wErrorID	92

5.2.3	Identification of Parameter Number, Index, Data Size during PTP_Drives_Parameter_EIP Configuration	93
5.3	Word_2_DWord	94
5.3.1	Input Interface	94
5.3.2	Output Interface.....	94
5.4	DWord_2_Word	94
5.4.1	Input Interface	94
5.4.2	Output Interface.....	94
5.5	“Tel001_Velocity_Drives_Festo_EIP” / Basic Velocity Control	95
5.5.1	“Tel001_Velocity_Drives_Festo_EIP” FB Introduction	95
5.5.2	Input and Output Interface.....	95
5.5.3	Diagnostic Messages / Function Block Faults	96
5.6	“Tel102_Velocity_Drives_Festo_EIP” / Advance Velocity Control	97
5.6.1	“Tel102_Velocity_Drives_Festo_EIP” FB Introduction.....	97
5.6.2	Input and Output Interface.....	97
5.6.3	Diagnostic Messages / Function Block Faults	98
6	CMMT Drive in Position Control Mode (PTP_Drives_Festo_EIP) FUNCTIONS	99
6.1	Axis Enabling & Stopping	99
6.2	Axis Software & Hardware Position Limits Enabling	99
6.3	Axis Actual Speed Monitoring.....	99
6.4	PtP_Drives_Festo_EIP Modes of Operation	100
6.4.1	Relative Positioning Mode [ModePos – 1]	100
6.4.2	Absolute Positioning Mode [ModePos – 2].....	101
6.4.3	Setup Mode [ModePos – 3]	101
6.4.4	Home/Reference Point using Festo Automation Suite Configuration [ModePos – 4]	102
6.4.5	Home to Current Position [ModePos – 5].....	102
6.4.6	Traversing/Record Selection Mode [ModePos – 6]	102
6.4.7	Jog Mode [ModePos – 7].....	103
6.4.8	Inching/Incremental Jog Mode [ModePos – 8].....	103
6.4.9	Procedure to set the CMMT-AS in “Travel to fixed Stop” mode with the PtP_Drive_Festo_EIP function block	103
7	Extended Process Data	106
7.1	Configuring RJ71EIP91 using EtherNet/IP Configuration Tool (EIP – CT).....	106
7.2	Set parameters in the FAS under Extended Process Data (EPD)	106
7.3	Add the EPD in PLC Logic.....	109
7.3.1	Configure the Extended Process Data - Exclusive Owner in PLC Logic by calling “M+RJ71EIP91_Class1GetInputData_00A and M+RJ71EIP91_Class1SetOutputData_00A” Input & Output Interface Tags (Refer Chapter 4.1.1 for configuring M_FBLib – RJ71EIP91)	109
7.3.2	Logic to Read and Write the EPD	110
7.3.3	Changing the Record Table Value from EPD	111
8	PTP_Drives_Parameter_EIP FUNCTIONS	114
8.1	Read Parameter.....	114
8.1.1	Example : Read parameter of the data type FLOAT.	114
8.2	Write Parameter	115
8.2.1	Example : Write parameter of the data type FLOAT.	116
9	CMMT Drive in Velocity Control Mode Functions	117
9.1	Hardware Configuration for Velocity Control Mode	117
9.1.1	CMMT Servo Drive Configuration for Velocity Control Mode	117
9.1.2	CMMT Drive Parameterisation for Integrate with PLC in Velocity Control Mode	124

9.2	Quick Procedure to Run the CMMT Drive in Velocity Control	128
9.2.1	Tel001_Velocity_Drives_Festo_EIP Functions.....	128
9.2.2	Tel001_Velocity_Drives_Festo_EIP Functions.....	131
10	Record table.....	134
10.1	Set the Task/Sequence in Record Table	134
10.2	Testing the Task/Sequence of control through FAS	136
11	How to control the torque control at the position.	139
11.1	Application description	139
11.1.1	Used Parameters for Torque	139
11.1.2	Travel top fixed stop (application class 3).....	139
11.1.3	General parameters for “Travel to fixed stop”	143
12	Downloading And Monitor Mode.....	144
12.1	Download.	144
12.2	Monitor Mode.....	146
13	Sample Code Description	153
13.1	Method 1 - Using PTP_Drives_Festo_EIP Modes of Operation.....	153
13.1.1	Set EPD and Dynamics Parameters in FAS	153
13.1.2	Execute the Logic in PLC using EPD.....	155
13.1.3	Procedure to control in Torque Mode using PTP_Drive_Festo_EIP function block (Refer Chapter 6.4.9)	155
13.2	Method 2 - Using FAS Record Table and PTP_Drives_Festo Modes of Operation	160
13.2.1	Set the Task/Sequence in Record Table	160
13.2.2	Procedure to control in Traversing/Record Table using PTP_Drive_Festo_EIP function block (Refer Chapter 6.4.9).....	162

1 Introduction

The main use of this application note is to describe the communication between CMMT Servo Drive and Mitsubishi PLC through Ethernet/IP protocol.

The application note has the description for the following:

1. Festo Automation Suite
 - Field bus communication settings between CMMT-AS and Mitsubishi PLC
 - How to change the Control Mode
 - How to set parameters in the Record table(with stroke limit function)
 - How to set parameters in the Extended Process Data (EPD)
2. PLC (Mitsubishi)
 - EtherNet/IP communication settings in GX Works3
 - Integration of Festo_CMMT_Mitsubishi_IQR_EIP_Lib library into GXWorks3 project.
 - Description on the working of the Festo_CMMT_Mitsubishi_IQR_EIP_Lib library.
 - CMMT Drive in Position Control Mode
 - Extended Process Data
 - How to control the torque control by Telegram 111 communication with PLC
 - CMMT Drive in Velocity Control Mode

Introduction of Position Control FB (PTP_Drives_Festo_EIP)

The below function block is used to give the Position control command, Start command, Stop command, Jogging, Incremental jogging, Set homing reference & Move the axis to set homing reference point.



Note

- Refer [Chapter - 7](#) for Extended Process Data



Introduction of Velocity Control FB (Tel001_Velocity_Drives_Festo_EIP” & “Tel102_Velocity_Drives_Festo_EIP)

The below function blocks is used to give the Velocity control command The “Tel001_Velocity_Drives_Festo_EIP” FB controls only speed of a servo motor with reference to its given speed setpoint but in “Tel102_Velocity_Drives_Festo_EIP” taking advantage by controlling both speed and torque of a servo motor.

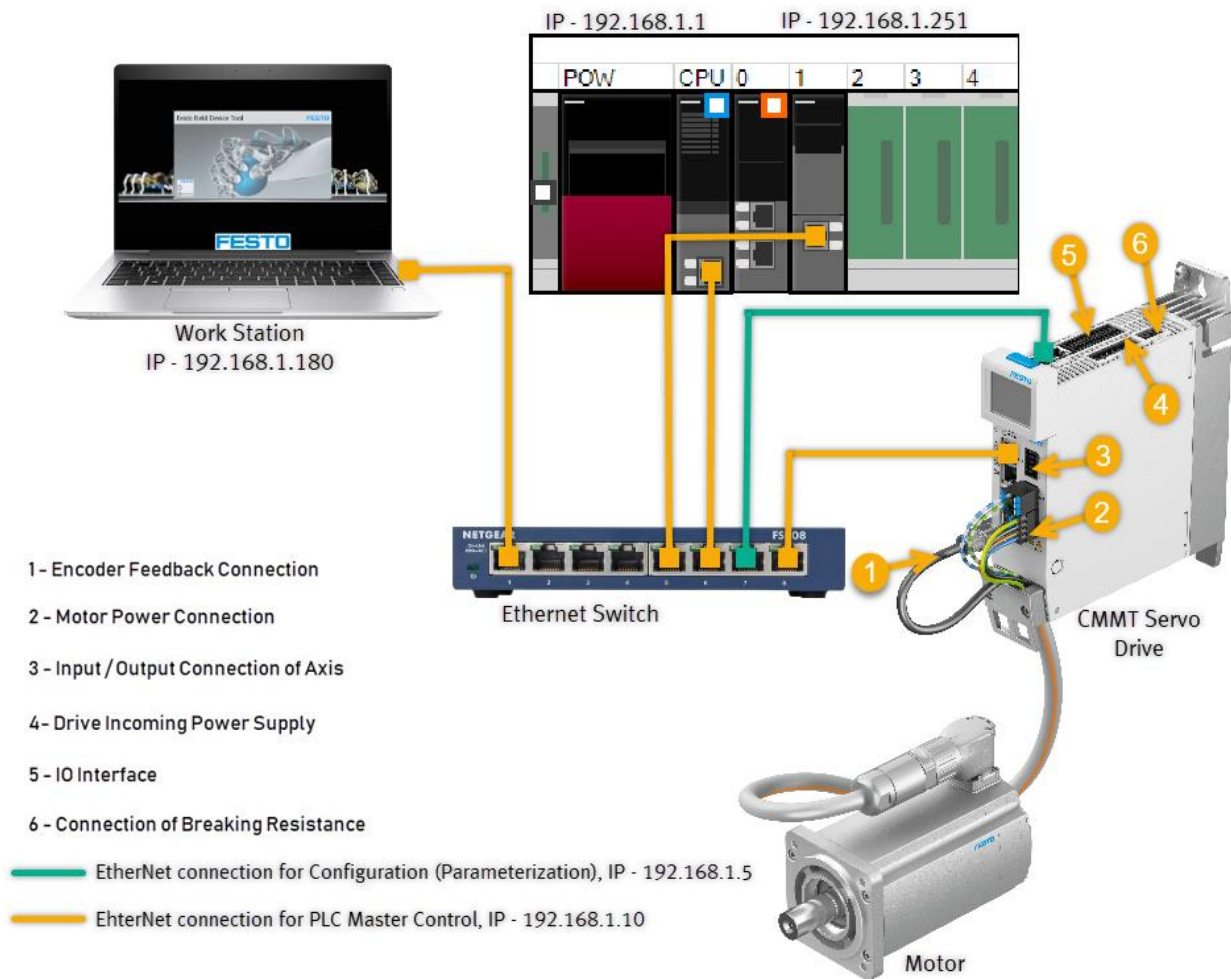


1.1 Hardware & Software requirements

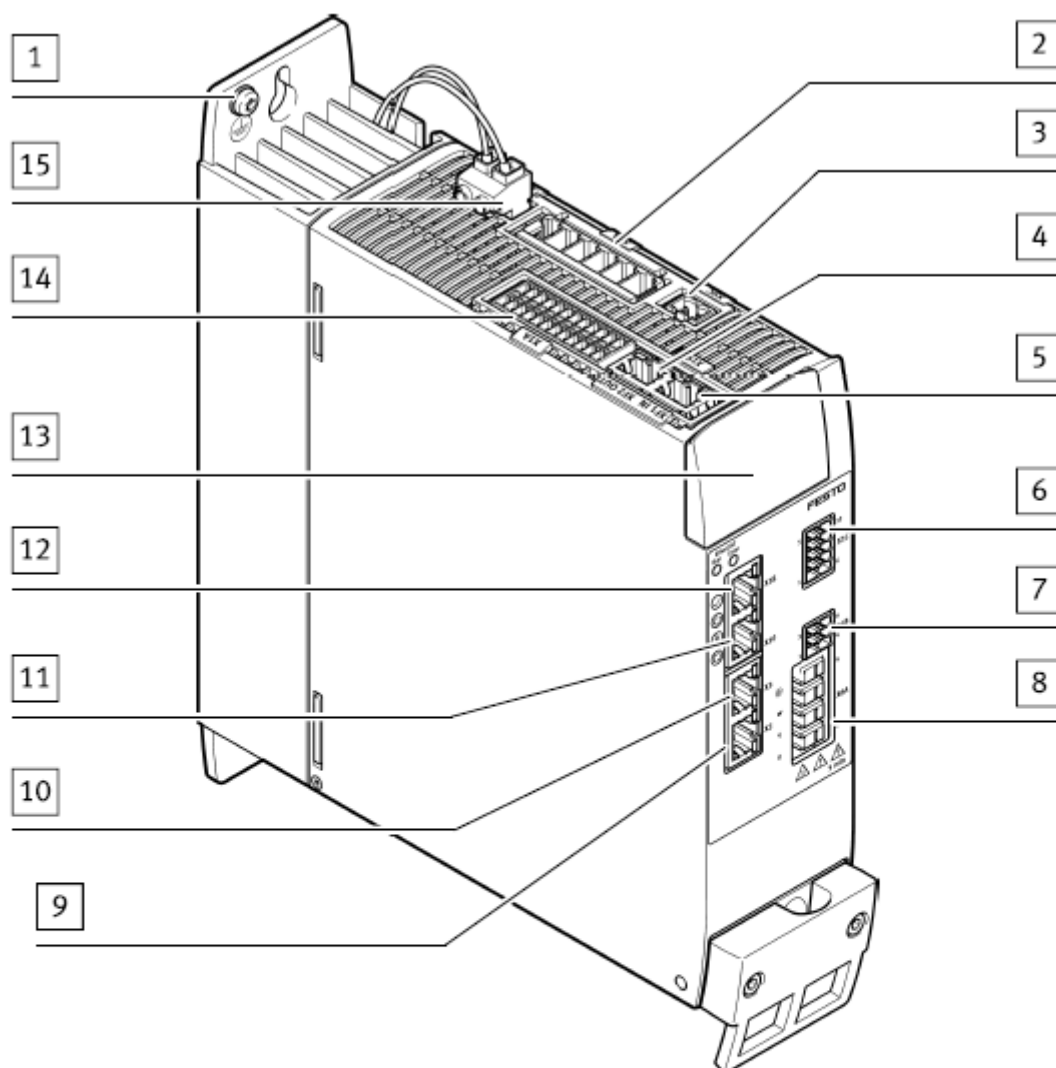
Device Name	Version (or) Part Number	Device Type
CMMT-AS-...-MP Drive Kit	V31.0.7.10 & above	Hardware
EtherNet Cable	NEBC-R3G4-ES-1-S-R3G4-ET	Hardware
R-__-ENCPU (Mitsubishi PLC)	Firmware V52 & above	Hardware
RJ71EIP91	-	Hardware
GxWorks3	V1.105K & Above	Software
EIP-CT for RJ71EIP91 1.00A	V 1.00A & Above	Software
Festo_CMMT_Mitsubishi_IQR_EIP_Lib	V1.4	Software
Festo Automation Suite	V2.5.0.635	Software
Festo Field Device Tool	V2.10.5.198	Software

1.2 Topology

When we want to communicate between Mitsubishi PLC and CMMT motor controller via Ethernet/IP protocol the following topology must be followed.



1.3 Hardware Over View of CMMT Servo Drive



- | | |
|---|--|
| 1 PE connection, housing | 9 [X2] Encoder connection 1 |
| 2 [X9A] Mains and intermediate circuit connection | 10 [X3] Encoder connection 2 |
| 3 [X9C] Logic voltage | 11 [X10] Device synchronisation |
| 4 [XF2 OUT] RTE interface port 2 | 12 [X18] Standard Ethernet |
| 5 [XF1 IN] RTE interface port 1 | 13 [X5] Connection for operating unit (behind the blind plate) |
| 6 [X1C] Inputs/outputs for the axis | 14 [X1A] I/O interface |
| 7 [X6B] Motor auxiliary connection | 15 [X9B] Connection for braking resistor |
| 8 [X6A] Motor phase connection | |

2 Hardware Configuration for Position Control Mode

2.1 IP Address Configuration of CMMT Controller Parameterization Port

The IP address of the CMMT Controller can be configured in 2 ways namely:

- Configuration of CMMT Drive Parameterization port X18 using Festo Field Device Tool.
- Configuration of CMMT Drive Parameterization port X18 using Festo Automation Suite.

2.1.1 Configuration of CMMT Drive Parameterization Port X18 using Festo Field Device Tool

This chapter describes about IP address configuration of CMMT drive parametrizing port using Festo Field Device Tool.



Note

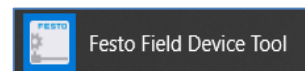
- Before starting CMMT drive IP address configuration process make sure the hardware connection between PC & servo drive is done as per communication architecture shown in [chapter 1.2](#). In the [chapter 1.2](#) green colour connection is only for parametrization and after parametrization to establish connection with PLC, connect the PC with Ethernet switch as shown orange colour line.

- User can download the Festo Field Device Tool from Festo support portal website using below link.

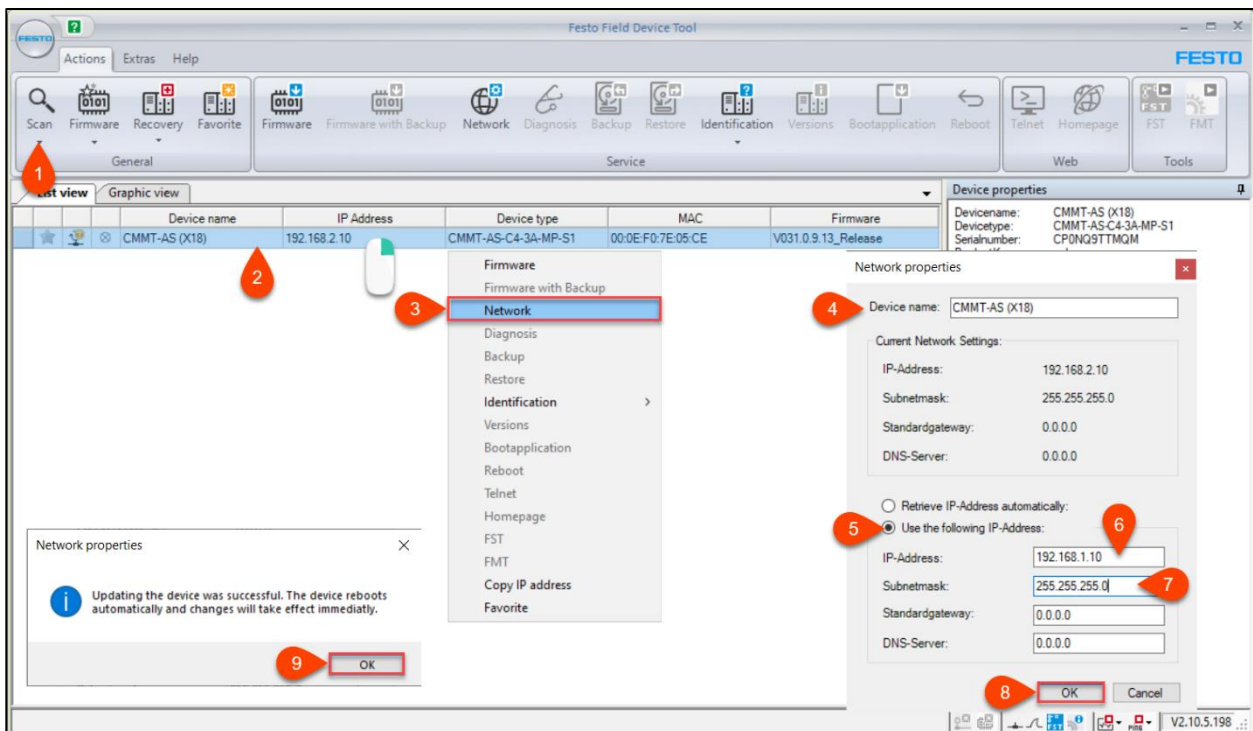
https://www.festo.com/net/en-gb_gb/SupportPortal/Default.aspx?tab=0&q=8004365

Step 1

- Open the Festo Field Device Tool from following path **Windows Start > Festo Software > Festo Field Device Tool**



Step 2



1. Select “**Scan Button**” to update the connected device list.
2. Select the correct drive from device list based on identifiers like Device name & MAC address of devices.
3. Right click the selected device & click the “**Network**” option from right click menu.

4. Enter the **“New name”** for the drive. It’s an optional point by default drive name will be device type name.
5. Select **“Use following IP-Address”** option for enable static IP mode. By default the IP address mode will in Dynamic mode.
6. Enter the **new IP address** for the drive. Make sure the drive IP series must meet series of PLC IP address.
7. Enter the **subnet mask** value to define capacity of the network range. Above network configuration done for 255 devices.
8. Click **“OK”** to confirm configuration.
9. Click **“OK”** button from pop-up window.

2.1.2 Configuration of CMMT Drive Parameterization Port X18 using Festo Automation Suite

This chapter describes about IP address configuration of CMMT drive parametrizing port using Festo Automation Suite.

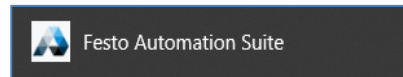


Note

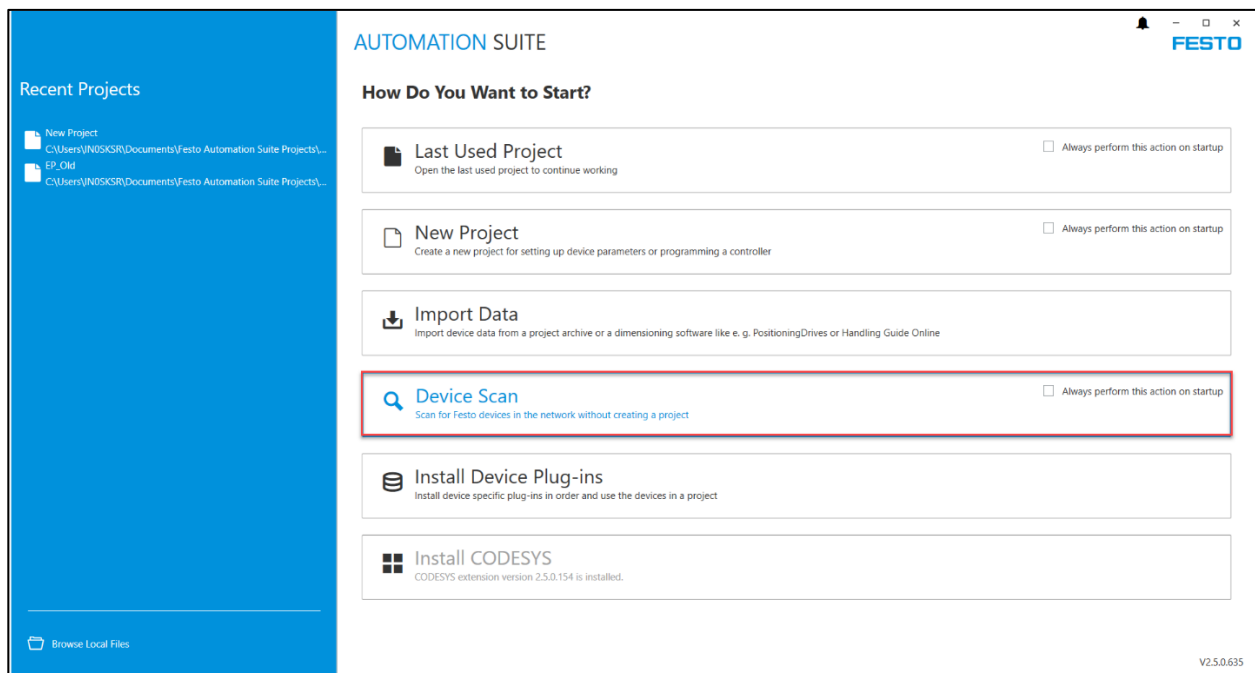
- Before starting CMMT drive IP address configuration process make sure the hardware connection between PC & servo drive is done as per communication architecture shown in [chapter 1.2](#). In the [chapter 1.2](#) green colour connection is only for parametrization and after parametrization to establish connection with PLC, connect the PC with EtherNet switch as shown orange colour line.
- User can download the Festo Field Device Tool from Festo support portal website using below link.
https://www.festo.com/net/en-gb_gb/SupportPortal/Default.aspx?tab=0&q=8004365

Step 1

- Open the Festo Automation Suite from following path **Windows Start > Festo Software > Festo Field Device Tool**

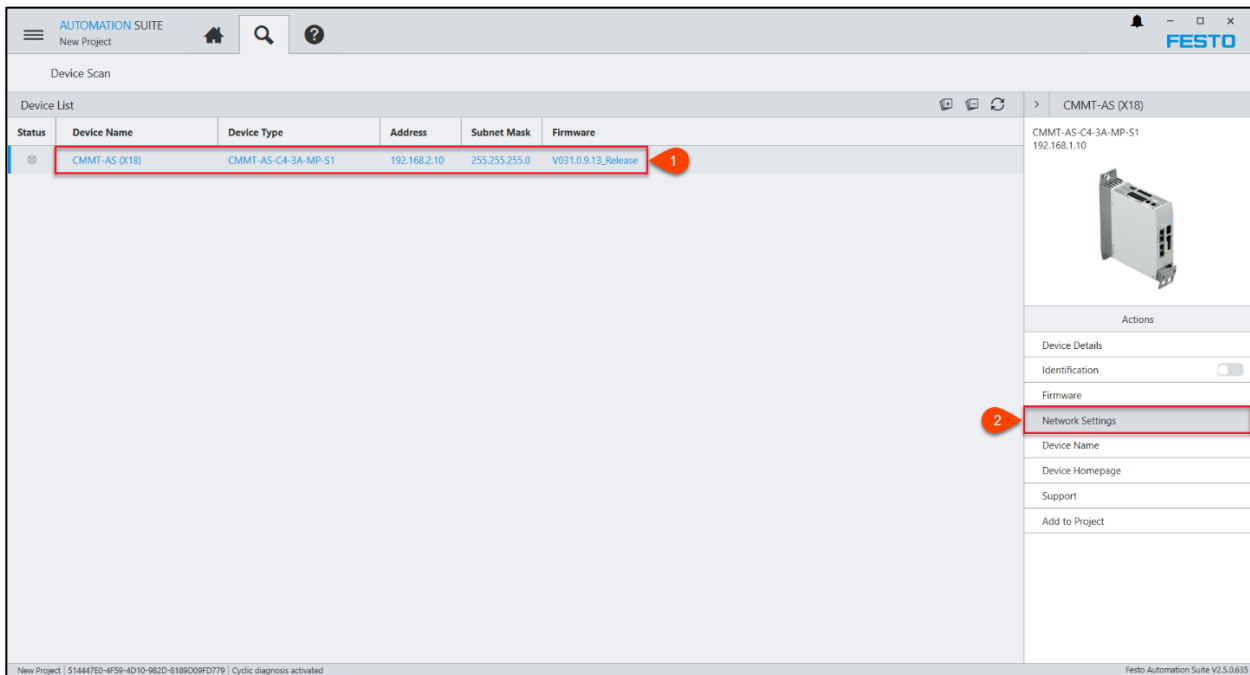


Step 2



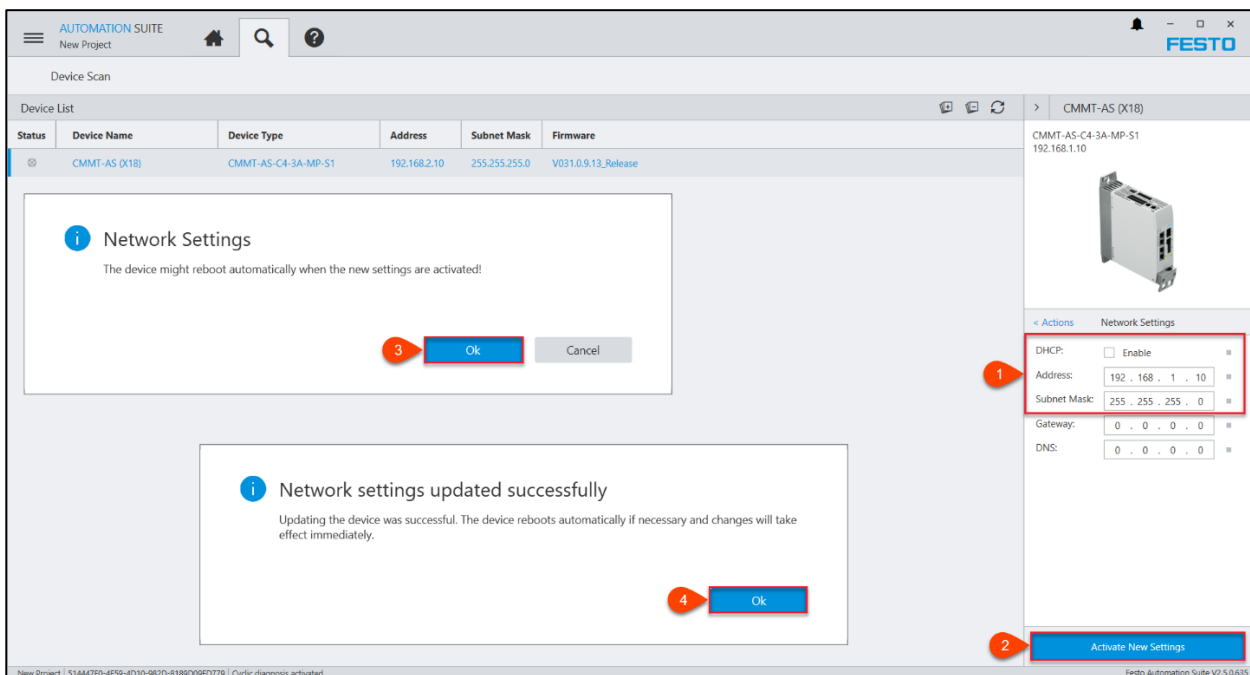
- Click **“Device Scan”** button from tool bar.

Step 3



1. Select correct device by its name.
2. Select “**Network Setting**” from action window.

Step 4



1. From network setting window **Uncheck the DHCP enable, Enter the new address** as per local network as highlighted in red colour.
2. Finally click “**Activate New Setting**” to update the changes.
3. Click “**OK**” to confirm the change of “**Network Settings**”.
4. Click “**OK**” to close the acknowledge pop up window.

2.2 CMMT-AS Servo Drive Fundamental Configuration

2.2.1 Servo Drive Selection



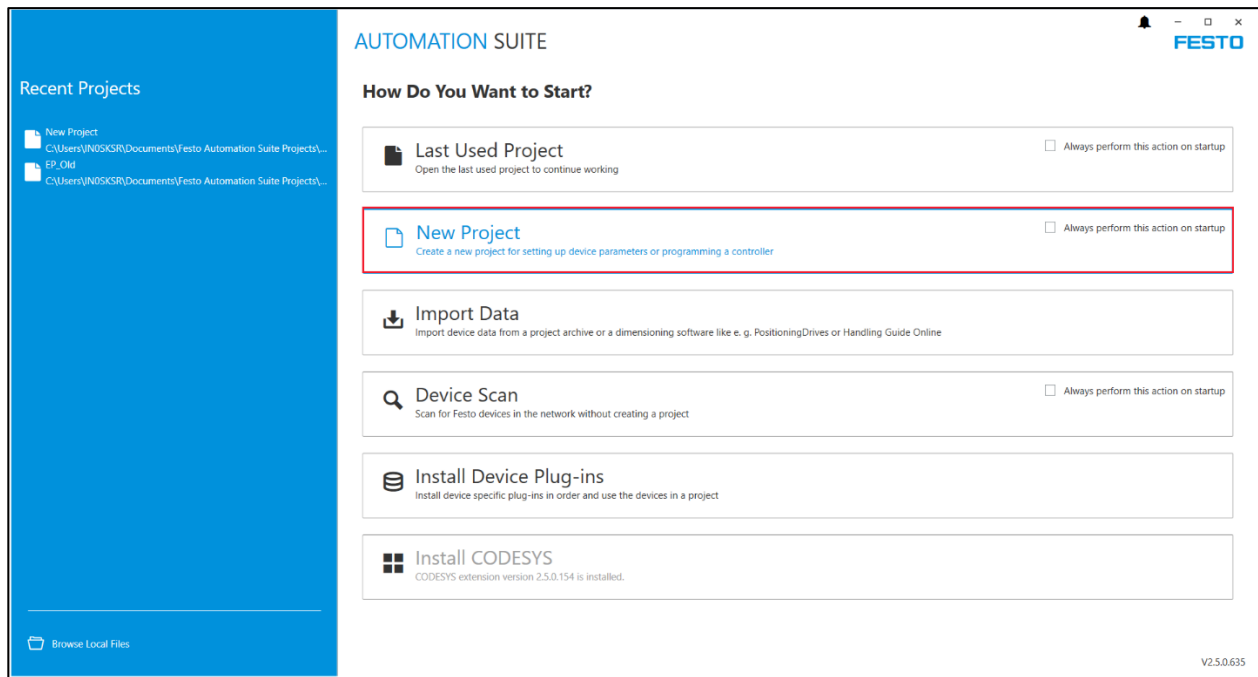
Note

- Follow the steps 1 to 4 **if drive is not in online**. If drive is connected with PC as shown in [Chapter 1.2](#), simply click Device scan from home as mentioned in [Chapter 2.1.2](#) Step 2 & jump to Step 5

Step 1

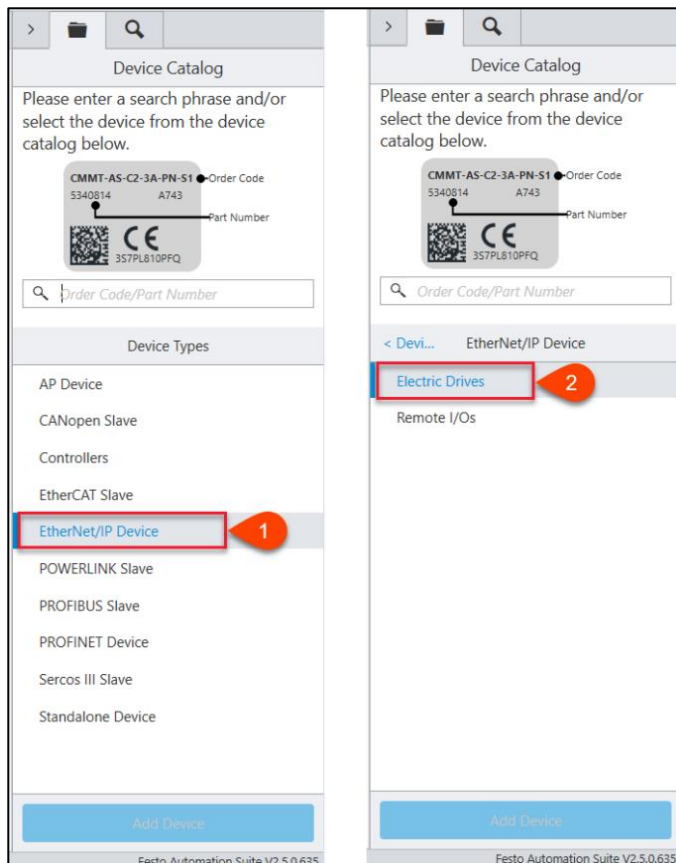
- Open **FAS** software as per [Chapter 2.1.2](#) Step 1.

Step 2



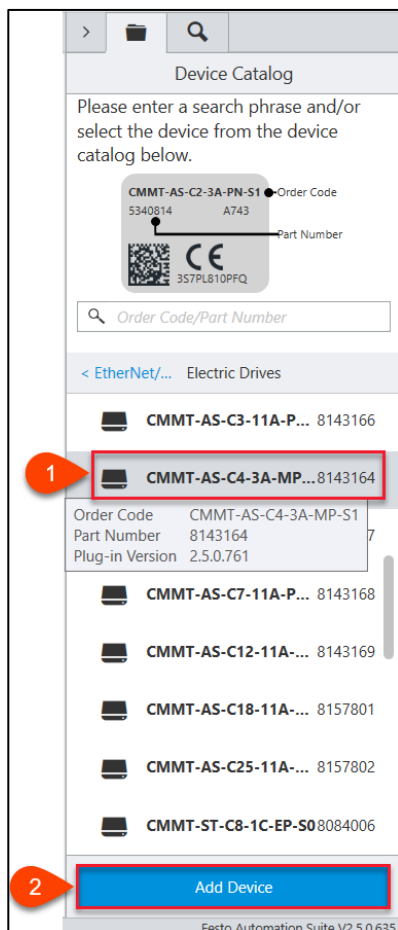
- Select **“New Project”** from start-up menu.

Step 3



1. Click Ethernet/IP Device from the device repository.
2. Select **“Electric Drives”**.

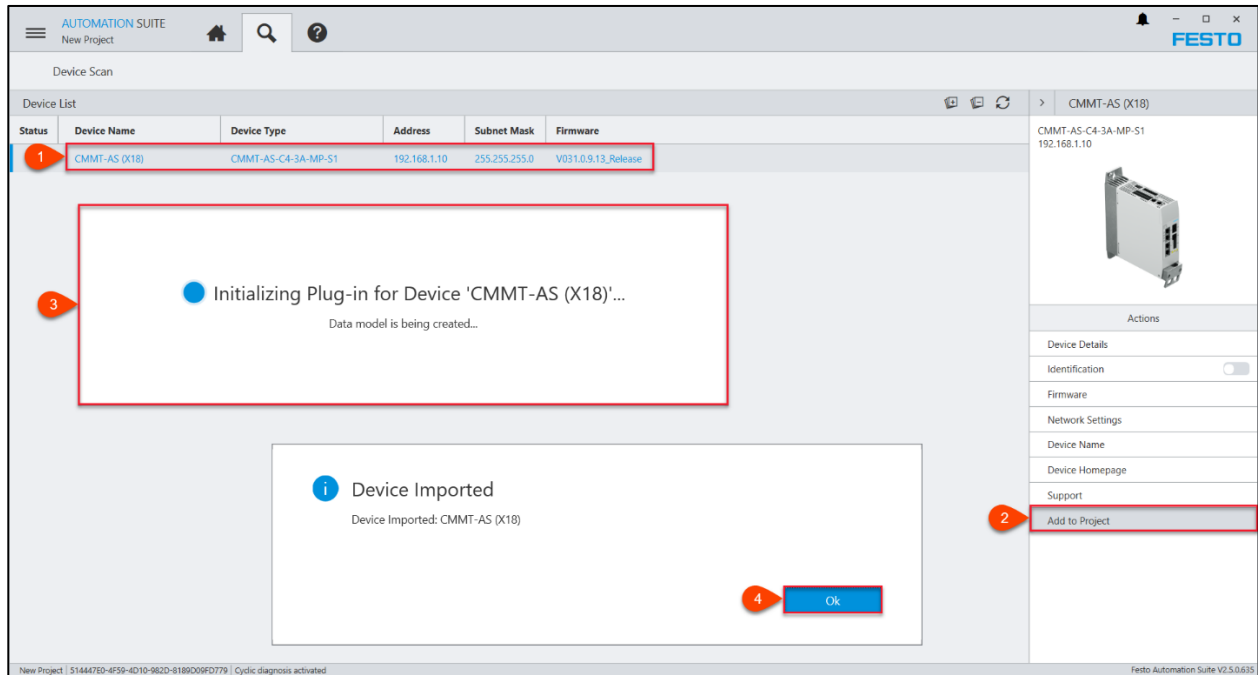
Step 4



1. **Select correct drive part number** from electric drives list . As per this sample application **CMMT-AS-C4-3A-MP-S1** is selected.
2. After selecting the drive click **Add Device**.

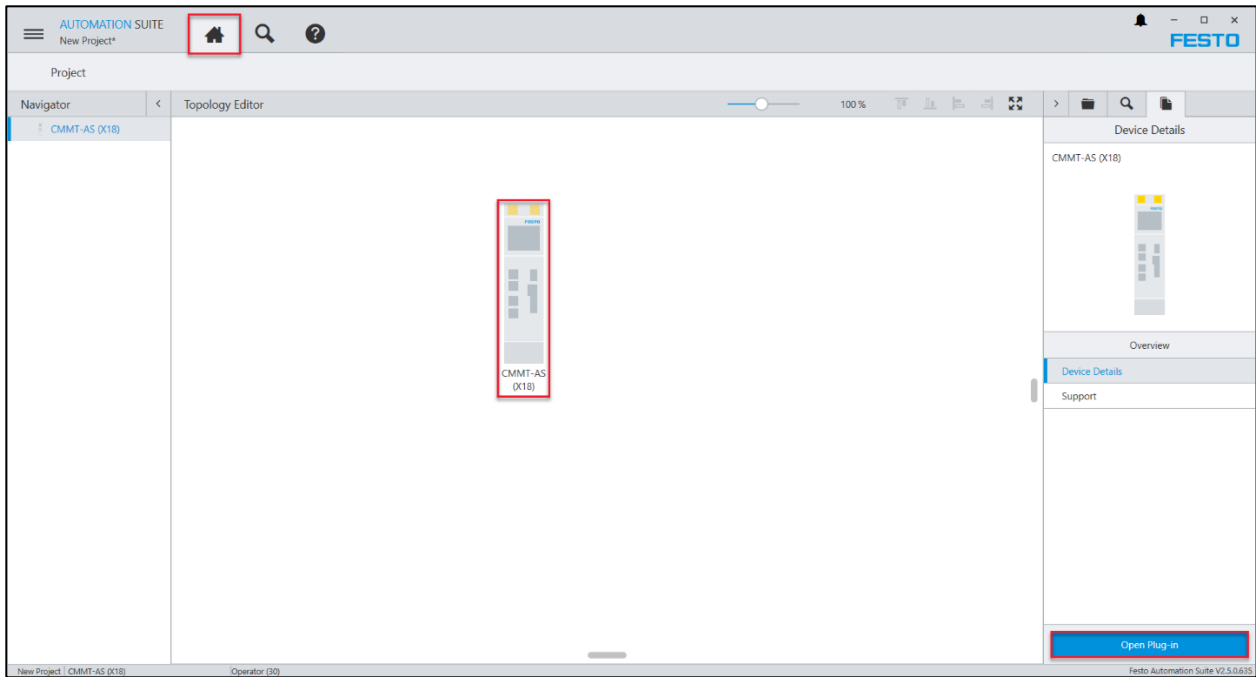
**Note**

- User can directly search the Drive through search bar without following previous steps.
- Jump to Step 6 if the user following offline configuration (Step 1 to Step 4).

Step 5

1. Select the correct device based on Its configured name & IP address.
2. Click **"Add to Project"**.
3. Device Plug-in initialization will start.
4. Click **"OK"** to confirm the successful device import.

Step 6



- Go to **Project Section** & double click the newly added drive or Click **Open Plug-in** to initialize configuration parameters.

2.2.2 Linear Axis Configuration



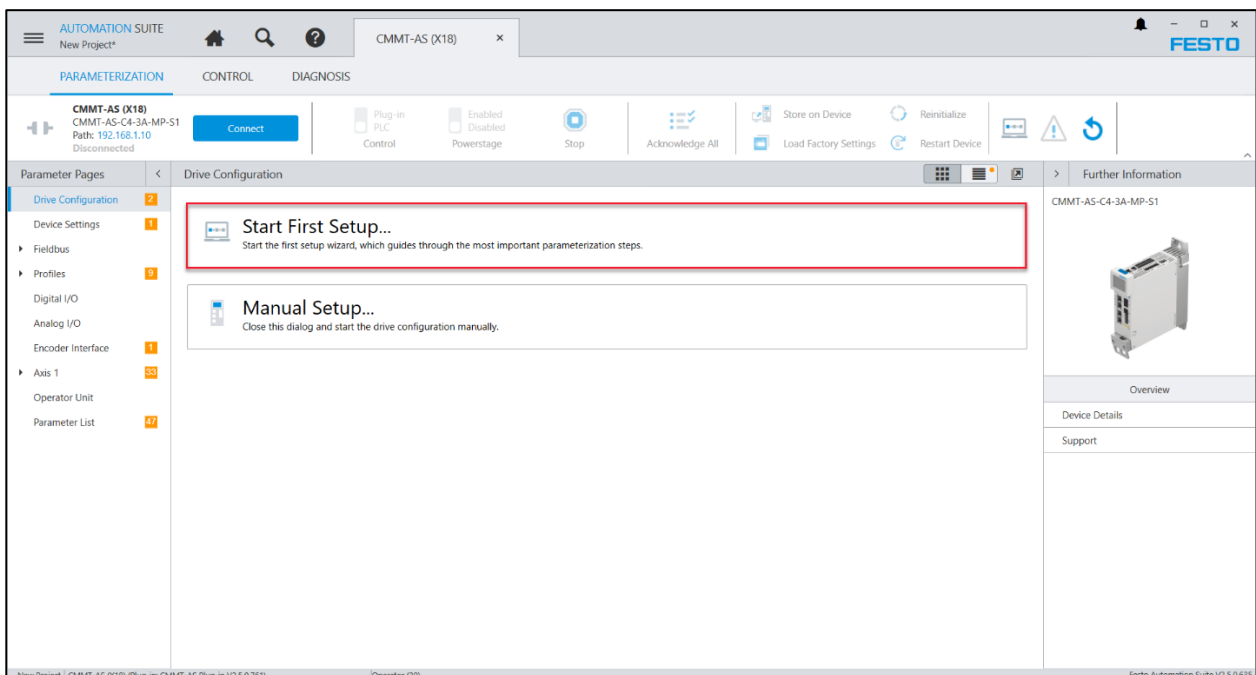
Note

- If the user is looking for Rotary Axis configuration skip this chapter and jump to [Chapter 2.2.3](#)

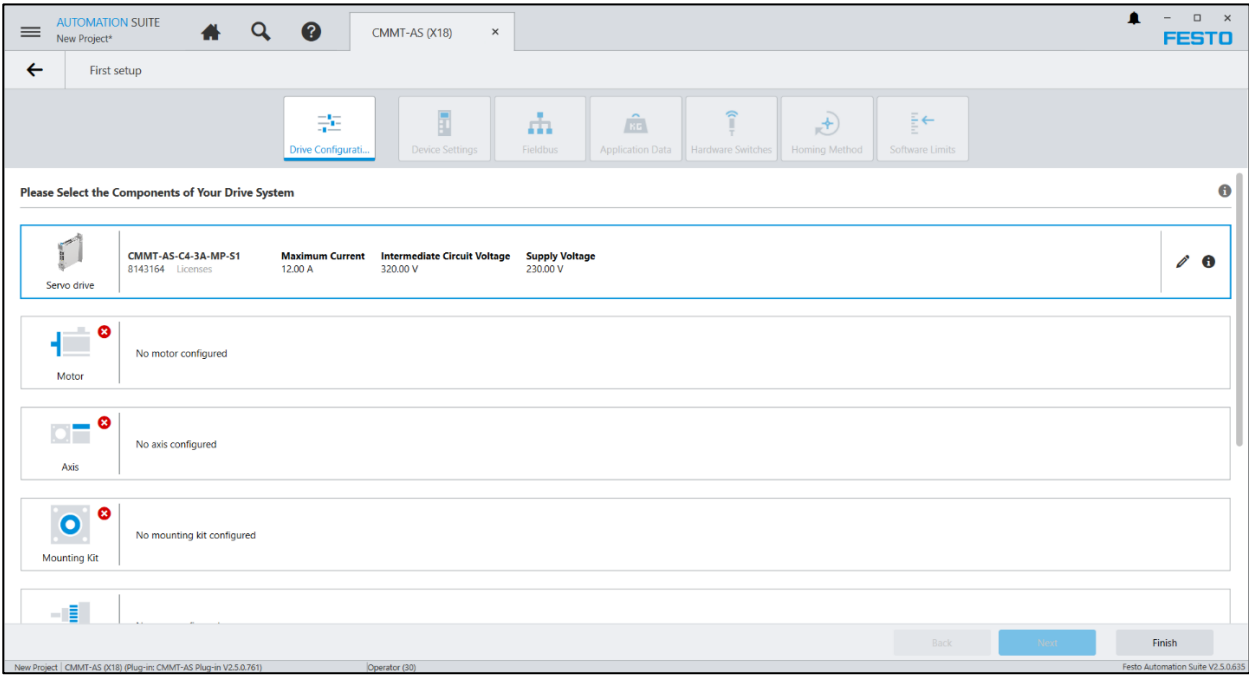
This chapter describes the parameterization of the CMMT drive for a linear axis to move 100 mm stock length with 0.50 m/s as base speed & current position as homing mode.

Step 1 – Motor Configuration

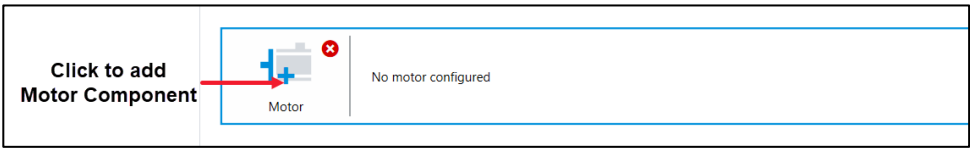
1. Select **“Start first setup”** from drive configuration menu.



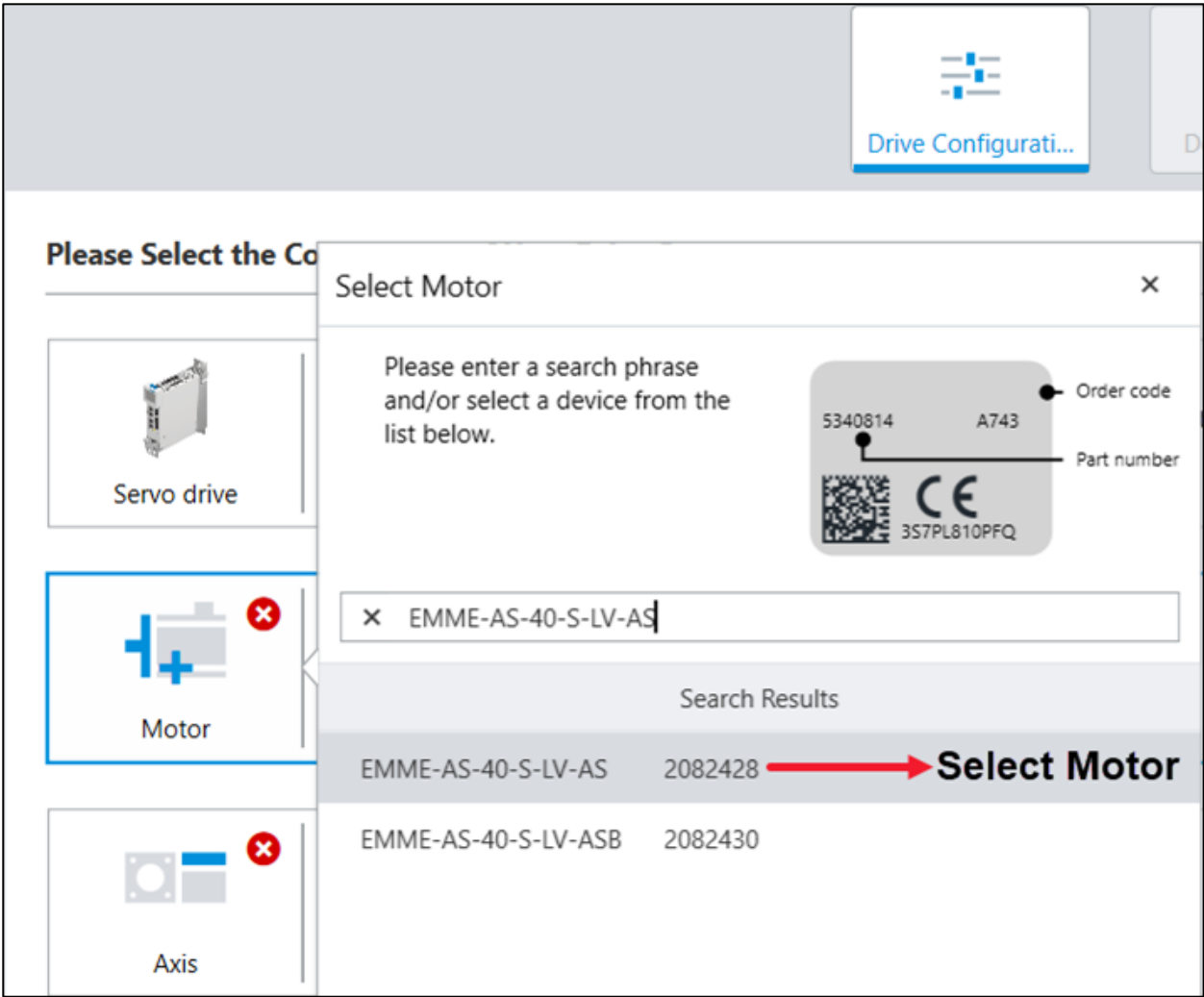
2. Once you click Start first setup following will be displayed.



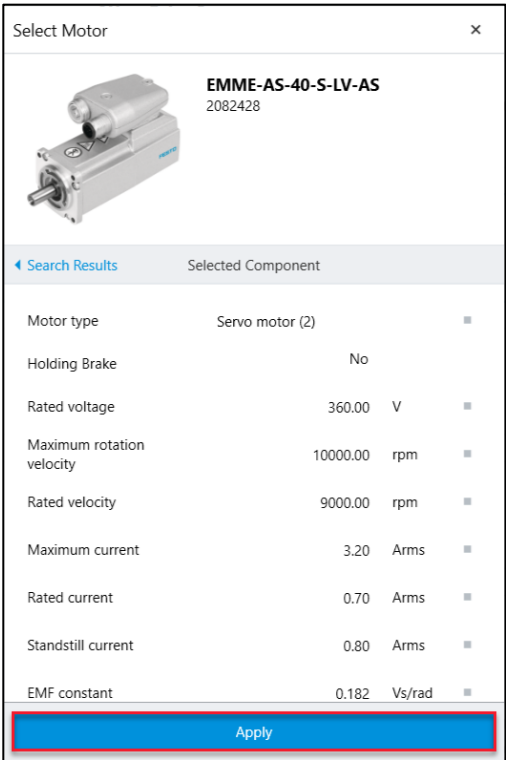
3. Click **Motor** to add the motor component.



4. Select the correct motor from the list of available motors. As per this sample application “**EMME-AS-40-S-LV-AS**” motor is used.



5. After selecting the motor click **Apply** to add the motor to project.



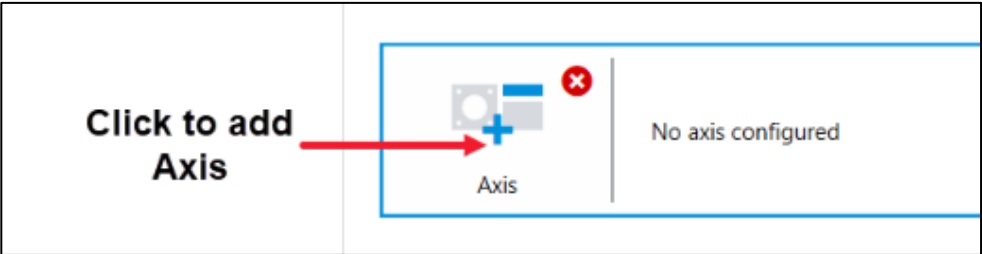
6. The motor tab after configuration will be viewed as shown below.

 Motor	EMME-AS-40-S-LV-AS 2082428	Type Servo motor (2)	Holding Brake No	Encoder Protocol Hiperface (0)	Encoder Type Single turn (1)	Voltage 360.00 V
--	--------------------------------------	--------------------------------	----------------------------	--	--	----------------------------

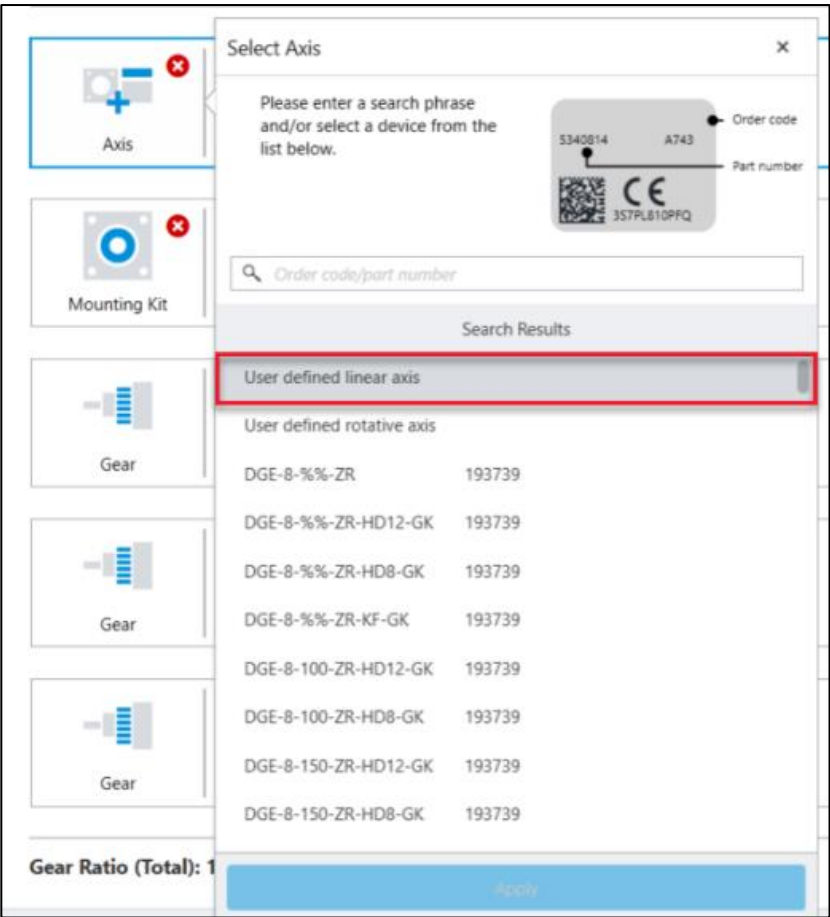
Step 2 – Axis Configuration

This step describes the axis type configuration. As specified initially here axis configuration is done for 100 mm in linear stock length. This configuration will change based on user application type.

1. Click **Axis** to add the required axis to project.



2. Select the needed axis from the available list. In our example we will select **User Defined Linear Axis**.



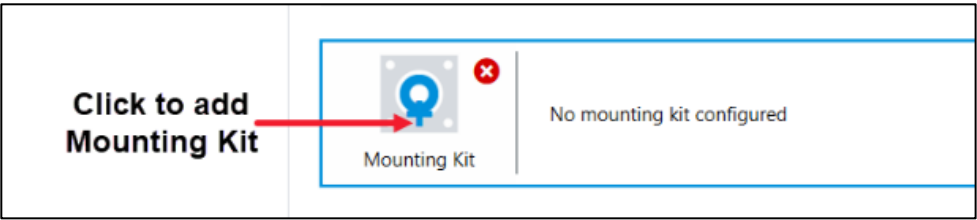
- After selecting the axis enter the needed configuration details. Click **Apply** to add the axis to project.

- The Axis tab will look after configuration as shown below.

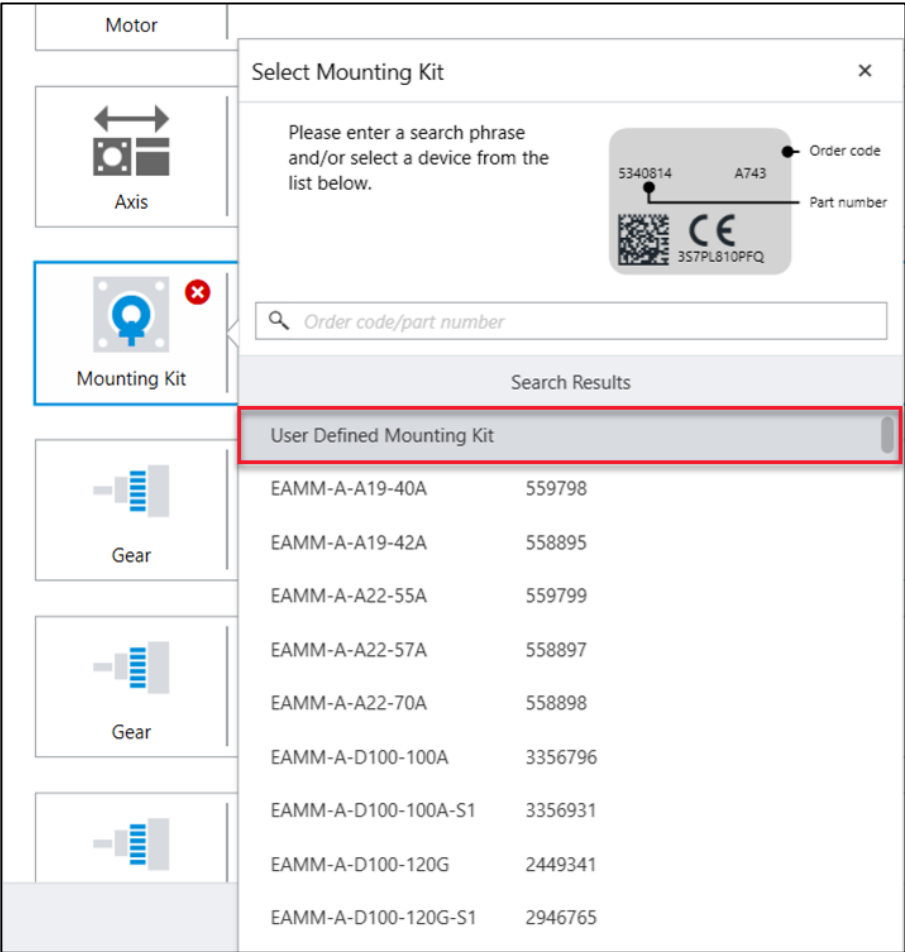
Step 3 – Mounting Kit Configuration

This step illustrates the mounting kit configuration. This configuration will vary depending on user application type. At the end of this step click next button to move to next level configuration.

- Click **Mounting Kit** to add a new Mounting Kit to project.



2. Select the needed mounting kit. In our example we have selected **User Defined Mounting Kit**.



3. After selecting the Mounting kit enter the needed configuration details. Click **Apply** to add mounting kit to project.

Select Mounting Kit



User Defined Mounting Kit

Search Results

Selected Component

Type

Axial

Inertia coupling

0.00000001

kgm²

Numerator

1

Denominator

1

Apply

4. The Mounting Kit tab will look after configuration as shown below.



Mounting Kit

User Defined Mounting Kit

Type

Axial

Gear Ratio

1:1

Step 4 – Device Settings

- 1. Go to **Device Settings** to do device configuration.

Enable Servo Drive

Activation via: I/O and fieldbus (0)

Supply Voltage

Mains voltage: 230.00 V

DC Link

Warning
If the DC circuit is linked to that of another drive, the rapid discharge function needs to be deactivated in order to protect the device.

Rapid discharge: ☒ Active

Back Next Finish

Operator (30) Festo Automation Suite V2.5.0.635

2. Select **Enable Servo Drive** method as **I/O and fieldbus** from drop down option **Activation Via**. Click **Next** button for next level configuration for next level configuration.

Enable Servo Drive

Activation via: I/O and fieldbus (0)

Supply Voltage

Mains voltage: 230.00 V

DC Link

Warning
If the DC circuit is linked to that of another drive, the rapid discharge function needs to be deactivated in order to protect the device.

Rapid discharge: ☒ Active

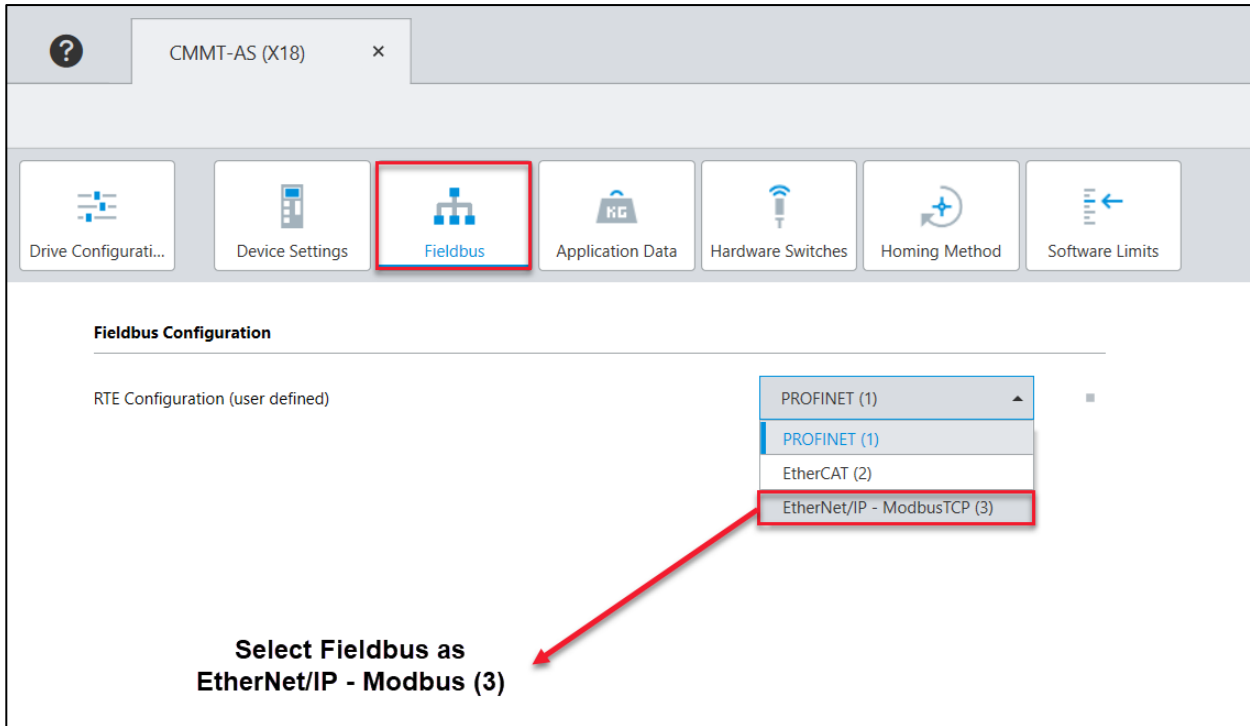
Back Next Finish

Operator (30) Festo Automation Suite V2.5.0.635

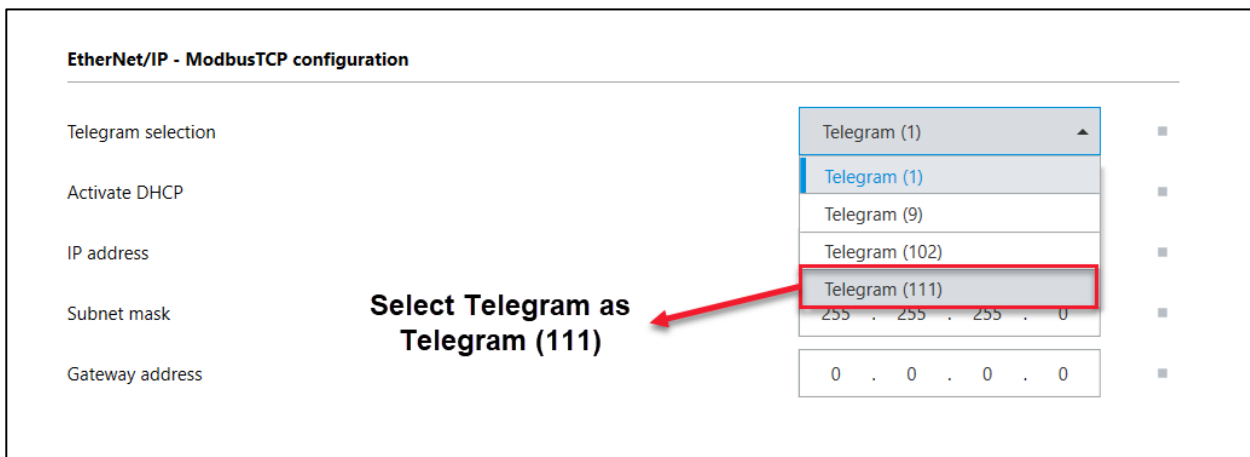
Select Enable Servo Drive Method as I/O and fieldbus

Step 5 – Fieldbus Configuration

1. Select **RTE Configuration** as **EtherNet/IP - Modbus** per application requirement.



2. In **EtherNet/IP – Modbus Configuration** select **Telegram Selection** as per user requirement. For this application **Telegram (111)** is used.



3. Enter the IP address & Subnet mask for **EtherNet/IP interface**(the same will be the IP address of the **Modbus TCP interface of CMMT**) as shown in below image. This communication interface is used to establish connection with PLC to receive control command & to send signal feedbacks. Click **Next** button for next level configuration.

EtherNet/IP - ModbusTCP configuration

Telegram selection: Telegram (1)

Activate DHCP: ☐ Active

IP address: 192 . 168 . 1 . 5

Subnet mask: 255 . 255 . 255 . 0

Gateway address: 0 . 0 . 0 . 0

Enter IP Address & Subnet Mask Values

Back Next Finish



Note

- Refer [Chapter 2.3](#) for more information.

Step 6 – Enter Inertia Value

- Go to **Application Data** to Specify inertia value of user application. This parameter value will vary depend on application type. Click **Next** button for next level configuration.

CMMT-AS (X18) x

FESTO

Drive Configurati... Device Settings Fieldbus **Application Data** Hardware Switches Homing Method Software Limits

Application Data

Axis mass: 0.00 kg

Application mass: 0.00 kg

Total mass: 0.00 kg

Enter Inertia Value

Rotation Polarity

Please select the mounting position of the motor (viewed from top):

- - - - - +

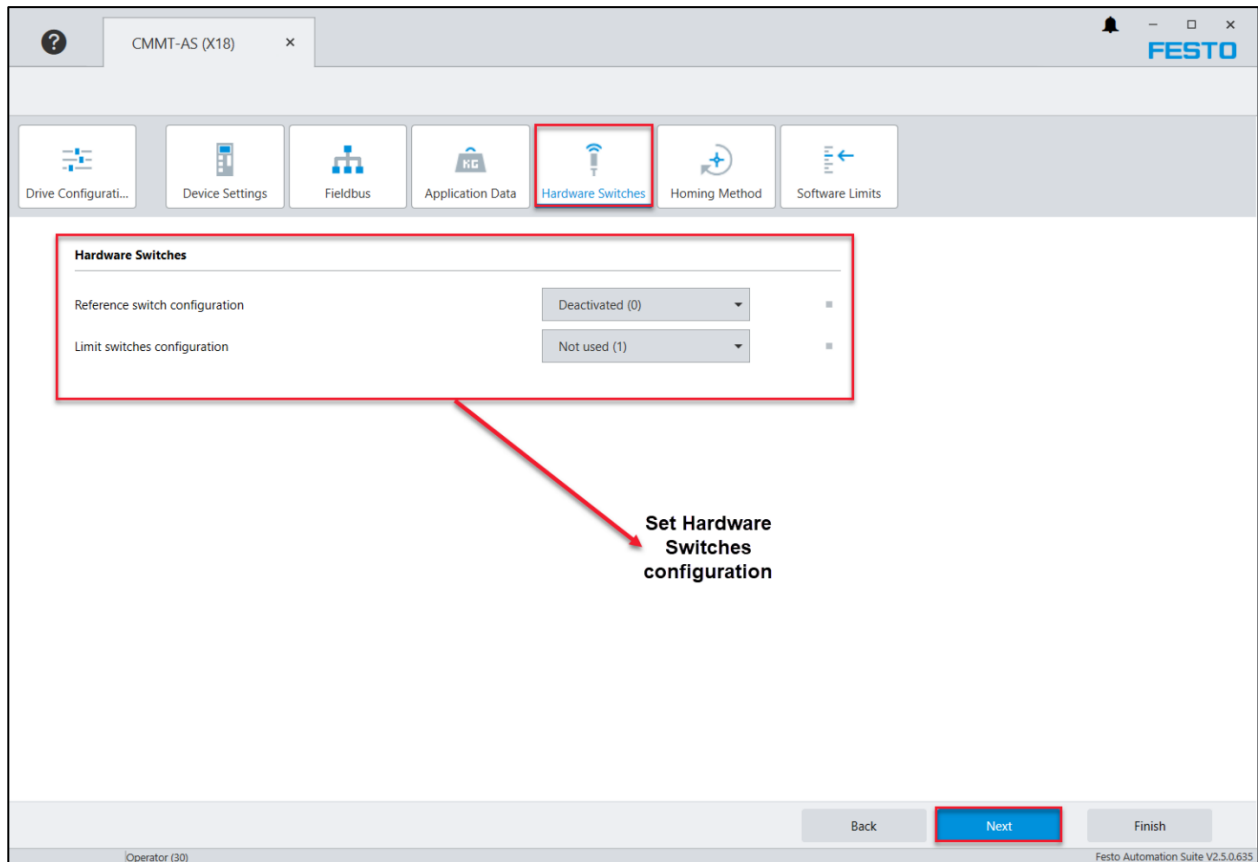
☐ Invert rotation polarity

Back Next Finish

Operator (30) Festo Automation Suite V2.5.0.635

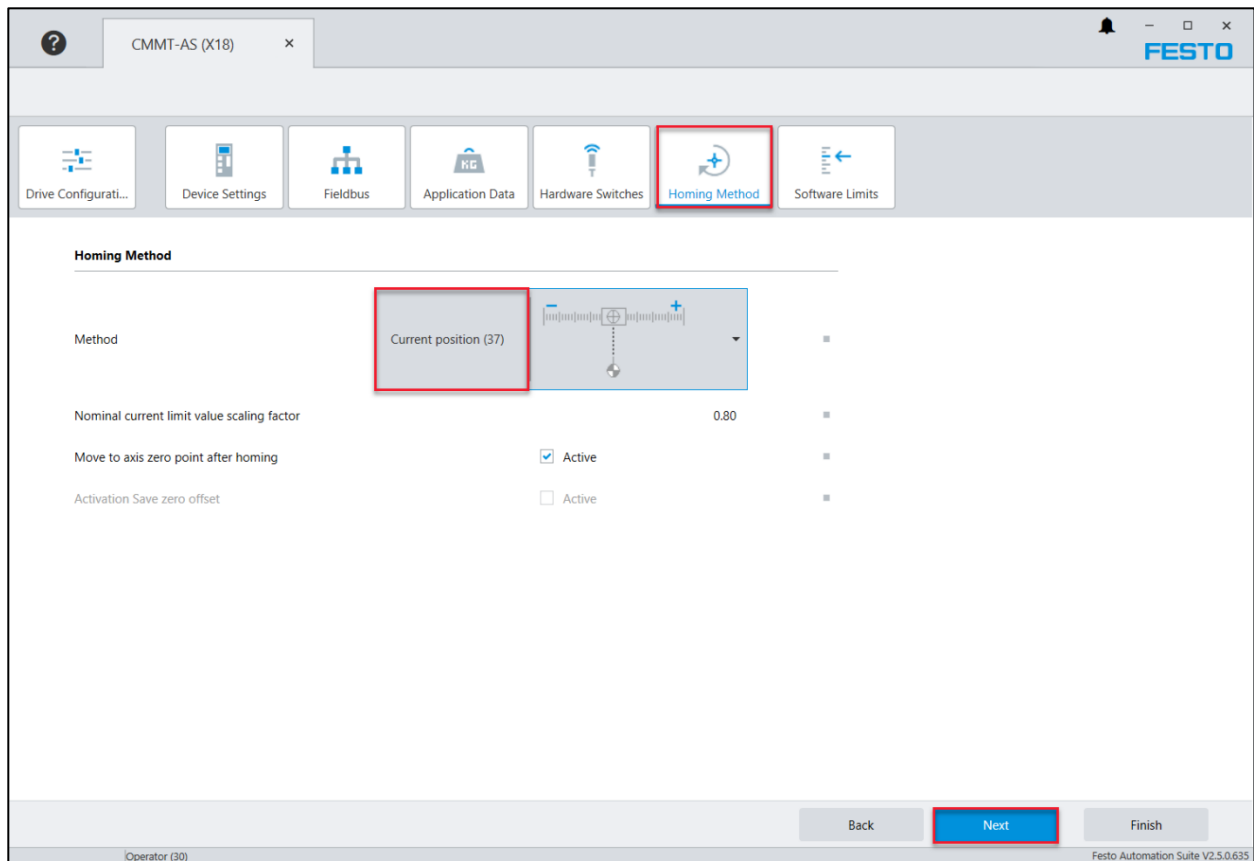
Step 7 – Hardware Switches Configuration

- Specify **Hardware Switches Configuration** of user application as per requirement. Click **Next** button for next level configuration.



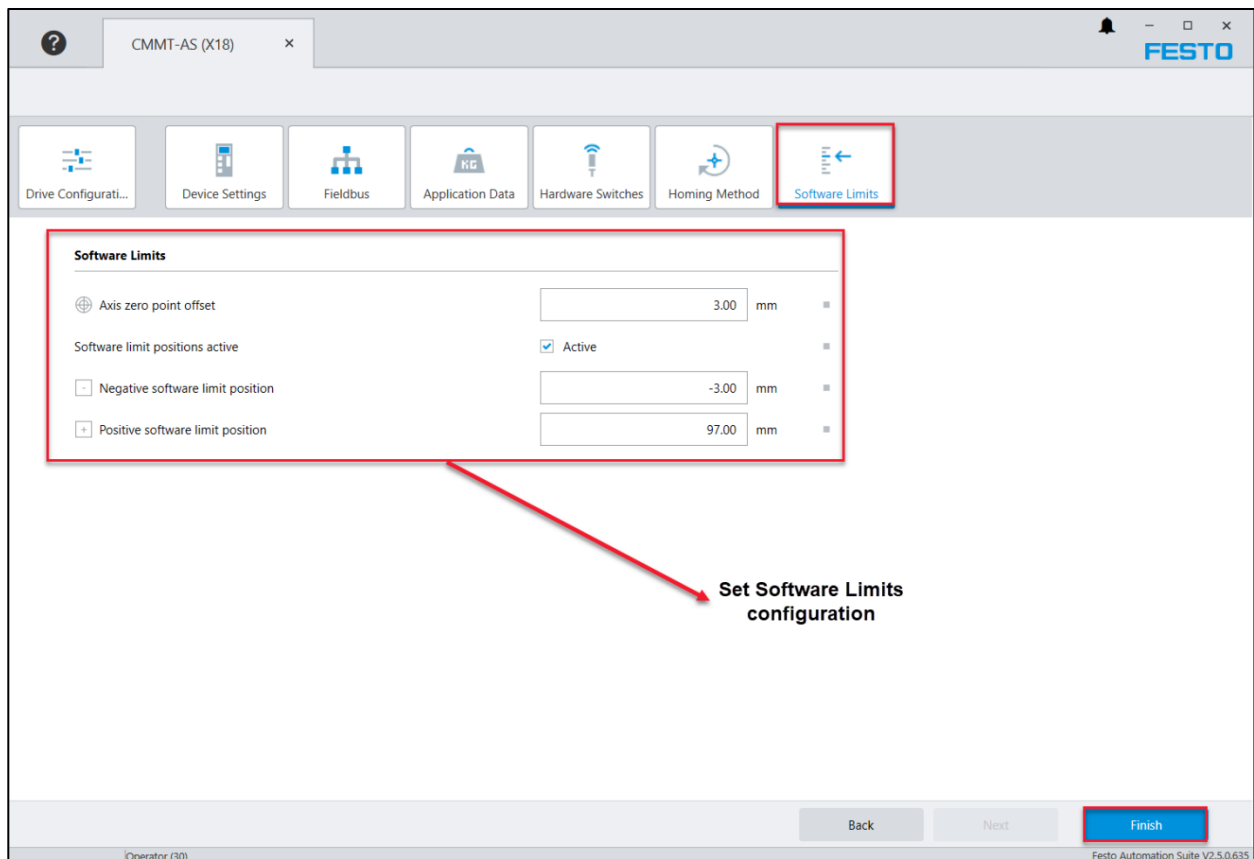
Step 8 - Homing Method

- Select the **Homing Method**. As per this sample application “**Current Position**” mode is selected, but the user should select the homing mode depending on the application requirement. Click **Next** button for next level of configuration.



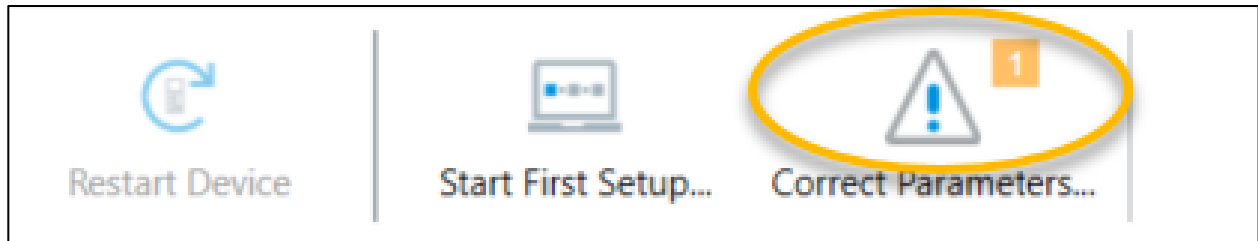
Step 9 – Software Limits Configuration

- Here user can enable the maximum & minimum values of **Software Limits** as per application requirements. Finally click **“Finish”** button to complete the configuration.



Step 10 – Correct Parameters

- At this stage all the fundamental configurations of drive system are done. If there are any corrections in the parameter software tool will indicate number of “**Correction Parameters**” (Right most top corner in FAS tool) as shown in the below picture. So to correct parameter value click the “**Correction parameters**” button and click **Apply** button to apply system generated recommended values

**Note**

- This step might not be required as per user application data.

2.2.3 Rotary Axis Configuration

This chapter describes the parameterization of the CMMT drive for a rotary axis to move 1000 revolution forward and reverse with 1000 RPM base speed & current position as homing mode.

**Note**

- If the user is looking for Linear Axis configuration skip this chapter and jump to [Chapter 2.2.2](#)

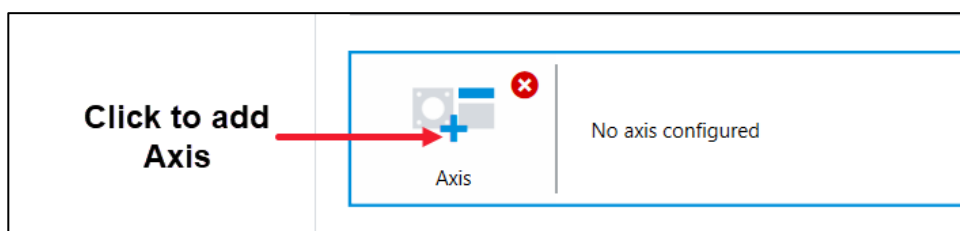
Step 1

- Please follow the **Step 1** for configuration described in [Chapter 2.2.2](#).

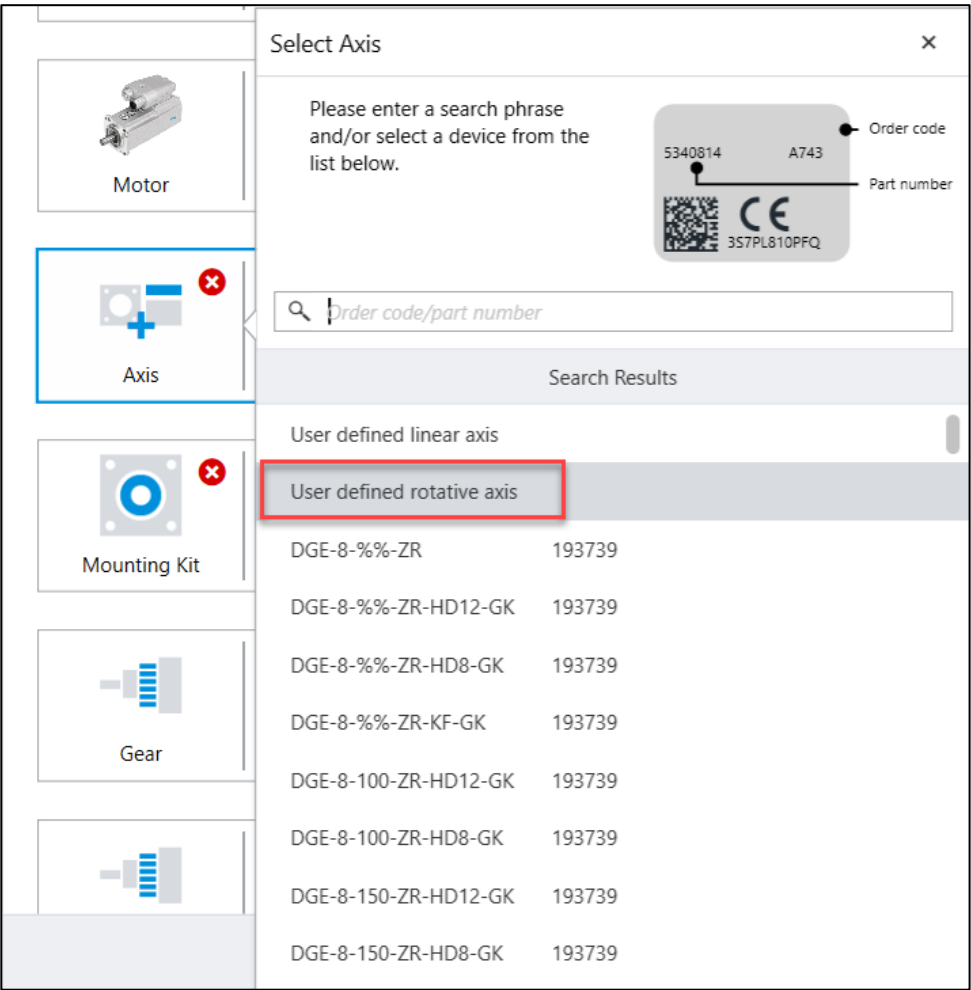
Step 2 – Axis Configuration

This step describes about the axis configuration. As specified initially here axis configuration is done for 1000 rotation in forward & reverse. This configuration will change based on user application type.

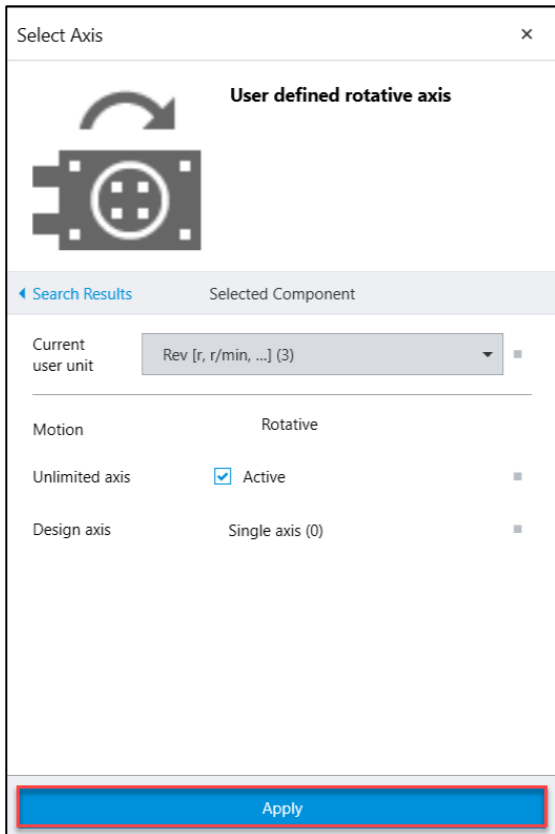
- Click **Axis** to add the required axis to project.



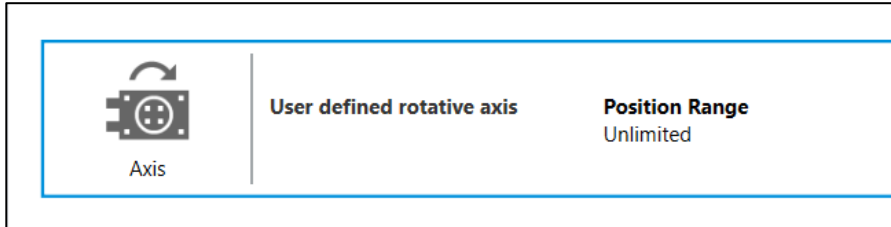
2. Select the needed axis from the available list. In our example we will select **User Defined Rotative Axis**.



3. After selecting the axis enter the needed configuration details. Click **Apply** to add the axis to project.



4. The Axis tab will look after configuration as shown below.



Step 3

- Please follow the **Step 3 to Step 10** for configuration described in [Chapter 2.2.2](#).
-

2.2.4 CMMT Drive Parameterization for Integrate with PLC

This chapter describes:

- Parameterisation of CMMT drive to establish communication with PLC through Telegram 111.
- Also it describes some additional parameters like Factor group, Reference values, EtherNet/IP interface, User defined limits.

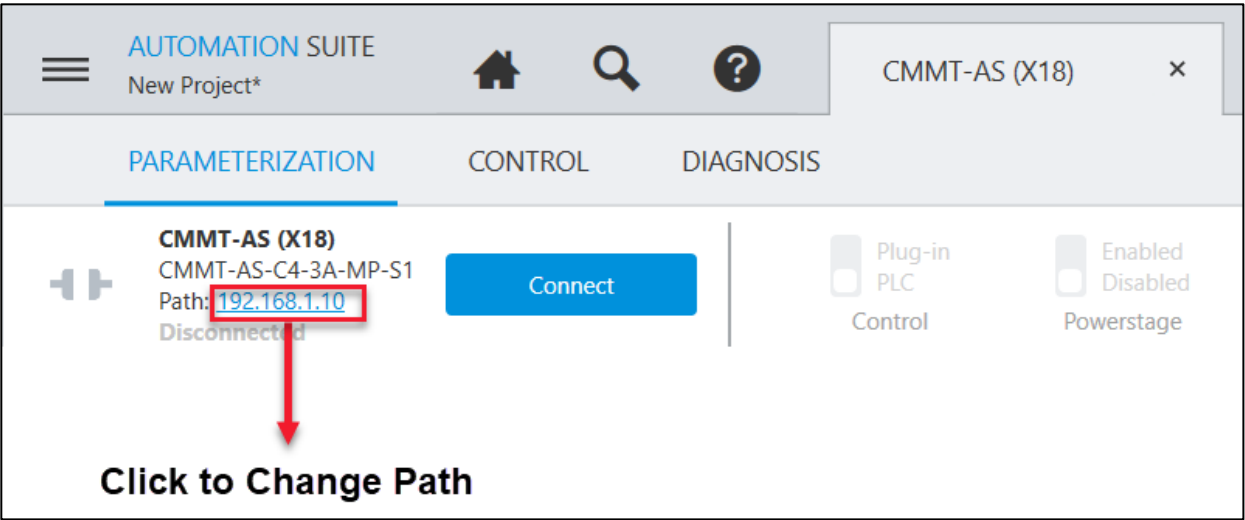
Step 1



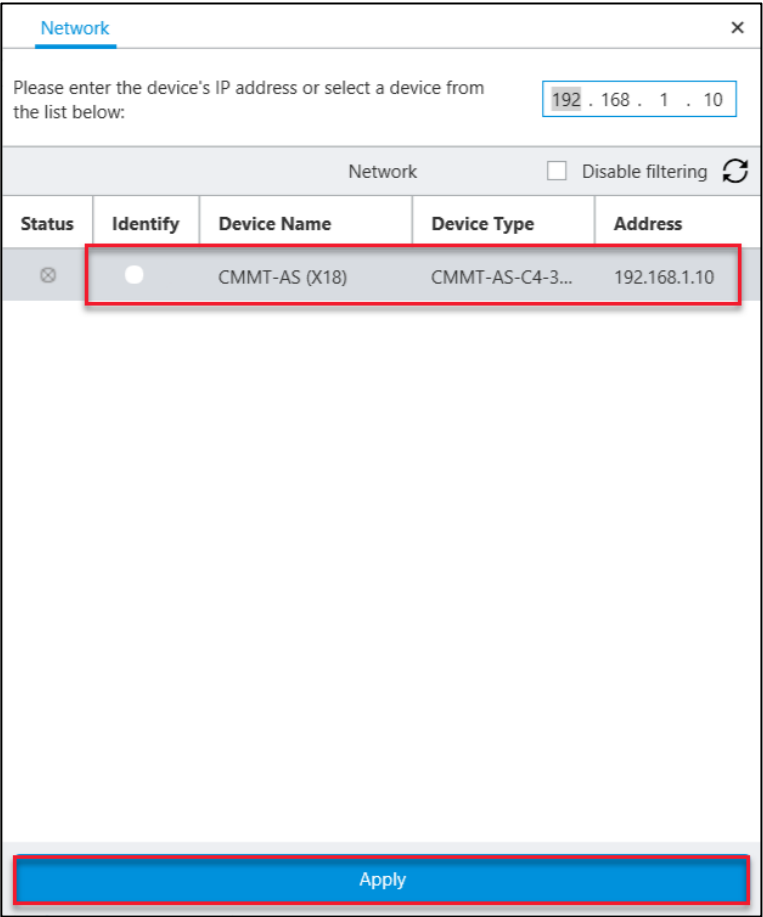
Note

- Follow this step when drive is available for online connection. If drive is not available simply skip this step and jump to next step.

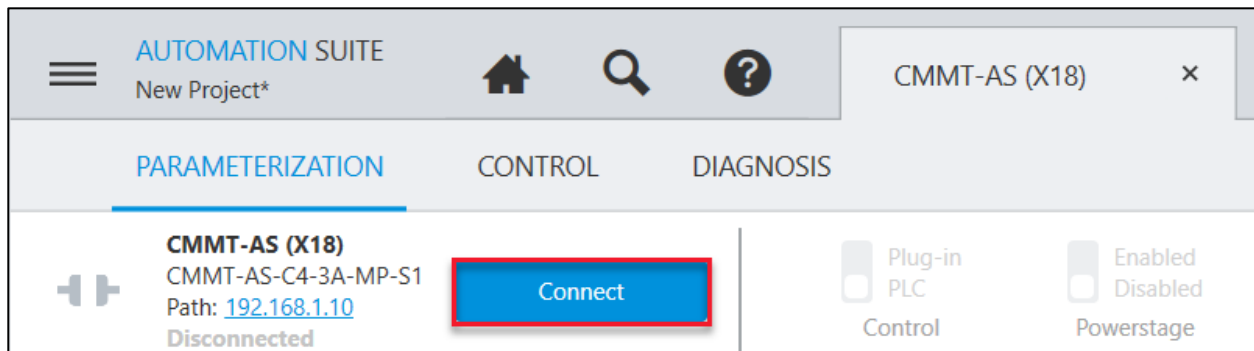
1. Click the IP Address **“Path”** from tool bar.



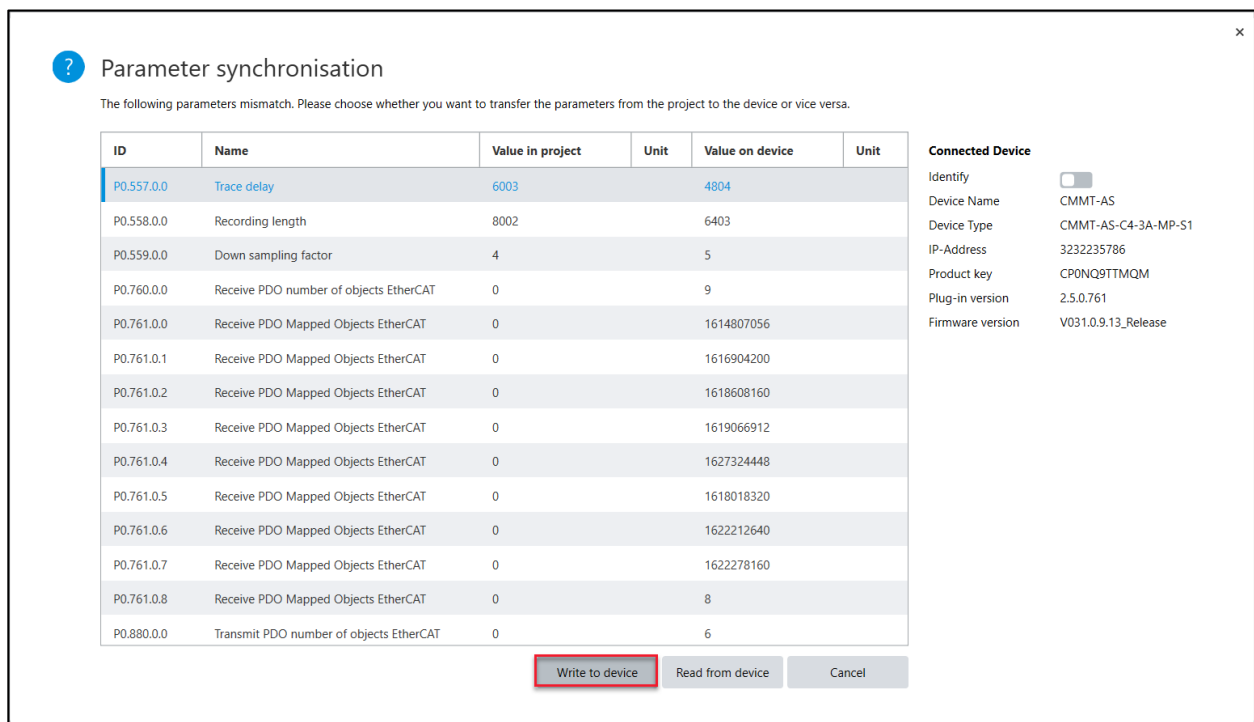
2. Select the correct drive based on IP address & Click **“Apply”** button to assign path address.



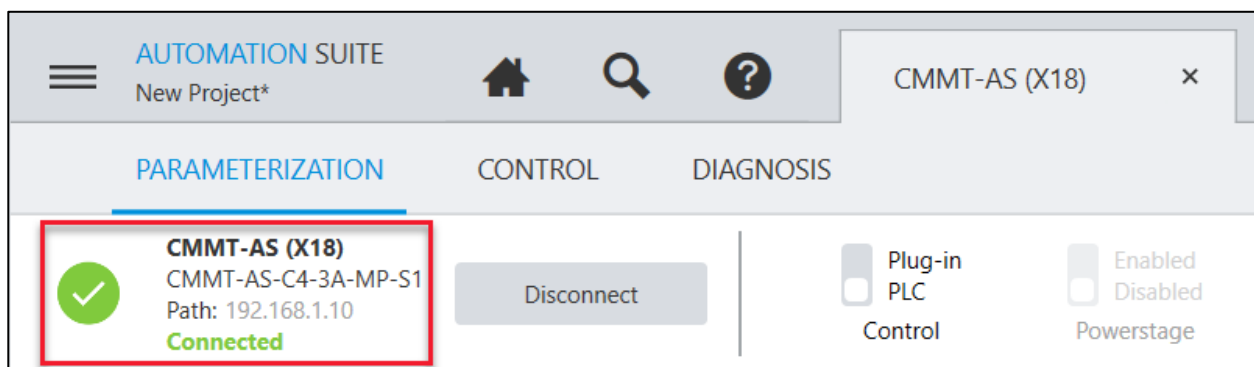
3. Click on **“Connect”** button to establish communication .



4. Click **Write to device** button on **Parameter Synchronisation** window.



5. The status of the drive changes from **Disconnected** to **Connected**.



Step 2 – Entering Factor Group Value from Position and Velocity

- Go to **Parameter Pages > Profiles > PROFIdrive > Factor Group** as shown below.

- Example - 1 PLC scale value is calculated below for move the linear axis 10 mm in 1 m/s speed.**

Position Factor (-6) = $10^{-6} = 0.000001 = 1\mu\text{m}$ Then For 10 mm = 1000000 Increment value in PLC

Velocity Factor (-3) = $10^{-3} = 0.001 = 1\text{mm/s}$ Then for 1 m/s = 1000 Increment value in PLC

- Example - 2 PLC scale value is calculated below for move the rotary axis 5 Rev in 100 RPM speed.**

Position Factor (-6) = $10^{-6} = 0.000001 \text{ Rev}$ Then For 5 Rev = 5000000 Increment value in PLC

Velocity Factor (-3) = $10^{-3} = 0.001 \text{ RPM}$ Then for 100 RPM = 100000 Increment value in PLC

**Note**

For more details about fieldbus scaling refer chapter name “Scaling of internal units for the fieldbus” in user manual.

- Different user unit can't be set for Position, Velocity, Acceleration & Jerk.
- The maximum value limit for 32 bit DINT data type variable is 2,147,483,647. So plan scaling factor with in the limit.

Step 3 – Store On Device

- Click the **“Store on device”** button to store all the parameters on drive memory.

**Note**

- The triangle warning symbol on both buttons indicates action required.
- All the modified parameters will automatically erase from drive during power restart If modified parameters are not stored in the drive.
- Its mandatory to reinitialize the drive for some parameter modification. If drive is not reinitialized the drive performance remains same as like previous configuration.

**Information**

Hardware configuration of CMMT-AS controller described in this chapters are only for this application note purpose.

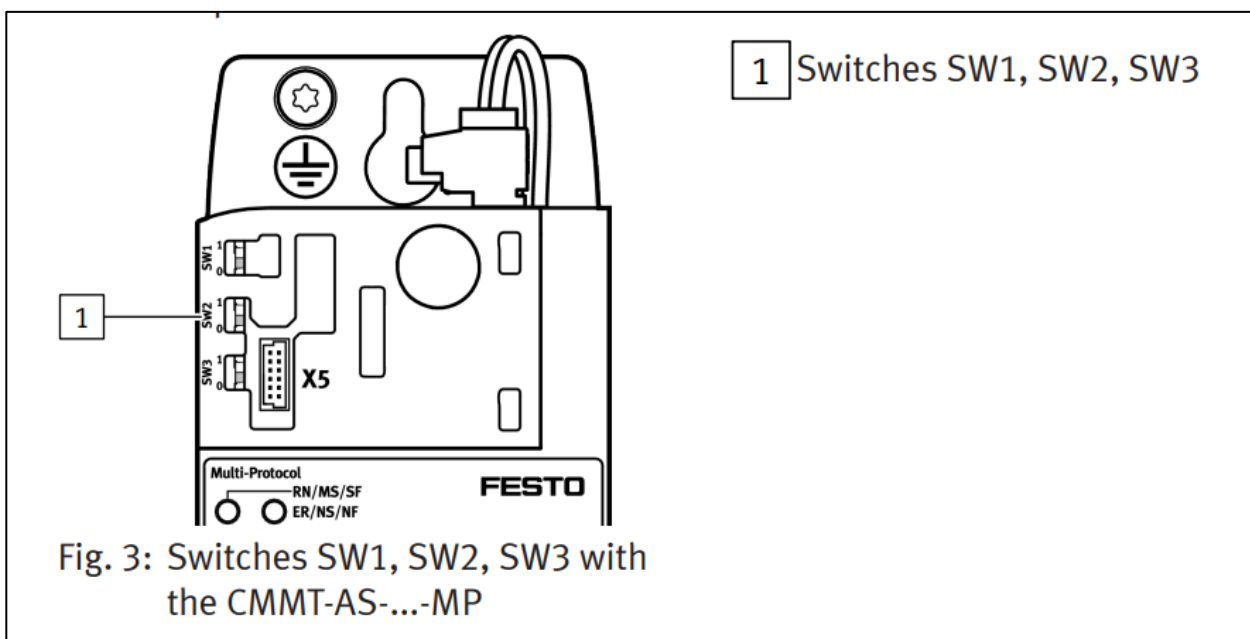
User must configure CMMT-AS controller as per their application requirement. For further clarification please refer the CMMT user manual from below link

https://www.festo.com/net/en_in/SupportPortal/Files/725657/CMMT-AS-SW_manual_2022-04i_8173126g1.pdf

2.3 CMMT-AS Servo Drive Fieldbus Configuration

The CMMT-AS-...-MP product variant supports several bus protocols. The following options are available for setting the protocol:

- Automatic detection by the CMMT-AS-...-MP
- Configuration in the CMMT-AS plug-in
- Direct specification via SW1 to SW3



**Information**

Protocols supported by the firmware version used ---> [Manual - Servo drive - Fieldbus - Function - Device profile - Software](#)

2.3.1 Switch Setting for Bus Protocol

Protocol	Size 3	Size 2	Size 1
Auto (detection or parameterisation)	0	0	0
PROFINET	0	0	1
EtherCAT	0	1	0
EtherNet/IP	0	1	1

Tab. 8: Switch setting bus protocol

The switches can be adjusted with a small slotted head screwdriver.
The switch position is evaluated once when the device is started.

**Note**

- For **Configuration in the CMMT-AS plug-in** all switches should be in 0 position.
- User can use either one method to configure fieldbus protocol for CMMT-AS-...-MP. For example if the user select **EtherNet/IP** protocol via switch setting then must select **EtherNet/IP or Modbus** as **RTE Configuration** in [Chapter 2.2.1 Step 11](#).

3 Starting a Project

3.1 Create New Project File & Configurations of IQ-R Series PLC in Gx Works 3

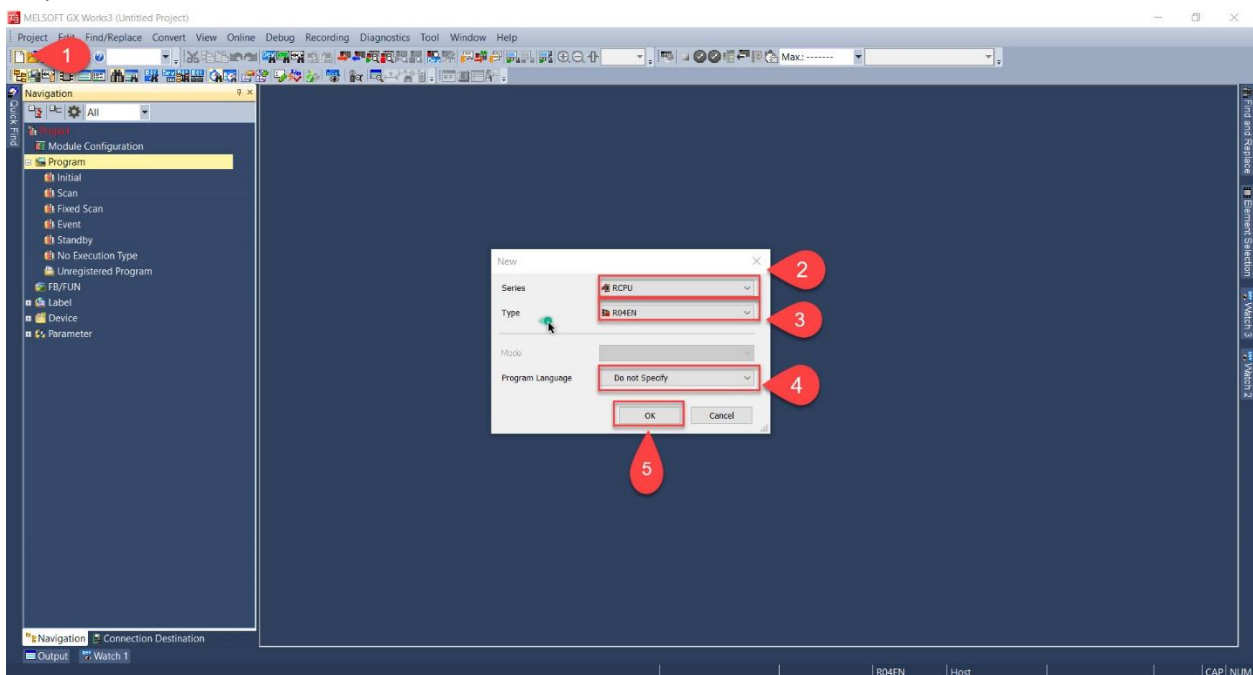
This chapter also covers how to develop projects step-by-step, how to set up communication settings, and how to configure software for things like CMMT EDS file import, EIP-CT for RJ71EIP91, adding Ethernet/IP Module, and adding libraries in the Gx Works 3 programming environment.



Note

- Ignore Chapter 3.1 in case integration of CMMT drive with existing running project file and jump to [Chapter 3.3](#) to set IP address as per requirement.

Step – 1



Open MELSOFT **GX Works3** development software and start a new project by clicking

1. [Project] => [New]
Select **"New Project"** to create a new project.
2. Select **"RCPU"** from **"Series"** drop down option.
3. Select your CPU model from **"Type"** drop down option.
4. Select **"Do not Specify"** as programming language from **"Programming Language"** drop down option.
Or Select preferred programming language, select your settings or use the ones in the example image.
5. Confirm your selection with the button **"OK"**.

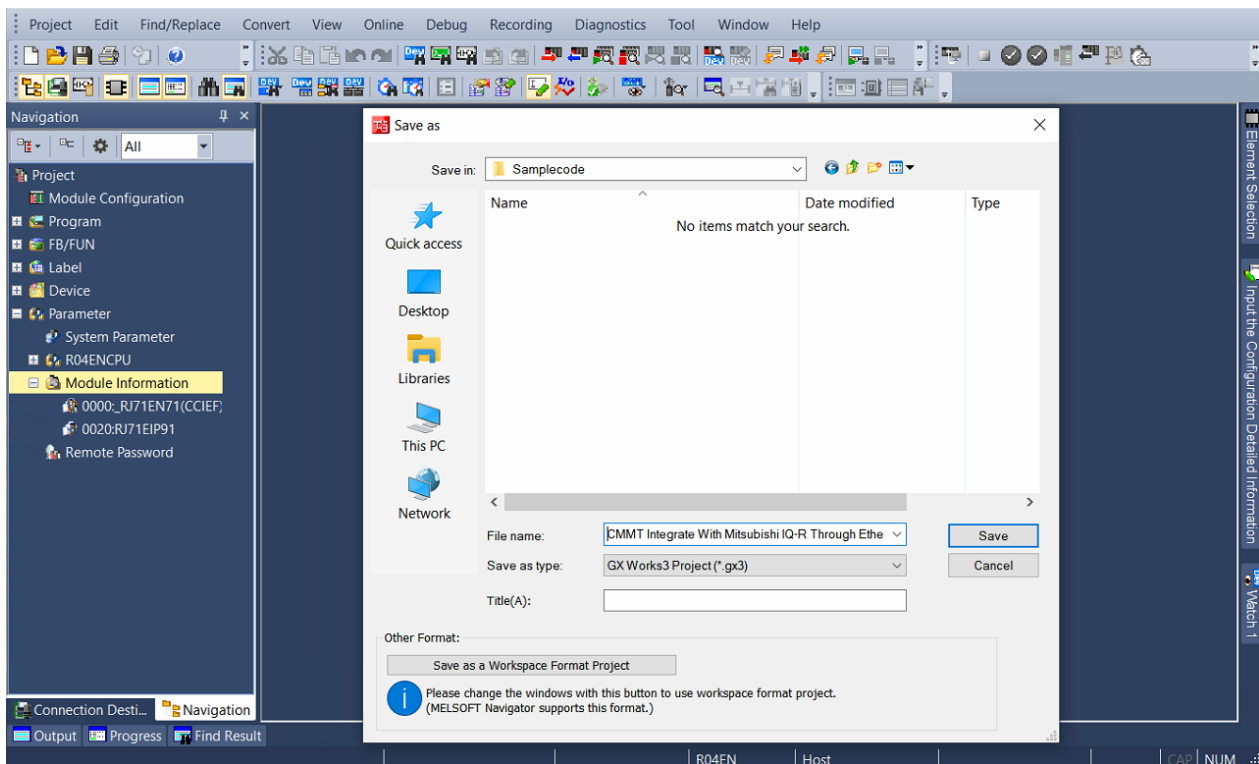


Note

- Users can also create **"New Project"** by clicking shortcut key **CTRL+N** or .
- CPU model and Programming language can be select as per user requirement.
- In this App note as **example** we have considered **R04EN** CPU.

Step – 2

Save the Project

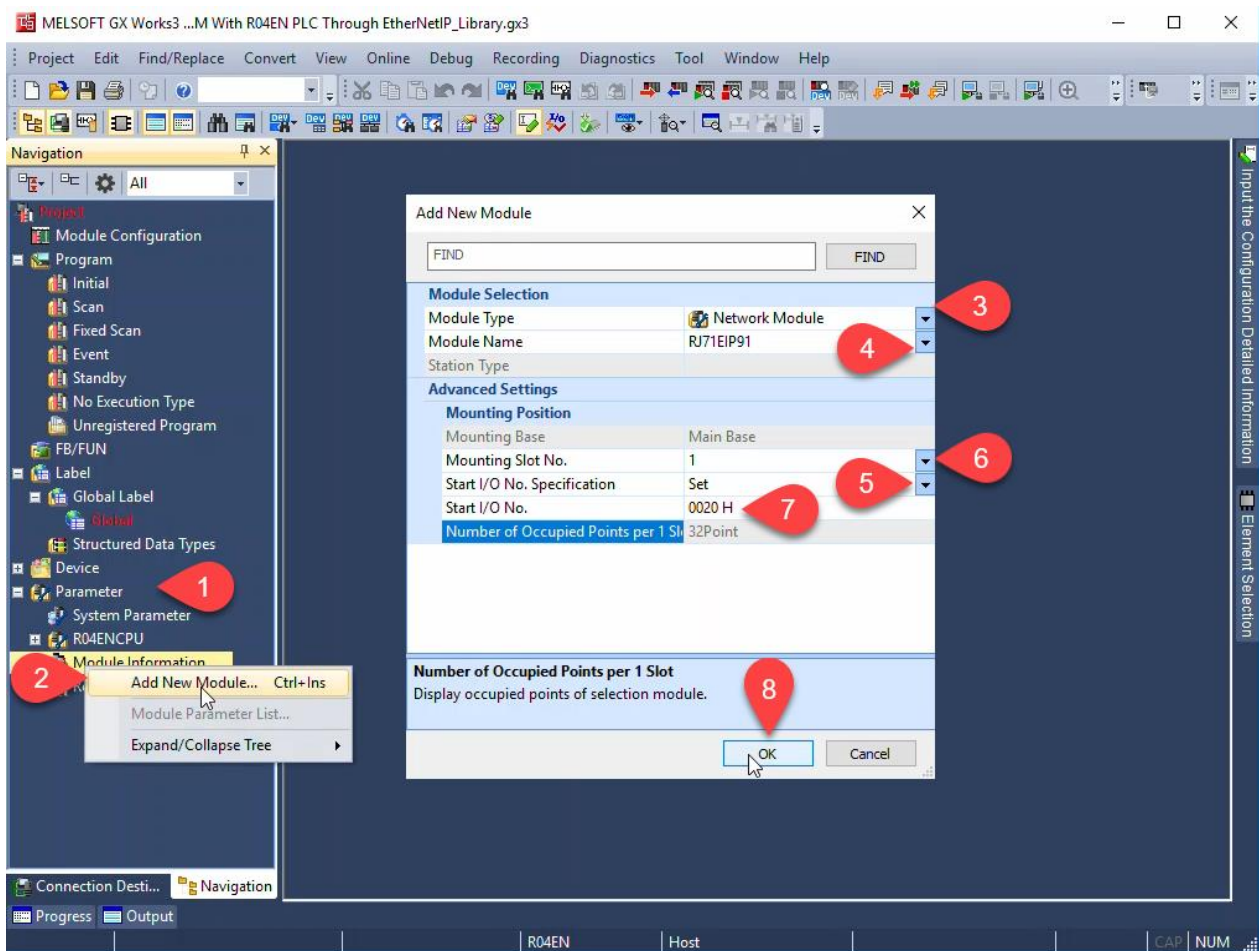


Save the project as desired.

3.2 Adding Ethernet Module in Project file

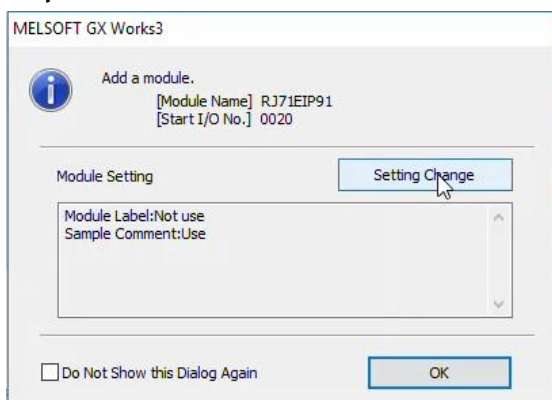
Method – 1

Step – 1



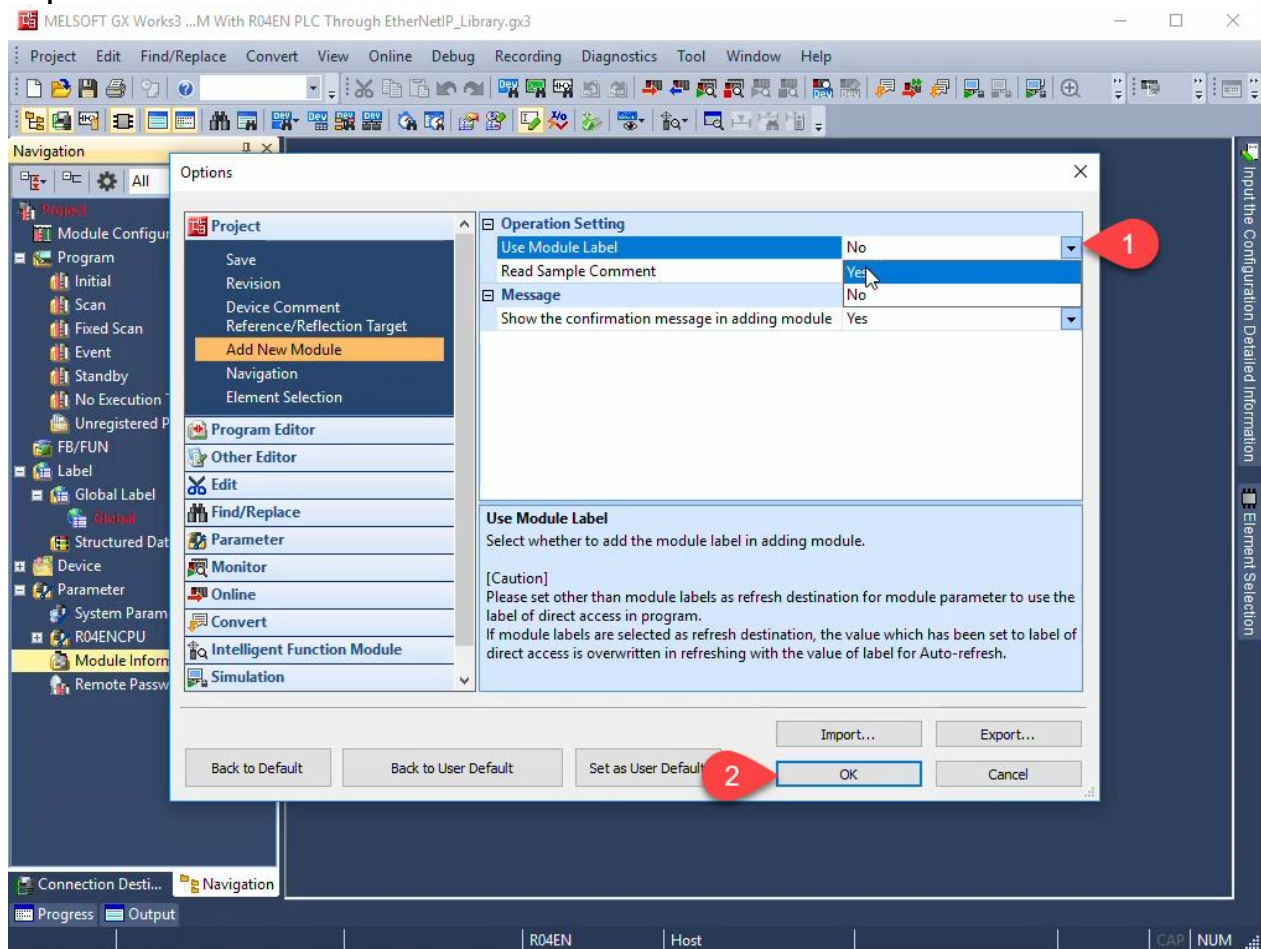
1. Expand **[Parameter]**=> **[R04ENCPU]** => **[Module Information]**
2. Right Click on **Module Information** – **Select Add New Module**.
3. Select **"Network Module"** under Module Type.
4. Select **"RJ71EIP91"** under Module Name.
5. Select **"Set"** under Start I/O No. Specification.
6. Select **"1"** under the Mounting Slot No. (Set the Slot No based on the User arrangement)
7. Enter value **"0020H"** under the Start I/O No. (If user knows the Start I/O No, only then enter the value)
8. Confirm your selection with the button **"OK"**.

Step – 2



Click on Setting Change

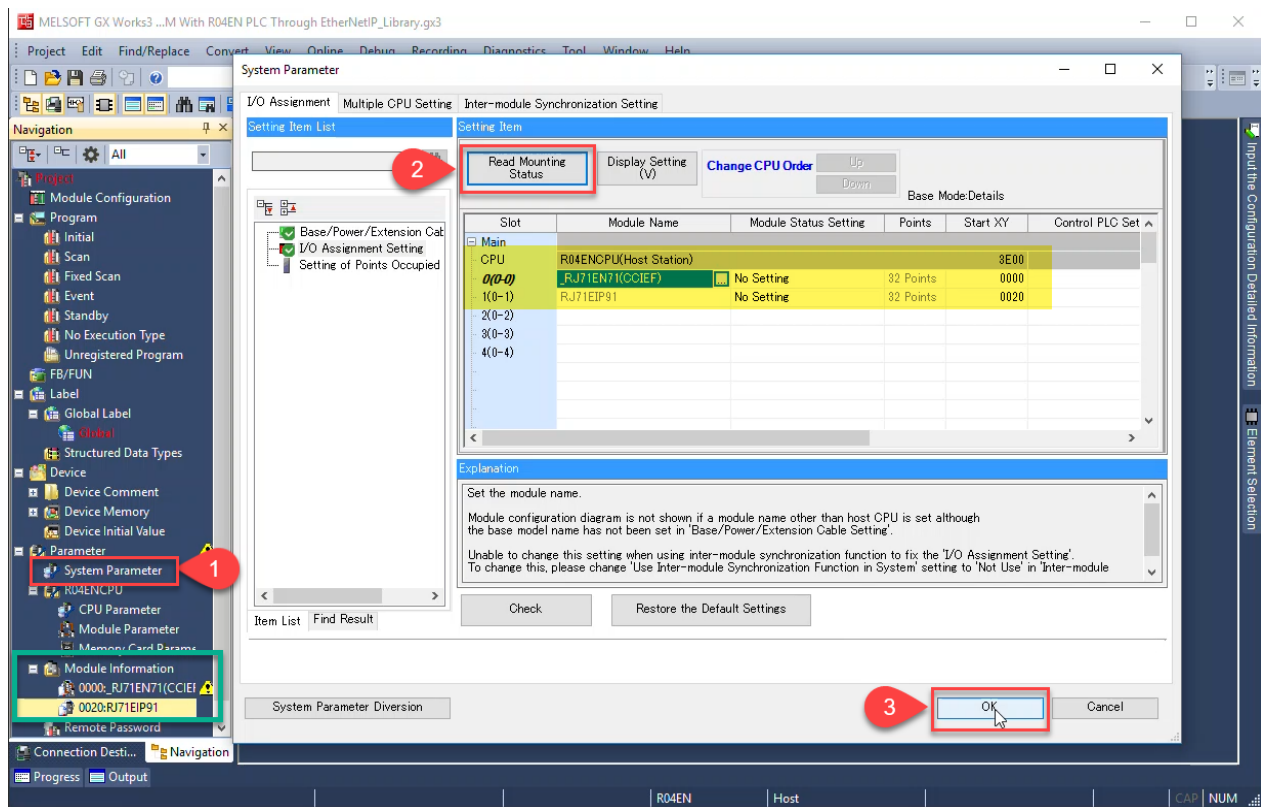
Step – 3



1. Select **“Yes”** under Use Module Label.
2. Confirm your selection with the button **“OK”**.

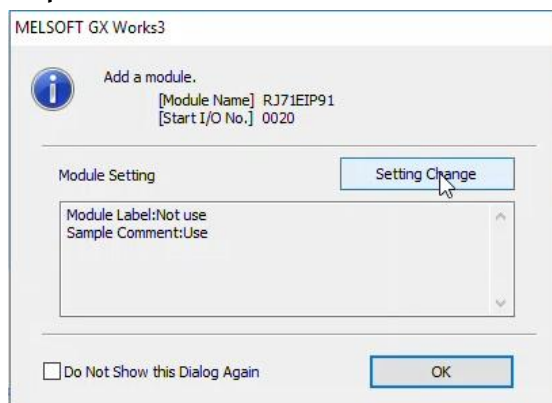
Method – 2

Step – 1



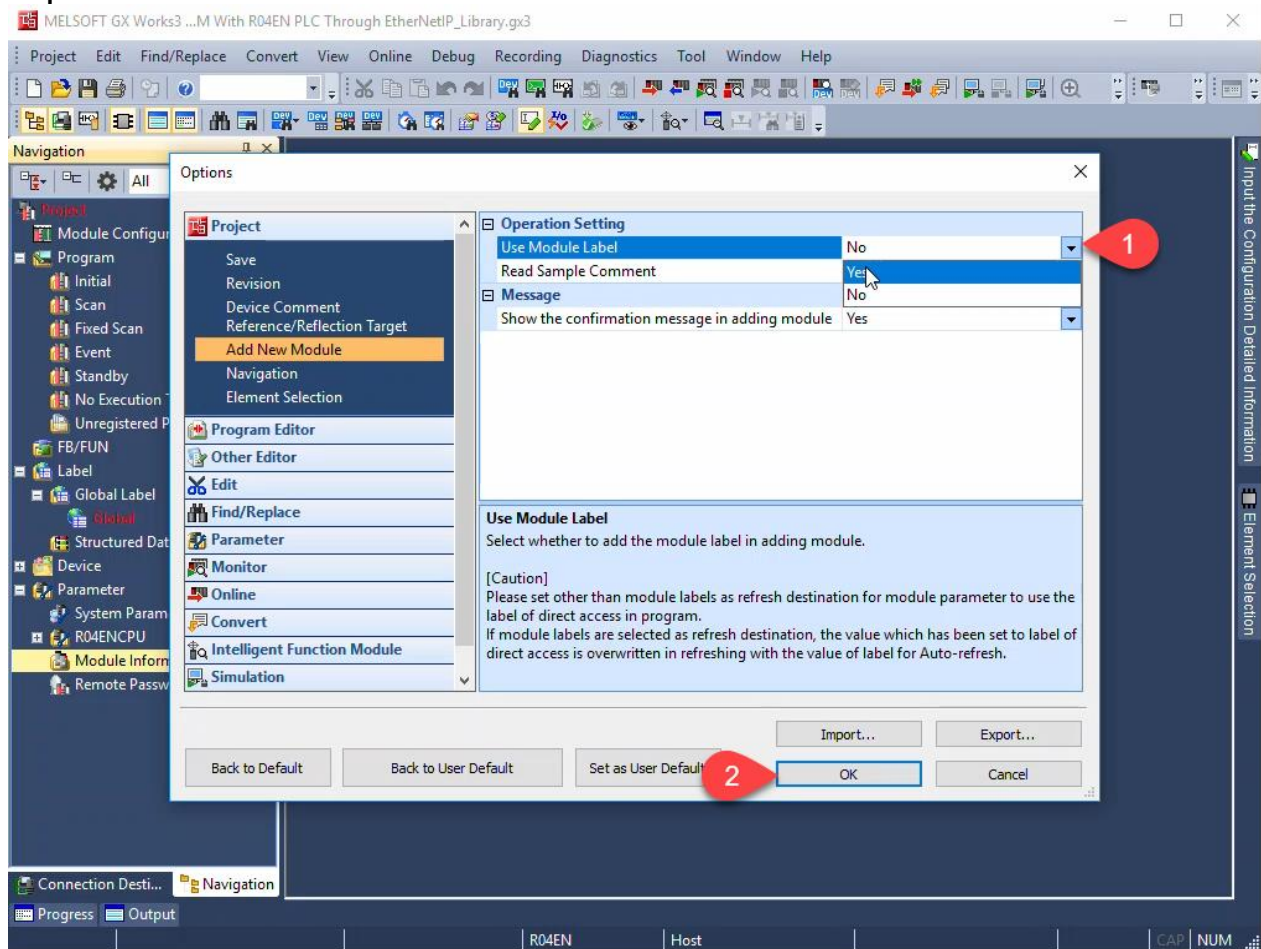
1. Expand **[Parameter]** Double Click on **“System Parameter”**.
2. Click **“Read Mounting Status”**.
 - a. Click **“Yes”**.
3. Confirm your selection with the button **“OK”**.

Step – 2



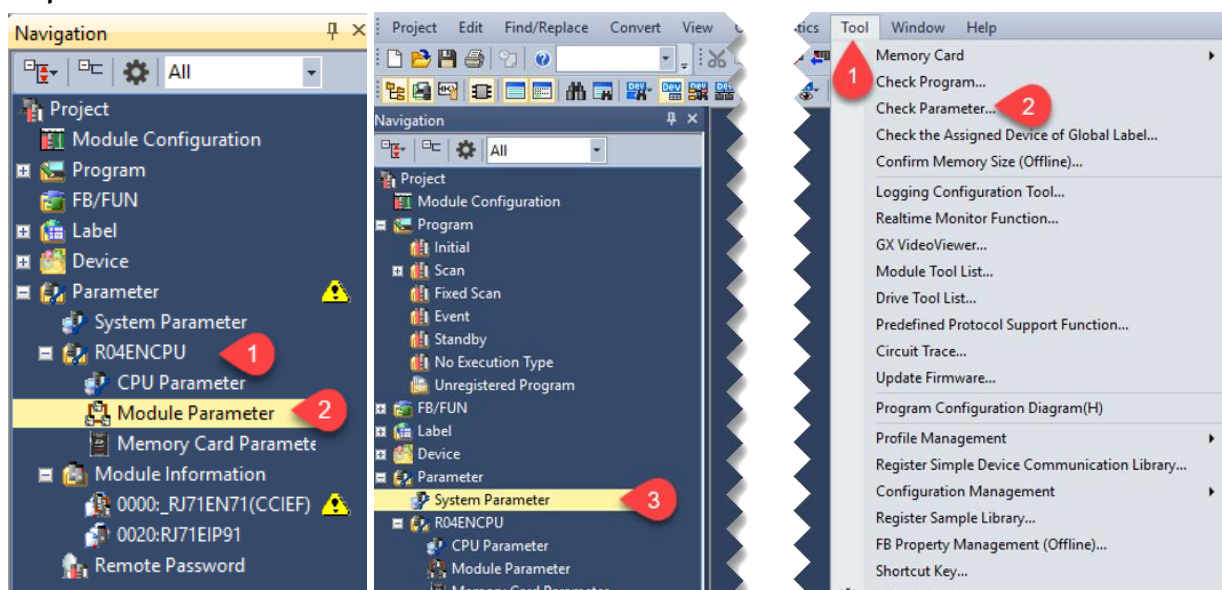
Click on Setting Change

Step – 3



1. Select “Yes” under Use Module Label.
2. Confirm your selection with the button “OK”.

Step – 4



1. Click on **Tool**.
2. Click **Check Parameter**.
3. Check whether Warning symbol cleared.

3.3 Ethernet port parameters

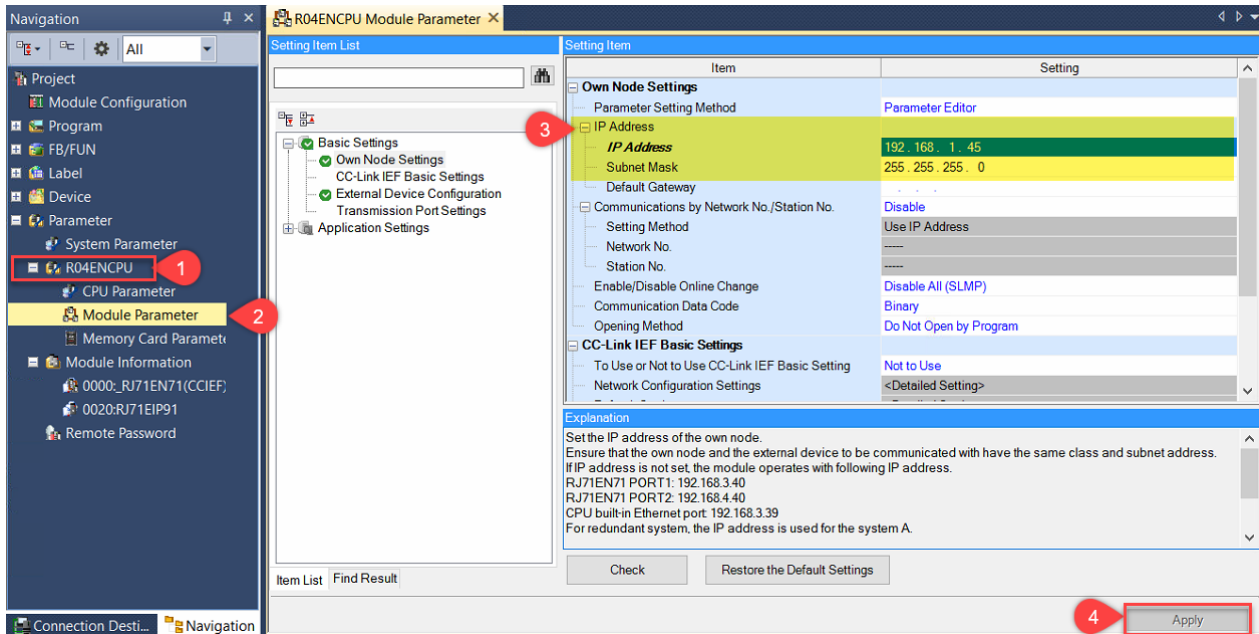


Note: Follow this step based on the PLC preferred.

R04ENCPU

- Double click the [Module Parameter] under [R04ENCPU] => [Parameter] from the project tree to access the Ethernet port Parameter page.

3.3.1 IP address of PLC



- Expand “R04ENCPU” from “Parameter” option.
- Double click “Module Parameter” option.
- Enter the “IP Address & Subnet Mask” of the PLC as per requirement. as example shown in above image.
- Click “Apply” button to confirm the configuration.

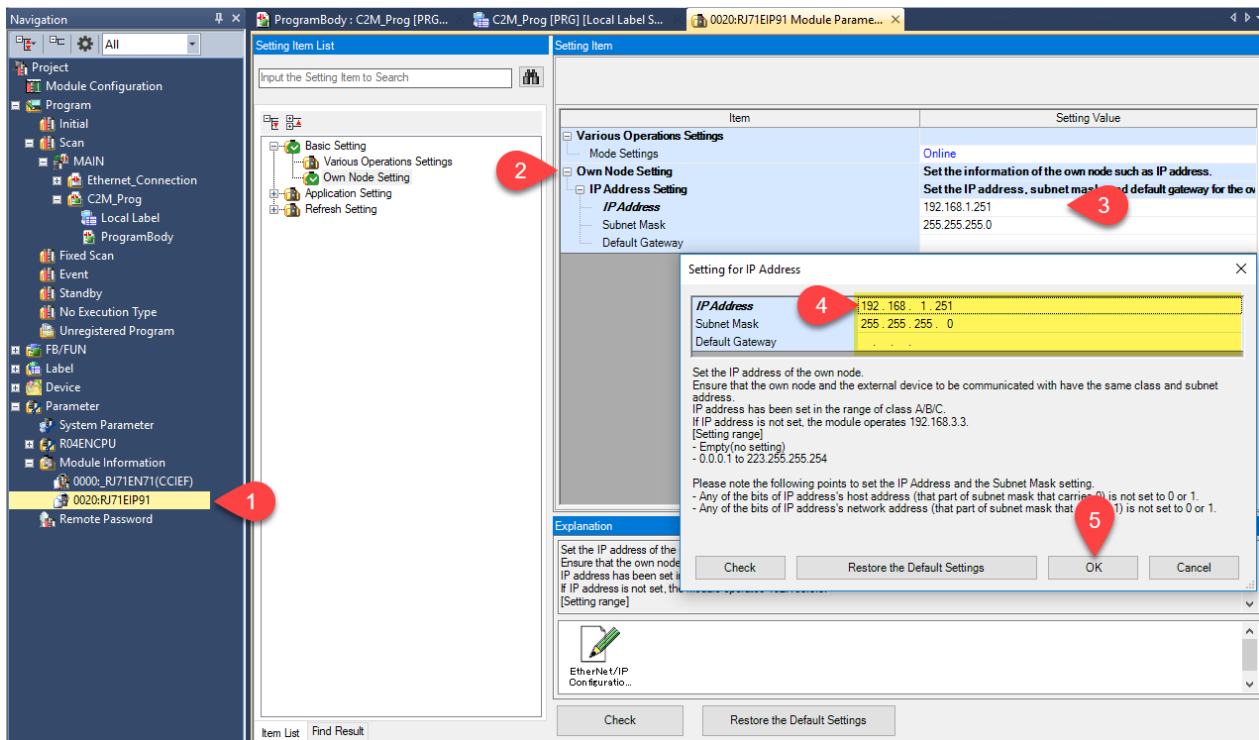
3.3.2 IP address of Ethernet Module



Note: Follow this step based on the PLC preferred.

R04ENCPU

- Double click the [RJ71EIP91] under [Module Information] from the project tree to access the Ethernet port Parameter page.



1. Expand **“Module Parameter”** and Double click **“RJ71EIP91”**.
2. Expand **“Own Node Setting”**
3. Double click **“IP Address blank area”**.
4. Setting for IP Address window will open
Enter the **“IP Address & Subnet Mask”** of the EtherNet Module as per requirement. as example shown in above image.
5. Click **“Ok”** button to confirm the configuration

3.4 Configuring RJ71EIP91 using EtherNet/IP Configuration Tool (EIP – CT) by Adding EDS File in EtherNet/IP Devices.



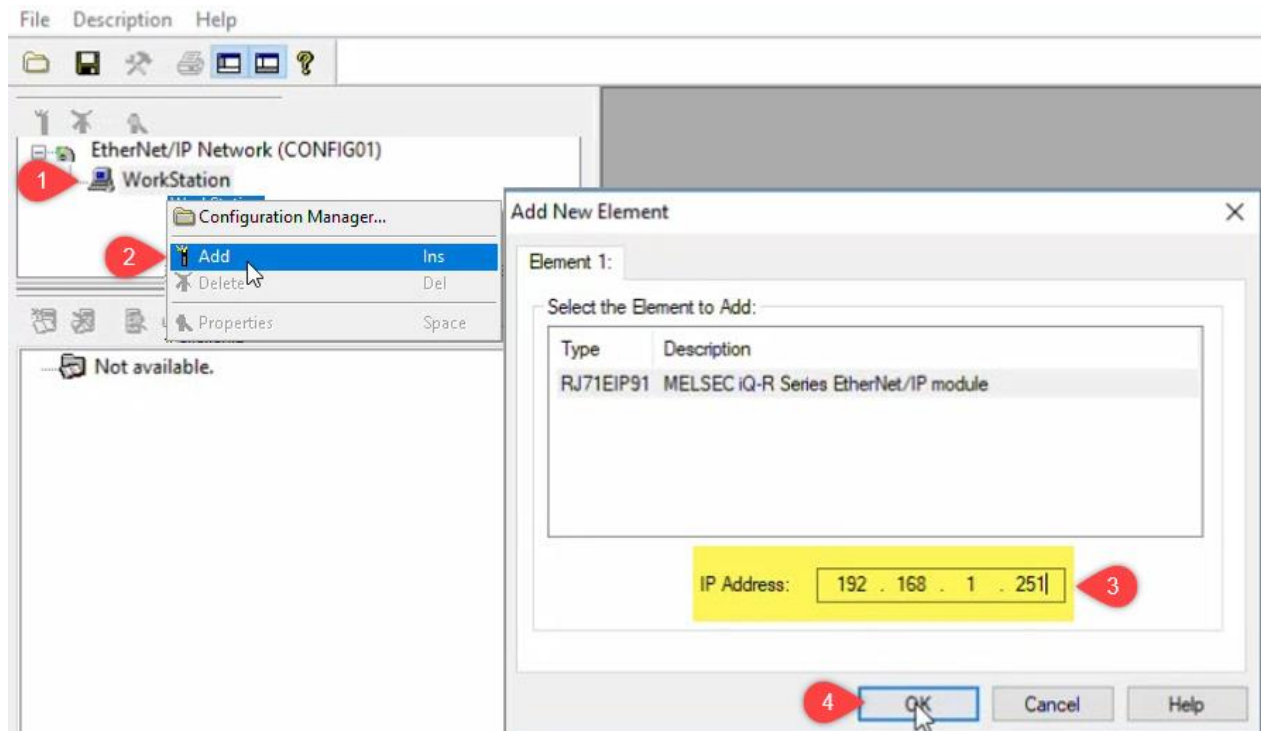
Note

- CMMT EDS file is available with this application note folder & Also user can able to download it from Festo Support Portal website.
- Skip this chapter if Generic EDS will be used to configure RJ71EIP91 and proceed to [Chapter 3.5](#)

Open the EIP-CT for RJ71EIP91 1.00A from following path **Windows Start > Melsoft > EIP-CT for RJ71EIP91 1.00A**

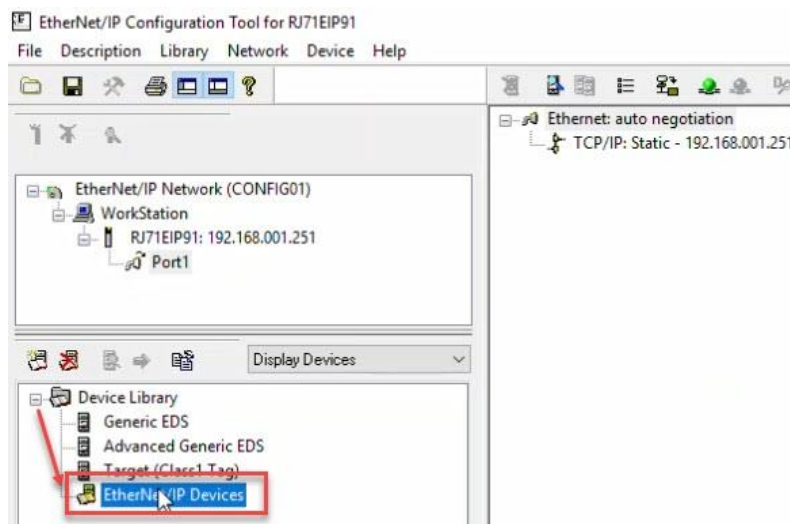
EIP Configuration Tool for RJ71EIP91

Step – 1

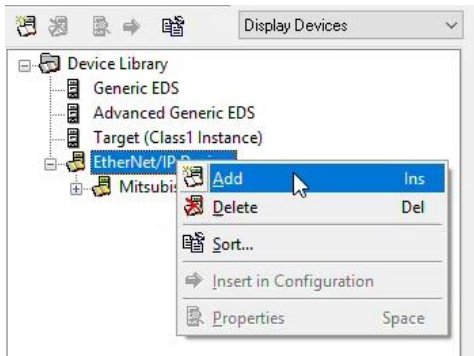


1. Right Click on Workstation under EtherNet/IP Network tree.
2. Click on “Add”
3. Enter the same IP Address used for RJ71EIP91. (Refer [Chapter 3.3.2](#))
4. Click the “OK”.

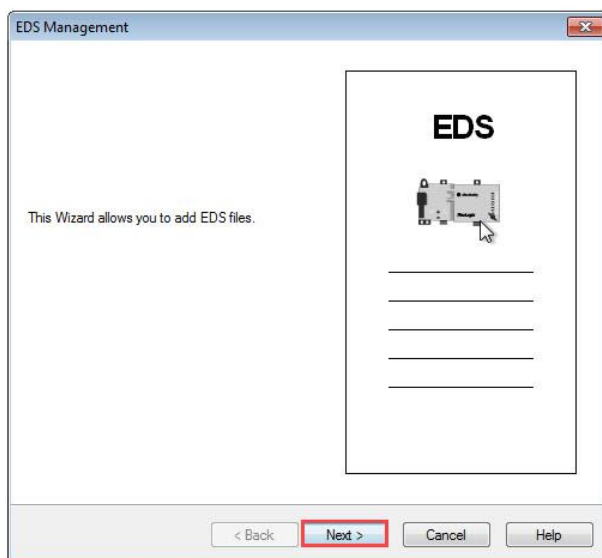
Step – 2



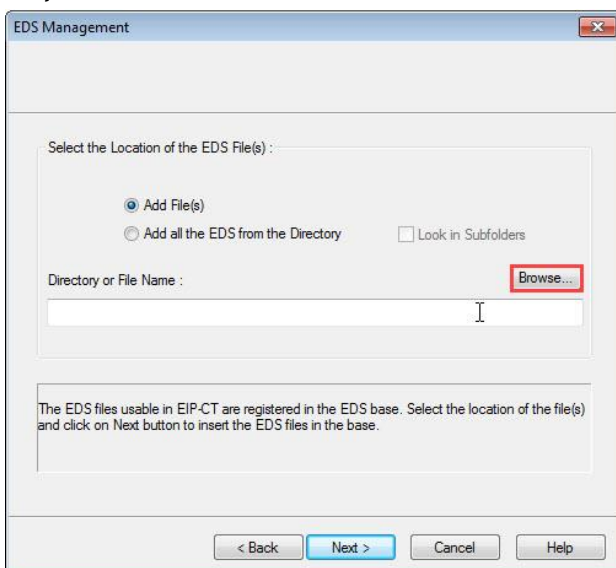
- Expand the “**Device Library**” from tool bar.
- Select on “**EtherNet/IP Devices**” from tools drop-down menu.

Step – 3

- Right Click on **“EtherNet/IP Devices”**
- Select **“Add”** for Add EDS file.

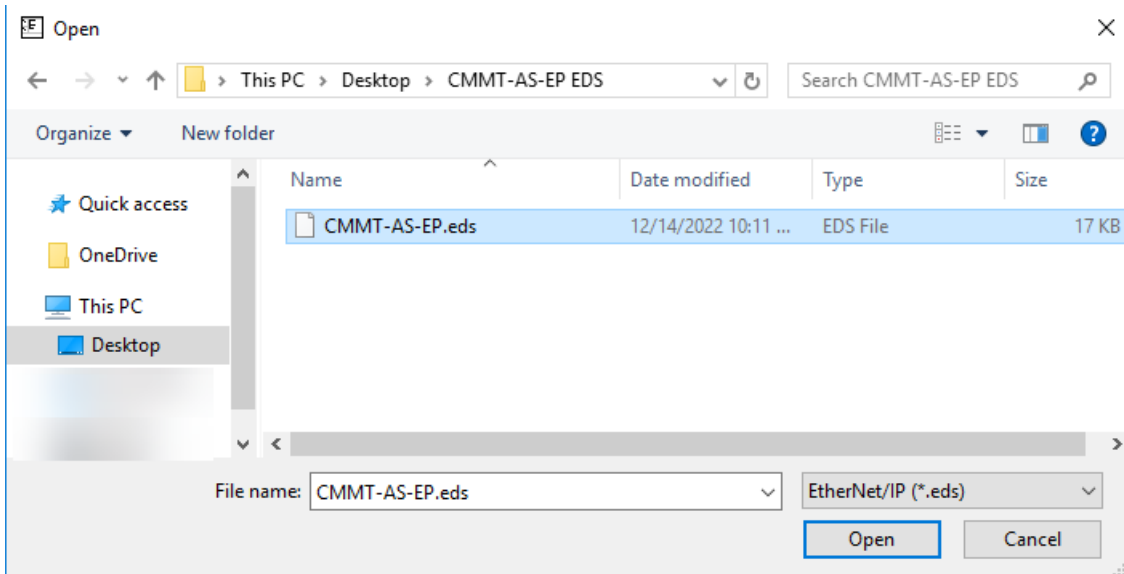
Step – 4

- Click the **“Next”**.

Step – 5

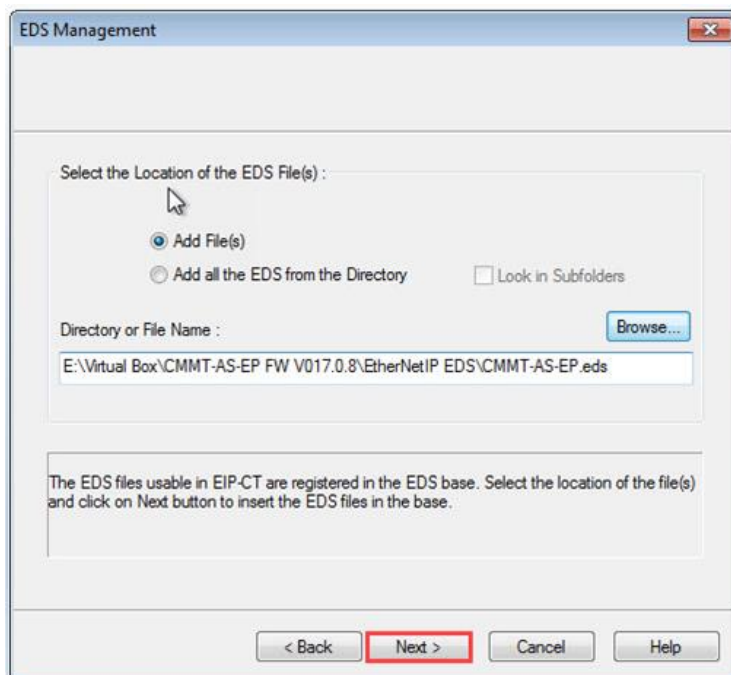
- Click the **“Browse”** for selecting the EDS file from the Drive.

Step – 6

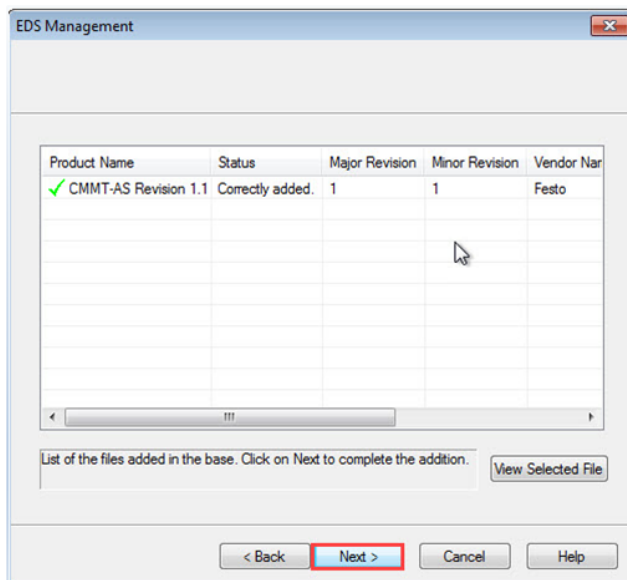


- Select “**CMMT-AS-EP.eds**” from the Drive.
- Click “**Open**”.

Step – 7

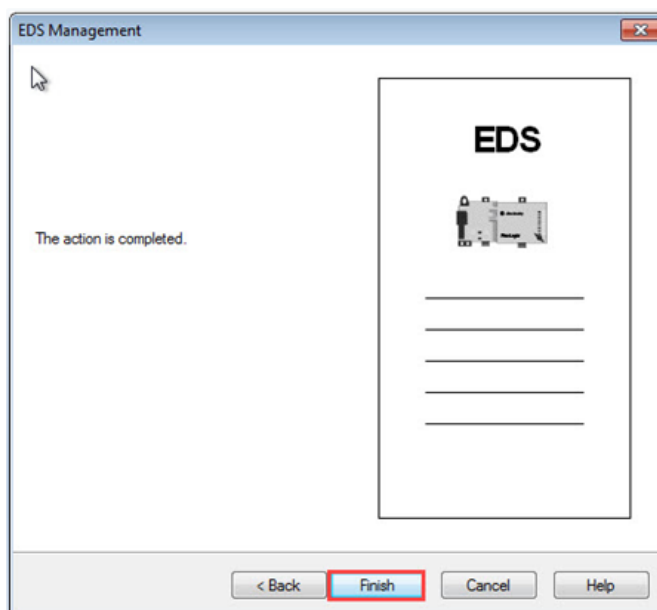


- Click “**Next**” from EDS installation wizard.

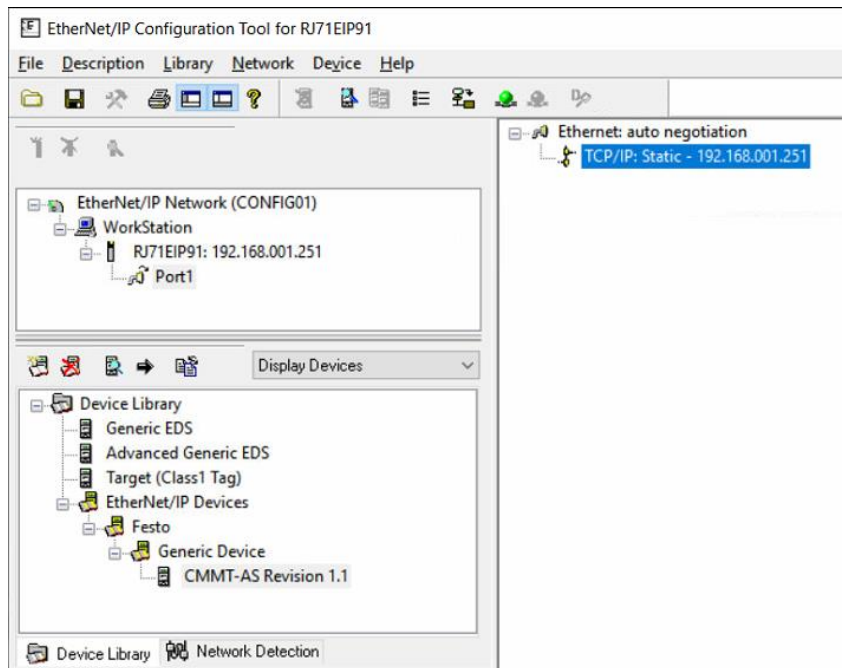


- Then Click “**Next**”.

Step – 8

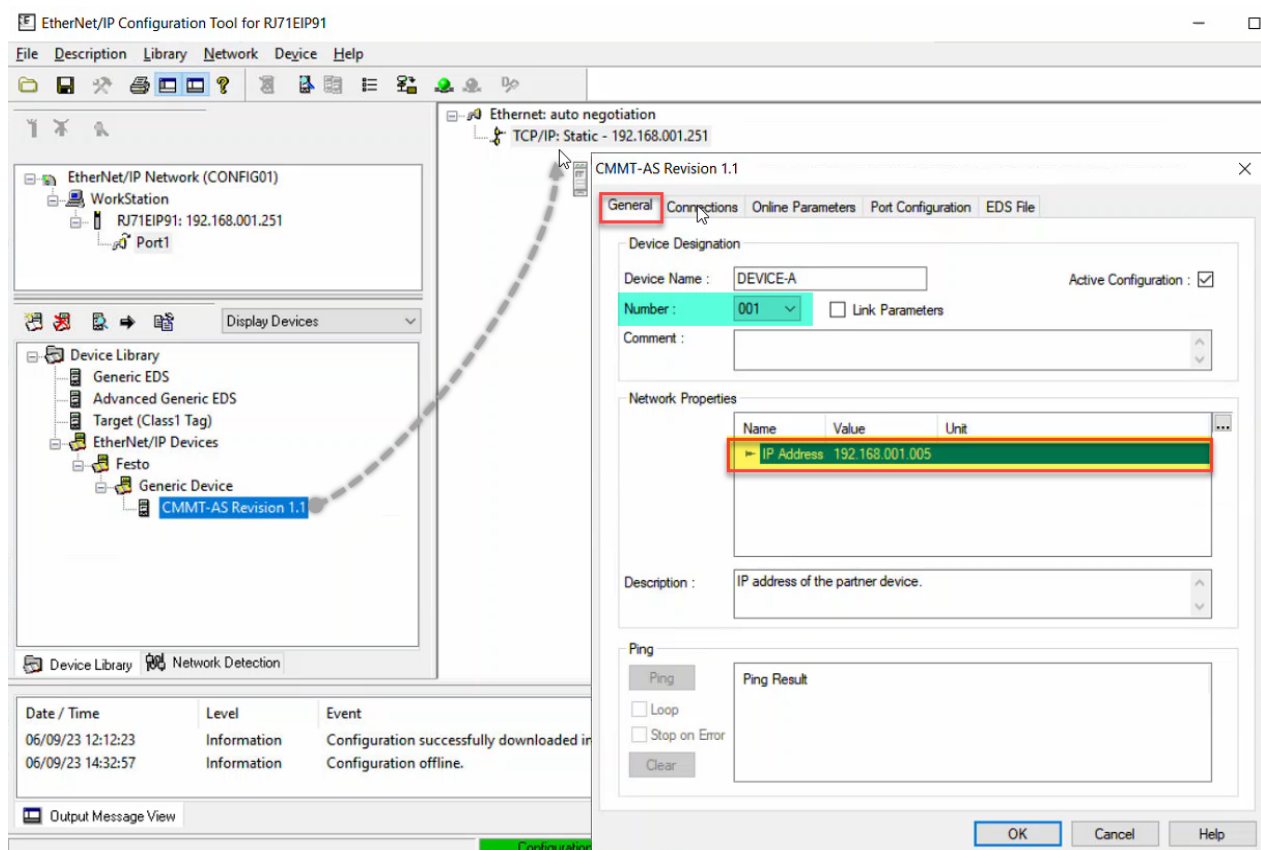


Starting a Project



- Click **“Finish”** from EDS installation wizard.
- Now **“CMMT-AS”** EDS File added to the EIP Configuration Tool.
- Set the **“TCP/IP :Static”** address (Which is your RJ71EIP91 Module IP Address)

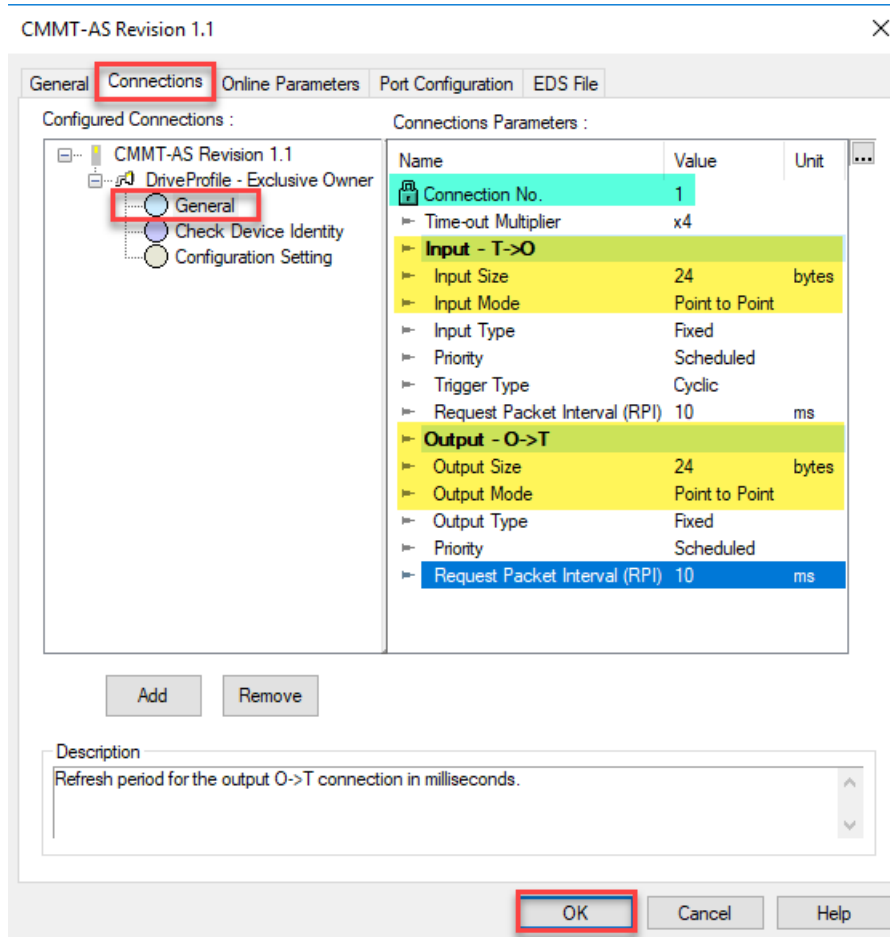
Step – 9



- Drag the **“CMMT-AS”** from tool bar.
- Select **“General”** and Set the **“IP Address”** of CMMT Ethernet/IP Address. Refer [Chapter 2.2.2 Step – 5, Point 3](#)
- Click on **“OK”**.

**Note**

- Here the IP address to be noted for using in [chapter 5.2.1](#) the Input “dwTargetIP”
- Here the Number to be noted for using in [chapter 5.2.1](#) the Input “wTargetDeviceNO”

Step – 10

- Click the “**Connections**” from tool bar.
- Select “**General**” from DriveProfile - Exclusive Owner drop-down menu.
- Set **Point to Point** Input mode.
- Set minimum value in Request Packet Interval(RPI), Example – 10 ms (Note: not to keep 0.5ms).
- Click on “**OK**”.

**Note**

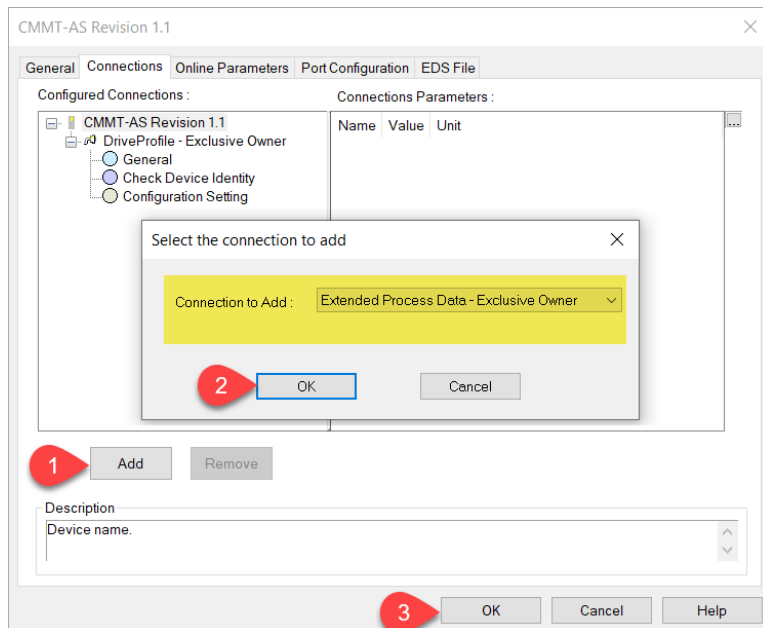
- Here the Connection No to be noted for using in [chapter 4.1.1](#)

Step – 11: Extended Process Data Connection



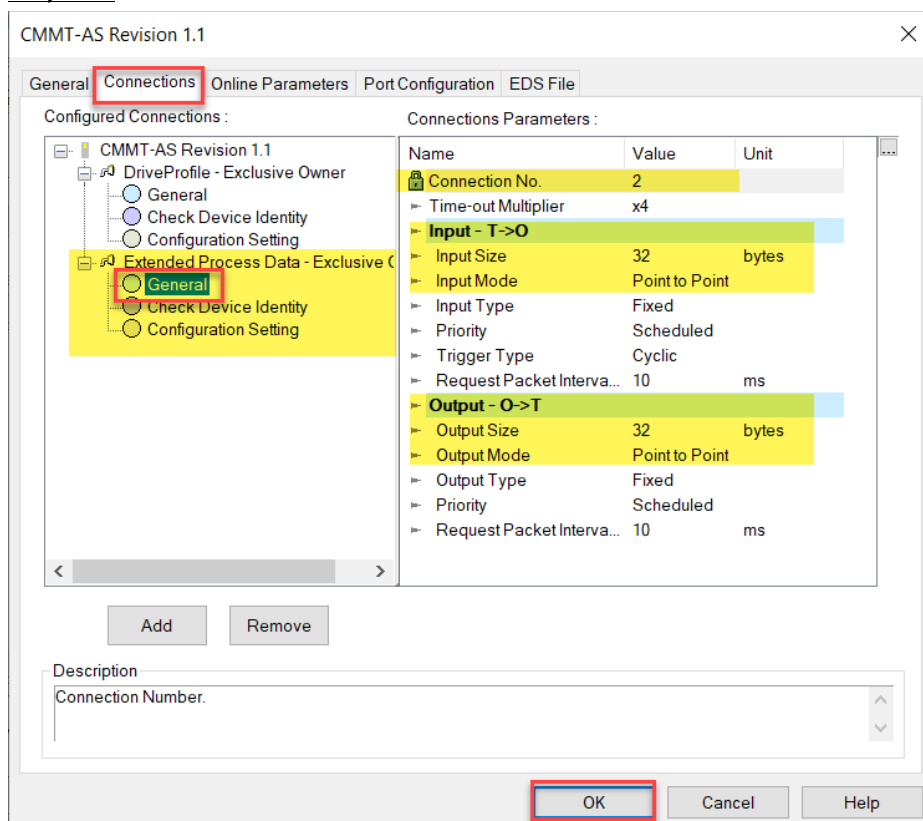
Note

- follow this Step 11 only in case of using Extended Process Data, else jump to Step 13



1. Click on Add.
2. On click, Select the connection to add pop-up window will appear on screen. Make sure Extended Process Data - Exclusive Owner is selected, Click on “OK”.
3. Click on “OK”.

Step – 12



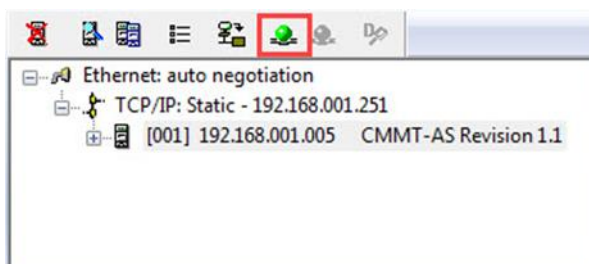
- Click the “**Connections**” from tool bar.
- Select “**General**” from Extended Process Data - Exclusive Owner drop-down menu.
- Set **Point to Point** Input mode.
- Set minimum value in Request Packet Interval(RPI), Example – 10 ms (Note: not to keep 0.5ms)
- Click on “**OK**”.

**Note**

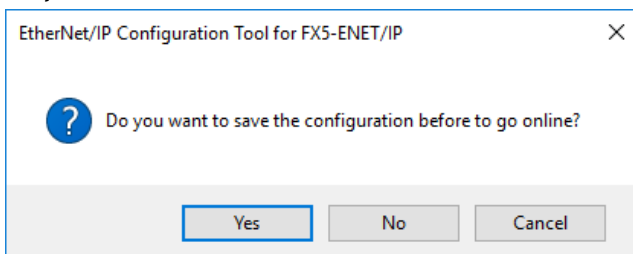
- Here the Connection No to be noted for using in [chapter 7.3.1](#)

Step – 13**Note**

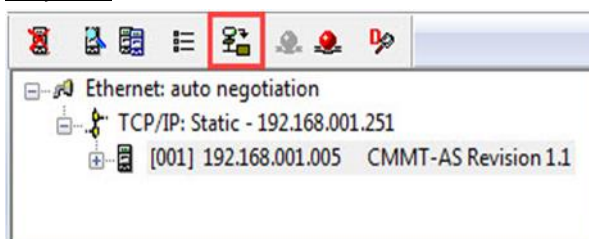
- Ensure the New IP address entered is correct.
- First establish connection with PLC, download the IQ-R EtherNet/IP setting through GX work3.



- Click the “**Online Command**” from tool bar.

Step – 14

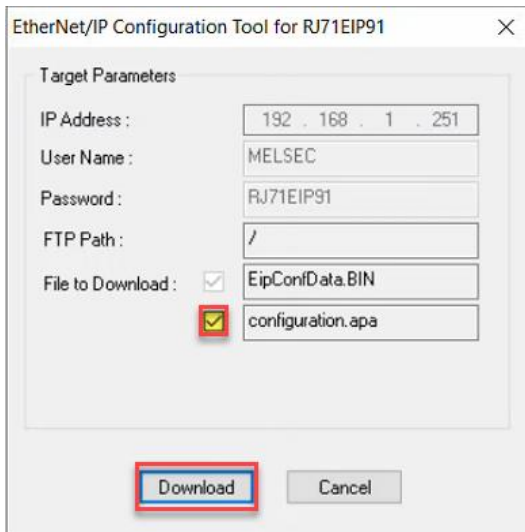
- Click the “**YES**”.

Step – 15

- Click the “**Download**” from tool bar.

Step – 16

If connection is established then below image will pop-up on screen.



- Select **“Configuration.apa”**.
- Click the **“Download”**.

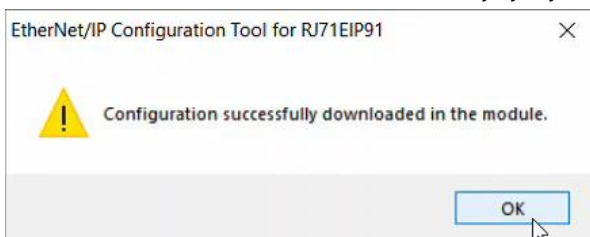
If connection is not established then below window will pop-up on screen.



- Follow the note given in step – 13 and repeat step – 15.

Step –17

After successful download below window pop-up on screen



- Click the **“OK”**.

3.5 Configuring RJ71EIP91 using EtherNet/IP Configuration Tool (EIP – CT) by Adding Generic EDS.

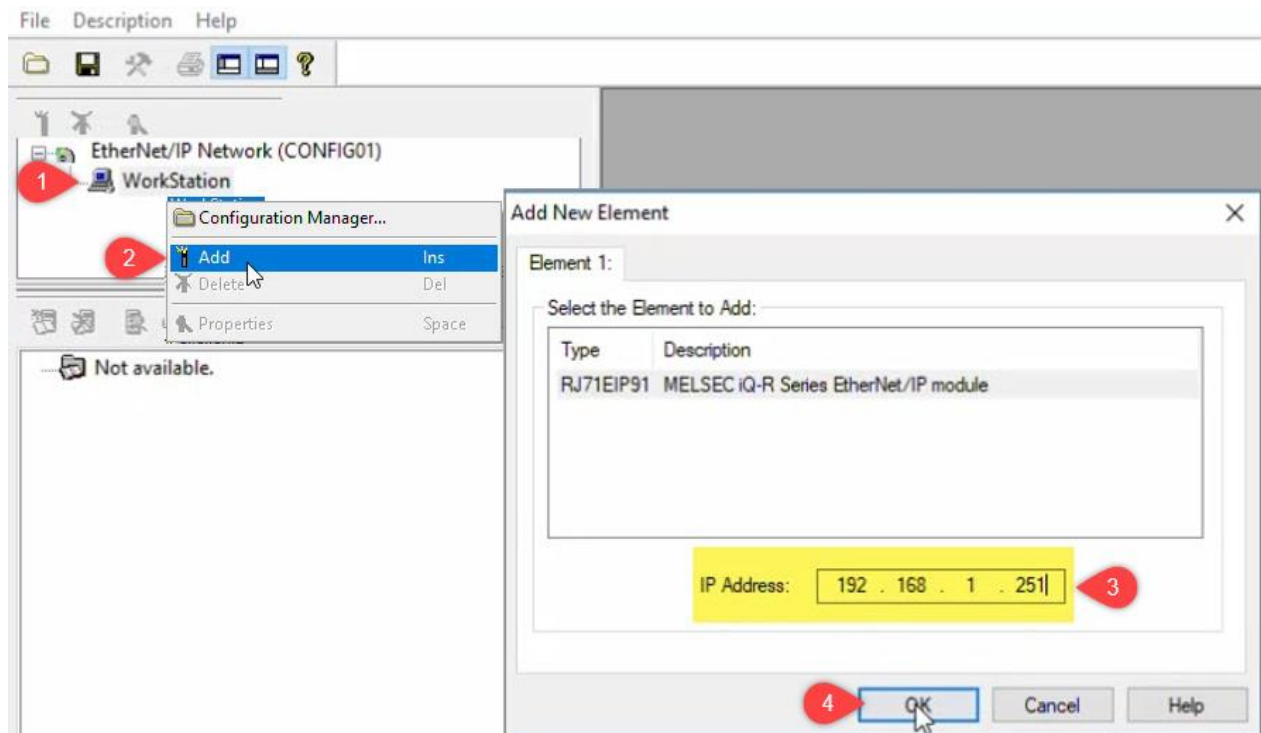


Note

- Skip this chapter if using EDS file to add in EtherNet/IP Device to configure RJ71EIP91 and proceed to [Chapter 3.4](#)

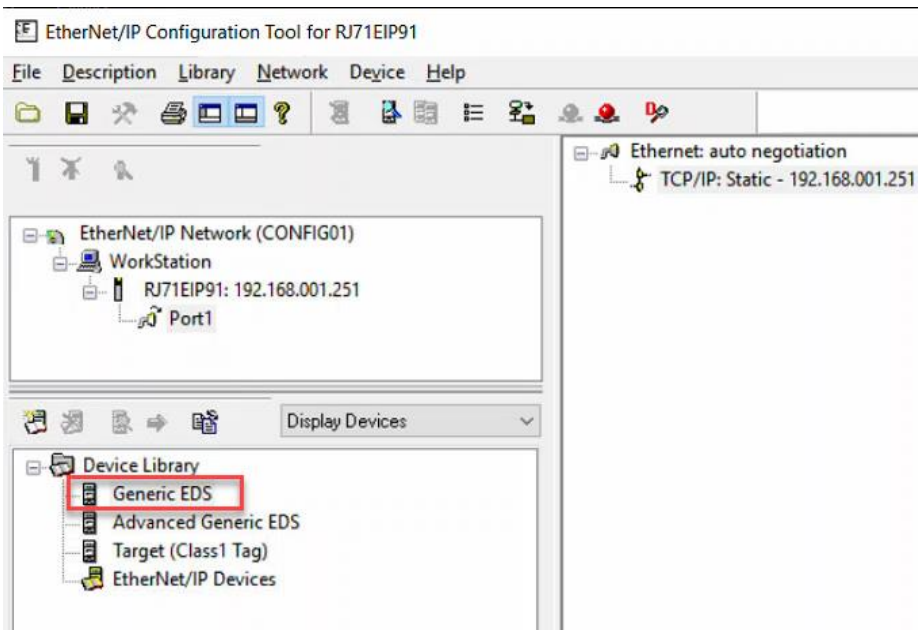
Open the EIP-CT for RJ71EIP91 1.00A from following path **Windows Start > Melsoft > EIP-CT for RJ71EIP91 1.00A**

Step – 1



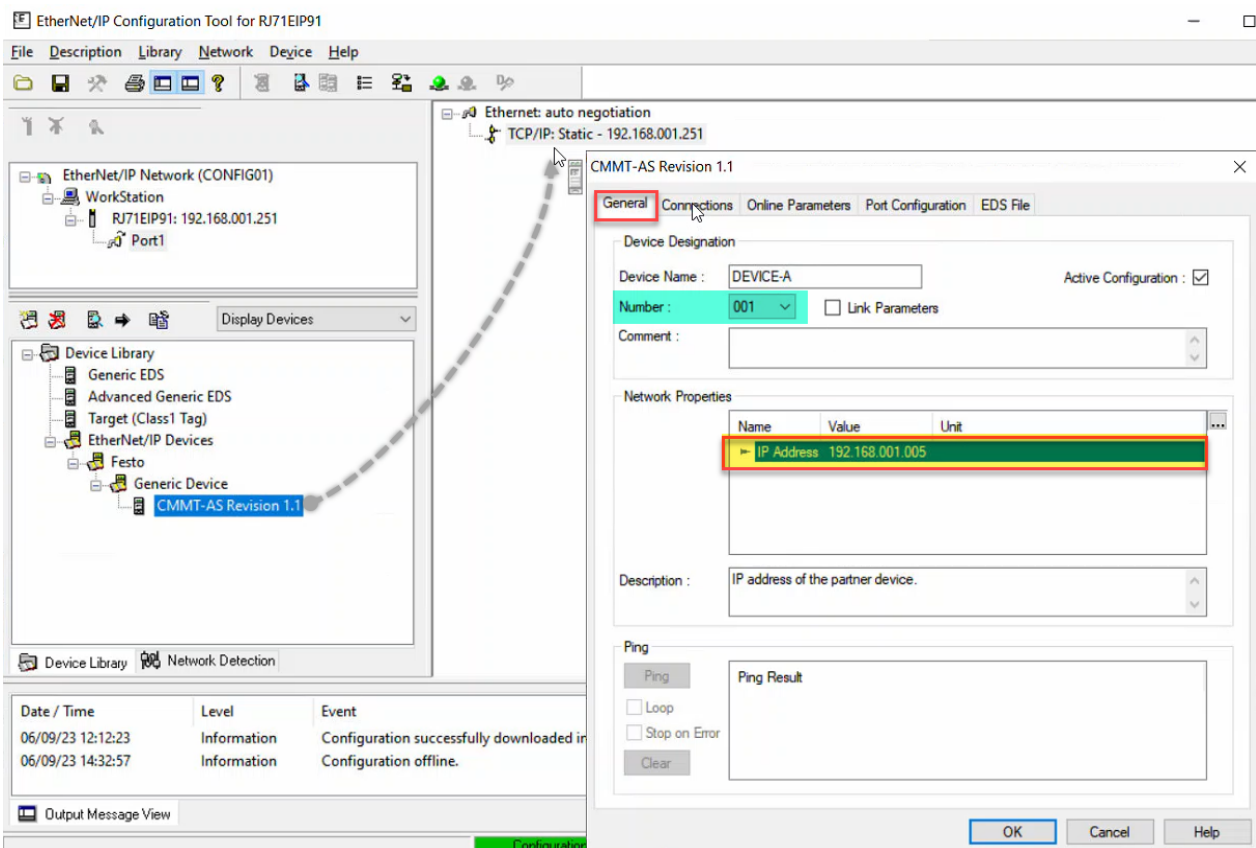
1. Right Click on Workstation under Ethernet/IP Network tree.
2. Click on “Add”
3. Enter the same IP Address used for FX5-ENET/IP. (Refer [Chapter 3.3.2](#))
4. Click the “OK”.

Step – 2



- Expand the “**Device Library**” from tool bar.
- Select on “**Generic EDS**” from tools drop-down menu.

Step – 3

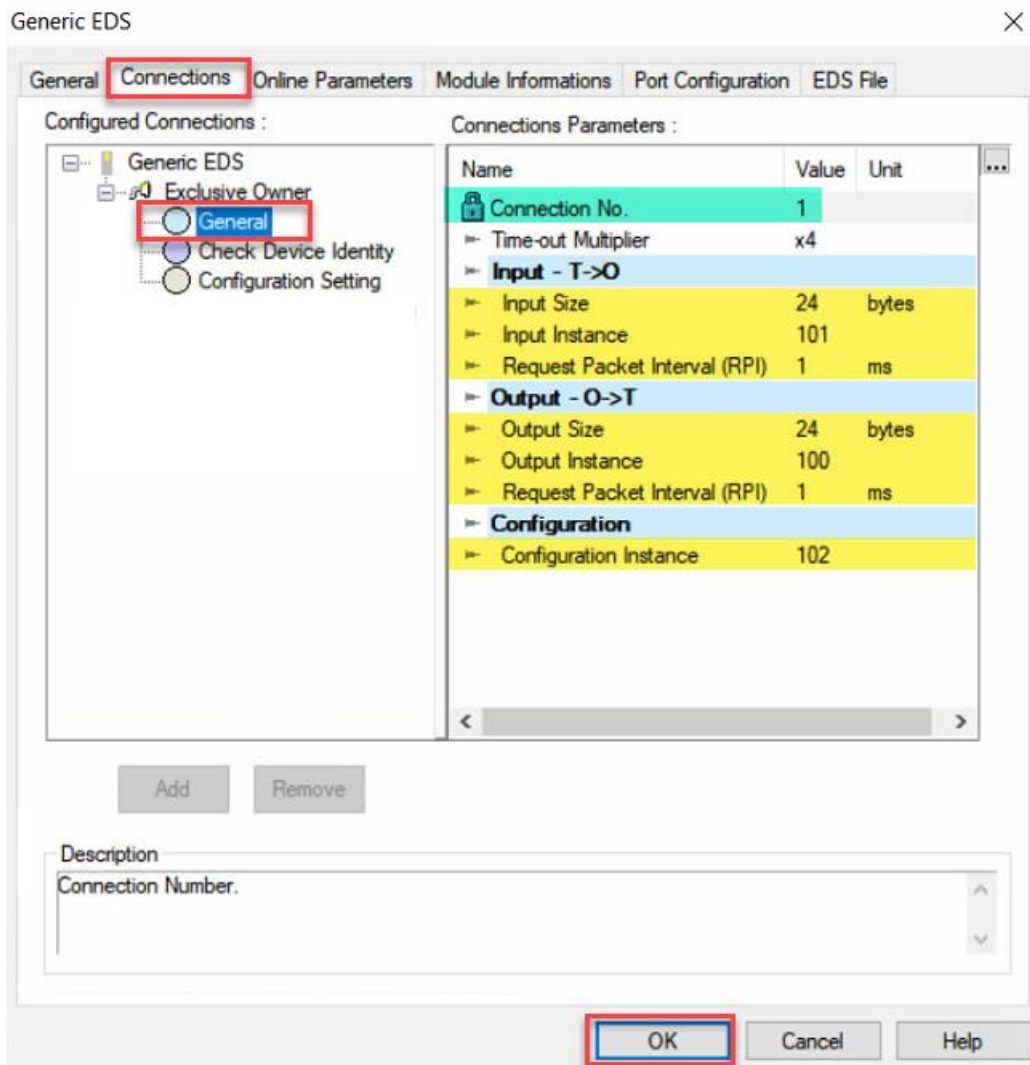


- Drag the “**CMMT-AS**” from tool bar.
- Select “**General**” and Set the “**IP Address**” of CMMT Ethernet/IP Address. Refer [Chapter 2.2.2 Step – 5, Point 3](#)
- Click on “**OK**”.



Note

- Here the Number highlighted in green colour to be noted for using in [chapter 4.1.1](#)

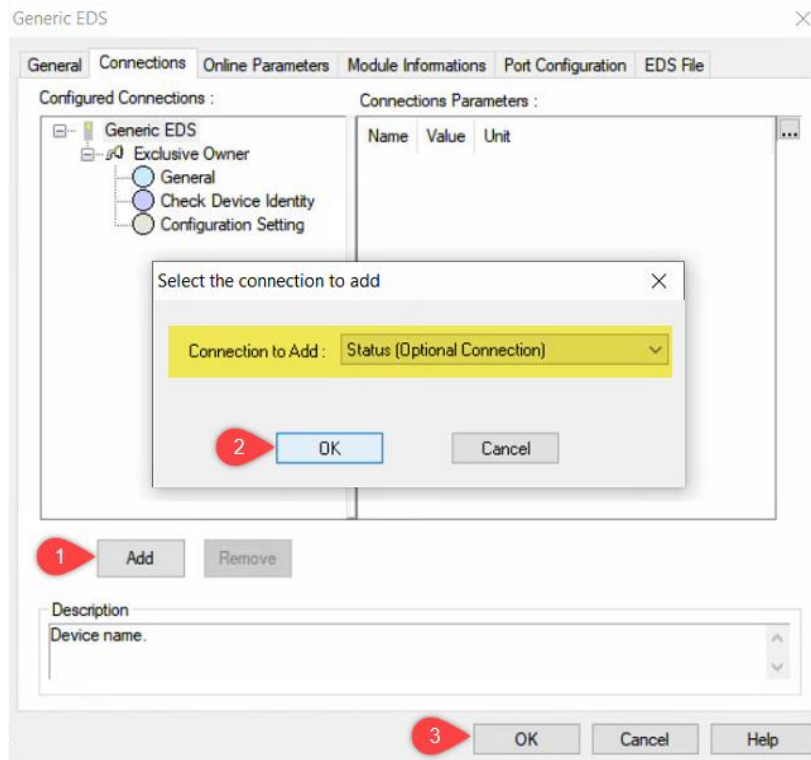
Step – 4

- Click the **“Connections”** from tool bar.
- Select **“General”** from Exclusive Owner drop-down menu.
- Input- T->O:
Input Size : 24 Bytes
Input Instance : 101
- Output- O->T:
Input Size : 24 Bytes
Input Instance : 100
- Configuration:
Configuration Instance : 102
- Set minimum value in Request Packet Interval(RPI), Example – 1 ms (Note: not to keep 0.5ms)
- Click on **“OK”**.

**Note**

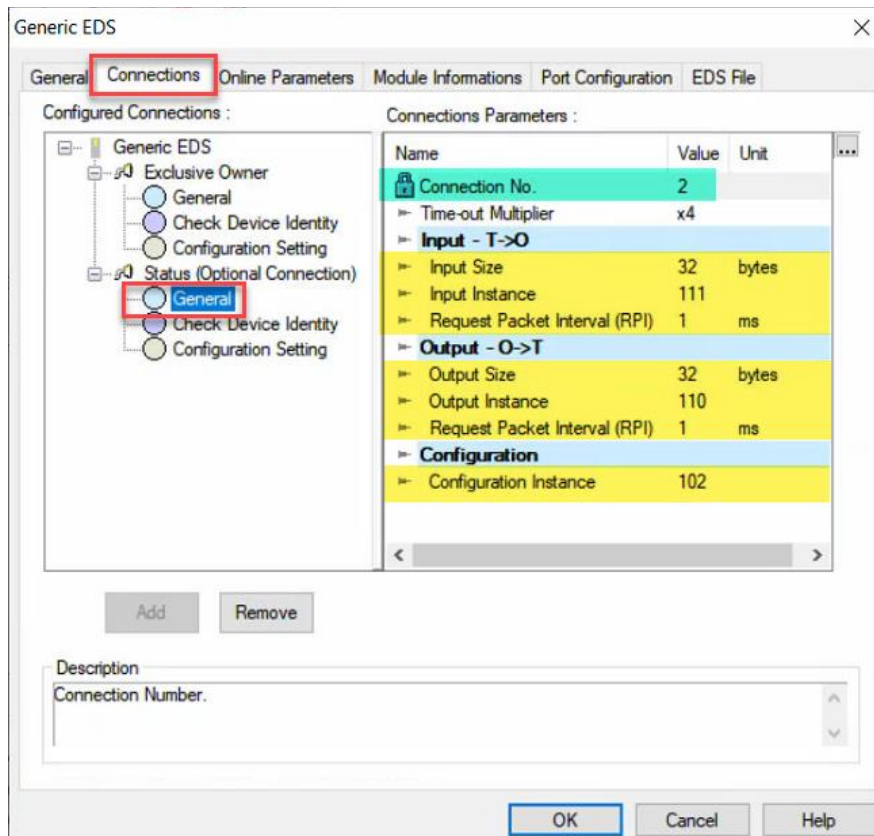
- Here the Connection No to be noted for using in [chapter 4.1.1](#)

Step – 5: Extended Process Data Connection



1. Click on Add.
2. On click, Select the connection to add pop-up window will appear on screen. Make sure Extended Process Data - Exclusive Owner is selected, Click on “OK”.
3. Click on “OK”.

Step – 6



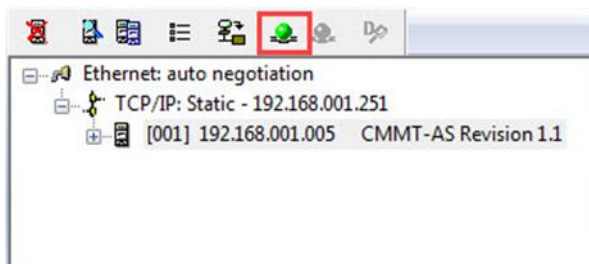
- Click the “**Connections**” from tool bar.
- Select “**General**” from Status (Optional Connection) drop-down menu.
- Input- T ->O:
Input Size : 32 Bytes
Input Instance : 111
- Output- O ->T:
Input Size : 32 Bytes
Input Instance : 110
- Configuration:
Configuration Instance : 102
- Set minimum value in Request Packet Interval(RPI), Example – 1 ms (Note: not to keep 0.5ms)
- Click on “**OK**”.

**Note**

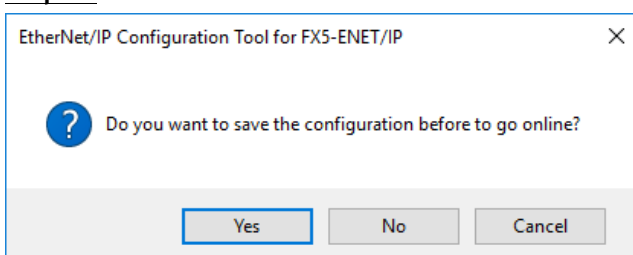
- Here the Connection No highlighted in green colour to be noted for using in [chapter 7.3.1](#)

Step – 7**Note**

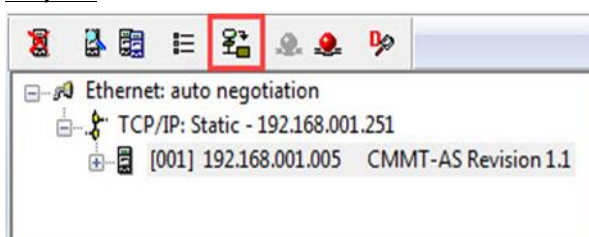
- Ensure the New IP address entered is correct.
- First establish connection with PLC, download the IQ-R EtherNet/IP setting through GX work3.



- Click the “**Online Comand**” from tool bar.

Step – 8

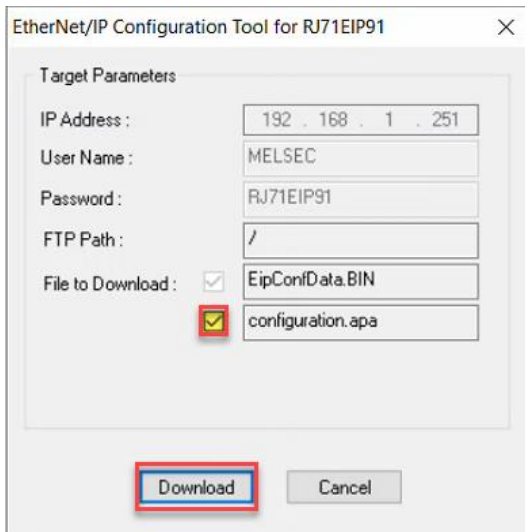
- Click the “**YES**”.

Step – 9

- Click the “**Download**” from tool bar.

Step – 10

If connection is established then below image will pop-up on screen.



- Select **“Configuration.apa”**.
- Click the **“Download”**.

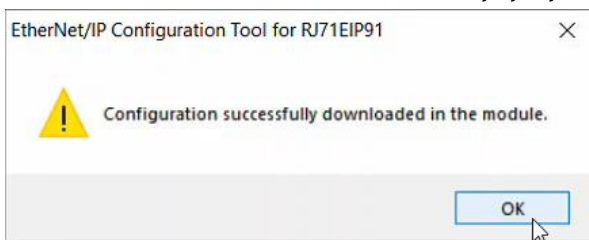
If connection is not established then below window will pop-up on screen.



- Follow the note given in step – 7 and repeat step – 9.

Step –11

After successful download below window pop-up on screen

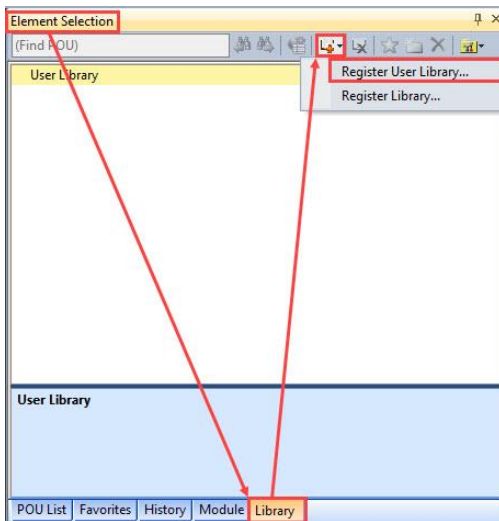


- Click the **“OK”**.

3.6 Adding Library to the project

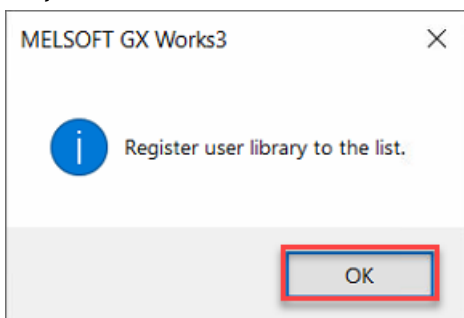
3.6.1 Import Festo_CMMT_Mitsubishi_IQR_EIP_Lib in FB/Fun under program

Step – 1



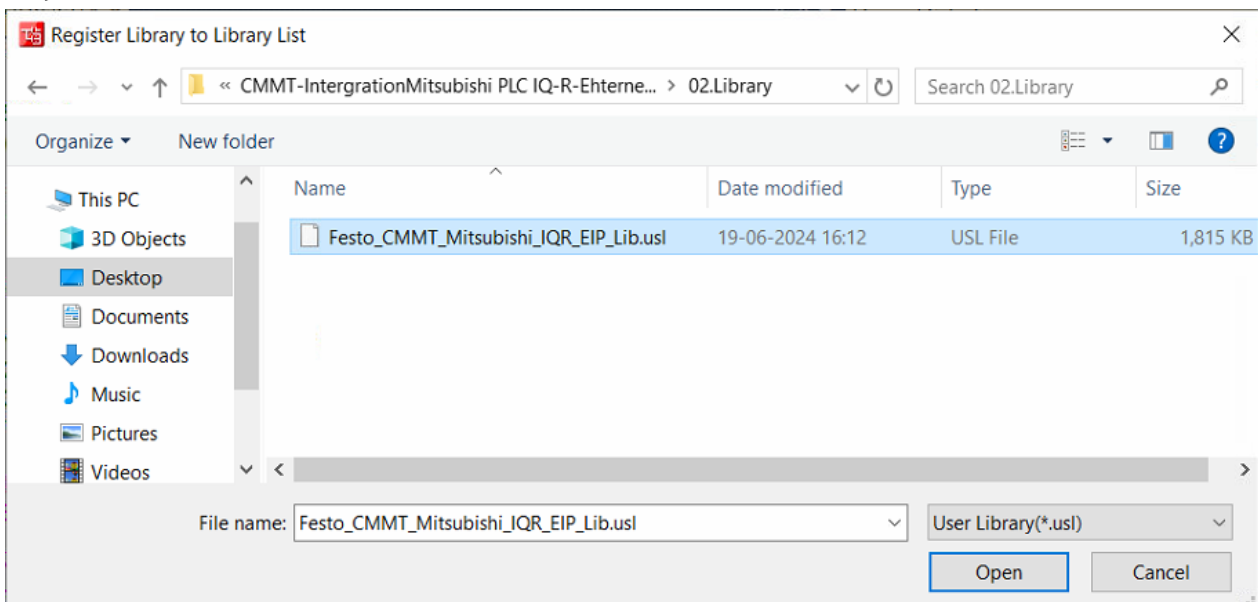
- In **Element Selection** Select **Library**.
- Select on “**Register to Library List**”
- Click on “**Register User Library**”

Step – 2



- Click “**OK**” button in import configuration window.

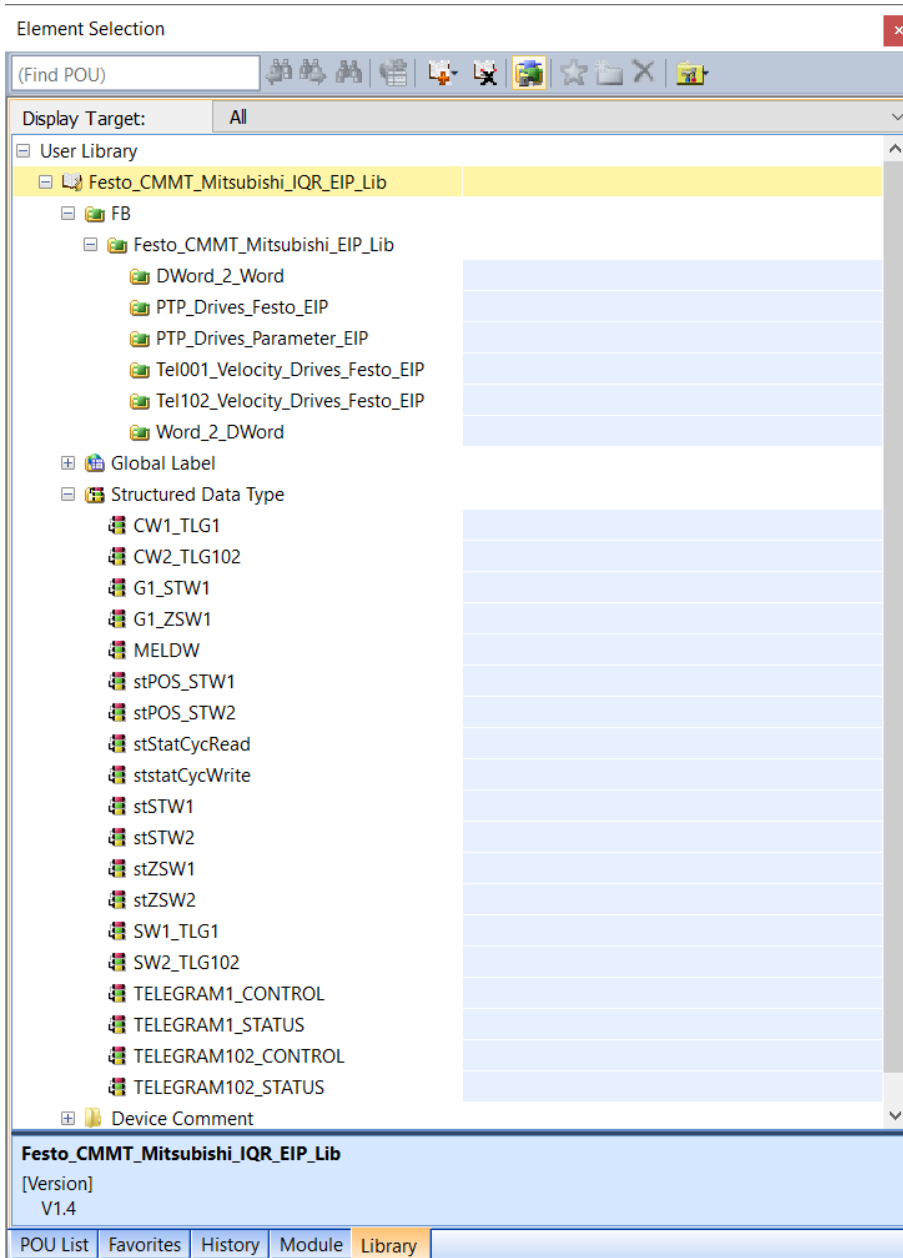
Step – 3



Starting a Project

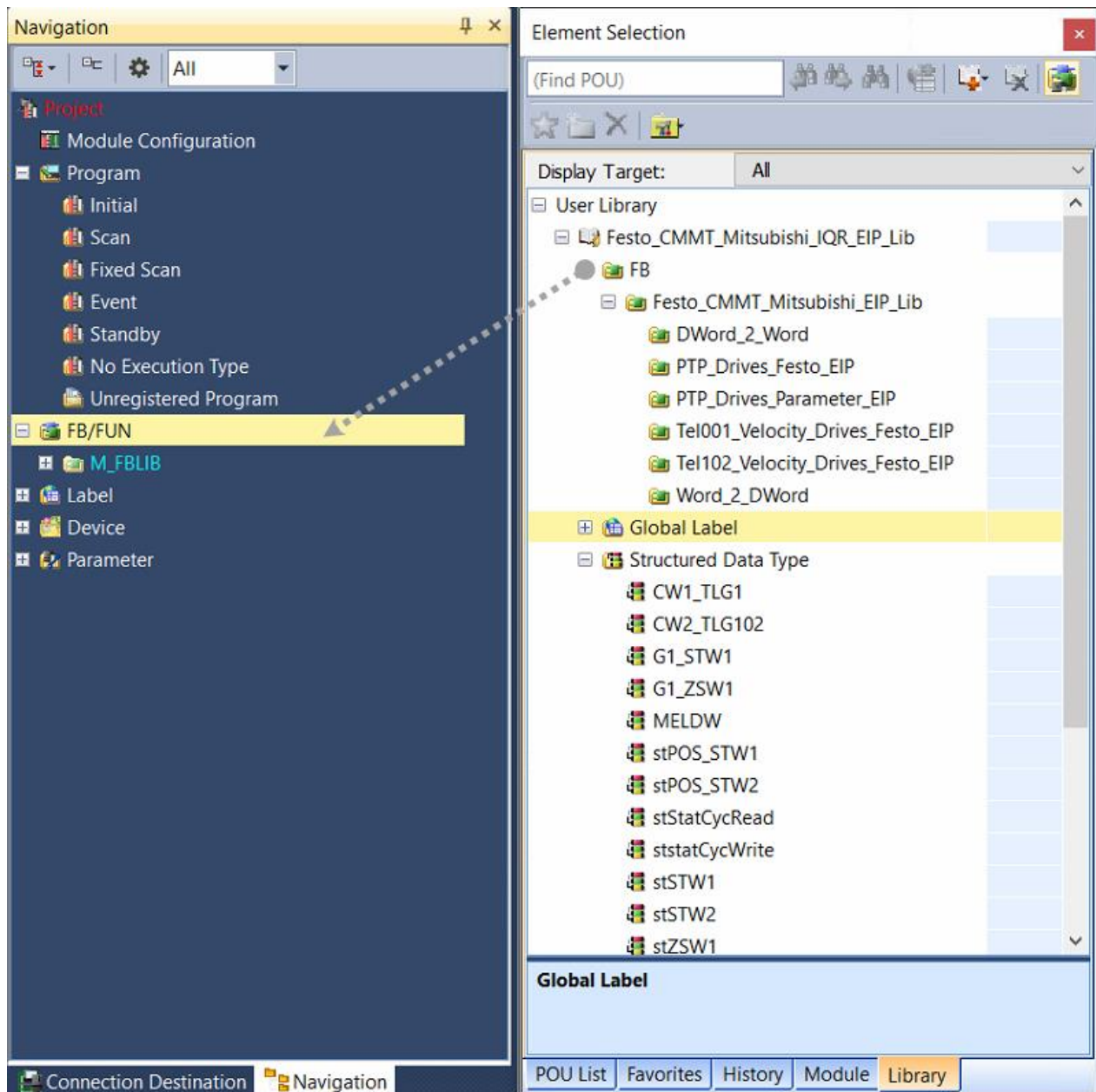
- Select **"Festo_CMMT_Mitsubishi_IQR_EIP_Lib.usl"**.
- Click **"Open"**

After successful import of Festo_CMMT_Mitsubishi_IQR_EIP_Lib will be displayed in User Library folder as shown in below picture.



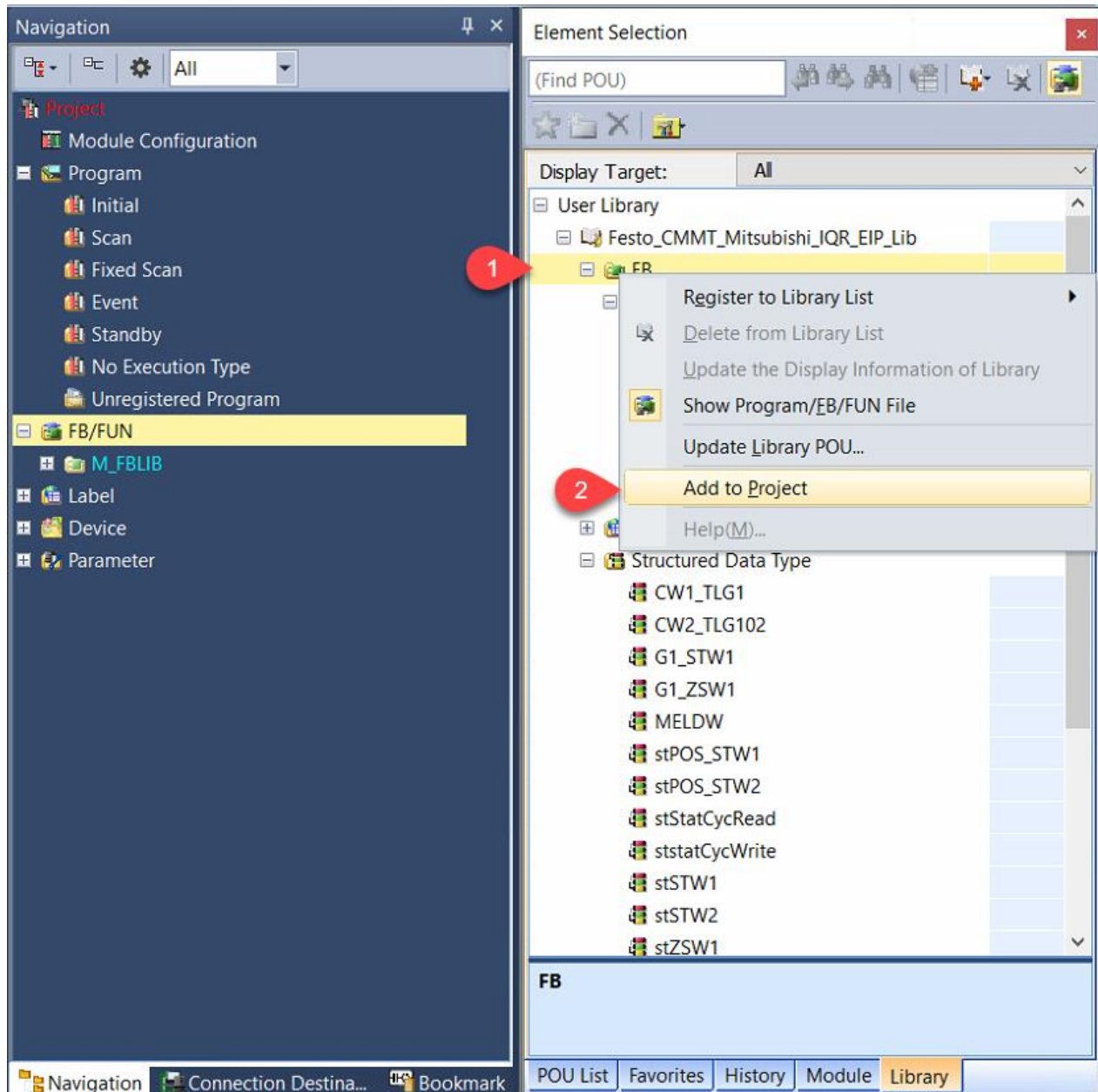
Note

- All user defined libraries will be under User Library Folder.

Step – 5 Add function block to Program.**Method 1:-**

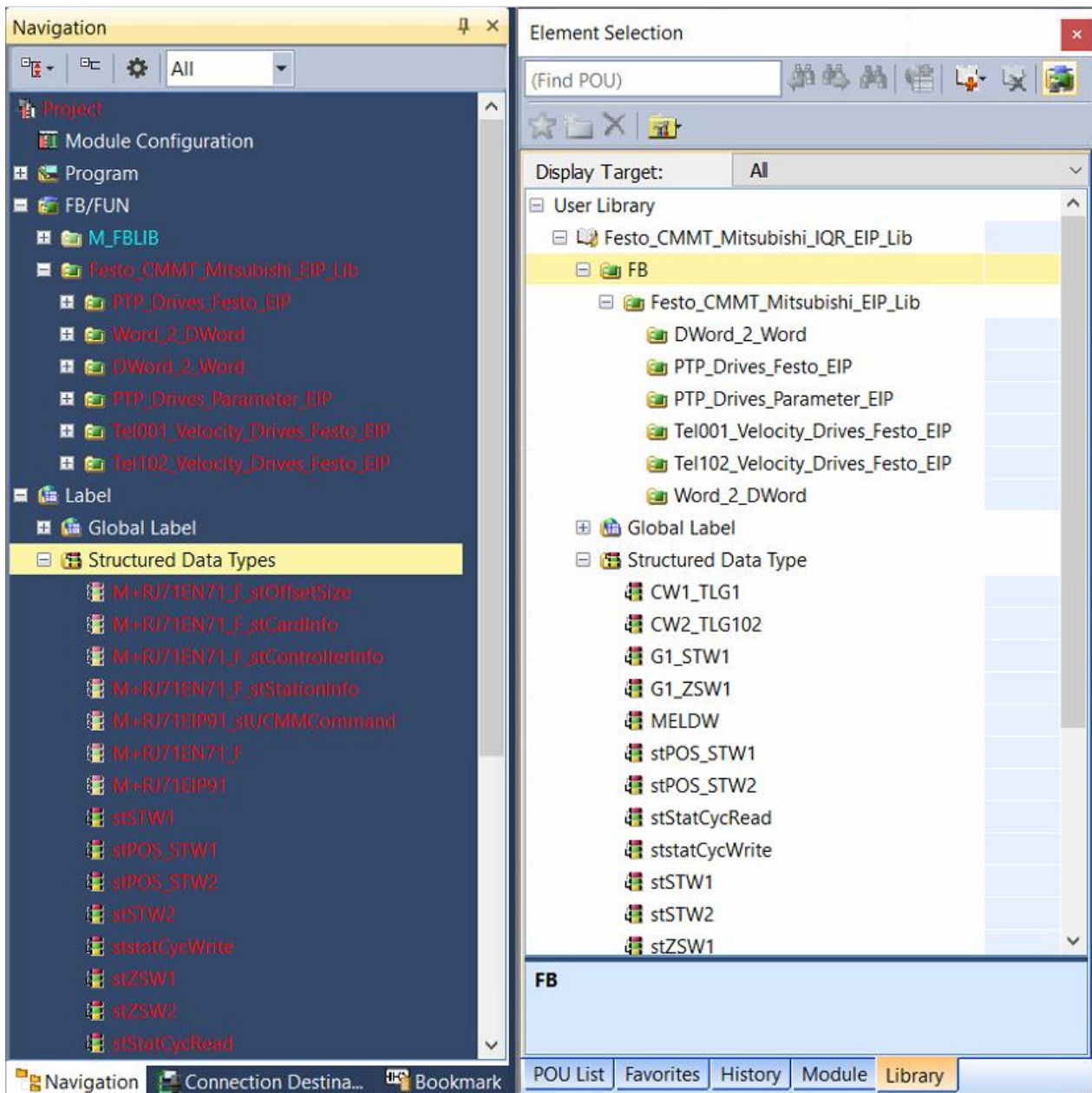
Drag and drop the **All FB's** from User Library folder to the **FB/FUN** as shown in below image.

Method 2:-



1. Right Click on **FB** under Festo_CMMT_Mitsubishi_IQR_EIP_Lib as shown in below image .
2. Click on **Add to Project**

After successful addition of Festo_CMMT_Mitsubishi_IQR_EIP_Lib will be displayed in FB/FUN folder as shown in below picture.



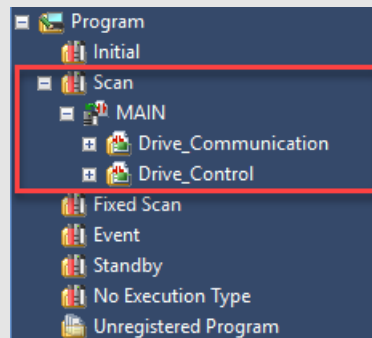
4 Library Configuration



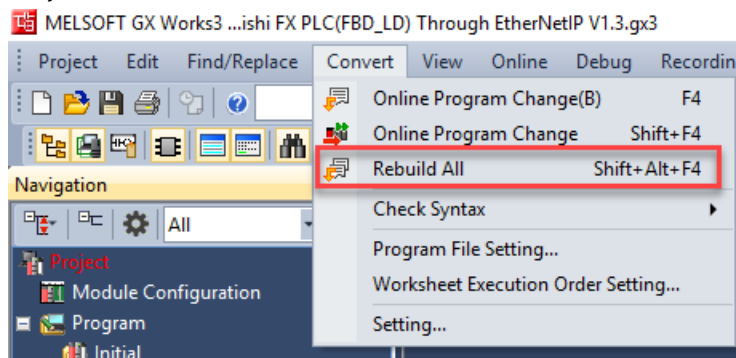
Note

The function blocks must be called cyclically with a separate instance for every CMMT Drives integrated in the user program. Simultaneous use of multiple instances of the same device is not permitted.

Any Function Block that must be called in **Scan** under Program file is called cyclically. Refer Image below.



Step – 1



– [Convert] => [Rebuild All]

Step – 2

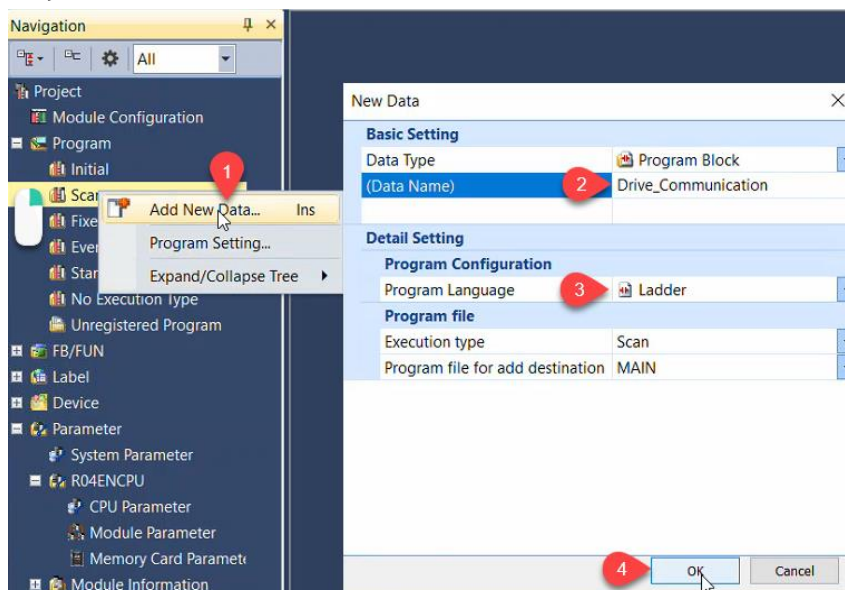


Fig – 1

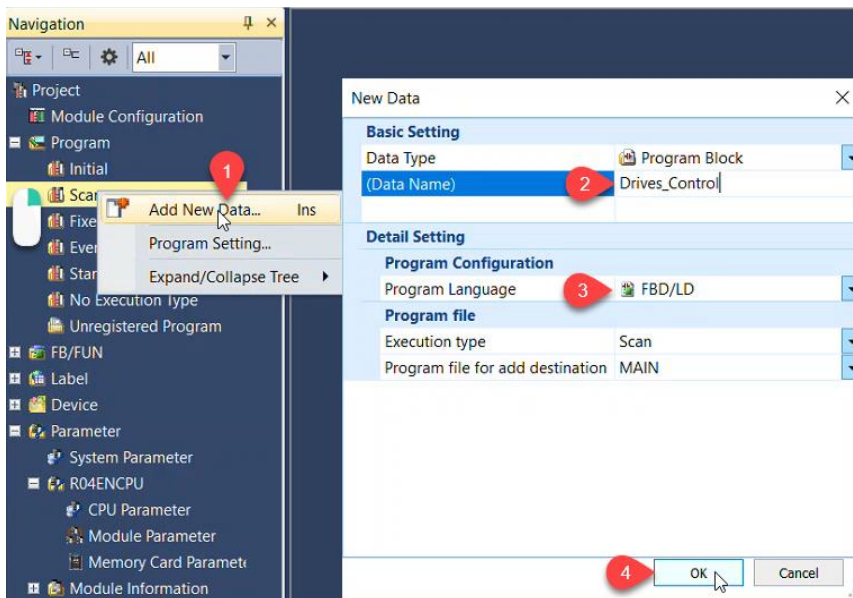
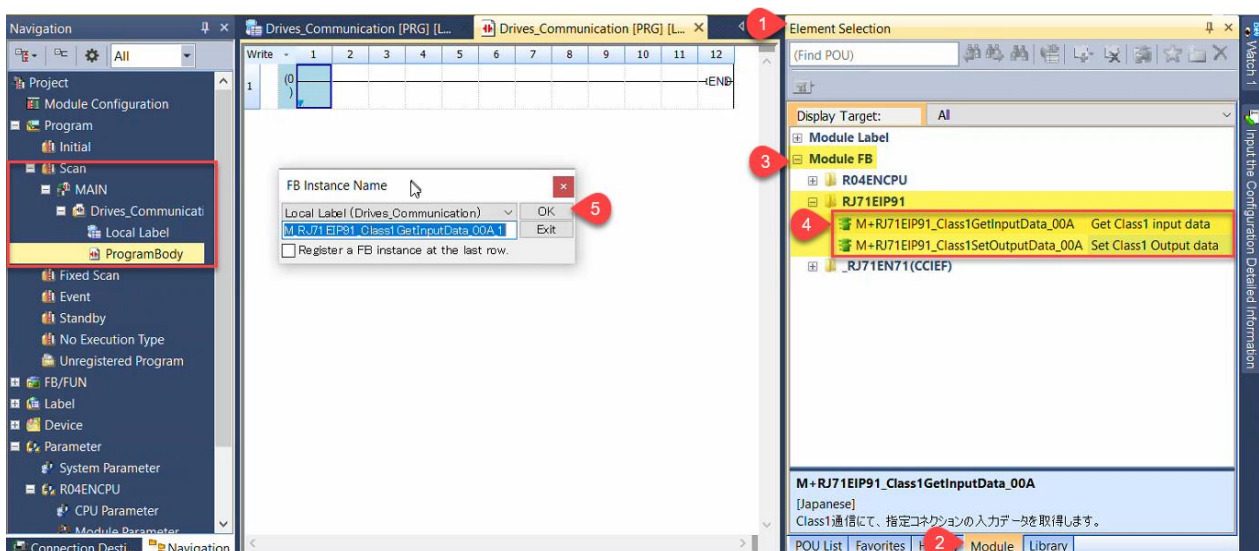


Fig – 2

1. [Program] => [Scan]
 - a. Right-Click on “Scan” in the “Project” side bar.
 - b. Left-Click on “Add new Data.”
 - c. A new window “New Data” will be opened.
2. [Data Name] > [User Defined] or use the ones in the example image.
3. [Program Language] > [Ladder] (Refer Fig – 1)
[Program Language] > [FBD/LD] (Refer Fig – 2)
4. Confirm your selection with the button “OK”.

4.1 M_FBLib – RJ71EIP91 Library Configuration

This chapter describes about adding “M_FBLib – RJ71EIP91” library in program & assigning input, output tags.



Insert the following code inside the implementation window.

1. Go to **Element Selection**.
2. Select the **Module** in Element Selection.
3. Under Display Target => [ModuleFB]=>[**RJ71EIP91**].
4. Drag and drop the FB to the empty ladder rung as shown in above image
 - a. **M+RJ71EIP91_Class1GetInputData_00A**
 - b. **M+RJ71EIP91_Class1SetOutputData_00A**
5. Confirm your selection with the button “OK”.

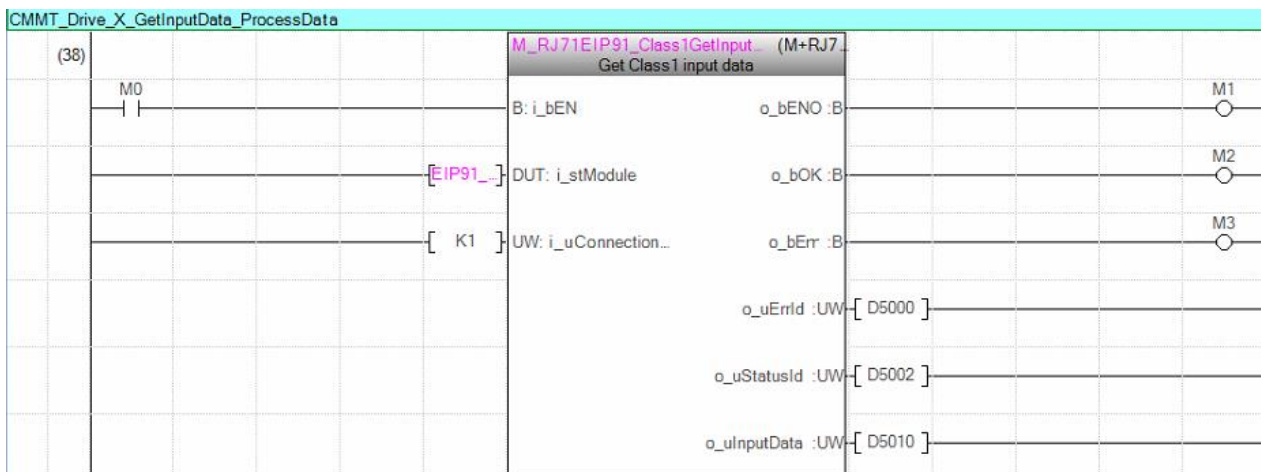
4.1.1 Configure “M+RJ71EIP91_Class1GetInputData_00A and M+RJ71EIP91_Class1SetOutputData_00A” Input & Output Interface Tags

Add a new program to initialize the Get Input Data & Set Output Data of RJ71EIP91 Module.

- Initialize the control words as follows to parameterize the RJ71EIP91 Module.
- If the mapping was changed in the Get Input Data & Set Output Data settings, remember to initialise the correct memory areas.

M+RJ71EIP91_Class1GetInputData_00A

- Click “OK”.
- Fill the Function Block.



Input and Output Tags

Tag Name	Data Type	Function Description
i_bEN	Bit	Get Input Data - Execution command
i_stModule	Structure	EIP91_1
i_uConnectionNo	Word [Un-signed]/Bit String [16-bit]	Device Connection No (Refer Chapter 3.4 Step – 10 & 12 or Chapter 3.5 Step – 4 & 6)
o_bENO	Bit	Get Input Data - Execution status
o_bOK	Bit	Get Input Data - Normal completion
o_bErr	Bit	Get Input Data - Error completion
o_uErrID	Word [Un-signed]/Bit String [16-bit]	Get Input Data - Error code
o_uStatusID	Word [Un-signed]/Bit String [16-bit]	Get Input Data - Error code of connection communication error
o_uInputData	Word [Un-signed]/Bit String [16-bit]	Get Input Data - Input data storage device (head number of the device)

M+FX5ENETIP_Class1SetOutputData_00A

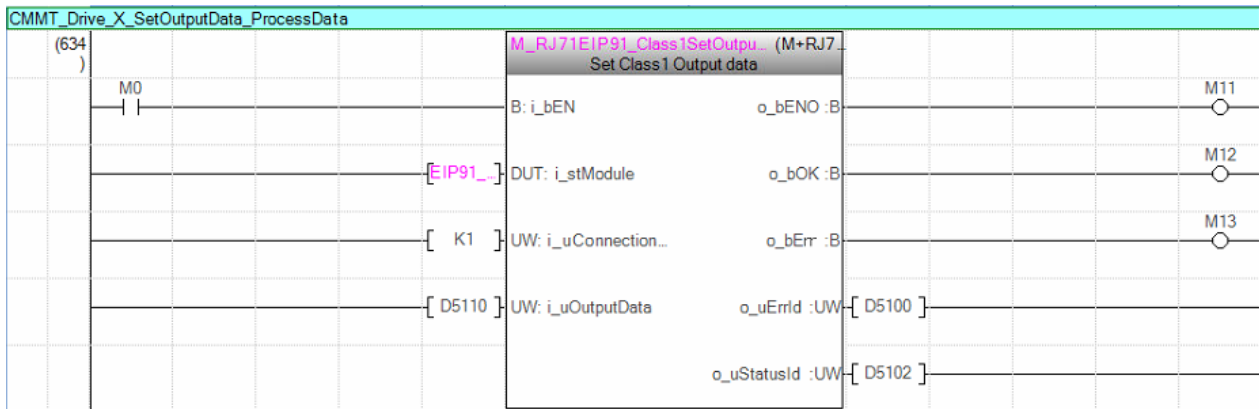
FB Instance Name

Local Label (EthernetIP_Connection) OK

M_RJ71EIP91_Class1SetOutputData_00A_1 Exit

☒ Register a FB instance at the last row.

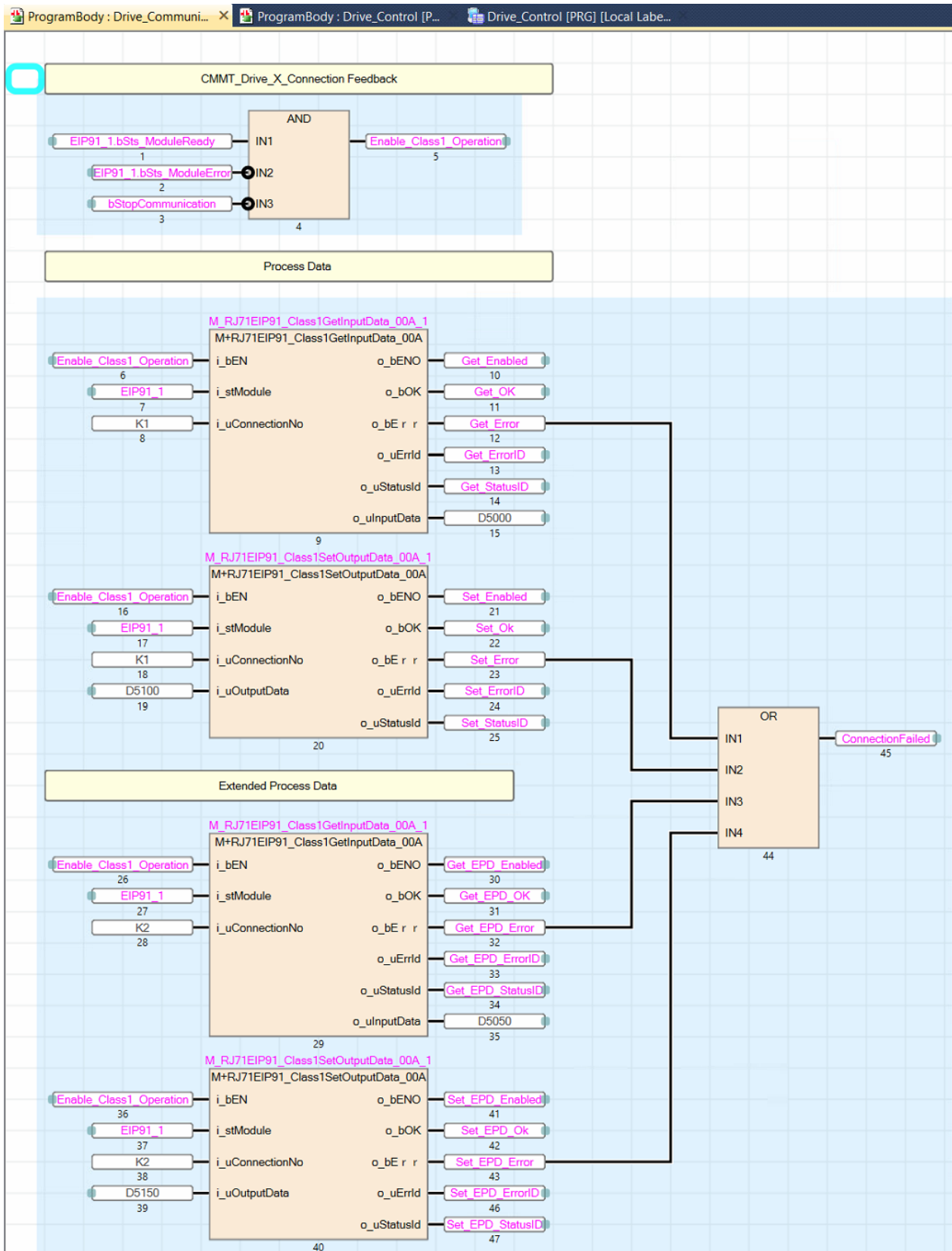
- Click “OK”.
- Fill the Function Block

**Input and Output Tags**

Tag Name	Data Type	Function Description
i_bEN	Bit	Set Output Data - Execution command
i_stModule	Structure	EIP91_1
i_uConnectionNo	Word [Unsigned]/Bit String [16-bit]	Device Connection No (Refer Chapter 3.4 Step – 10 & 12 or Chapter 3.5 Step – 4 & 6)
o_uOutputData	Word [Unsigned]/Bit String [16-bit]	Set Output Data - Output data storage device (head number of the device)
o_bENO	Bit	Set Output Data - Execution status
o_bOK	Bit	Set Output Data - Normal completion
o_bErr	Bit	Set Output Data - Error completion
o_uErrID	Word [Unsigned]/Bit String [16-bit]	Set Output Data - Error code
o_uStatusID	Word [Unsigned]/Bit String [16-bit]	Set Output Data - Error code of connection communication error

- Once the FB tags successfully configured then the view of FB looks like below.





4.2 Festo_CMMT_Mitsubishi_IQR_EIP_Lib Library Configuration

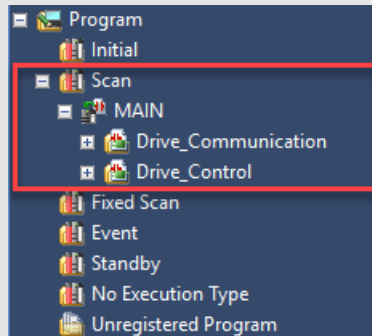
This chapter describes about adding “Festo_CMMT_Mitsubishi_IQR_EIP_Lib” library in program & assigning input, output tags.



Note

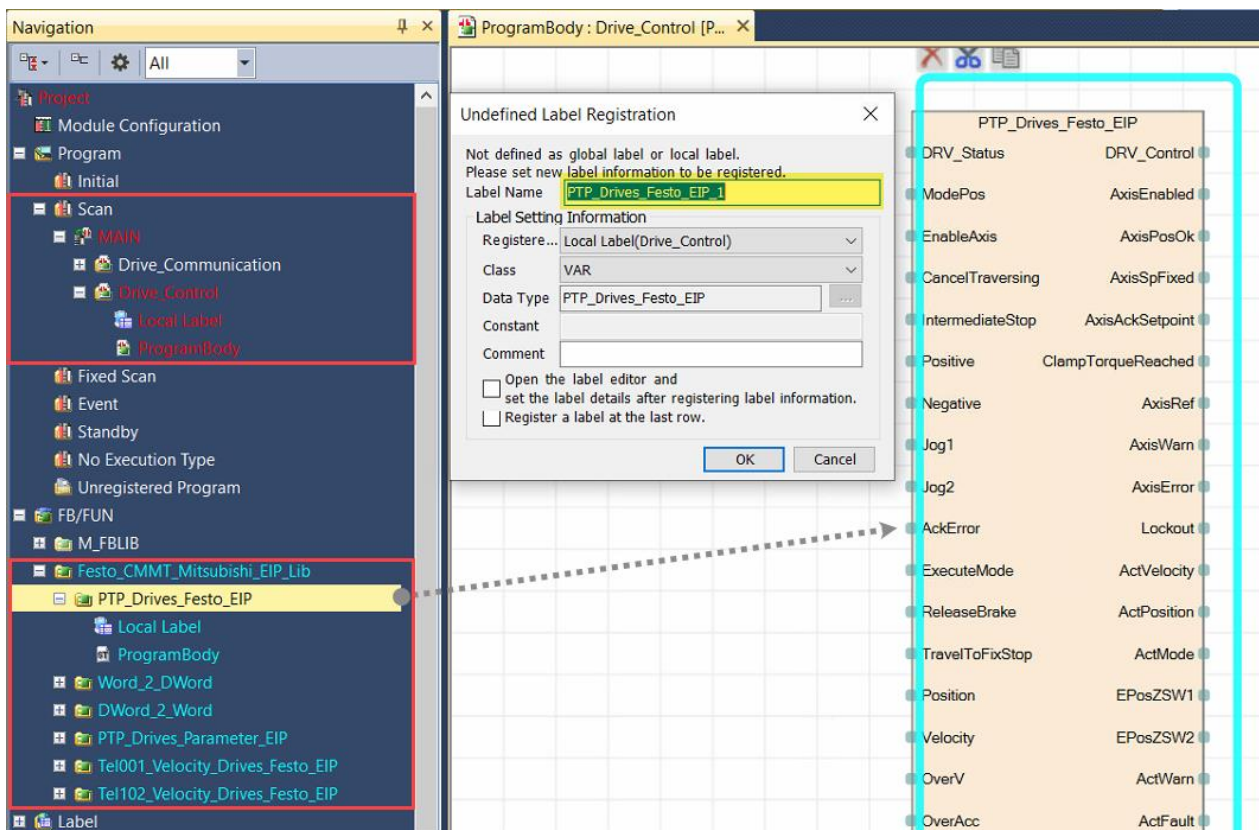
The function blocks must be called cyclically with a separate instance for every CMMT Drives integrated in the user program. Simultaneous use of multiple instances of the same device is not permitted.

Any Function Block that must be called in **Scan** under Program file is called cyclically. Refer Image below.



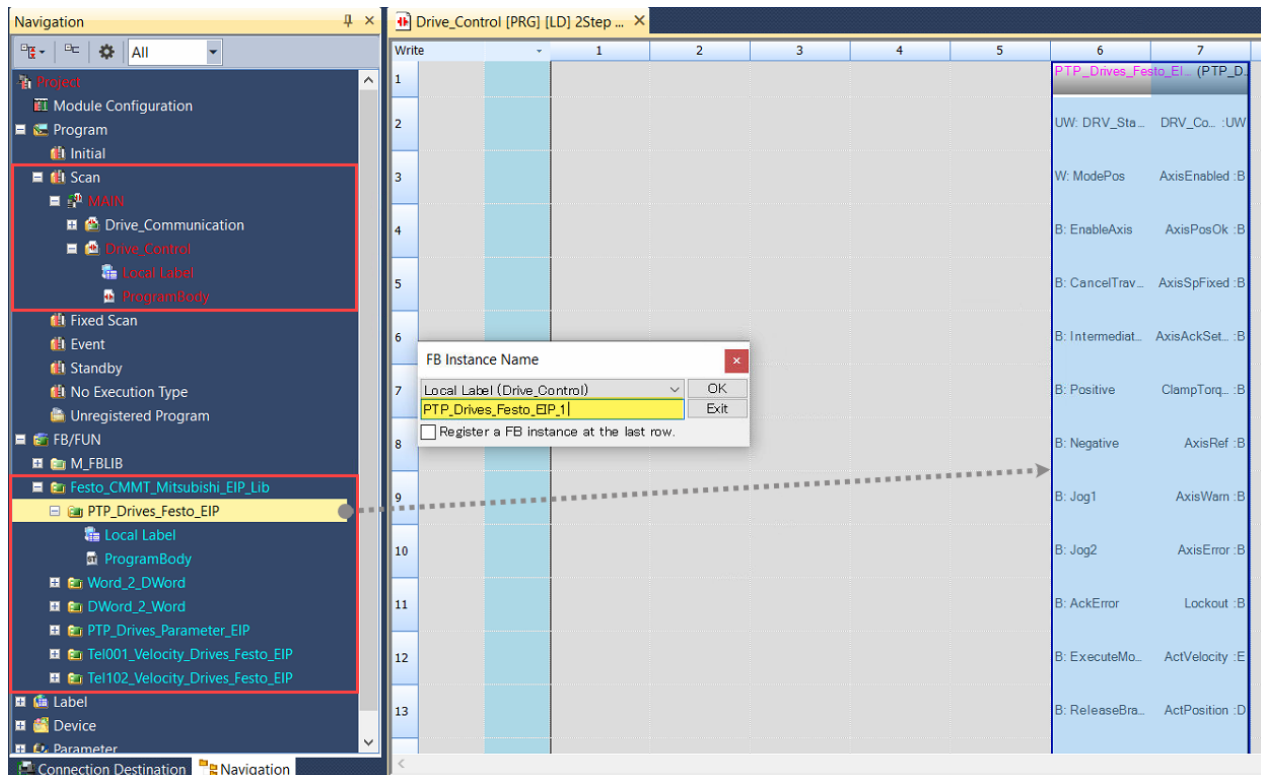
4.2.1 Add “PTP_Drives_Festo_EIP” in PLC Program

This chapter describes about adding “PTP_Drives_Festo_EIP” FB in program & assigning input, output tags.



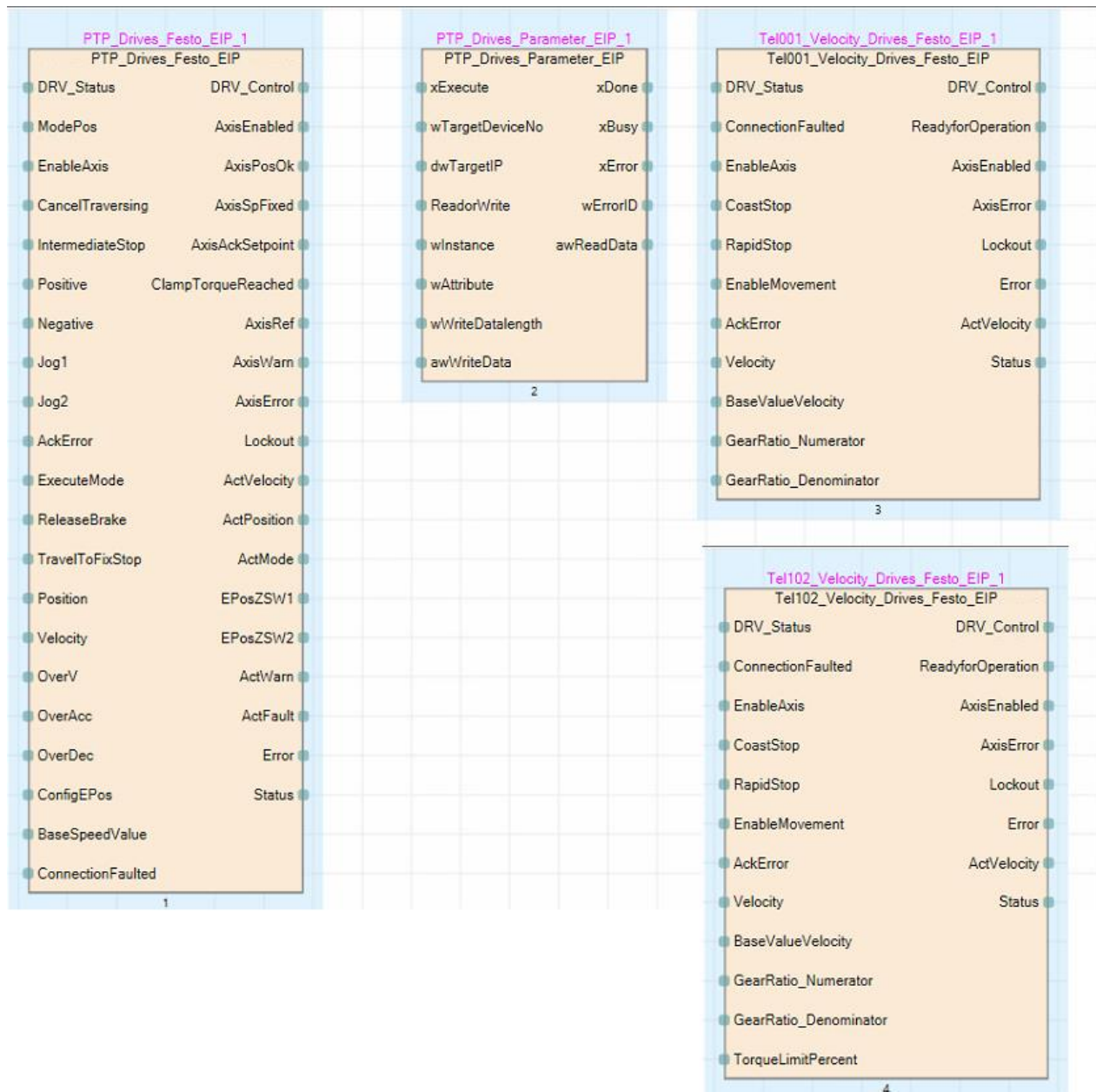
- Insert the following code inside the implementation window.
- Drag and drop the, PTP_Drives_Festo_EIP, PTP_Drives_Parameter_EIP, Tel001_Velocity_Drives_Festo_EIP and Tel102_Velocity_Drives_Festo_EIP from FB/FUN to the empty space as shown in above image.
- Confirm your selection with the button “OK”.

Or



- Insert the following code inside the implementation window.
- Drag and drop the, PTP_Drives_Festo_EIP, PTP_Drives_Parameter_EIP, Tel001_Velocity_Drives_Festo_EIP and Tel102_Velocity_Drives_Festo_EIP from FB/FUN to the empty Ladder Rung as shown in above image.
- Confirm your selection with the button **“OK”**.

After successful addition of the Drag and drop the, PTP_Drives_Festo_EIP, PTP_Drives_Parameter_EIP, Tel001_Velocity_Drives_Festo_EIP and Tel102_Velocity_Drives_Festo_EIP will be displayed in FBD rung (program folder) as shown in below picture.



After successful addition of the Drag and drop the, PTP_Drives_Festo_EIP, PTP_Drives_Parameter_EIP, Tel001_Velocity_Drives_Festo_EIP and Tel102_Velocity_Drives_Festo_EIP will be displayed in Ladder rung (program folder) as shown in below picture.

PTP_Drives_Festo_EIP_1 (PTP_Drives_Festo_EIP)	
UW: DRV_Status	DRV_Control :UW
W: ModePos	AxisEnabled :B
B: EnableAxis	AxisPosOk :B
B: CancelTraversing	AxisSpFixed :B
B: IntermediateStop	AxisAckSetpoint :B
B: Positive	ClampTorqueReached :B
B: Negative	AxisRef :B
B: Jog1	AxisWarn :B
B: Jog2	AxisError :B
B: AckError	Lockout :B
B: ExecuteMode	ActVelocity :E
B: ReleaseBrake	ActPosition :D
B: TravelToFixStop	ActMode :W
D: Position	EPosZSW1 :UW
D: Velocity	EPosZSW2 :UW
W: OverV	ActWarn :UW
W: OverAcc	ActFault :UW
W: OverDec	Error :B
UD: ConfigEPos	Status :UW
E: BaseSpeedValue	
B: ConnectionFaulted	
PTP_Drives_Parameter_EIP_1 (PTP_Drives_Parameter_...)	
B: xExecute	xDone :B
UW: wTargetDeviceNo	xBusy :B
UD: dwTargetIP	xError :B
B: ReadorWrite	wErrorID :UW
UW: wInstance	awReadData :UW
UW: wAttribute	
UW: wWriteDataLength	
UW: awWriteData	
Tel001_Velocity_Drives_Festo_EIP_1 (Tel001_Velocity_...)	
UW: DRV_Status	DRV_Control :UW
B: ConnectionFaulted	ReadyforOperation :B
B: EnableAxis	AxisEnabled :B
B: CoastStop	AxisError :B
B: RapidStop	Lockout :B
B: EnableMovement	Error :B
B: AckError	ActVelocity :E
E: Velocity	Status :UW
E: BaseValueVelocity	
W: GearRatio_Numerator	
W: GearRatio_Denominator	
Tel102_Velocity_Drives_Festo_EIP_1 (Tel102_Velocity_...)	
UW: DRV_Status	DRV_Control :UW
B: ConnectionFaulted	ReadyforOperation :B
B: EnableAxis	AxisEnabled :B
B: CoastStop	AxisError :B
B: RapidStop	Lockout :B
B: EnableMovement	Error :B
B: AckError	ActVelocity :E
E: Velocity	Status :UW
E: BaseValueVelocity	
W: GearRatio_Numerator	
W: GearRatio_Denominator	
E: TorqueLimitPercent	

Linking Tags to Function Block

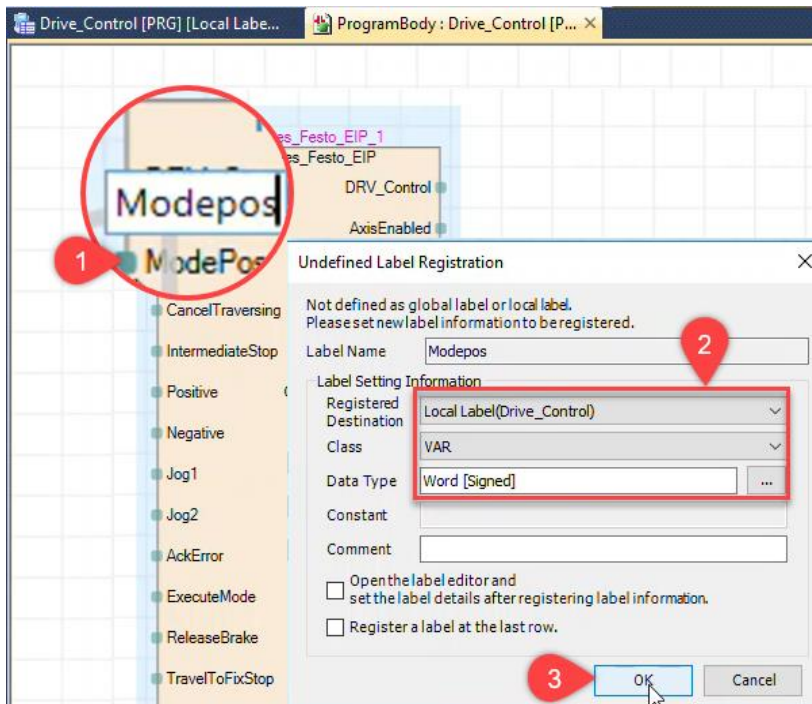
Step – 1



Note

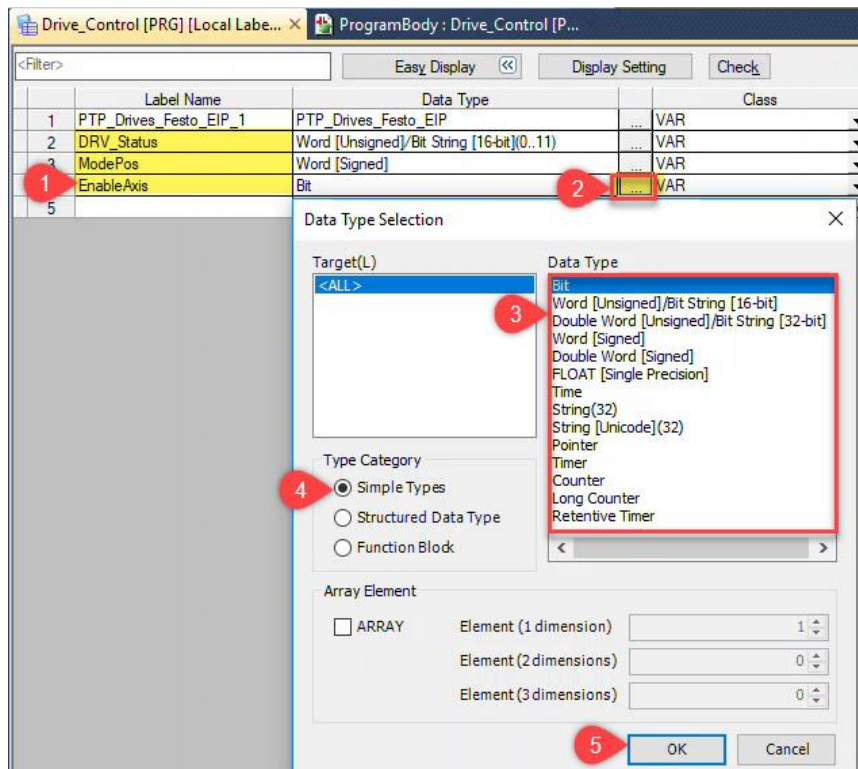
- In the tables of [Chapter 5.1.1 Input and Output Interface](#) Data Table, [Chapter 5.2.1 Input and Output Interface](#) Data Table, [Chapter 5.5.2 Input and Output Interface](#) Data Table, [Chapter 5.6.2 Input and Output Interface](#) Data Table the user must create the appropriate tag according to the Data Type and Class.
- Insert the following code inside the declaration window. As described in Step – 2. Follow Method 1 or Method 2

Method - 1



1. Click /Select the interface tag. Enter the **“Tag Name”** for input **“xEnable”** and press **“Enter”** button to open **“Undefined Label Registration”** window.
2. Under Label Setting Information (This will be auto declared,)
 - a. Registered Destination => Local Label(Drive_Controls)
 - b. Class => Var
 - c. Data Type => Word[Signed] (Refer [Chapter 5.1.1 Input and Output Interface](#))
3. click **“OK”** button to declare the variable.
4. Repeat the previous step for rest of the interface tag for all function block.

Method-2



Double-Click on “Local Label” of the program “**Drive_Control**” in the “Program” side bar. A new window “Drive_Control Local Label Setting” will be opened as shown in above image.

1. Write under [Label Name] => [Used defined name] or [As given in].
2. Click on the highlighted box, then Data Type pop-up window will open as shown in below image.
3. Select **Simple Type** in **Type Category**.
4. Select the **Data Type** for the Label. [Ref [Chapter 5.1.1 Input and Output Interface](#)].
5. Confirm your selection with the button “**OK**”.



Note

- Repeat the [Chapter 4.2.1 - Linking Tags to Function Block, Step - 1](#) follow either Method 1 or Method 2 the same procedure for rest of the Tag that's need for Function Block interface.
- Fill In and Out Tags of the Function Blocks.

Drive_Communication [PRG] [L... Drive_Control [PRG] [LD] 584Ste... Drive_Control [PRG] [Local Labe... X					
<Filter>		Easy Display <<		Display Setting	
				Check	
	Label Name	Data Type		Class	
1	PTP_Drives_Festo_EIP_1	PTP_Drives_Festo_EIP	...	VAR	▼
2	DRV_Status	Word [Unsigned]/Bit String [16-bit](0..11)	...	VAR	▼
3	ModePos	Word [Signed]	...	VAR	▼
4	EnableAxis	Bit	...	VAR	▼
5	CancelTraversing	Bit	...	VAR	▼
6	IntermediateStop	Bit	...	VAR	▼
7	Positive	Bit	...	VAR	▼
8	Negative	Bit	...	VAR	▼
9	Jog1	Bit	...	VAR	▼
10	Jog2	Bit	...	VAR	▼
11	AckError	Bit	...	VAR	▼
12	ExecuteMode	Bit	...	VAR	▼
13	ReleaseBrake	Bit	...	VAR	▼
14	TravelToFixStop	Bit	...	VAR	▼
15	Position	Double Word [Signed]	...	VAR	▼
16	Velocity	Double Word [Signed]	...	VAR	▼
17	OverV	Word [Signed]	...	VAR	▼
18	OverAcc	Word [Signed]	...	VAR	▼
19	OverDec	Word [Signed]	...	VAR	▼
20	ConfigEPos	Double Word [Unsigned]/Bit String [32-bit]	...	VAR	▼
21	BaseSpeedValue	FLOAT [Single Precision]	...	VAR	▼
22	DRV_Control	Word [Unsigned]/Bit String [16-bit](0..11)	...	VAR	▼
23	AxisEnabled	Bit	...	VAR	▼
24	AxisPosOk	Bit	...	VAR	▼
25	AxisSpFixed	Bit	...	VAR	▼
26	AxisAckSetpoint	Bit	...	VAR	▼
27	ClampTorqueReached	Bit	...	VAR	▼
28	AxisRef	Bit	...	VAR	▼
29	AxisWam	Bit	...	VAR	▼
30	AxisError	Bit	...	VAR	▼
31	Lockout	Bit	...	VAR	▼
32	ActVelocity	FLOAT [Single Precision]	...	VAR	▼
33	ActPosition	Double Word [Signed]	...	VAR	▼
34	ActMode	Word [Signed]	...	VAR	▼
35	EPosZSW1	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
36	EPosZSW2	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
37	ActWam	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
38	ActFault	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
39	Error	Bit	...	VAR	▼
40	Status	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
41	PTP_Drives_Parameter_1	PTP_Drives_Parameter	...	VAR	▼
42	Execute	Bit	...	VAR	▼
43	Enable_Write	Bit	...	VAR	▼
44	ParameterID	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
45	Attribute	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
46	WriteDataLength	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
47	WriteData	Word [Unsigned]/Bit String [16-bit](0..127)	...	VAR	▼
48	Done	Bit	...	VAR	▼
49	Busy	Bit	...	VAR	▼
50	ParaError	Bit	...	VAR	▼
51	ParaErrorID	Word [Unsigned]/Bit String [16-bit]	...	VAR	▼
52	ReadData	Word [Unsigned]/Bit String [16-bit](0..127)	...	VAR	▼
53	DRV_EPD_Status	Word [Unsigned]/Bit String [16-bit](0..15)	...	VAR	▼
54	DRV_EPD_Control	Word [Unsigned]/Bit String [16-bit](0..15)	...	VAR	▼
55	Word_2_DWord_1	Word_2_DWord	...	VAR	▼
56	SetPointTorque_DW	Double Word [Unsigned]/Bit String [32-bit]	...	VAR	▼
57	SetPointTorque_Real	FLOAT [Single Precision]	...	VAR	▼
58	ActMotorShaftTorque_DW	Double Word [Unsigned]/Bit String [32-bit]	...	VAR	▼
59	ActMotorShaftTorque_Real	FLOAT [Single Precision]	...	VAR	▼
60	DWord_2_Word_1	DWord_2_Word	...	VAR	▼
61	SetClampingTorque_Real	FLOAT [Single Precision]	...	VAR	▼
62	SetClampingTorque_DW	Double Word [Unsigned]/Bit String [32-bit]	...	VAR	▼
63			...		▼

List of all the tag for Function Blocks of Festo_CMMT_Mitsubishi_IQR_EIP_Lib and codes for the same in Local Label window as shown in above picture.

Step – 3: Configure Interface Tags in Global Label.**Note**

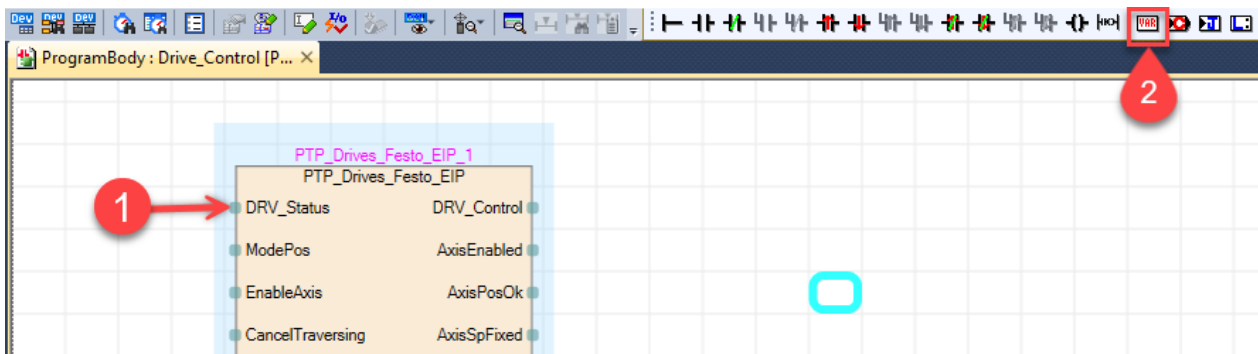
- Ensure the tag are listed in Global [Global Label Setting] as shown in above figure, if not the create as described below.
- Skip **Step – 3** if all the tags are already added in Global Tags upon adding the Library “Festo_CMMT_Mitsubishi_IQR_EIP_Lib”.

Global [Global Label Setting]				
<Filter> Easy Display Display Setting Check				
	Label Name	Data Type	Class	Assign (Device/Label)
1	bStopConnection	Bit	VAR_GLOBAL	
2	Connection_Faulted	Bit	VAR_GLOBAL	
3				

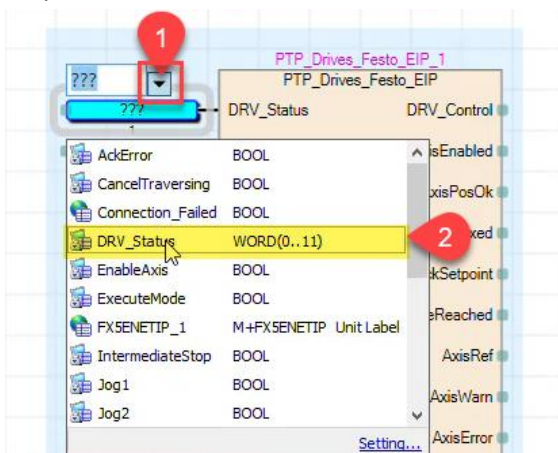
- **EIP91_1** this tag will be automatically created in “**M+Global**” when “**RJ71EIP91**” module is added Module Information. [As shown in [Chapter 4.1](#)]

After Step – 2 & Step – 3 follow the below steps to Map the correct Communication interface tag.

Ignore Step – 4 in case program language is Ladder, jump to step – 6

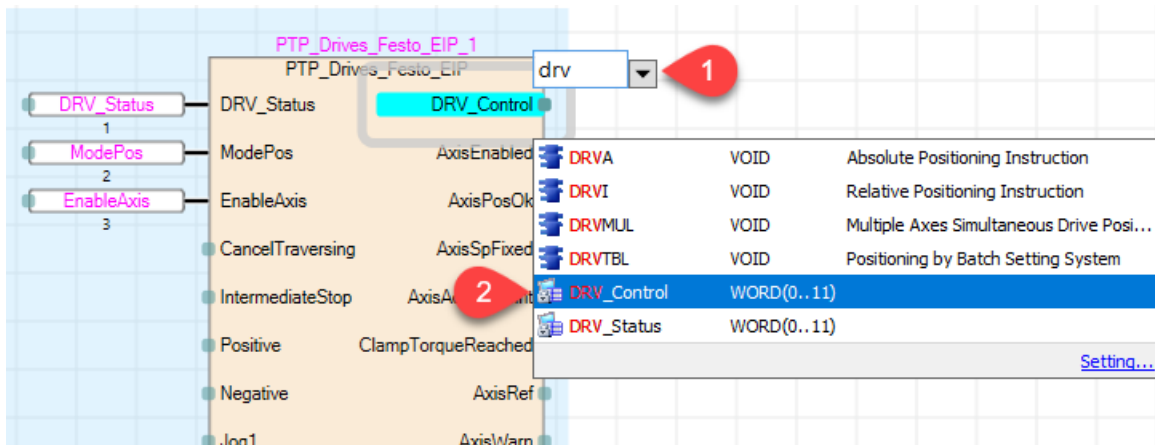
Step – 4 Variables to be filled in the function block, if using FBD/LD language.

1. Click/Select the interface tag.
2. Select the “**Variable**” from the “**FBD/LD**” menu buttons, as shown in below image.
 - 2.1. Repeat the previous step for rest of the interface tag on communication module’s function block.

Step – 5

1. Double click on the Variable, to add the tag.
2. Click on drop/down arrow button or the highlighted box, a selection window opens, select the right tag to FB, as shown in above image.

Or



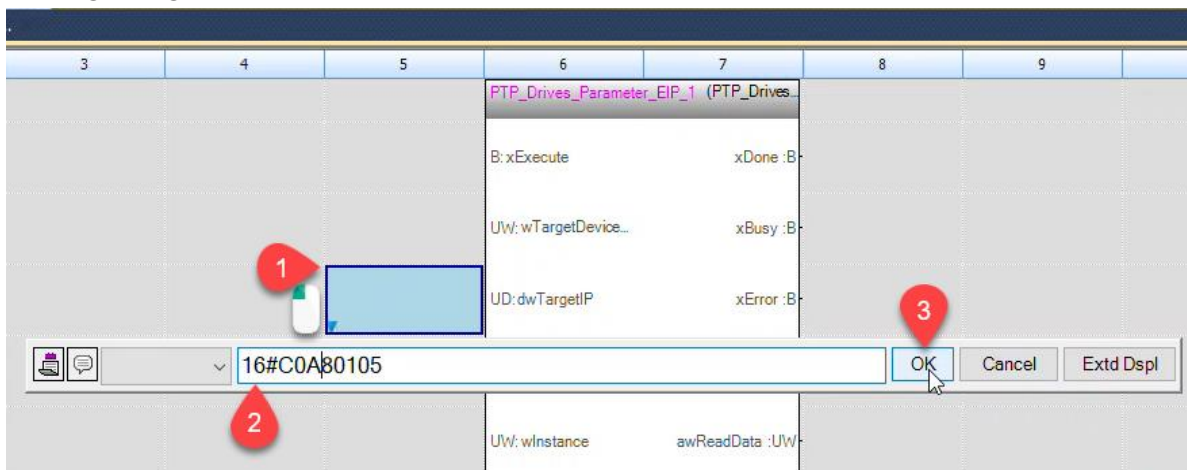
1. Double click on the Variable, to add the tag.
2. Write the correct tag name and select by left clicking mouse button or press enter, , as shown in above image.

After Step – 2 & Step – 3 follow the below steps to Map the correct Communication interface tag.

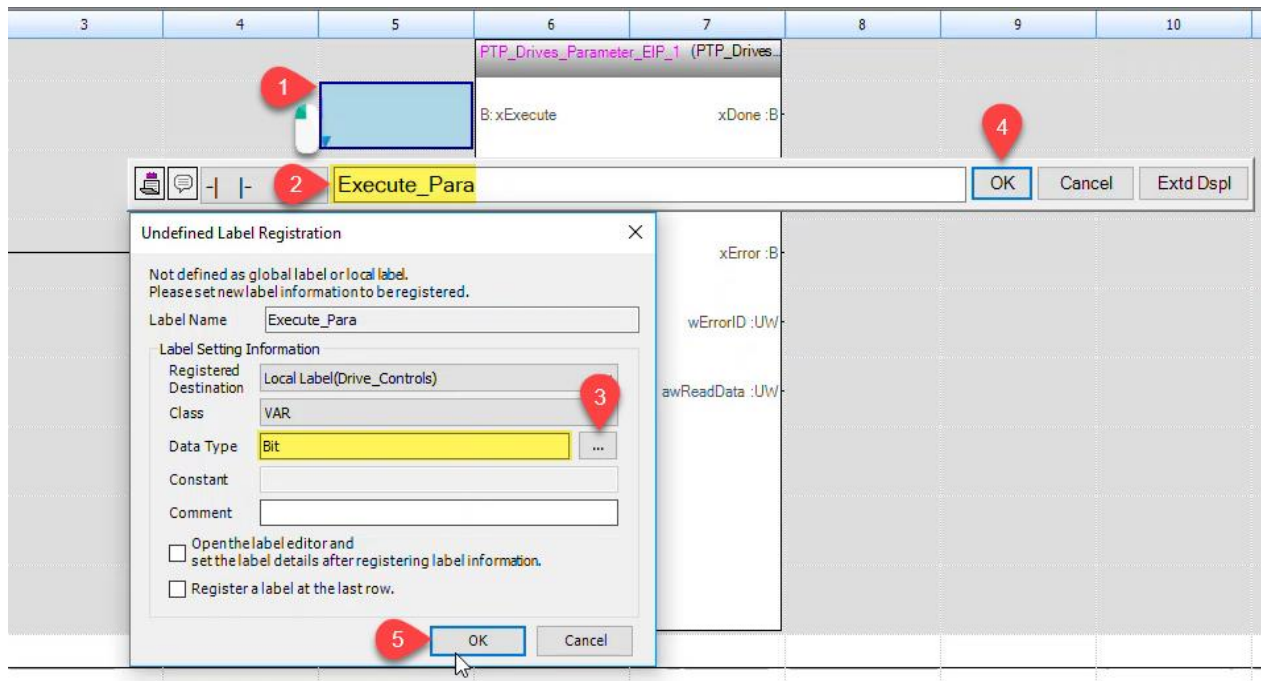
Ignore Step – 6 in case program language is FBD/LD

Step – 6 Variables to be filled in the function block if using ladder language.

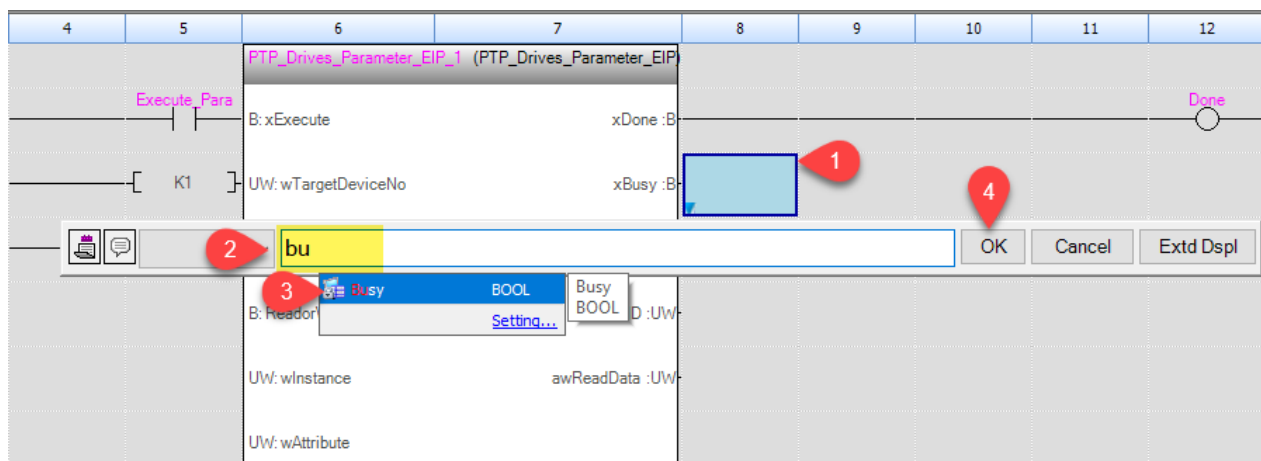
To assign Integer



1. Double-Click on near the Tag to be assigned.
2. A Tag Entry Pop-Ups on screen as shown in above image, enter the respectively Integer value.
3. Confirm your selection with the button "OK".

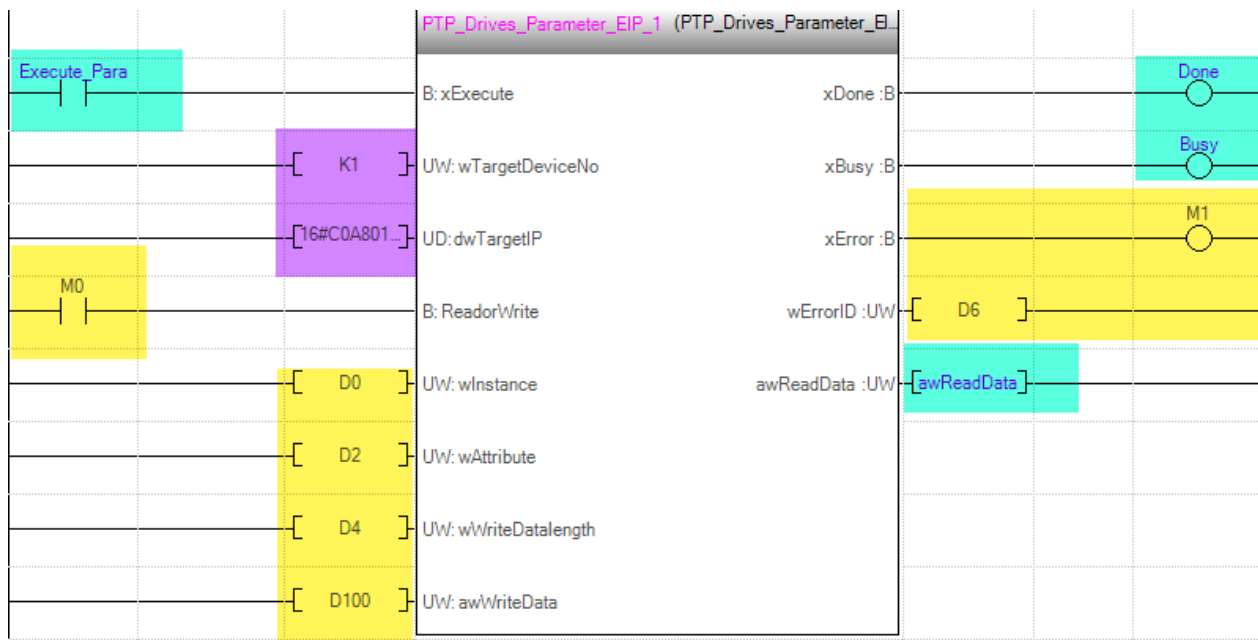
Method – 1

1. Double-Click on near the Tag to be assigned.
2. A Tag Entry Pop-Ups on screen as shown in above image, enter the [User Defined Tag Name] or use the ones in the example image. And Press Enter Button
3. Confirm the Data Type.
4. Confirm your selection with the button “OK”.
5. Confirm your selection with the button “OK”.

Method – 2

1. Double-Click on near the Tag to be assigned.
2. A Tag Entry Pop-Ups on screen as shown in above image, enter the [User Defined Tag Name] or use the ones in the example image. Double click on the Variable, to add the tag.
3. Click on drop/down arrow button or the highlighted box, a selection window opens. As shown in below image.
Or
Write the correct tag name and left click mouse button or press enter.
4. Confirm your selection with the button “OK”.

Method – 3

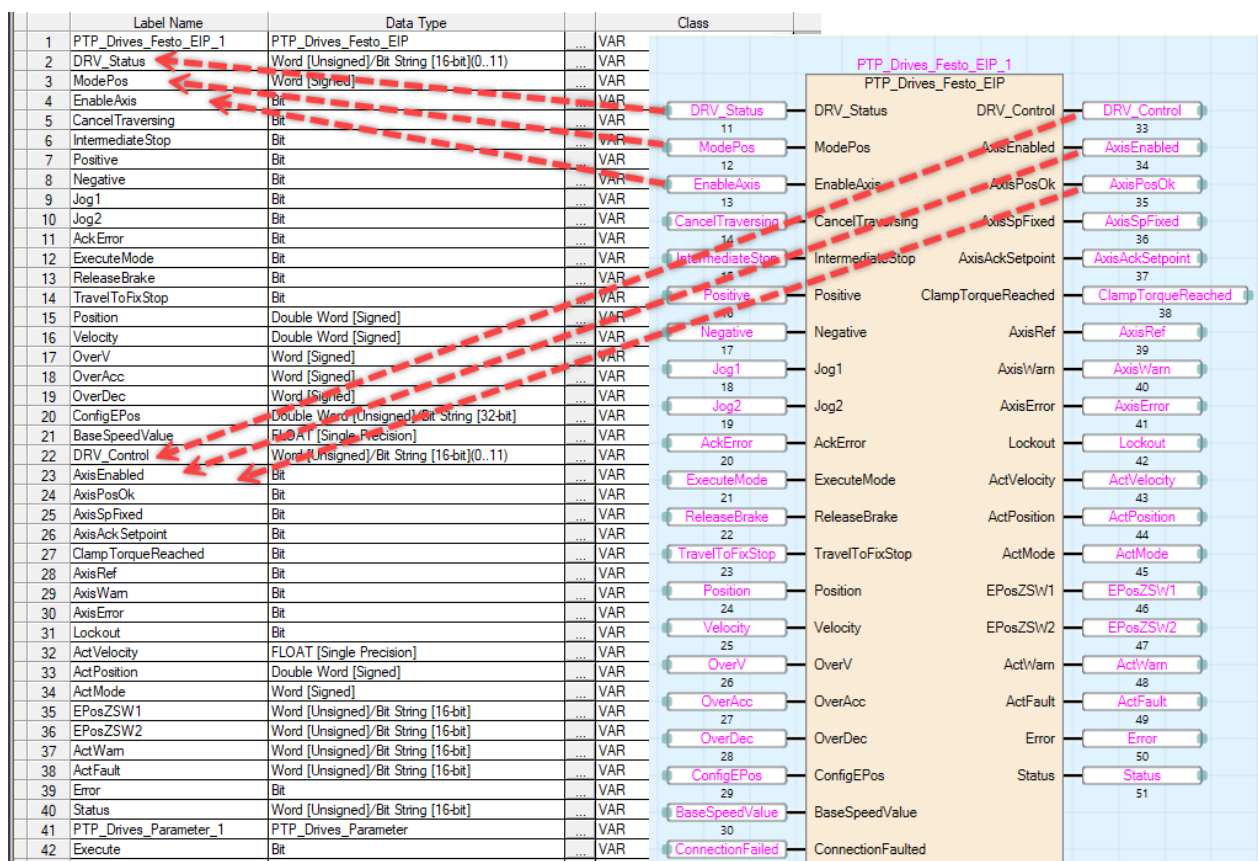


- User can also use Memory Bit & Device Memory which is highlighted in Yellow colour, or Tag name highlighted in Green colour and Integer highlighted in Purple colour.

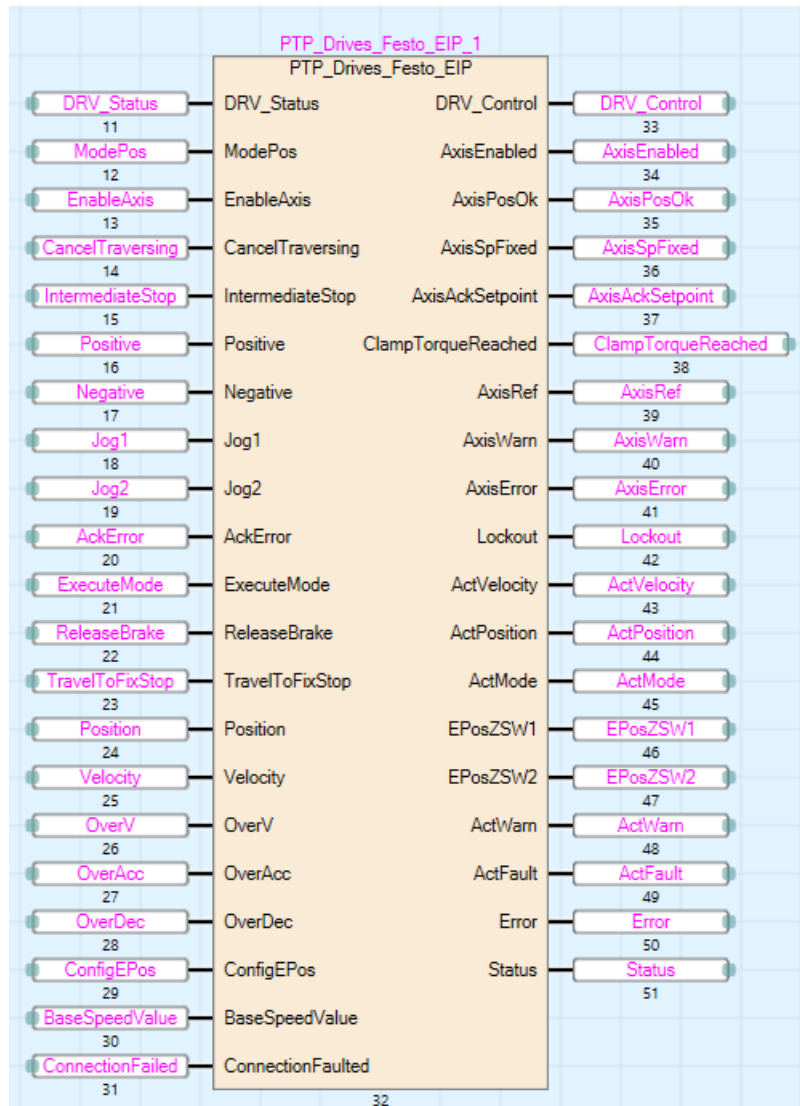


Note

- User have to map the correct communication module interface tag to FB module which are created.

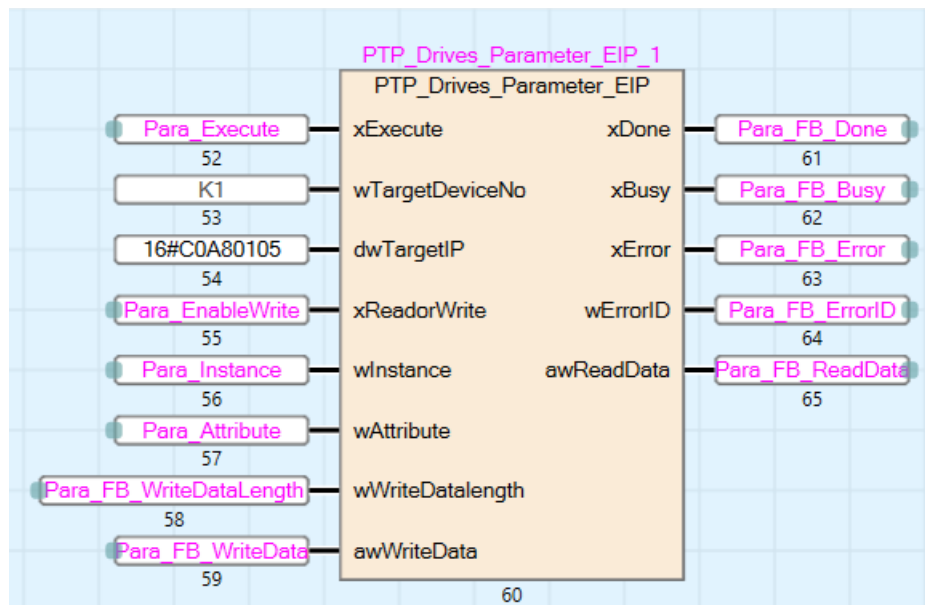


Mapping the Interface tag for PTP_Drives_Festo_EIP is as shown in above picture. Once the FB tags successfully configured then the view of FB looks like below.



4.2.2 Add “PTP_Drives_Parameter_EIP” in PLC Program

This chapter describes about adding “PTP_Drives_Parameter_EIP” FB in program & assigning input, output tags.

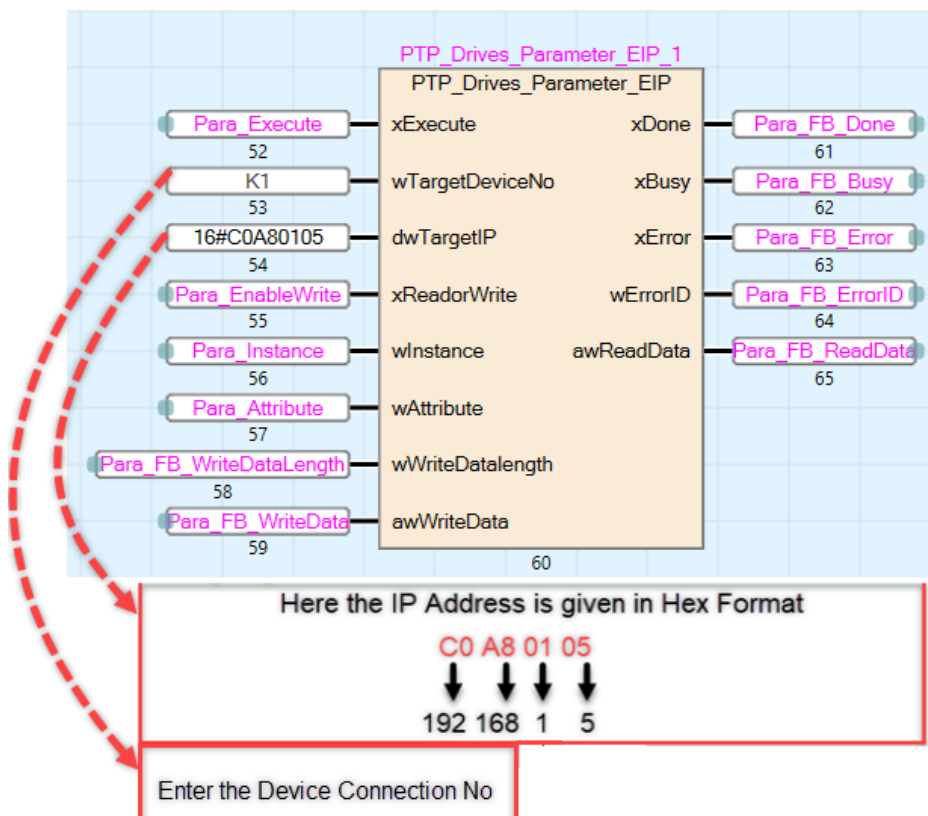


Configure the “PTP_Drives_Parameter_EIP” FB interface tags as shown above. Refer [Chapter 4.2.1 - Linking Tags to Function Block, Step - 1](#) to know how to create external tags and to map the tags.



Note

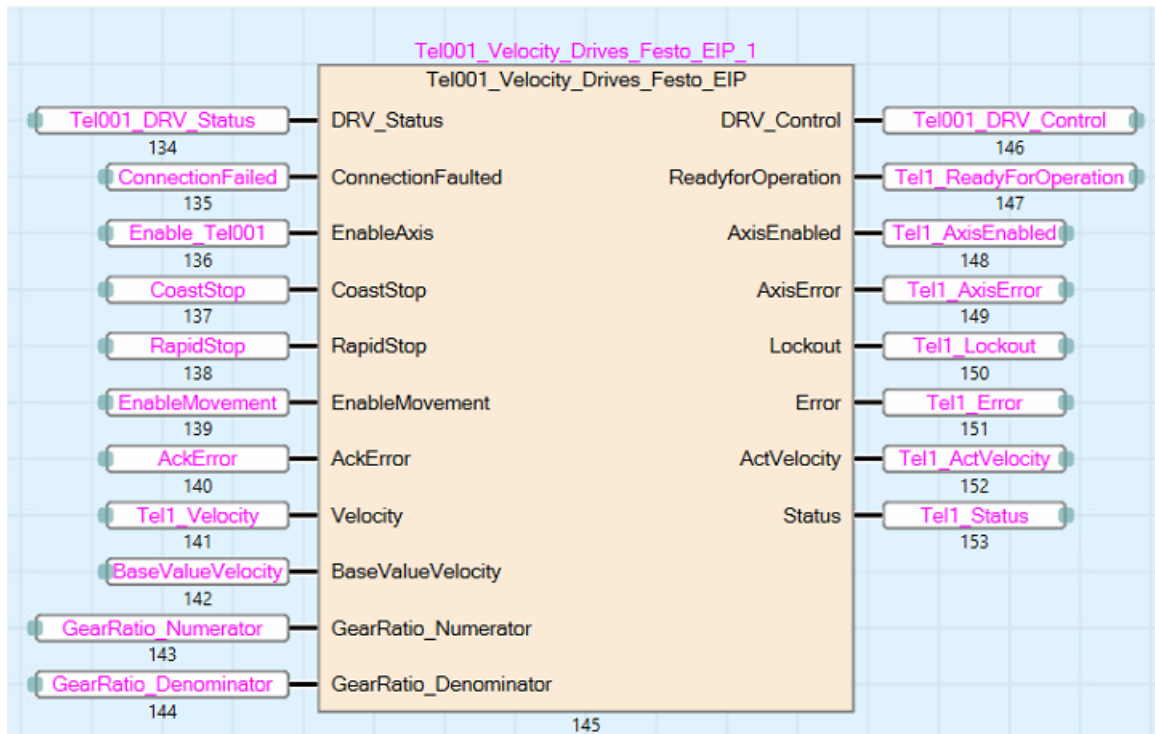
- Repeat the [Chapter 4.2.1 - Linking Tags to Function Block, Step - 1](#) follow either Method 1 or Method 2 the same procedure for rest of the Tag that's need for Function Block interface.
- Fill In and Out Tags of the Function Blocks.



Match the IP-Address “dwTargetIP” and “wTargetDeviceNo” of the function block to your “CMMT Drives” IP-Address and Device Connection No (Refer [Chapter 3.4 Step - 9](#) or [Chapter 3.5 Step - 3](#))

4.2.3 Add “Tel001_Velocity_Drives_Festo_EIP” in PLC Program

This chapter describes about adding “Tel001_Velocity_Drives_Festo_EIP” FB in program & assigning input, output tags.



Configure the “Tel001_Velocity_Drives_Festo_EIP” FB interface tags as shown above. Refer [Chapter 4.2.1 - Linking Tags to Function Block, Step - 1](#) to know how to create external tags and to map the tags.

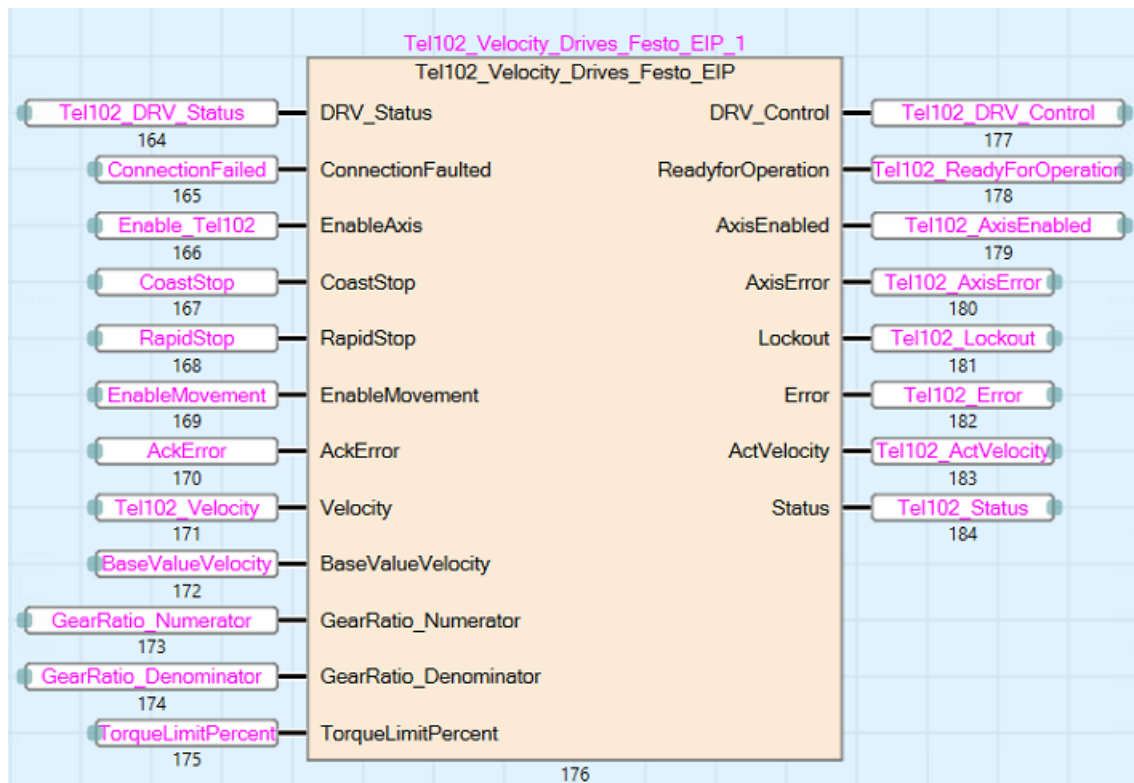


Note

- Repeat the [Chapter 4.2.1 - Linking Tags to Function Block, Step - 1](#) follow either Method 1 or Method 2 the same procedure for rest of the Tag that's need for Function Block interface.
- Fill In and Out Tags of the Function Blocks.

4.2.4 Add “Tel102_Velocity_Drives_Festo_EIP” in PLC Program

This chapter describes about adding “Tel102_Velocity_Drives_Festo_EIP” FB in program & assigning input, output tags.



Configure the “Tel102_Velocity_Drives_Festo_EIP” FB interface tags as shown above. Refer [Chapter 4.2.1 - Linking Tags to Function Block, Step - 1](#) to know how to create external tags and to map the tags.



Note

- Repeat the [Chapter 4.2.1 - Linking Tags to Function Block, Step - 1](#) follow either Method 1 or Method 2 the same procedure for rest of the Tag that's need for Function Block interface.
- Fill In and Out Tags of the Function Blocks.

5 Elements in the Festo_CMMT_Mitsubishi_IQR_EIP_Lib Library

Function Blocks

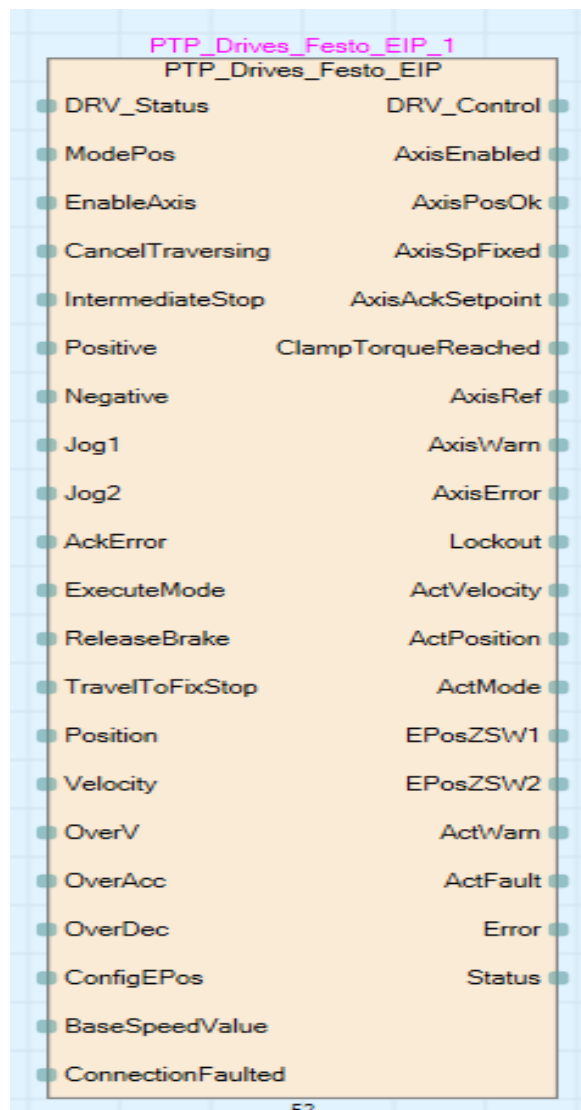
There are six function blocks in the library namely:

- PTP_Drives_Festo_EIP
- PTP_Drives_Parameter_EIP
- Word_2_DWord
- DWord_2_Word
- Tel001_Velocity_Drives_Festo_EIP
- Tel102_Velocity_Drives_Festo_EIP

5.1 PTP_Drives_Festo_EIP

PTP_Drives_Festo_EIP Function Block is used for controlling the drive in the following modes.

- **ModePos – 1:** Relative Positioning
- **ModePos – 2:** Absolute Positioning
- **ModePos – 3:** Setup Mode
- **ModePos – 4:** Home/Reference using Festo Automation Suite Configuration
- **ModePos – 5:** Home to Current Position
- **ModePos – 6:** Traversing/Record Select Mode
- **ModePos – 7:** Jog
- **ModePos – 8:** Inching/Incremental Jog Mode



The following tables gives a detailed explanation of the inputs, outputs data types of the function block.

5.1.1 Input and Output Interface

INPUT DATA TAGS

Tag Name	Data Type	Function Description
DRV_Status	Word [Unsigned]/Bit String [16-bit](0..11)	Gives the drive feedback.
ModePos	Word [Signed]	Gives the Operating mode selection command. It has the following values. 1=MDI Relative Pos. 2=MDI Absolute Pos. 3= Setup Mode 4=Home FAS Config. 5=Home Current Pos. 6=Record Select 7=Jogging 8=Inching
EnableAxis	Bit	TRUE - Enable the axis FALSE – DISABLES the axis.
CancelTraversing	Bit	Cancel travel task (0 – Cancel Travel) Ramp Stop @ 100% Set Deceleration Task is Rejected
IntermediateStop	Bit	Interrupt travel task (0 – Interrupt Travel) Ramp Stop @ Current Deceleration Task is NOT Rejected
Positive	Bit	TRUE -Travel positive direction.
Negative	Bit	TRUE -Travel negative direction
Jog1	Bit	TRUE – Drive jogs in Positive direction
Jog2	Bit	TRUE – Drive jogs in negative direction
AckError	Bit	Clear error command (1 – Clear error)
ExecuteMode	Bit	Execute the task (1 – Execute)
ReleaseBrake	Bit	Release Servo Motor Brake (1 – Release)
TravelToFixStop	Bit	Enable Travel to Fix Stop mode (1 – Enable)
Position	Double Word [Signed]	Position Target Setpoint or Record table selection in traversing block mode Position Target Value x FAS=P1.7841.0.0 Factor Group for Position = 10-6 = FAS P1.11280605.0.0 Target Position MDI Example: Position Target Setpoint= $2500000 \times 10^{-6} = 2500.000 \text{ mm}$

Tag Name	Data Type	Function Description
Velocity	Double Word [Signed]	Velocity Target Setpoint Velocity in User Units While using MDI (Manual Data Input) Mode Velocity Target Setpoint $\times$ FAS=P1.7842.0.0 Factor Group for Velocity = 10^{-3} $\times$ OverV FAS=P1.1309.0.0 Velocity Override = FAS P1.11280605.0.0 Profile Velocity MDI Example1: Velocity Target Setpoint = $50 \times 10^{-3} = 0.05 \text{ m/s}$ Example2: Velocity Target Setpoint = $250 \times 10^{-3} = 0.25 \text{ m/s}$
OverV	Word [Signed]	Velocity override in % (0 to 199 %) OverV Value = 0-100% FAS=P1.1309.0.0 Velocity Override
OverAcc	Word [Signed]	Acceleration override in % (0 to 100 %) OverAcc Target Value = 0-100% $\times$ FAS P1.11280702.0.0 Base Value Acceleration = FAS P1.11280606.0.0 Acceleration MDI
OverDec	Word [Signed]	Deceleration override in % (0 to 100 %) OverDec Target Value = 0-100% $\times$ FAS P1.11280703.0.0 Base Value Deceleration = FAS P1.11280607.0.0 Deceleration MDI
ConfigEPos	Double Word [Unsigned]/Bit String [32-bit]	Configuration command (Bit – Granular) See Chapter 5.1.2 for further details
BaseSpeedValue	FLOAT [Single Precision]	Base speed value for actual speed display. Enter the same value given for parameter (P1.11280701.0.0)
ConnectionFaulted	Bit	TRUE – Communication is disconnected. FALSE – Communication is not disconnected.

OUTPUT DATA TAGS

Tag Name	Data Type	Function Description
DRV_Control	Word [Unsigned]/Bit String [16-bit](0..11)	Drive control hardware interface.
AxisEnabled	Bit	TRUE - Axis enabled feedback Drive Ready and Switched On/Enabled
AxisPosOk	Bit	TRUE - Axis reached target position Reached & Motion Complete
AxisSpFixed	Bit	TRUE -Setpoint is stationary feedback
AxisAckSetpoint	Bit	TRUE - Acknowledge setpoint

Tag Name	Data Type	Function Description
ClampTorqueReached	Bit	TRUE - ClampTorqueReached
AxisRef	Bit	TRUE -Referencing/Homing is Completed
AxisWarn	Bit	TRUE - Drive has a warning message (Warning Active)
AxisError	Bit	TRUE -Drive has a error message (Error Active)
Lockout	Bit	TRUE -Switching ON inhibited (ON Inhibited)
ActVelocity	FLOAT [Single Precision]	Actual velocity feedback (Feedback value scaled by base speed value)
ActPosition	Double Word [Signed]	Axis actual position feedback
ActMode	Word [Signed]	Actual mode selection feedback 1=MDI Relative Pos. 2=MDI Absolute Pos. 4=Home FAS Config. 5=Home Current Pos. 6=Record Select 7=Jogging 8=Inching
EposZSW1	Word [Unsigned]/Bit String [16-bit]	Status of Epos ZSW1 (Bit – Granular) See chapter 5.1.3 for further details
EposZSW2	Word [Unsigned]/Bit String [16-bit]	Status of Epos ZSW2 (Bit – Granular) See chapter 5.1.4 for further details
ActWarn	Word [Unsigned]/Bit String [16-bit]	Actual warning number
ActFault	Word [Unsigned]/Bit String [16-bit]	Actual fault number
Error	Bit	TRUE -FB Fault Active
Status	Word [Unsigned]/Bit String [16-bit]	FB Status Code: See Chapter 5.1.5 for further details

5.1.2 ConfigEPos Bit Description

ConfigEPos Bits	Function Description
Bit – 0	OFF2 (Pulse Enable)
Bit – 1	OFF3 (Pulse Enable)
Bit – 2	Software Limits Enable FAS=P1.112414140.0.0
Bit – 3	Hardware Limits Enable FAS=P1.112414150.0.0
Bit – 4	Probe Edge Evaluation
Bit – 5	Select Probe
Bit – 6	External Block Change (via BUS)
Bit – 7	-FW Pending – Signal Source Reference Mark
Bit – 8	-FW Pending – Continuous Setpoint Transfer MDI
Bit – 9	- FW Pending – DDS Bit 0
Bit – 10	- FW Pending – DDS Bit 1
Bit – 11	- FW Pending – DDS Bit 2

ConfigEPos Bits	Function Description
Bit – 12	- FW Pending – DDS Bit 3
Bit – 13	- FW Pending - DDS Bit 4
Bit - 14	- FW Pending - Parking Axis Selection
Bit - 15	Motor Brake 0=Close 1=Open
Bit – 16 to Bit - 29	Reserved
Bit - 30	STW2.8 Travel to Fixed Stop
Bit - 31	Reserved

5.1.3 Epos ZSW1 Bit Details

EPos ZSW1 Bits	Function Description
Bit - 0	Position Record Selected Bit 0
Bit - 1	Position Record Selected Bit 1
Bit - 2	Position Record Selected Bit 2
Bit - 3	Position Record Selected Bit 3
Bit - 4	Position Record Selected Bit 4
Bit - 5	Position Record Selected Bit 5
Bit - 6	Position Record Selected Bit 6
Bit - 7	Reserved
Bit - 8	Negative Hardware Limit Switch is Activated
Bit - 9	Positive Hardware Limit Switch is Activated
Bit - 10	Jogging Task is Active
Bit - 11	Homing Task is Active
Bit - 12	- FW Pending - Flying Reference Active
Bit - 13	Position Record Task is Active
Bit - 14	- FW Pending - MDI Setup Active
Bit - 15	MDI Mode is Active

5.1.4 Epos ZSW2 Bit Details

EPos ZSW2 Bits	Function Description
Bit - 0	Tracking Mode Active
Bit - 1	Velocity Limit Active
Bit - 2	Ready to Accept new Setpoint
Bit - 3	- FW Pending - Print Mark Outside Outer Window
Bit - 4	Axis is Moving Positive
Bit - 5	Axis is Moving Negative
Bit - 6	Negative Software Limit Switch is Activated
Bit - 7	Positive Software Limit Switch is Activated
Bit - 8	Actual Position <= CAM Position 1

EPos ZSW2 Bits	Function Description
Bit - 9	Actual Position <= CAM Position 2
Bit - 10	Direct Output 1 Via Traversing Block
Bit - 11	Direct Output 2 Via Traversing Block
Bit - 12	Fixed STOP Reached
Bit - 13	Fixed STOP Clamping Torque Reached
Bit - 14	Travel to Fixed STOP Active
Bit - 15	Traversing Task Active

5.1.5 Diagnostic Messages

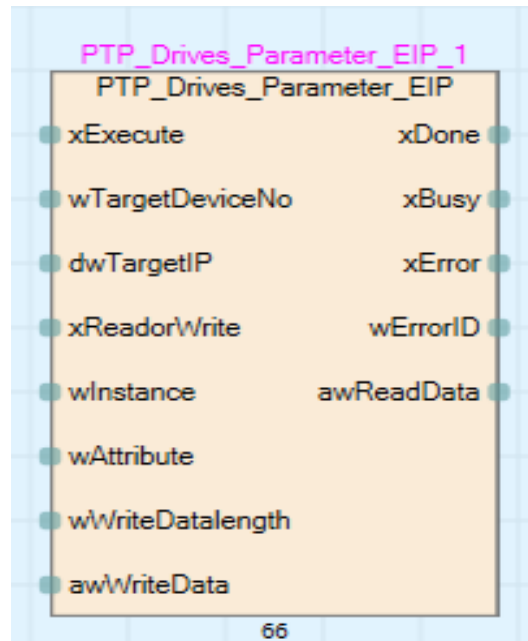
The function block gives multiple types of diagnostic messages. The following chapter will describes how to handle & solve them.

Function Block Faults

Status Error Number	Case	Remedy
16#7002	No Error	--
16#8401	Alarm message(s) in the drive	Check the error code in ActFaulted tag
16#8402	Switching ON is Inhibited .	Check whether axis is parked, Safety functions active & Parameter P10 <> 0
16#8403	- FW Pending - Flying Reference could NOT be Started	Check for pending alarms / faults in the drive
16#8600	Communication Disconnected	Check the communication connections/ setting between drive and PLC
16#8202	Incorrect Operating Mode Selected	Set ModePos tag value between 1 to 8
16#8203	Incorrect parameterization of the override inputs	Check the setting of override inputs
16#8204	Invalid traversing block number	Enter the traversing block number from 0 to 127

5.2 PTP_Drives_Parameter_EIP

This Function Block allows the user to read and write a single parameter of CMMT drive from Mitsubishi PLC via Ethernet/IP protocol.



5.2.1 Input and Output Interface

INPUT DATA TAGS

Tag Name	Data Type	Description
xExecute	Bit	TRUE = FB is executed once
wTargetDeviceNO	Word [Unsigned]/Bit String [16-bit]	Device Connection No. (Refer Chapter 3.4 Step - 9 or Chapter 3.5 Step – 3) Either enter Device No or number between 1 to 32
dwTargetIP	Double Word [Un- signed]/Bit String [32-bit]	IP address of the targeted device is “192.168.1.5” Give that in HEX Format 16#C0A80105
xReadorWrite	Bit	FALSE = Read TRUE = Write
wInstance	Word [Unsigned]/Bit String [16-bit]	Parameter Number
wAttribute	Word [Unsigned]/Bit String [16-bit]	Parameter Index
wWriteDataLength	Word [Unsigned]/Bit String [16-bit]	Length in Word of Data variable
awWriteData	Word [Unsigned]/Bit String [16-bit](0..127)	Array of Variable through which data is added to Write in Drive.

OUTPUT DATA TAGS

TagName	Data Type	Description
xDone	BOOL	TRUE = FB successfully executed
xBusy	BOOL	TRUE = FB being executed
xError	BOOL	TRUE = Error during execution

wErrorID	Word [Unsigned]/Bit String [16-bit]	Refer Chapter 5.2.2
awReadData	Word [Unsigned]/Bit String [16-bit](0..127)	Array of Variable through which data is Read from Drive.

5.2.2 wErrorID

Error Code (Hex)	Extended Status Error Code	Error Code (Dec)	Name	Description
0x01	0x012A	1	Invalid Consuming Application Path	This error will be returned by the CLICK CPU if the Consuming (O → T) Application Path is not present in the Forward Open message coming from the Scanner device or if the specified Connection Point is incorrect.
0x01	0x012B	1	Invalid Producing Application Path	This error will be returned by the CLICK CPU if the Producing (T → O) Application Path is not present in the Forward Open message coming from the Scanner device or if the specified Connection Point is incorrect.
0x01	0x012F	1	Inconsistent Application Path Combination	The combination of Configuration, Consuming and Producing application paths specified are inconsistent.
0x01	0x0132	1	Null Forward Open function not supported	The target device does not support the function requested in the NULL Forward Open request. The request could be such items as "Ping device", "Configure device application", etc.
0x01	0x0315	1	Invalid Segment in Connection Path	This error occurs when the target device cannot understand the segment type or segment value in the Connection Path. Consult the documentation of the device to verify the correct segment type and value. If a Connection Point greater than 255 is specified this error could occur.
0x01	0x0813	1	Not Configurable for Off-subnet Multicast	The producer has been requested to support a Multicast connection for a consumer on a different subnet and does not support this functionality.
0x04	N/A	4	Path Segment Error	The path segment identifier or the segment syntax was not understood by the target device.
0x05	N/A	5	Path Destination Error	The Class, Instance or Attribute value specified in the Unconnected Explicit Message request is incorrect or not supported in the Target device. Check the manufacturer's documentation for the correct codes to use.
0x08	N/A	8	Service not supported	The requested service was not implemented or was not defined for this Object.
0x09	Index to Error	9	Error in Data Segment	This error code is returned when an error is encountered in the Data segment portion of a Forward Open message. The Extended Status value is the offset in the Data segment where the error was encountered.
0x10	N/A	16	Device state conflict	CIP Status: 0x10. Device state conflict. The device current mode/state prohibits the execution of the requested service
0x0E	N/A	14	Attribute not settable	A request to modify a non-modifiable attribute was received.
0x0F	N/A	15	Privilege violation	A permission/privilege check failed

0x13	N/A	19	Not Enough Data	Not enough data was supplied in the service request specified.
0x14	N/A	20	Attribute not supported	The attribute specified in the request is not supported
0x15	N/A	21	Too Much Data	Too much data was supplied in the service request specified.
0x16	N/A	22	Object does not exist	The object specified does not exist in the device.
0x20	N/A	32	Invalid parameter	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
0x26	N/A	38	Path Size Invalid	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.
0x27	N/A	39	Connection Number Invalid	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.

5.2.3 Identification of Parameter Number, Index, Data Size during PTP_Drives_Parameter_EIP Configuration

- For reading or writing a CMMT drive parameter we need 3 important data namely:
 - Parameter Number
 - Index of the PNU
 - Data size to be read.
- The above 3 values can be obtained from the **PNUs Reference List table (Page No: 926) from CMMT drive user manual.**
- Suppose our Function Block has to read the **Status load voltage** of the CMMT drive. From the CMMT user manual we get the following information.

PNU	Name	Data type	Access	Parameter
2546.0	Storage option in error log	Unsigned8	rw	P0.4890.0.0
2549.0	Status load voltage	Boolean	ro	P0.4995.0.0

- From the below example we come to know that to read **Status Load voltage** we need the following parameter.
 Parameter Number = 2549 , Index of the PNU = 0 , Data size to be read.= 1 Bit., Access = Read Only.

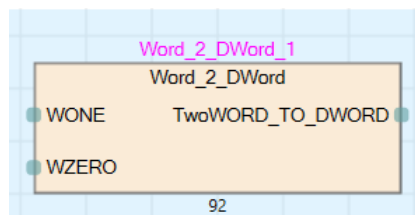


Note

- Parameters whose **Access is "RO"** can only be read and a new value to the parameter cannot be written using the PTP_Drives_Parameter_EIP.
- Parameters whose **Access is "RW"** can only be read and a new value to the parameter can be written also using the PTP_Drives_Parameter_EIP.

5.3 Word_2_DWord

This Function Block allows the user to Convert Array of word input to Double Word output. This Block Output is used give input data to TwoWORD_TO_DWORD of PARAMETER Block.



5.3.1 Input Interface

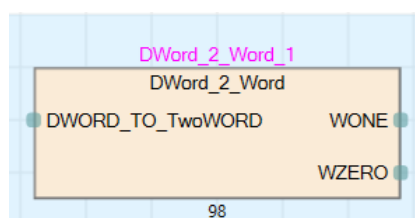
Tag Name	Data Type	Description
WONE	Word [Unsigned]/Bit String [16-bit]	Second word from Array of Two word.
WZERO	Word [Unsigned]/Bit String [16-bit]	First word from Array of Two word.

5.3.2 Output Interface

TagName	Data Type	Description
TwoWORD_TO_DWORD	Double Word [Unsigned]/Bit String [32-bit]	Double Word

5.4 DWord_2_Word

This Function Block allows the user to Convert Double Word Input to Array of word Output. This Block Output is used give input data to WONE AND WZERO of PARAMETER Block.



5.4.1 Input Interface

TagName	Data Type	Description
DWORD_TO_TwoWORD	Double Word [Unsigned]/Bit String [32-bit]	Word

5.4.2 Output Interface

Tag Name	Data Type	Description
WONE	Word [Unsigned]/Bit String [16-bit]	Second word from Array of Two word.
WZERO	Word [Unsigned]/Bit String [16-bit]	First word from Array of Two word.

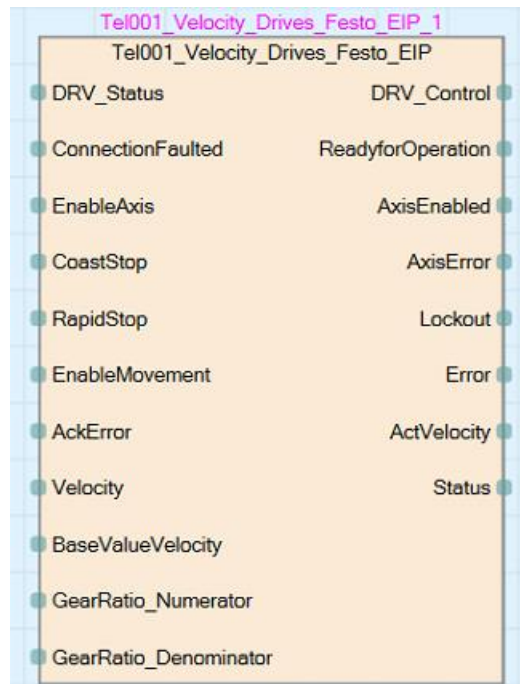
5.5 “Tel001_Velocity_Drives_Festo_EIP” / Basic Velocity Control

5.5.1 “Tel001_Velocity_Drives_Festo_EIP” FB Introduction

The drive is controlled by a main setpoint (Velocity Setpoint) via Telegram 1, the velocity is controlled completely in the drive.

The “Tel001_Velocity_Drives_Festo_EIP” FB is used to control the CMMT drive in velocity mode. This FB controls the drive by toggling appropriate control bits & speed has been controlled by setting RPM value in specified control word. This FB controls the enabling of the axis, coast stop, rapid stop & drive fault reset.

“Tel001_Velocity_Drives_Festo_EIP” FB is also designed to update the feedback status like Drive Enabled , Drive Ready, Drive Fault is Active, Drive in Inhibited status & Actual axis RPM feedback.



5.5.2 Input and Output Interface

IO Interface Area Tag Details

Interface Area Tags	Data Type	Function Description
DRV_Status	ARRAY OF INT [0.. 1]	Ethernet Module Status Variables
DRV_Control	ARRAY OF INT [0.. 1]	Ethernet Module Control Variables

INPUT DATA TAGS

Control Area Tags	Data Type	Meaning
ConnectionFaulted	BOOL	Communication disconnected acknowledgement for FB
EnableAxis	BOOL	Drive Enable (ON/OFF1) 1 = Enable Drive 0 = Ramp Down & Disable drive
CoastStop	BOOL	Drive coast (OFF 2) – 1: no coasting – 0: coasting. Power output stage is deactivated (OFF2). The drive coasts to a stop.

RapidStop	BOOL	Fast stop (OFF 3) – 1: no fast stop – 0: brake drive to standstill with fast stop and then deactivate the power output stage (OFF3).
EnableMovement	BOOL	Enable rotational speed setpoint value 1 = Enable
AckError	BOOL	Reset Active Faults 1 = Reset Faults
Velocity	REAL	Speed Setpoint as per Actual user Unit.
BaseValueVelocity	REAL	Base Value Velocity. User can find the value from the parameter P1.11280701.0.0
GearRatio_Numerator	INT	Determine the numerator of the overall gear ratio(drive side). User can find the value from the parameter P1.1242.0.0
GearRatio_Denominator	INT	Determine the denominator of the overall gear ratio(drive side). User can find the value from the parameter P1.1243.0.0

OUTPUT DATA TAGS

Output signal	Type	Meaning
ReadyforOperation	BOOL	Ready to State
AxisEnabled	BOOL	Pulses are Enabled
AxisError	BOOL	Fault is Active
Lockout	BOOL	Drive Inhibited
Error	BOOL	Fault Active
ActVelocity	REAL	Actual velocity as per Actual user Unit.
Status	INT	Status Code: For more details refer Chapter 5.5.3.

5.5.3 Diagnostic Messages / Function Block Faults

The function block gives multiple types of diagnostic messages. The following chapter will describes how to handle & solve them.

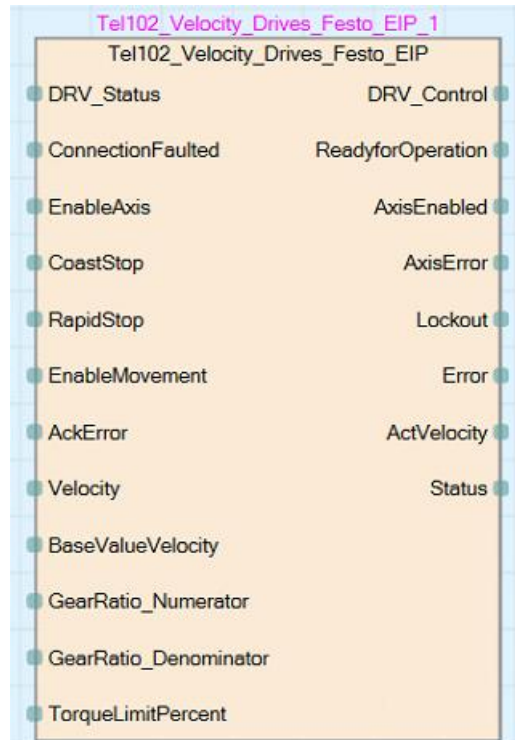
Status Error Number	Case	Remedy
7002	No Error	--
8401	Alarm message(s) in the drive	Check the error code in ActFaulted tag
8402	Switching ON Inhibited	The switch-on lock is active. The drive is in the status S1 Switching On Inhibited. Switching on is only possible with the following command sequence: OFF (OFF1) and no coasting (no OFF2) and no fast stop (no OFF3) and then ON.
8403	No Entry Error	Set the Base Value Velocity or Gear Ratio as per FAS.
8600	Communication Disconnected	Check the communication connections/ setting between drive and PLC.

5.6 “Tel102_Velocity_Drives_Festo_EIP” / Advance Velocity Control

5.6.1 “Tel102_Velocity_Drives_Festo_EIP” FB Introduction

The drive is controlled by a main setpoint (Velocity Setpoint) via Telegram 102, the velocity is controlled completely in the drive.

The “Tel102_Velocity_Drives_Festo_EIP” is used to control the CMMT drive in velocity mode with specified torque limit. This is the major different between advanced & basic velocity modes.



5.6.2 Input and Output Interface

IO Interface Area Tag Details

Interface Area Tags	Data Type	Function Description
DRV_Status	ARRAY OF INT [0.. 10]	Ethernet Module Status Variables
DRV_Control	ARRAY OF INT [0.. 6]	Ethernet Module Control Variables

INPUT DATA TAGS

Control Area Tags	Data Type	Function Description
ConnectionFaulted	BOOL	Communication disconnected acknowledgement for FB
EnableAxis	BOOL	Drive Enable (ON/OFF1) 1 = Enable Drive 0 = Ramp Down & Disable drive
CoastStop	BOOL	Drive coast (OFF 2) – 1: no coasting – 0: coasting. Power output stage is deactivated (OFF2). The drive coasts to a stop.
RapidStop	BOOL	Fast stop (OFF 3) – 1: no fast stop

		– 0: brake drive to standstill with fast stop and then deactivate the power output stage (OFF3).
EnableMovement	BOOL	Enable rotational speed setpoint value 1 = Enable
AckError	BOOL	Reset Active Faults 1 = Reset Faults
Velocity	REAL	Speed Setpoint as per Actual user Unit.
BaseValueVelocity	REAL	Base Value Velocity. User can find the value from the parameter P1.11280701.0.0
GearRatio_Numerator	INT	Determine the numerator of the overall gear ratio(drive side). User can find the value from the parameter P1.1242.0.0
GearRatio_Denominator	INT	Determine the denominator of the overall gear ratio(drive side). User can find the value from the parameter P1.1243.0.0
TorqueLimitPercent	REAL	Torque Limit Setpoint in Percentage 0 = No Torque Limit, 100 = Full Torque Limit

OUTPUT DATA TAGS

Monitor Area Tags	Data Type	Function Description
ReadyforOperation	BOOL	Ready to Start
AxisEnabled	BOOL	Pulses are Enabled
AxisError	BOOL	Fault is Active
Error	BOOL	Fault Active
ActVelocity	REAL	Actual velocity as per Actual user Unit.
Status	INT	Status Code: For more details refer Chapter 5.6.3.

5.6.3 Diagnostic Messages / Function Block Faults

The function block gives multiple types of diagnostic messages. The following chapter will describes how to handle & solve them.

Status Error Number	Case	Remedy
7002	No Error	--
8401	Alarm message(s) in the drive	Check the error code in ActFaulted tag
8402	Switching ON Inhibited	The switch-on lock is active. The drive is in the status S1 Switching On Inhibited. Switching on is only possible with the following command sequence: OFF (OFF1) and no coasting (no OFF2) and no fast stop (no OFF3) and then ON.
8403	No Entry Error	Set the Base Value Velocity or Gear Ratio as per FAS.
8404	Incorrect parameterization of the TorqueLimitPercent input	Check the setting of TorqueLimitPercent Range, set the value within the Range (0% - 100%).
8600	Communication Disconnected	Check the communication connections/ setting between drive and PLC.

6 CMMT Drive in Position Control Mode (PTP_Drives_Festo_EIP) FUNCTIONS

This chapter describes about the configuration of **PtP_Drives_Festo_EIP** Function Block for different operation modes, Enabling & Stopping the axis, Enabling hardware & software position limits as well as Actual speed monitoring.

6.1 Axis Enabling & Stopping

For axis enabling following tags configuration must be done.

- “**ConfigEPos**” Bit 0 & Bit1(OFF2 & OFF3) must be **TRUE**.
- “**AxisError**” & “**Lockout**” output tags must be **FALSE** to ensure drive is in ready state.
- “**ModePos**” value must be between 1 to 8.
- “**CancelTraversing**”(Reject traversing task) & “**IntermediateStop**”(intermediate stop) tags must be **TRUE**.
- “**EnableAxis**” must be **TRUE** & “**AxisEnabled**” feedback should become **TRUE** after enabling the drive.
- Toggle “**AckError**” control bit to clear any error in the Function Block.

6.2 Axis Software & Hardware Position Limits Enabling

“**ConfigEPOS tag’s Bit 2 & 3**” are used to enable the software & hardware position limits respectively

- “**ConfigEPOS.2 = TRUE**” This means software position limit is active (Festo Automation Suite Parameter **P1.112414140.0.0**)
- “**ConfigEPos.3 = TRUE**” This means hardware position limit is active (Festo Automation Suite Parameter **P1.112414150.0.0**)

When software position limit is active please pay attention if the below parameters are configured correctly as per application requirement.

The screenshot shows the 'Axis configuration' window. On the left, there is a tab labeled 'Axis'. The main area contains several configuration options:

- Reversing the direction of rotation:** A circular arrow icon and a checkbox labeled 'Active'.
- Axis zero point offset:** A text input field containing '0.03' and a unit 'r'.
- Software limit positions active:** A checkbox labeled 'Active' which is checked.
- Negative software limit position:** A text input field containing '-0.03' and a unit 'r'.
- Positive software limit position:** A text input field containing '999.97' and a unit 'r'.

6.3 Axis Actual Speed Monitoring

User can monitor the actual speed of the axis in the variable “**ActVelocity**”. The actual speed of the axis is scaled from 0 to 100% depending on the base speed value of parameter number (**P1.11280701.0.0**). So it is mandatory to enter the same value in input tag “**BaseSpeedValue**” of the Function Block as like parameter number (**P1.11280701.0.0**).



Note

- Make sure the Function Block input tag “**BaseSpeedValue**” & Festo Automation Suite parameter number (**P1.11280701.0.0**) values are same. Function Block will display wrong velocity value If both values are not equal.

6.4 PtP_Drives_Festo_EIP Modes of Operation

The PtP_Drives_Festo will control the drive in the following modes

- **ModePos – 1:** Relative Positioning
- **ModePos – 2:** Absolute Positioning
- **ModePos – 3:** Not Supported
- **ModePos – 4:** Home/Reference using Festo Automation Suite Configuration
- **ModePos – 5:** Home to Current Position
- **ModePos – 6:** Traversing/Record Select Mode
- **ModePos – 7:** Jog
- **ModePos – 8:** Inching/Incremental Jog Mode

6.4.1 Relative Positioning Mode [ModePos – 1]

1. Configurations for Relative Positioning Mode

- Set the “**ModePos**” value to 1.
- Check “**AxisRef**” status tag is True. If not true, the user could also use other modes like mode 4 or 5 to set the reference point for an axis.
- Enable axis by toggling “**EnableAxis**” command tag.
- “**CancelTraversing**” and “**IntermediateStop**” command tags must be in true state.
- Enter the percentage of Override acceleration, Override deceleration & Override velocity in “**Over-Acc**”, “**OverDec**” & “**OverV**” respective command tags.
- The status tags “**AxisEnabled**” & “**AxisRef**” must be true and status tag “**Error**” must be false.

2. Sequence Command

- Enter the Position command value in “**Position**” tag & Velocity command value in “**Velocity**” tag. To move axis in forward direction, give position command in positive value & for moving axis in reverse direction give position command in negative value.
- Toggle the “**ExecuteMode**” command tag to start motion task for configured parameter. Rising edge of ExecuteMode tag will start the command. So, for every new position command tag want to toggle. Check the feedback tag “**ActMode**” has the same value as ModePos.
- User will receive execute command accepted handshake signal from “**AxisAckSetpoint**” status tag.
- If task is completed successfully, it is acknowledge by rising edge of “**AxisPosOk**” status variable. Also if task completed unsuccessfully with an error “**AxisError**” status variable will indicate it by going false to true.
- “**AxisSpFixed**” Status variable indicates the axis mobility status. If the axis is standby the “**Axis-SpFixed**” status variable will be high.



Note

- Refer the document “**CMMT-xx-EP PTP_Drives_Festo_EIP Direct Mode 2 EtherNet/IP Timing Sequence**” document available with this download for execution command sequence.
- The existing running task command can be replaced by new task command only on **Mode 1 and 2**.

6.4.2 Absolute Positioning Mode [ModePos – 2]

1. Configuration for Absolute Positioning Mode

- Set the “**ModePos**” value to 2 .
- Check “**AxisRef**” status tag is True. If not true user could also use other modes like mode 4 or 5 to set the reference point for an axis.
- Enable axis by toggling “**EnableAxis**” command tag .
- “**CancelTraversing**” and “**IntermediateStop**” command tags must be in true state.
- Enter the percentage of Override acceleration, Override deceleration & Override velocity in “**OverAcc**”, “**OverDec**” & “**OverV**” respective command tags.
- The status tags “**AxisEnabled**” & “**AxisRef**” must be true. Also status tag “**Error**” must be false.

2. Sequence Command

- Enter the Position command value in “**Position**” tag & Velocity command value in “**Velocity**” tag. Give the absolute position value. The drive will decide forward & reverse direction based on its current position.
- Toggle the “**ExecuteMode**” command tag to start motion task for configured parameter. Rising edge of ExecuteMode tag will take for task start command. So for every new position command tag want to toggle. Check the feedback tag “**ActMode**” has the same value as ModePos.
- User will receive execute command accepted handshake signal from “**AxisAckSetpoint**” status tag.
- If commanded task is completed successfully, it is acknowledge by rising edge of “**AxisPosOk**” status variable. Also if task completed unsuccessfully with an error “**AxisError**” status variable will indicate it by going false to true.
- “**AxisSpFixed**” Status variable indicates the axis mobility status . If the axis is standby the “**AxisSpFixed**” status variable will be high.



Note

- Refer the document “**CMMT-xx-EP PTP_Drives_Festo_EIP Direct Mode 2 EtherNet/IP Timing Sequence**” document available with this download for execution command sequence.
- The existing running task command can be replaced by new task command only on **Mode 1 and 2**.

6.4.3 Setup Mode [ModePos – 3]

In the setup mode, the drive is traversed at a constant speed by means of position control. There’s no need to specify a target position. It must be noted that continuous motion (e.g. conveyor belt, fan etc.) cannot be implemented in this mode.

1. Configuration for Setup Mode

- Set the “**ModePos**” value to 3.
- Check “**AxisRef**” status tag is True. If not true user could also use other modes like mode 4 or 5 to set the reference point for an axis.
- “**CancelTraversing**” and “**IntermediateStop**” command tags must be in true state.
- Enter the percentage of Override acceleration, Override deceleration & Override velocity in “**OverAcc**”, “**OverDec**” & “**OverV**” respective command tags.
- Select the axis moving direction by the inputs “**Positive**” or “**Negative**”.
- Enable axis by toggling “**EnableAxis**” command tag.

2. Sequence Command

- Motion dynamics are defined via the “**Position**”, “**Velocity**”, “**OverV**”, “**OverAcc**” and “**OverDec**” inputs.
- Motion is started by means of a positive edge at “**ExecuteMode**”. The motion command is acknowledged via the “**AxisAckSetpoint**” output. A handshake mechanism can be programmed in this way. Check the feedback tag “**ActMode**” has the same value as ModePos.
- If a drive error has occurred during motion, this is indicated via the “**AxisError**” output. Any error occurring at the function block is indicated via the “Error output”.

6.4.4 Home/Reference Point using Festo Automation Suite Configuration [ModePos – 4]

The reference point approach mode is used to moves the axis to home position.

1. Configuration for Reference Point Approach Mode

- Set the “**ModePos**” value to 4 . The status tags “**AxisEnabled**” must be true & “**Error**” must be in false to ensure no error.
- Axis must be in standstill state.

2. Sequence Command

- All the required parameters for homing will be considered from Automation Suite configuration in “Homing Parameters” like Homing mode, velocity, acceleration & deceleration.
- Toggle the “**ExecuteMode**” to start task. Check the feedback tag “**ActMode**” has the same value as ModePos.
- “**AxisRef**” will become true once the axis finds reference point. If an error occurs during traveling motion, The output signal “**Error**” will become true .

6.4.5 Home to Current Position [ModePos – 5]

This mode marks current position as reference point. This will be used when homing mode is selected as current position(37).

1. Configuration for Set Reference Point Mode

- Set the “**ModePos**” value to 5. The status tags “**AxisEnabled**” must be true & “**Error**” must be in false to ensure no error.
- Axis must be in standstill state.

2. Sequence Command

- Toggle the “**ExecuteMode**” to start task. Check the feedback tag “**ActMode**” has the same value as ModePos.
- “**AxisRef**” will become true after execute command. If an error occurs, The output signal “**Error**” will be true.

6.4.6 Traversing/Record Selection Mode [ModePos – 6]

Traversing Block mode is implemented via the “Record Table” of drive function. This mode retrieves parameters like Velocity, Position, Acceleration & Deceleration values from record table saved in the CMMT drive. It enables the creation of automatic programs, velocity mode, force mode, travel to fixed stop and outputs to be set and reset.

1. Configuration for Traversing Block Mode

- Set the “**ModePos**” value to 6.
- Check “**AxisRef**” status tag is True. If not true user could also use other modes like mode 4 or 5 to set the reference point for an axis.
- Enable axis by toggling “**EnableAxis**” command tag .
- “**CancelTraversing**” and “**IntermediateStop**” command tags must be in true state.
- Enter the percentage of Override acceleration, Override deceleration & Override velocity in “**Over-Acc**”, “**OverDec**” & “**OverV**” respective command tags.
- The status tags “**AxisEnabled**” & “**AxisRef**” must be true. Also status tag “**Error**” must be false.

2. Sequence Command

- Select the record table by entering the table number in “**Position**” command tag. The value must be between 0 to 127.
- Toggle the “**ExecuteMode**” command tag to start motion task for configured parameter. Rising edge of ExecuteMode tag will start the command. So for every new position command tag want to toggle. Check the feedback tag “**ActMode**” has the same value as ModePos.
- User will receive execute command accepted handshake signal from “**AxisAckSetpoint**” status tag.
- If task is completed successfully, it is acknowledge by rising edge of “**AxisPosOk**” status variable. Also if task completed unsuccessfully with an error “**AxisError**” status variable will indicate it by going false to true.

- “**AxisSpFixed**” Status variable indicates the axis mobility status . If the axis is standby the “**AxisSpFixed**” status variable will be high.

**Note**

Refer the Chapter – 4 from the CMMT user manual to know the behaviour of the drive in different operation modes. And also (Refer [Chapter – 10](#))

6.4.7 Jog Mode [ModePos – 7]

1. Configuration for Jog Mode

- Set the “**ModePos**” value to 7 & check the feedback tag “**ActMode**” has the same value as ModePos.
- Enable axis by toggling “**EnableAxis**” command tag.
- Axis must be in standstill .
- Velocity, Acceleration & Jerk values are taken from Automation Suite configuration under **Jog Mode** tab.

2. Sequence Command

- “**Jog1**” & “**Jog2**” tags are used for Positive & Negative direction commands respectively.
- The “**ExecuteMode**” signal is not required for jog mode.

**Note**

- Refer the Chapter – 4 from the CMMT user manual to know the behaviour of the drive in different operation modes.

6.4.8 Inching/Incremental Jog Mode [ModePos – 8]

Incremental jog mode will not work like ordinary jog mode. It will rotate the axis step by step forward or reverse based on direction selection, But ordinary jog mode rotates the axis continuously. All the configuration and command configurations are same as **ModePos – 7** but only difference is that **ModePos** value must be 8 for incremental jog mode.

**Note**

- The inching distance of positive & negative direction will be based on values set in parameters Relative Position Jog1 (P1.214530.0.0) & Relative Position Jog2 (P1.214538.0.0) respectively.

6.4.9 Procedure to set the CMMT-AS in “Travel to fixed Stop” mode with the PTP_Drive_Festo_EIP function block

- Using a positioning mode absolute or relative (ModePos = 1 or 2 (positioning absolute/relative))

Travel to fixed Stop using Absolute Positioning Mode

1. Configuration for Travel to fixed Stop using Absolute Positioning Mode

- Set the “**ModePos**” value to 2 & check the output tag “**ActMode**” has the same value as ModePos.
- Enable the axis by toggling “**EnableAxis**” input tag.
- Check “**AxisRef**” status tag is True. If not true user could also use other modes like mode 4 or 5 to set the reference point for an axis.
- “**CancelTraversing**” and “**IntermediateStop**” input tags must be **TRUE**.
- Enable the axis by toggling “**TravelToFixStop**” input tag.
- Enter the percentage of Override acceleration, Override deceleration & Override velocity in “**OverAcc**”, “**OverDec**” & “**OverV**” respective input tags.
- The status tags “**AxisEnabled**” & “**AxisRef**” must be **TRUE**. Also status tag “**Error**” must be **FALSE**.
- Set the “**Clamping torque**” as per user defined, this is taken from the EPD.

2. Sequence Command

- Enter the Position command value in “**Position**” tag & Velocity command value in “**Velocity**” tag. Give the absolute position value. The drive will decide forward & reverse direction based on its current position.
- Toggle the “**ExecuteMode**” command tag (Rising Edge) to start a motion task using the configured parameters (Every new position task would require a toggle of the “**ExecuteMode**” command tag).
- The “**AxisAckSetpoint**” status tag will be true to acknowledge the task has been started.
- If the task is successfully completed, it is acknowledge by rising edge of “**AxisPosOk**” status variable. Also if task is unsuccessful “**AxisError**” output tag will be **TRUE**..
- “**AxisSpFixed**” Status variable indicates the axis mobility status . If the axis is standby the “**AxisSpFixed**” status variable will be high.
- The “**ClampTorqueReached**” status tag will be true to acknowledge the “**Clamping torque**” has been reached.

The screenshot displays the PTP_Drives_Festo_EIP_1 function block interface. It features a central table of input and output tags with their current values. To the right, a 'Watch 1[Watching]' panel lists the same tags with their current values.

Name	Current Value
Drives_Controls/ModePos	2
Drives_Controls/Position	10,000,000
Drives_Controls/Velosity	50,000
Drives_Controls/OverV	100
Drives_Controls/OverAcc	100
Drives_Controls/OverDec	100
Drives_Controls/ConfigEPos	3
Drives_Controls/BaseSpeedValue	3000.000000
Drives_Controls/Para_Instance	12,168
Drives_Controls/Para_Attribute	0
Drives_Controls/Para_FB_WriteDataLength	4
Drives_Controls/Para_FB_WriteData	
Drives_Controls/Para_FB_ReadData	
Drives_Controls/Parameter_Write	15.000000

The function block interface includes the following tags and values:

Input Tag	Value	Output Tag	Value
14903	K12	1151	K12
ModePos	2	AxisEnabled	TRUE
EnableAxis	TRUE	AxisPosOk	FALSE
CancelTraversing	TRUE	AxisSpFixed	TRUE
IntermediateStop	TRUE	AxisAckSetpoint	TRUE
Positive	FALSE	ClampTorqueReached	TRUE
Negative	FALSE	AxisRef	TRUE
Jog1	FALSE	AxisWarn	FALSE
Jog2	FALSE	AxisError	FALSE
AckError	FALSE	Lockout	FALSE
ExecuteMode	TRUE	ActVelocity	1.360622
ReleaseBrake	FALSE	ActPosition	3600104
TravelToFixStop	TRUE	ActMode	2
Position	10000000	EPosZSW1	32768
Velocity	50000	EPosZSW2	57360
OverV	100	ActWarn	0
OverAcc	100	ActFault	0
OverDec	100	Error	FALSE
ConfigEPos	3	Status	7002
BaseSpeedValue	3000.000000		
ConnectionFailed	FALSE		



Note

- If Clamping torque is reached before the position is reached then there will be no error displaced



Information

when activating the “Travel to fixed stop” mode than the following error will be disabled automatically.

The feedback can be read out on EPosZSW2 Bit14 or in ZSW2 Bit8. The PtP_Drives_Festo_EIP FB has the output interface EPosZSW1 and EPosZSW2. That means the feedback can be read-out easily at Bit14 on EPosZSW2.

Like on the picture above to see the Ptp_Drives_Festo_EIP does not offer an input for a torque setpoint. The torque setpoint is called Clamping Torque [Nm] and can be found at P1.526801.0.0 in Festo Automation Suite.

The max. allowed clamping torque is related to the “User defined limits” P1.852.0.0 for Torque: lower limit value and P1.853.0.0 for Torque: upper limit value in tab Axis, see below:

Axis																	
Homing Method Method Move to axis zero point after homing ☒ Active Activation Save zero offset ☐ Active																	
Axis Configuration Reversing the direction of rotation ☐ Active Axis zero point offset 3.00 mm Software limit positions active ☐ Active Negative software limit position -3.00 mm Positive software limit position 97.00 mm																	
User Defined Limits Velocity: lower limit value -16.66667 m/s Velocity: upper limit value 16.66667 m/s Torque: lower limit value -0.5472 Nm Torque: upper limit value 0.5472 Nm Active current: lower limit value -3.20 Arms Active current: upper limit value 3.20 Arms																	
Homing Parameters <table> <thead> <tr> <th></th> <th>Velocity [m/s]</th> <th>Accel. [m/s²]</th> <th>Jerk [m/s³]</th> </tr> </thead> <tbody> <tr> <td>Crawl</td> <td>0.005</td> <td>1.00</td> <td>100.00</td> </tr> <tr> <td>Search</td> <td>0.01</td> <td>1.00</td> <td>100.00</td> </tr> <tr> <td>Running</td> <td>0.05</td> <td>1.00</td> <td>100.00</td> </tr> </tbody> </table> Homing timeout 60.00 s			Velocity [m/s]	Accel. [m/s ²]	Jerk [m/s ³]	Crawl	0.005	1.00	100.00	Search	0.01	1.00	100.00	Running	0.05	1.00	100.00
	Velocity [m/s]	Accel. [m/s ²]	Jerk [m/s ³]														
Crawl	0.005	1.00	100.00														
Search	0.01	1.00	100.00														
Running	0.05	1.00	100.00														
Stop Deceleration Deceleration (stop ramp) 20.00 m/s² Jerk (stop ramp) 1000.00 m/s³																	

Cut-out from FestoAutomationSuite

To be able to change the clamping torque the EPD could be used to change the value cyclically.
(Refer [Chapter 7](#))

7 Extended Process Data

7.1 Configuring RJ71EIP91 using EtherNet/IP Configuration Tool (EIP – CT)



Note

- Based on earlier RJ71EIP91 setting follow either [Chapter 3.4 Steps 11](#) or [Chapter 3.5 Steps 5](#) for configuring for Extended Process Data in RJ71EIP91 module.

7.2 Set parameters in the FAS under Extended Process Data (EPD)



Information

Maximum of 32 Number of bytes can be called in Sent Data and Received Data

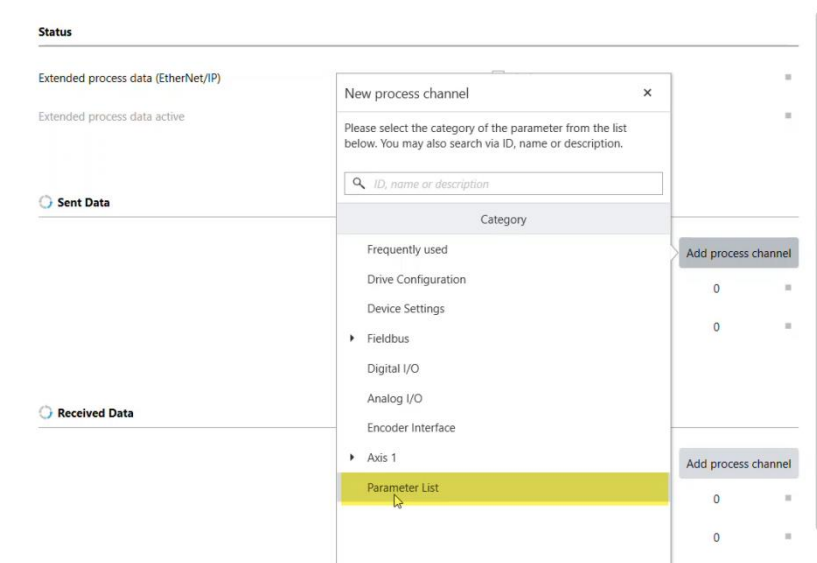
- Before adding the EPD to PLC, list out the Send and Receive Data (EPD) and byte position to modify the Read Holding Registers and Write Holding Registers
- For Instance: For Torque Control Mode, user has to add following EPD

Step – 1

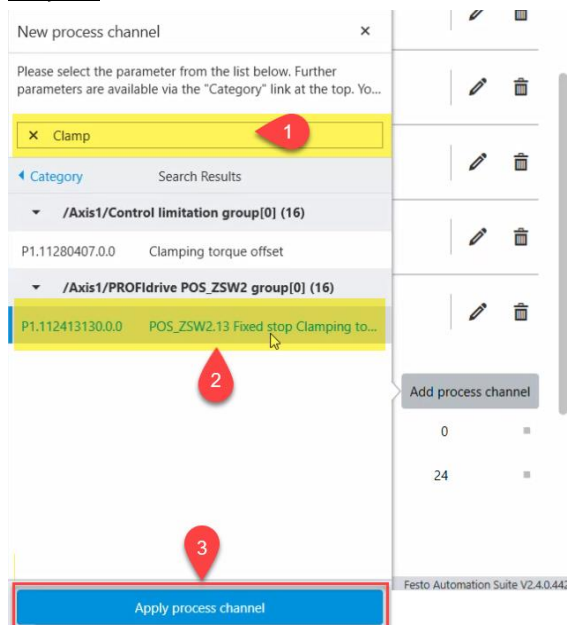
- In **Parameter pages** expand **PROFIdrive** under Profile tree.
- Click on the **Extended Process. Data.**
- Enable Extended Process Data (EtherNet/IP)
- Under the **Sent Data**, Click **Add process channel**
- Under the **Received Data**, Click **Add process channel**

Step – 2

Sent Data – This will send the data to the PLC.



- Select **Parameter List**.

Step – 3

1. Under the search option, either enter the **P1 number** or enter the **name of the parameter**.
2. Select the P1 / Parameter which needs to be added in EPD.
3. Click on Apply process channel or Double click on the parameter that will be added to the channel.

Extended Process Data :- Sent Data from FAS (Read Data)

ID	Parameter	Type	Byte Position
P1.2220.0.0	Setpoint value torque	REAL	0
P1.150.0.0	Actual torque value motor shaft	REAL	4
P1.526801.0.0	Clamping Torque	REAL	8

Extended Process Data :- Received Data to FAS from PLC (Receive Data)

ID	Parameter	Type	Byte Position
P1.526821.0.0	Clamping Torque	REAL	0

After adding the EPD.

Extended Process Data

Delete All Sent Data
Delete All Received Data

☒ Active
☐ Active

Extended process data (EtherNet/IP)

Extended process data active

Sent Data

	ID	Parameter	Type	Byte position	Function Block ID
0	P1.2220.0.0	Setpoint value torque	REAL	0	9
1	P1.150.0.0	Actual torque value motor shaft	REAL	4	9
2	P1.526801.0.0	Clamping torque	REAL	8	9

Add Process Channel

Number of bytes Tx: 0
Number of bytes Tx (Offline): 12
PLC output configuration: 999

Received Data

	ID	Parameter	Type	Byte position	Function Block ID
0	P1.526801.0.0	Clamping torque	REAL	0	9

Add Process Channel

Number of bytes Rx: 0
Number of bytes Rx (Offline): 4
PLC input configuration: 9

**Note**

- The data type and the byte position to be noted to read and write in the PLC program.

Step – 4

Store on Device



Reinitialize

- Click the **“Store on device”** button to store all the parameters on drive memory.
- Click the **“Reinitialize”** the button to initialize the drive.

**Note**

- The triangle warning symbol on both buttons indicates action required.
- All the modified parameters will automatically erase from drive during power restart If modified parameters are not stored in the drive.
- Its mandatory to reinitialize the drive for some parameter modification. If drive is not reinitialized the drive performance remains same as like previous configuration.

7.3 Add the EPD in PLC Logic



Information

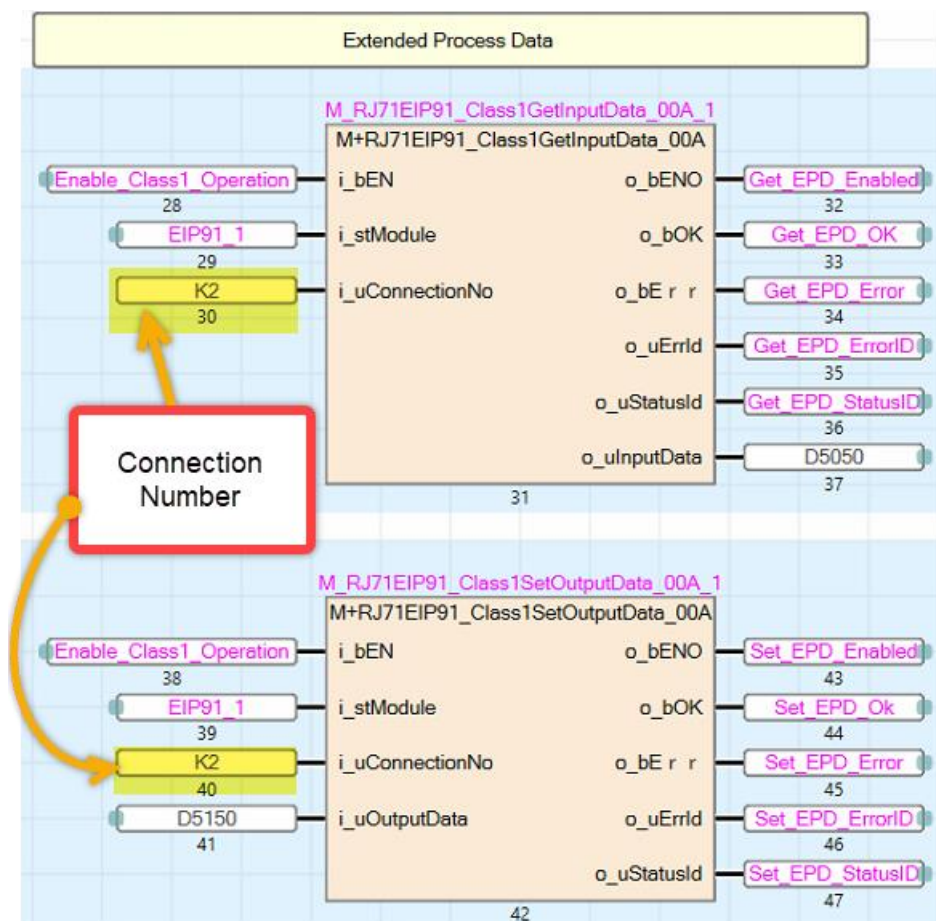
Maximum of 32 Number of bytes can be called in Sent Data and Received Data



Note

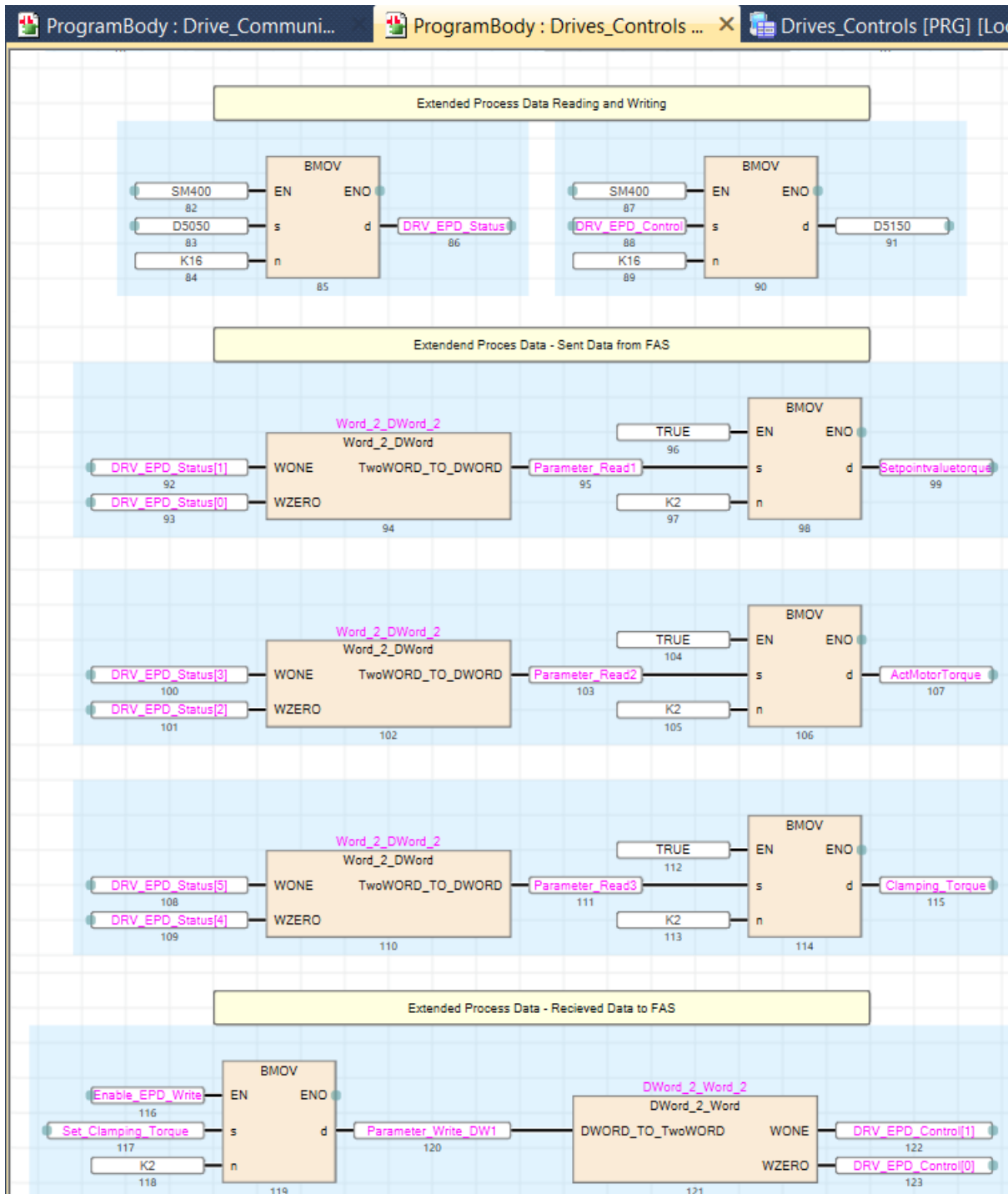
- For PTP_Drives_Festo_EIP FB (Position Mode) 12 Word Data Read and Data Write is used.
- To Read or Write data of EPD from user has use ReadData[12] & WriteData[12] onwards.
- If Data is Float use the instruction as shown in [chapter 7.3.2](#).
- If Data is Boolean user need to extract the data from the respective byte position.
- Maximum ReadData[27] & WriteData[27] can be used which includes Process Data and EPD.

7.3.1 Configure the Extended Process Data - Exclusive Owner in PLC Logic by calling “M+RJ71EIP91_Class1GetInputData_00A and M+RJ71EIP91_Class1SetOutputData_00A” Input & Output Interface Tags (Refer [Chapter 4.1.1](#) for configuring M_FBLib – RJ71EIP91)

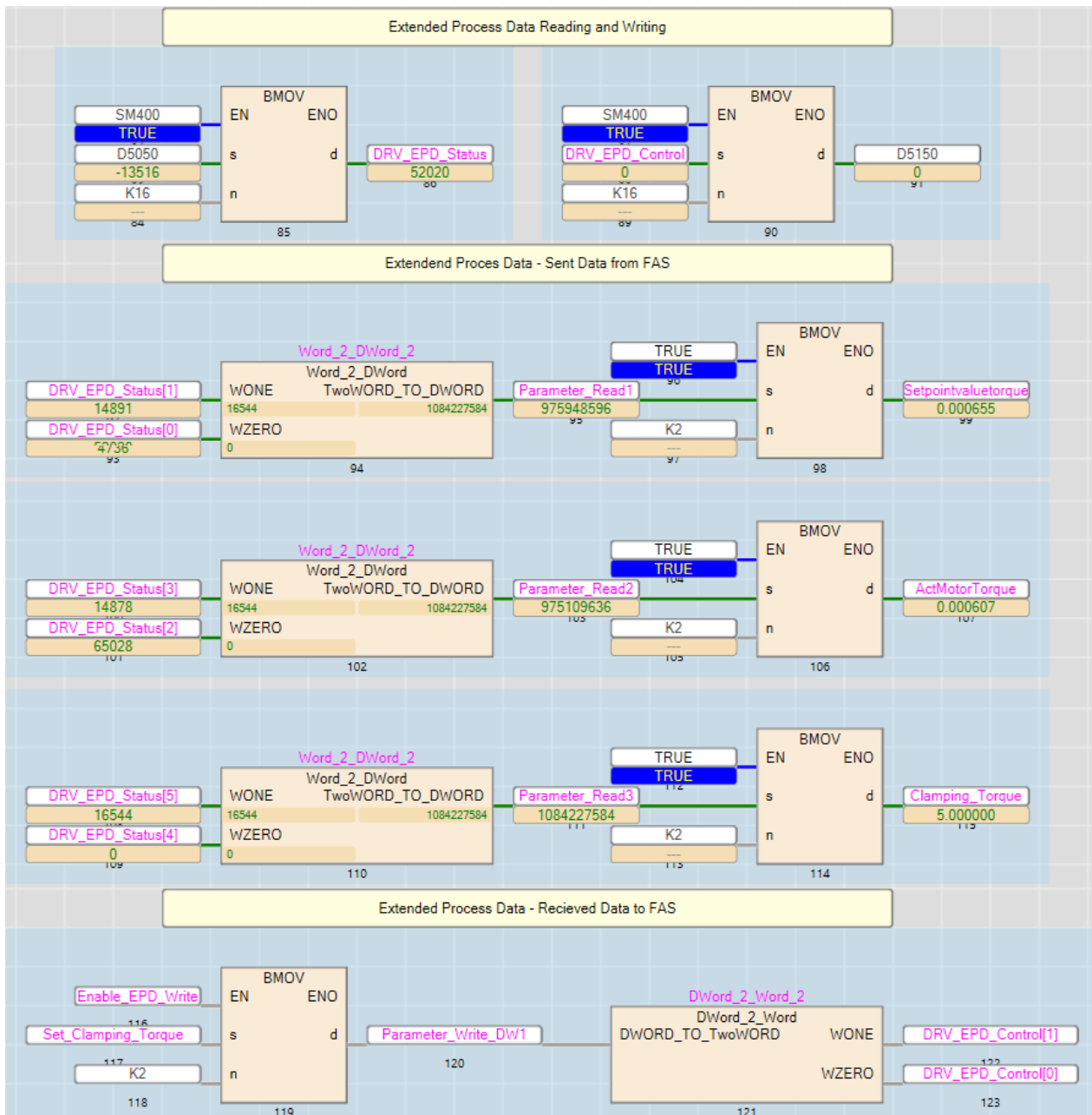


7.3.2 Logic to Read and Write the EPD

- Based on the Number of EPD & the Data type, write the logic as shown in below image.



- Read EPD value
 - Use the Word_2_DWord Function Block to if the EPD is 4 Byte, so that value can be read in DOUBLE WORD.
 - Use BMOV Function Element to convert DWord to Float value. As parameter which is read is FLOAT.
- Write EPD value
 - Use BMOV Function Element to convert Float value to DWord.
 - Use the DWord_2_Word Function Block to convert DWord to Word, so that value can be written in WORD.



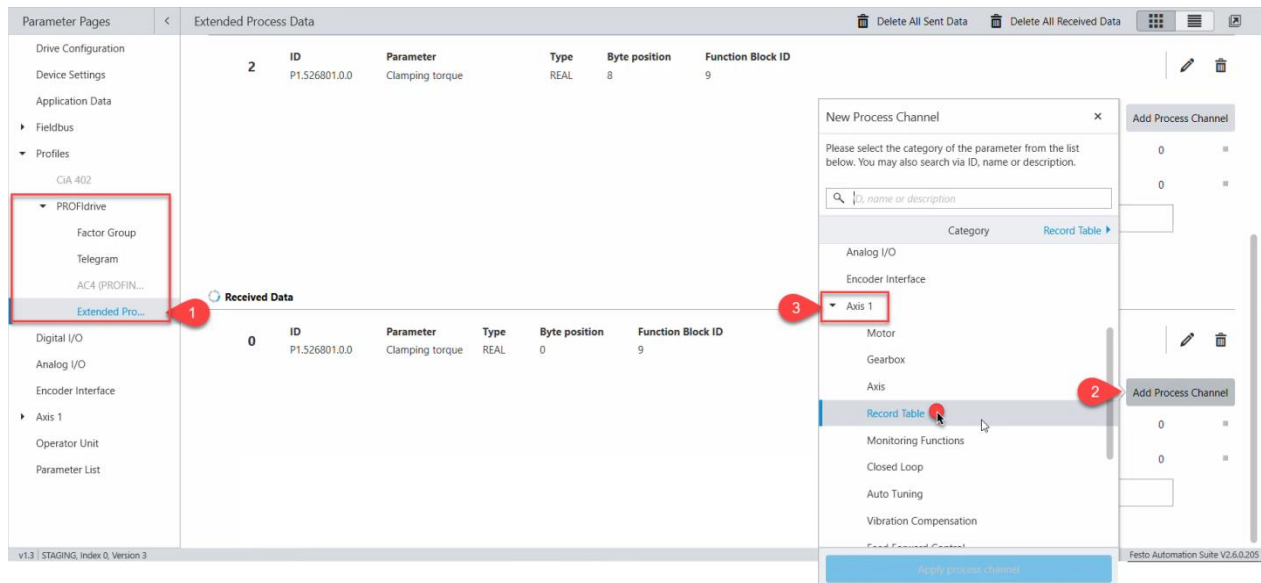
7.3.3 Changing the Record Table Value from EPD



Caution

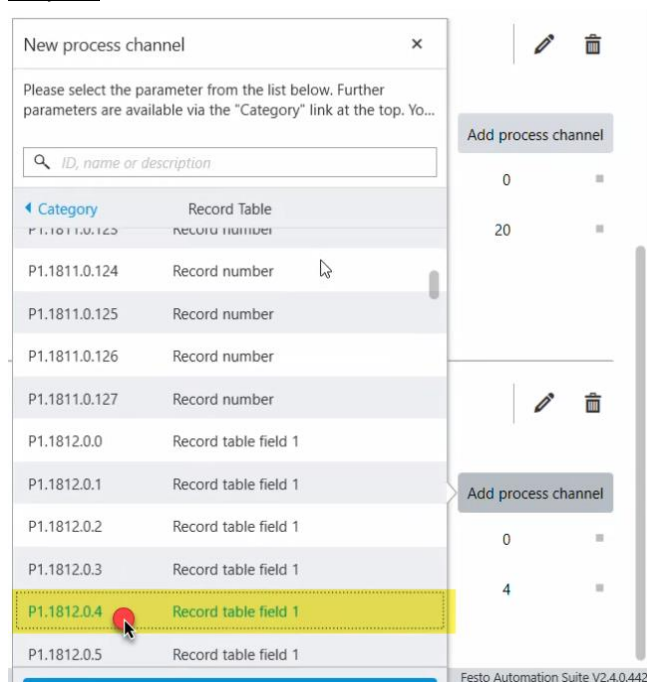
1. Do not create a Record Table from PLC through EPD, this may cause error in connection between CMMT and PLC.
2. If changing the value of Record Table from PLC through EPD, make sure to latch the default value of respective parameter in PLC, else at every power cycle of PLC the value of parameter will be zero. (Use Retain Memory for writing the parameter value).
3. Not all parameters of Record Table Field can be changed through EPD, and also read in PLC through EPD. Max of 7 Record Table Field can be called in EPD.

Step – 1



1. In **Parameter pages** expand **PROFdrive** under Profile tree and click on the **Extended Process. Data**.
2. Under the **Received Data**, Click **Add process channel**
3. Under the **Axis 1**, Click on **Record Table**

Step – 2



- Select the right parameter to change the Record Table value. (Refer below image)

Record Table

Delete All

1

Homing

Record type
Homing (3)

Method
Current position (37)

2

Position_Mode_1

Record type
Position (5)

Type
Positioning relative to current actual position (1)

Target position
1000.00 mm

Velocity
1.00 m/s

Acceleration
5.00 m/s²

Deceleration
5.00 m/s²

Jerk
150.00 m/s³

End ve
0.00 m/s

3

Position_Mode_2

Record type
Position (5)

Type
Positioning relative to current actual position (1)

Target position
2000.00 mm

Velocity
10.00 m/s

Acceleration
5.00 m/s²

Deceleration
5.00 m/s²

Jerk
150.00 m/s³

End v
0.00 m/s

4

Torque_1

Record type
Torque (7)

Target torque
2.00 Nm

Torque rise ramp
1.00 Nm/s

Limit speed
0.40 m/s

Activation stroke limit
☒

Stroke limit negative
-2500.00 mm

Stroke limit positive
2500.00 mm

5

Untitled

Record type
Torque (7)

Target torque
4.20 Nm

Torque rise ramp
1.00 Nm/s

Limit speed
0.40 m/s

Activation stroke limit
☒

Stroke limit negative
-50.00 mm

Stroke limit positive
50.00 mm

P1.1811.0.4

Record Number

P1.1810.0.4

Command Record Type

P1.1812.0.4

Record Table Field 1

P1.1813.0.4

Record Table Field 2

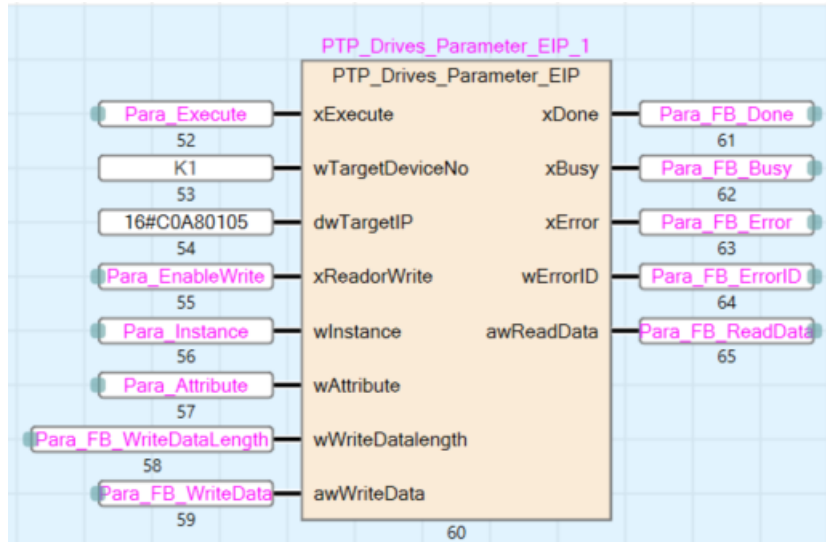
<

- User needs to note which record table value needs to change from PLC through EPD.
- The last digit of parameter refers to the Records Table in the list. (Note – Last digit starts from 0 hence user needs to count the record table from 0 onwards).

8 PTP_Drives_Parameter_EIP FUNCTIONS

The PTP_Drives_Parameter_EIP Function Block is used for single parameter read and write. In this chapter we explain with some examples on how to read and write CMMT parameters using this Function Block.

Call the **PTP_Drives_Parameter_EIP** Function Block.



8.1 Read Parameter

1. Configuration for Reading Parameter

- xReaderWrite input tag must be FALSE.
- Enter parameter number in **"xInstance"**, parameter Index in **"xAttribute"** (Refer CMMT Manual - Chapter 13.5 PNUs reference list)

2. Sequence Command

- Toggle the **"xExecute"** command tag (Rising Edge) to start reading parameter (Every new task would require a toggle of the **"xExecute"** command tag).
- The **"xBusy"** status tag will be true to acknowledge the task has been started.
- If the task is successfully completed, it is acknowledge by rising edge of **"xDone"** status variable. Also if task is unsuccessful **"xError"** output tag will be TRUE and **"wErrorID"** shows the error code.
- **awReadData** is the variable which holds the Read data, here UCMM_Data is a ARRAY of Word.
 - If Read Data is DWord or Float then call the **Word_2_DWord** Function Block



Note

- For detailed information on the working of awWriteData pointed input, awReadData pointed output, Refer [Chapter 5.2](#).

8.1.1 Example : Read parameter of the data type FLOAT.

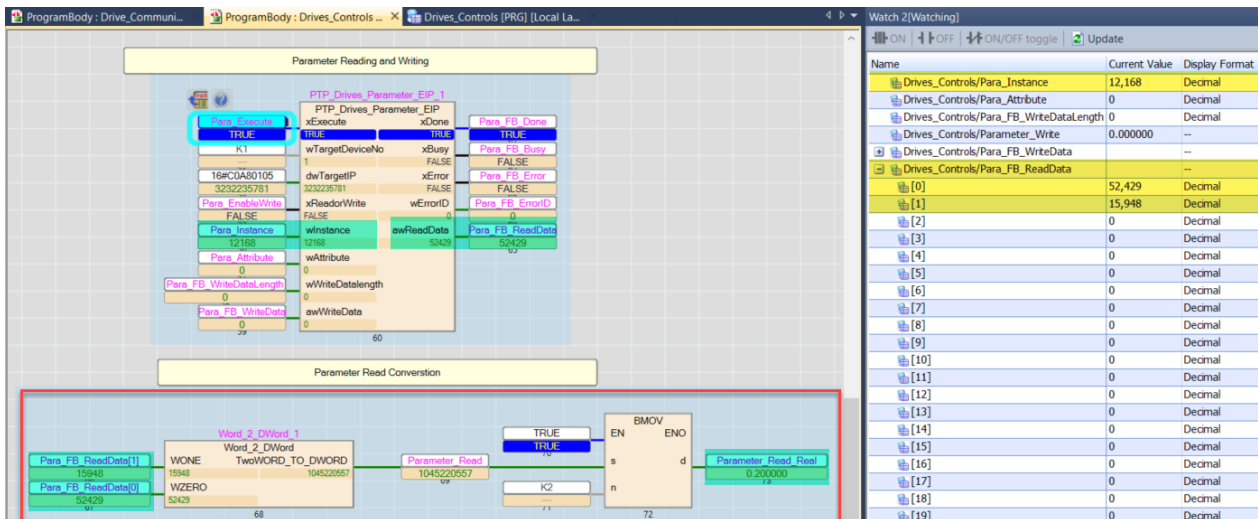
Consider we want to read/write the parameter named **"Clamping Torque"**.

PNU	Name	Data type	Access	Parameter
12168.0	Clamping torque	REAL	rw	P1.526801.0.0

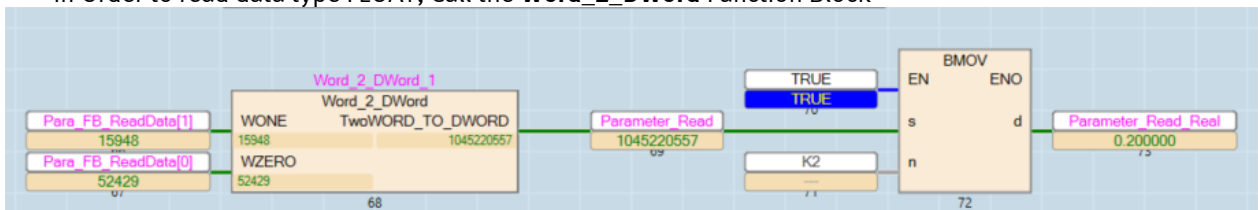
Parameter Index

Parameter Number

Data Type



In Order to read data type FLOAT, Call the **Word_2_DWord** Function Block



- Enter the Input Array of Word in **Word_2_DWord** and get Double Word Value Output.
- Enter the same Double Word Value Output in BMOV block and get Real value Output.

8.2 Write Parameter

1. Configuration for Reading Parameter

- xReadorWrite input tag must be TRUE.
- Enter parameter number in "**xInstance**", parameter Index in "**xAttribute**" (Refer CMMT Manual - Chapter 13.5 PNUs reference list)
- Enter the data length in "**wWriteDataLength**". (Example: 4 bytes (Since the data to be write is FLOAT). Refer Data length Table given below.
- Enter the parameter value in "**awWriteData**" tag, is the variable which holds the Write data, here UCMM_Data is a ARRAY of Word.
 - If Data is DWord or Float then call the **DWord_2_Word** Function Block

2. Sequence Command

- Toggle the "**xExecute**" command tag (Rising Edge) to start reading parameter (Every new task would require a toggle of the "**xExecute**" command tag).
- The "**xBusy**" status tag will be true to acknowledge the task has been started.
- If the task is successfully completed, it is acknowledge by rising edge of "**xDone**" status variable. Also if task is unsuccessful "**xError**" output tag will be TRUE and "**wErrorID**" shows the error code.
- awWriteData is the variable which holds the Write data, here UCMM_Data is a ARRAY of Word.
 - If write Data is DWord or Float then Call the **Word_2_DWord** Function Block



Note

- For detailed information on the working of awWriteData pointed input, awReadData pointed output, Refer [Chapter 5.2](#).

Data Length Table.

Data Type Rockwell	Data Type Festo Automation Suite	Size
Bit	Bool	1 Byte
Word [Signed]	SINT16	2 Byte
Double Word [Signed]	SINT32	4 Byte
Word [Unsigned]/Bit String [16-bit]	UINT16	2 Byte
Double Word [Unsigned]/Bit String [32-bit]	SINT32	4 Byte
FLOAT [Single Precision]	FLOAT32	4 Byte

8.2.1 Example : Write parameter of the data type FLOAT.

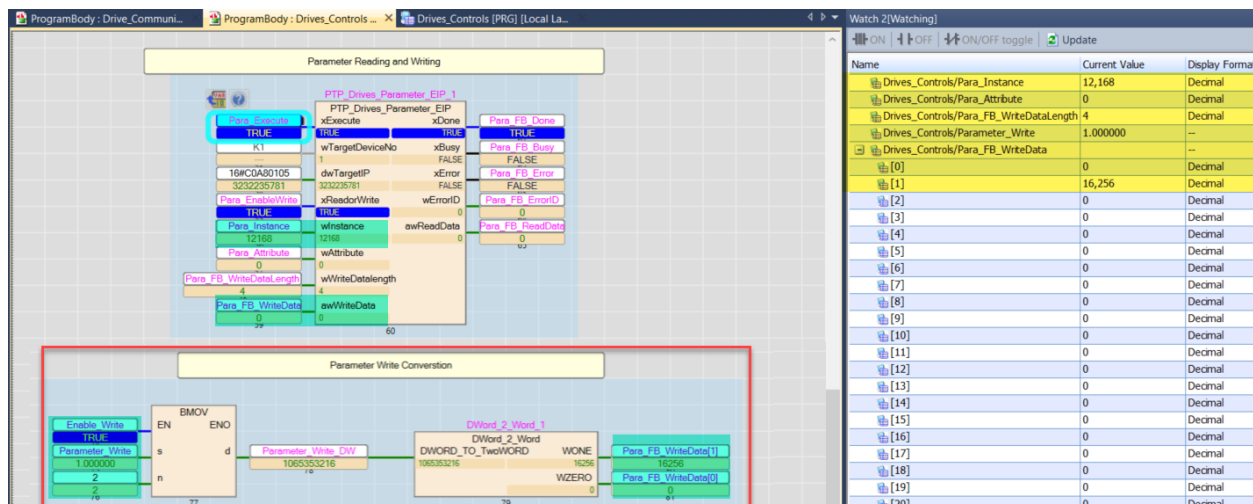
Consider we want to read/write the parameter named “Clamping Torque”.

PNU	Name	Data type	Access	Parameter
12168.0	Clamping torque	REAL	rw	P1.526801.0.0

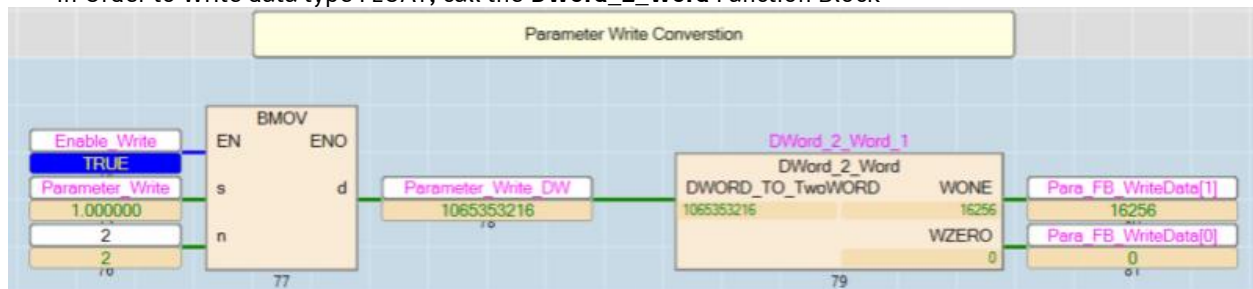
Parameter Index

Parameter Number

Data Type



In Order to Write data type FLOAT, call the **DWord_2_Word** Function Block



- Enter the Input Real Value Output in BMOV block and get Double Word Value Output.
- Enter the DWord Output value in **DWord_2_Word** and get Array of Word Value Output.

9 CMMT Drive in Velocity Control Mode Functions

The velocity control mode of CMMT drive has been designed with two different FBs. They are named as “Tel001_Velocity_Drives_Festo_EIP” & “Tel102_Velocity_Drives_Festo_EIP”. The “Tel001_Velocity_Drives_Festo_EIP” FB controls only speed of a servo motor with reference to its given speed setpoint but in “Tel102_Velocity_Drives_Festo_EIP” taking advantage by controlling both speed and torque of a servo motor.



Note

- Refer [Chapter - 1.1](#) for Hardware & Software Requirements, [Chapter – 1.2](#) for Communication Architecture & [Chapter – 1.3](#) for CMMT Drive Hardware Overview.

9.1 Hardware Configuration for Velocity Control Mode



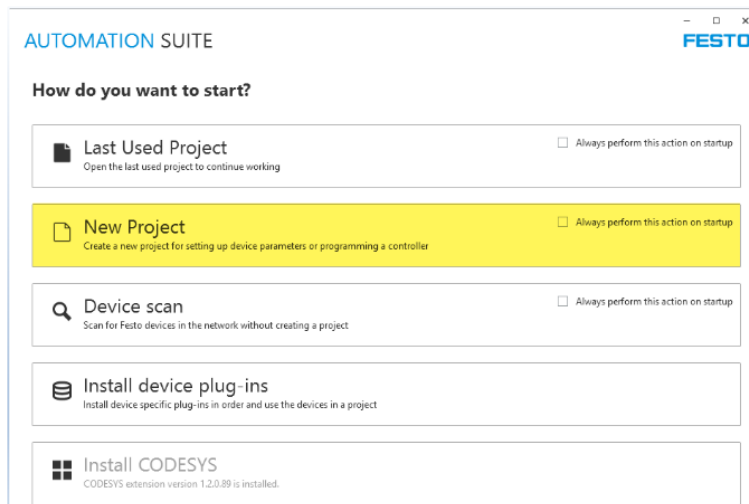
Note

- Refer entire [Chapter – 2](#) and [Chapter – 3](#) for IP address configuration of CMMT servo drive & Mitsubishi PLC.
- This chapter Hardware configuration procedures are similar for both Basic & Advance Velocity control modes.

9.1.1 CMMT Servo Drive Configuration for Velocity Control Mode

This chapter describe about parameterization of CMMT drive for rotary axis in set RPM & current position as homing mode.

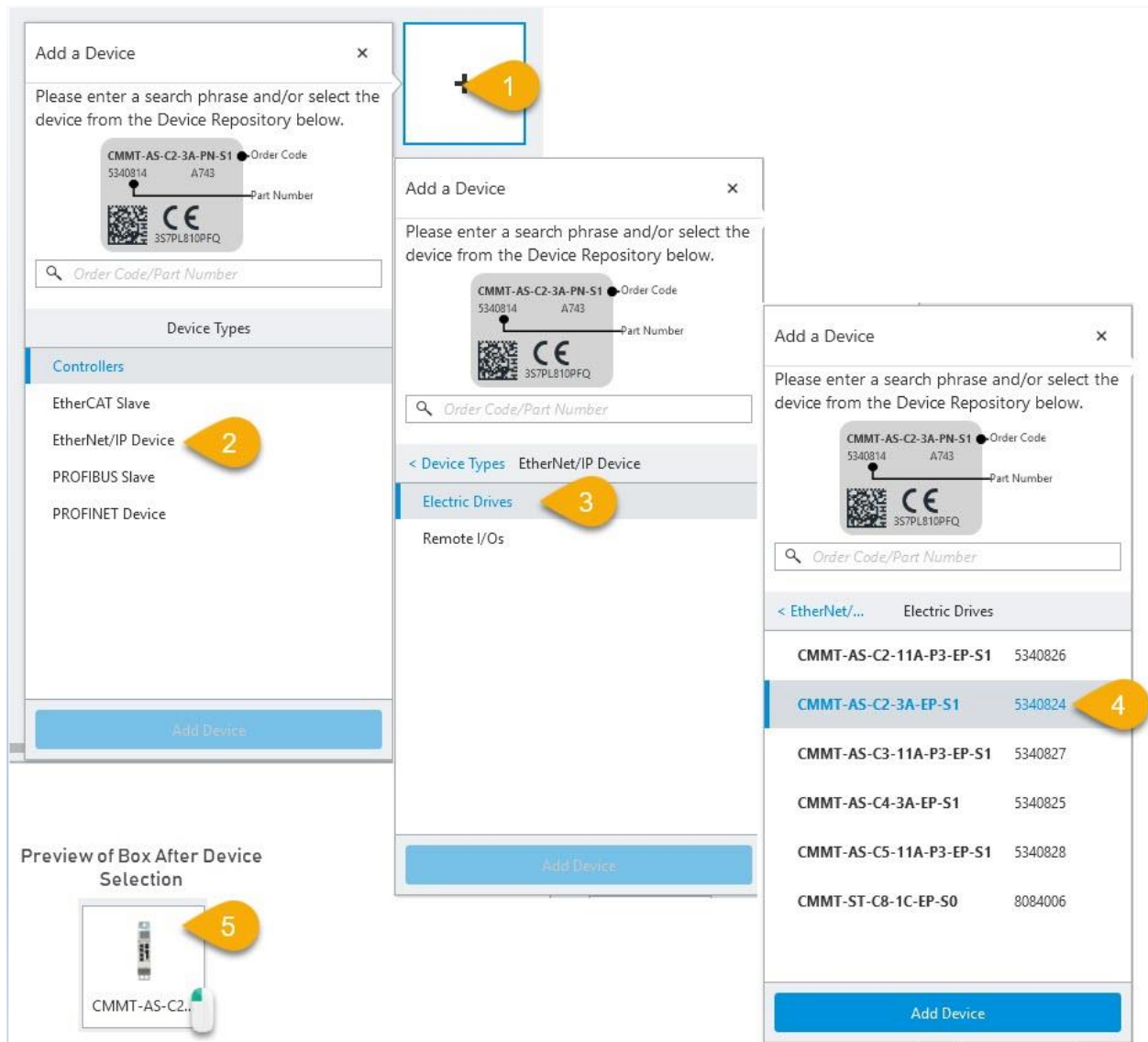
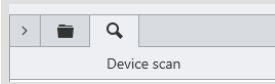
Step – 1



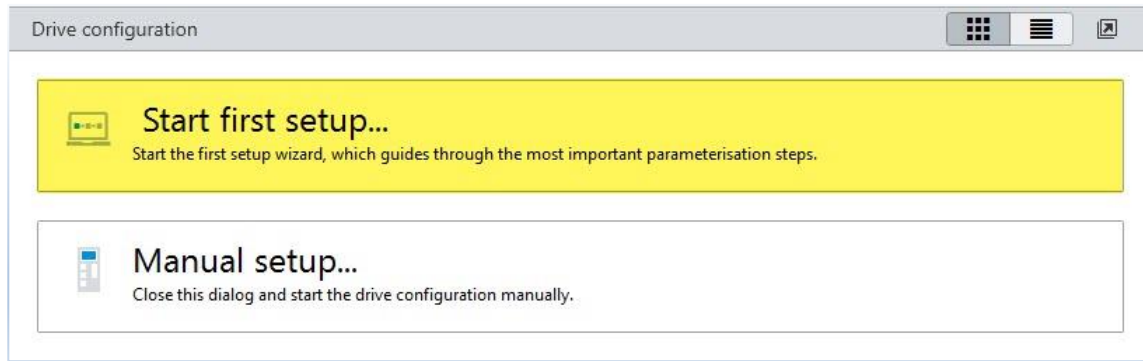
Open the Automation Suite software from Windows **Start > Festo Software > AUTOMATION SUITE** and select “**New Project**” from start-up menu.

Step – 2**Note**

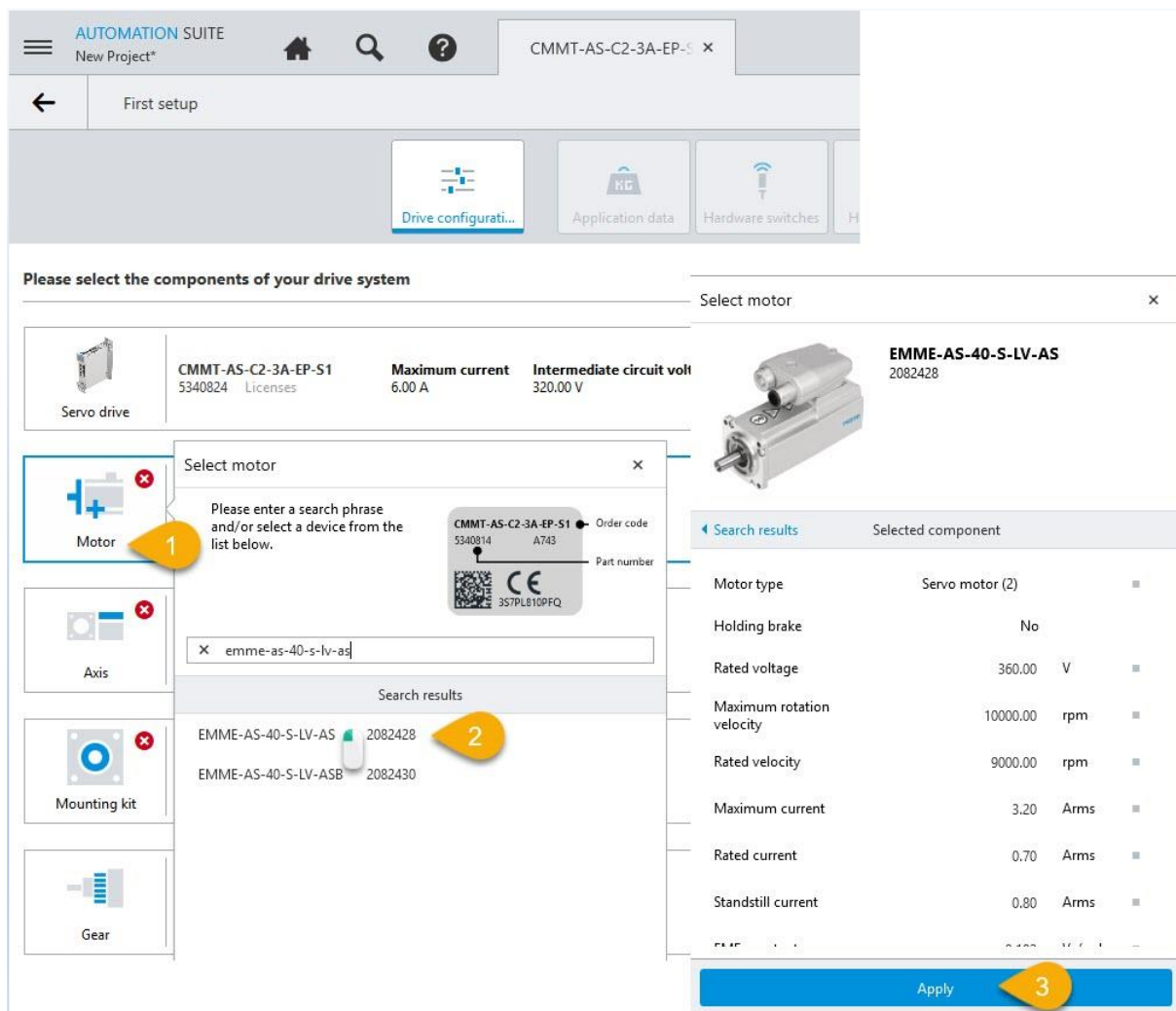
- Follow this step if drive is not in online. If the drive is connected with the PC as shown in [Chapter 2.1.2](#), simply click Device scan from home (Next to device Repository) and online devices will be visible in available devices list. Select the correct device based on Its configured name & IP address then double click.



- Select any empty device box and click add button.
- Select **"EtherNet/IP Device"** from Add a Device menu.
- Select **"Electric Drives"** from device type menu.
- Select correct drive part number** from electric drives list . As per this sample application CMMT-AS-C2-3A-EP-S1 is selected.
- Double click the newly added drive to initialize configuration parameters.

Step – 3

Select **“Start first setup”** from drive configuration menu.

Step – 4

1. Click **“Motor”** from components of drive system menu for specific motor model.
2. **Select correct motor** based on its **part number** and double click. As per this sample application “EMME-AS-40-S-LV-AS” motor is used.
3. Cross verify the motor parameters & click apply button to ensure motor configuration.

View of motor tab after configuration

	EMME-AS-40-S-LV-AS 2082428	Type Servo motor (2)	Holding brake No	Encoder protocol HiPerface (0)	Encoder type Single turn (1)	Voltage 360.00 V	 
---	-------------------------------	-------------------------	---------------------	-----------------------------------	---------------------------------	---------------------	---

Step – 5

Please select the components of your drive system



Servo drive

CMMT-AS-C2-3A-EP-S1
5340824 Licenses

Maximum current
6.00 A

Intermediate circuit v
320.00 V



Motor

EMME-AS-40-S-LV-AS
2082428

Type
Servo motor (2)

Holding brake
No

Encoder
HiPer



Axis

Please enter a search phrase and/or select a device from the list below.

Order code
CMMT-AS-C2-3A-EP-S1
5340814 A743

Part number
3S7PL810PFQ

CE
3S7PL810PFQ



Mounting kit

Search results

User defined linear axis

User defined rotative axis

DGE-8-%-ZR 193739



This step describes about the axis type configuration. As specified initially here axis configuration is done for rotate the axis in velocity mode for set RPM.



Note

- Make sure the “**Unlimited axis**” is activated as highlighted in yellow color. This ensures the axis does not have position limits.

Step – 6

Select mounting kit

Please enter a search phrase and/or select a device from the list below.

CMMT-AS-C2-3A-EP-S1
5340814 A743
357PL810PFQ

Order code
Part number

Search results

User defined mounting kit

EAMM-A-A19-40A	559798
EAMM-A-A19-42A	558895
EAMM-A-A22-55A	559799
EAMM-A-A22-57A	558897
EAMM-A-A22-70A	558898
EAMM-A-D100-100A	3356796
EAMM-A-D100-100A-S1	3356931

Gear ratio (total): 1:1

Select mounting kit

User defined mounting kit

Search results

Selected component

Type

Inertia Coupling

Axial

10 kgm²

Apply

Back Next Finish

New Project - CMMT-AS-C2-3A-EP-S1 (Plug-in: CMMT-AS Plug-in V1.2.2.4) Festo Automation Suite V1.2.1.1

This step illustrates about mounting kit configuration. This configuration will vary depending on user application type. At the end of this step click next button to move to next level configuration.

Step – 7

Application data

Axis mass moment of inertia

0.00 kgcm²

Application moment of inertia

1.00 kgcm²

Total mass moment of inertia

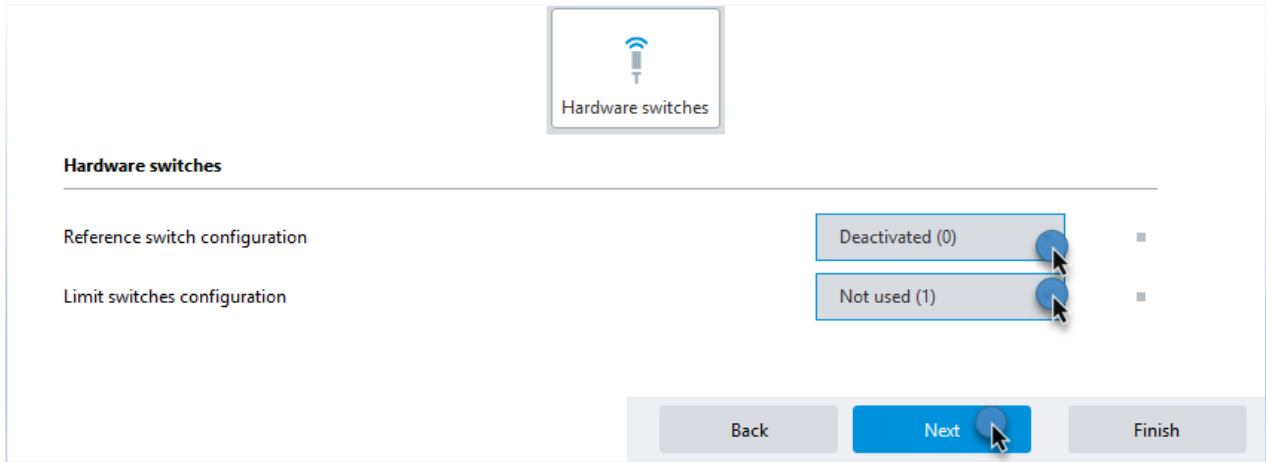
1.00 kgcm²

Back Next Finish

New Project - CMMT-AS-C2-3A-EP-S1 (Plug-in: CMMT-AS Plug-in V1.2.2.4) Festo Automation Suite V1.2.1.1

Specify inertia value of user application. This parameter value will vary depend on application type. Click next button for next level configuration.

Step -8



Hardware switches

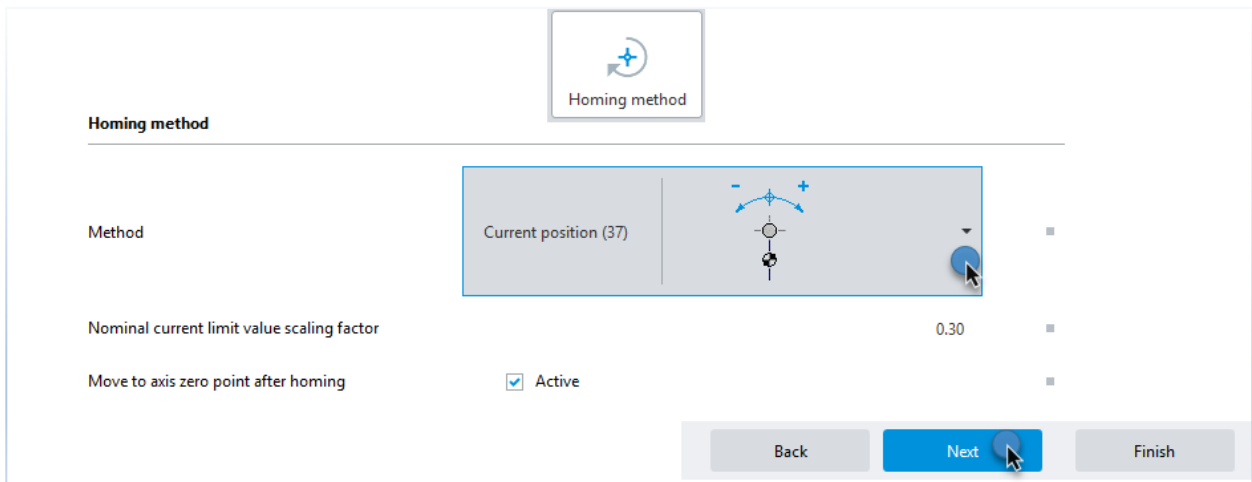
Reference switch configuration: Deactivated (0)

Limit switches configuration: Not used (1)

Back Next Finish

Select the reference & limit switch configuration types. This sample application has both disabled. This is the reason “Current Position (37)” homing mode selected for this sample application. Click next button for next level configuration.

Step – 9



Homing method

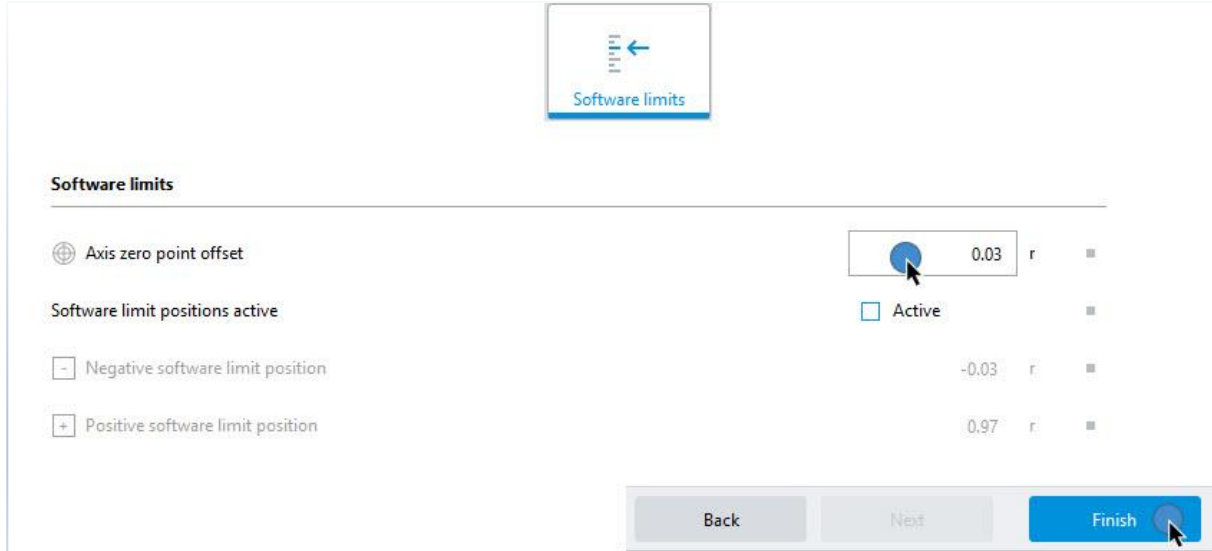
Method: Current position (37)

Nominal current limit value scaling factor: 0.30

Move to axis zero point after homing: ☒ Active

Back Next Finish

Select homing method. As per this sample application “**Current Position (37)**” mode is selected. But user want to select homing mode depending on application requirement. Click next button for next level of configuration.

Step – 10


Software limits

Axis zero point offset 0.03 r ■

Software limit positions active ☐ Active ■

☐ Negative software limit position -0.03 r ■

☐ Positive software limit position 0.97 r ■

Back Next Finish

Here user can select offset position of home reference point. This parameter is not valid for velocity mode, so simply leave default mode & click **“Finish”** button to complete the configuration.

**Note**

- Don't activate the “Software Limit Positions Active” option.

Step – 11

At this stage all the fundamental configurations of drive system are done. In case of any correction in parameter is required software tool will indicate number of **“Correct parameters”** (Right most top corner in FAS tool) as shown in the below picture. So to correct parameter value click the **“Correct parameters”** button and click apply button to apply system generated recommended values.



9.1.2 CMMT Drive Parameterisation for Integrate with PLC in Velocity Control Mode

This chapter will describe about parameterisation of CMMT drive to establish communication with PLC through Telegram-1 & Telegram-102. Also it describes some additional parameters like Factor group, Reference values, Ethernet/IP interface.



Note

- This chapter explains parameterisation for both Basic Speed Control & Advance Speed Control modes .

Step – 1



Note

- Follow **step – 1** when drive is available for online connection. If drive is not available simply skip this step and jump to next step.

The screenshot shows the 'PARAMETERISATION' tab of the software. On the left, a drive named 'CMMT1' (CMMT-AS-C2-3A-EP-S1) is shown with a 'Path: 192.150.3.7' and a 'Disconnected' status. A 'Connect' button is visible. Below this, a 'Status When Drive is in On-line' section shows the drive as 'Connected' with a 'Disconnect' button. On the right, a 'Device Communication' window is open, displaying a table of 'Available Devices'. The table has columns for Status, Identify, Device Name, Device Type, and IP Address. One device is listed: CMMT1 (CMMT-AS) with IP address 192.168.1.7. A blue circle with the number '2' is placed over the 'Identify' column of this device. At the bottom of the window, a blue button labeled '2 Use IP Address for Device Communication' is highlighted.

Status	Identify	Device Name	Device Type	IP Address
⊗	☐	CMMT1	CMMT-AS	192.168.1.7

- Click the IP address “**Path**” from tool bar.
- Select the correct drive based on IP address & select “**Use IP Address for Device Communication**” button to assign path address.
- Click on “**Connect**” button to establish communication & status of the drive changes from Disconnected to Connected.

Step – 2

Factor group		
Current user unit	Rev [rev, rpm, ...] (3)	■
Position	-6	■
Velocity	-3	■

Enter the Velocity command scale factor value of fieldbus as highlighted in yellow color field. Here scale factor is set for 1000. User can reach the Factor group in Automation Suite from **PARAMETERISATION > Fieldbus > Factor group**.

Example - 1 PLC scale value is calculated below for 1 RPM speed.

Velocity Factor (-3) = $10^{-3} = 0.001 \text{ RPM}$ Then for 1 RPM = 1000 Increment value in PLC

Example - 2 PLC scale value is calculated below for 100 RPM speed.

Velocity Factor (-3) = $10^{-3} = 0.001 \text{ RPM}$ Then for 100 RPM = 100000 Increment value in PLC

**Note**

- For more details about fieldbus scaling refer chapter name “**Scaling of internal units for the fieldbus**” in user manual.
- Different user unit can't be set for Velocity, Acceleration & Jerk.
- The maximum value limit for 32 bit DINT data type variable is 2,147,483,647. So, plan scaling factor within the limit.

Step – 3**Dynamic Values**

Coast stop

Acceleration	3000.00	r/min/s
Deceleration	3000.00	r/min/s
Jerk	300000.00	rpm/s ²
Deceleration (system stop AC1/AC3)	600.00	r/min/s
Jerk (system stop AC1/AC3)	600000.00	rpm/s ²
Deceleration (stop ramp)	12000.00	r/min/s
Jerk (stop ramp)	600000.00	rpm/s ²
Base value acceleration	600.00	r/min/s
Base value deceleration	600.00	r/min/s
Base value velocity	3000.00	r/min
Base value velocity (PLC, motorside)	3000.00	r/min

Enter the actual Base value velocity, Acceleration & Deceleration values of the application in the respective numeric fields. The base value velocity specifies the base value for all Application Classes. The base value in user units (drive side without gearbox) is multiplied by the normalized value in the process data and then results in the internal velocity setpoint.

**Information**

- As a recommendation, the base value should be half of the maximum velocity in order to use the full value range 0..200% from the control.
- Base value velocity does not affect the speed of the drive, but it will effect on display value of the drive.
- The FB tag **“BaseValueVelocity”** tag must have same value as the FAS parameter value that is (FAS = P1.11280701.0.0).

Step – 4

EtherNet/IP interface

Configuration

Activate DHCP	☐ Active	■
IP address	192 . 168 . 1 . 5	■
Subnet mask	255 . 255 . 255 . 0	■
Gateway address	0 . 0 . 0 . 0	■

Enter the IP address & Subnet mask for EtherNet/IP interface. This communication interface is used to establish connection with PLC to receive control command & to send signal feedbacks.

**Note**

- IP address series of EtherNet/IP interface must match with PLC IP address series to establish connection.

Step – 5

Telegram - EtherNet/IP - ModbusTCP		Tel001_Velocity_Drives_Festo_EIP
Telegram selection	Telegram (1)	
Actual application class	Application class 1 (1)	
Telegram - EtherNet/IP - ModbusTCP		Tel102_Velocity_Drives_Festo_EIP
Telegram selection	Telegram (102)	
Actual application class	Application class 1 (1)	

Select the connection properties as “**Telegram(1)**” for “Tel001_Velocity_Drives_Festo_EIP” FB as highlighted in yellow color & select “**Telegram (102)**” for “Tel102_Velocity_Drives_Festo_EIP” FB as highlighted in red color. User can reach Connection properties from **PARAMETERISATION > Fieldbus > Connection properties**.

Step – 6

Enable servo drive

Activation via

- I/O and fieldbus (0)
- I/O and fieldbus (0)
- I/O (1)
- Fieldbus (2)
- I/O and Plug-in (3)
- Plug-in (4)

Select the medium to enable the servo drive. The selection must be either “**I/O and fieldbus (0)**” or “**Fieldbus**” to control the drive through fieldbus.

Step – 7

Store on device

1

Load factory settings

Reinitialize

2

Restart device

1. Click the “**Store on device**” button to store all the parameters on drive memory.
2. Click the “**Reinitialize**” the button to initialize the drive.

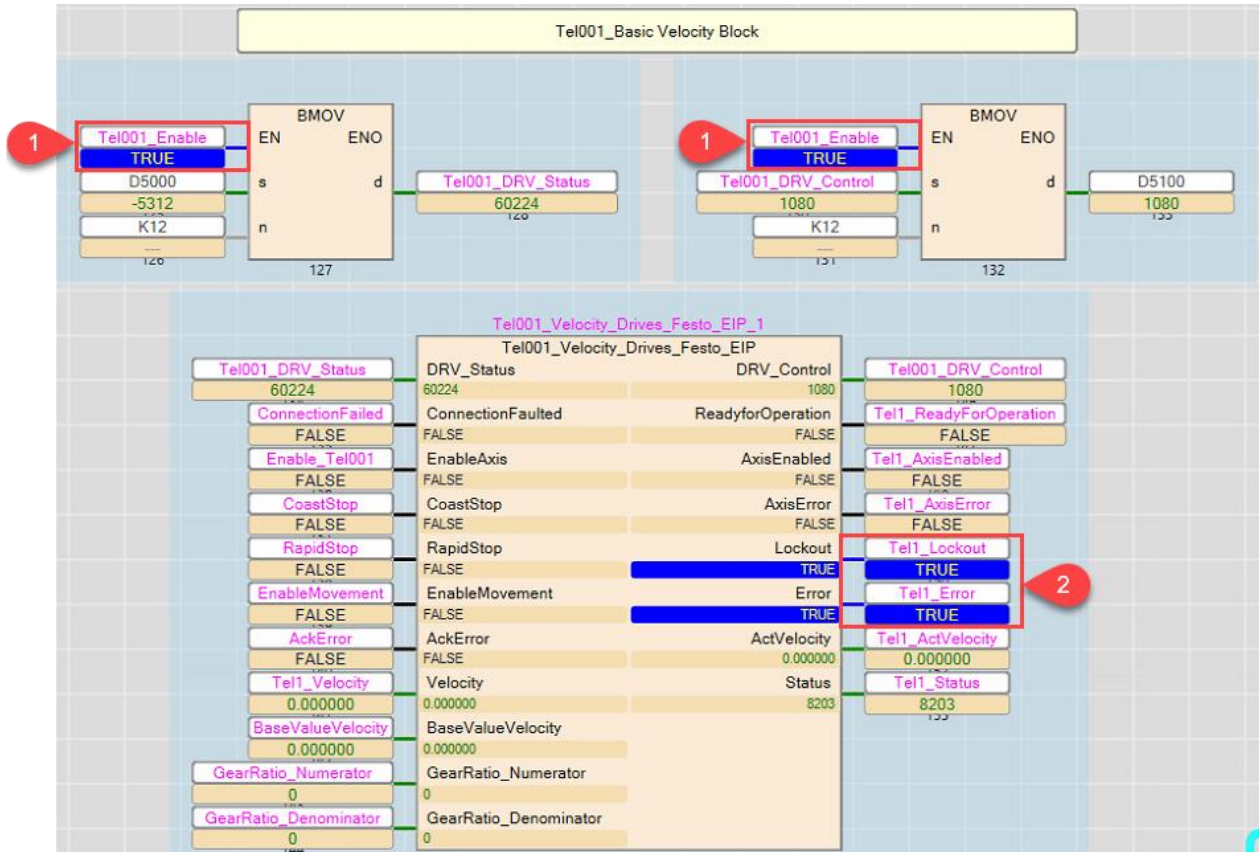
**Note**

- The triangle warning symbol on both buttons indicates action required.
- All the modified parameters will automatically erase from drive during power restart If modified parameters are not stored in the drive.
- Its mandatory to reinitialize the drive for some parameter modification. If drive is not reinitialized the drive performance remains same as like previous configuration.

9.2 Quick Procedure to Run the CMMT Drive in Velocity Control

9.2.1 Tel001_Velocity_Drives_Festo_EIP Functions

Step – 1: FB View at First Time Online



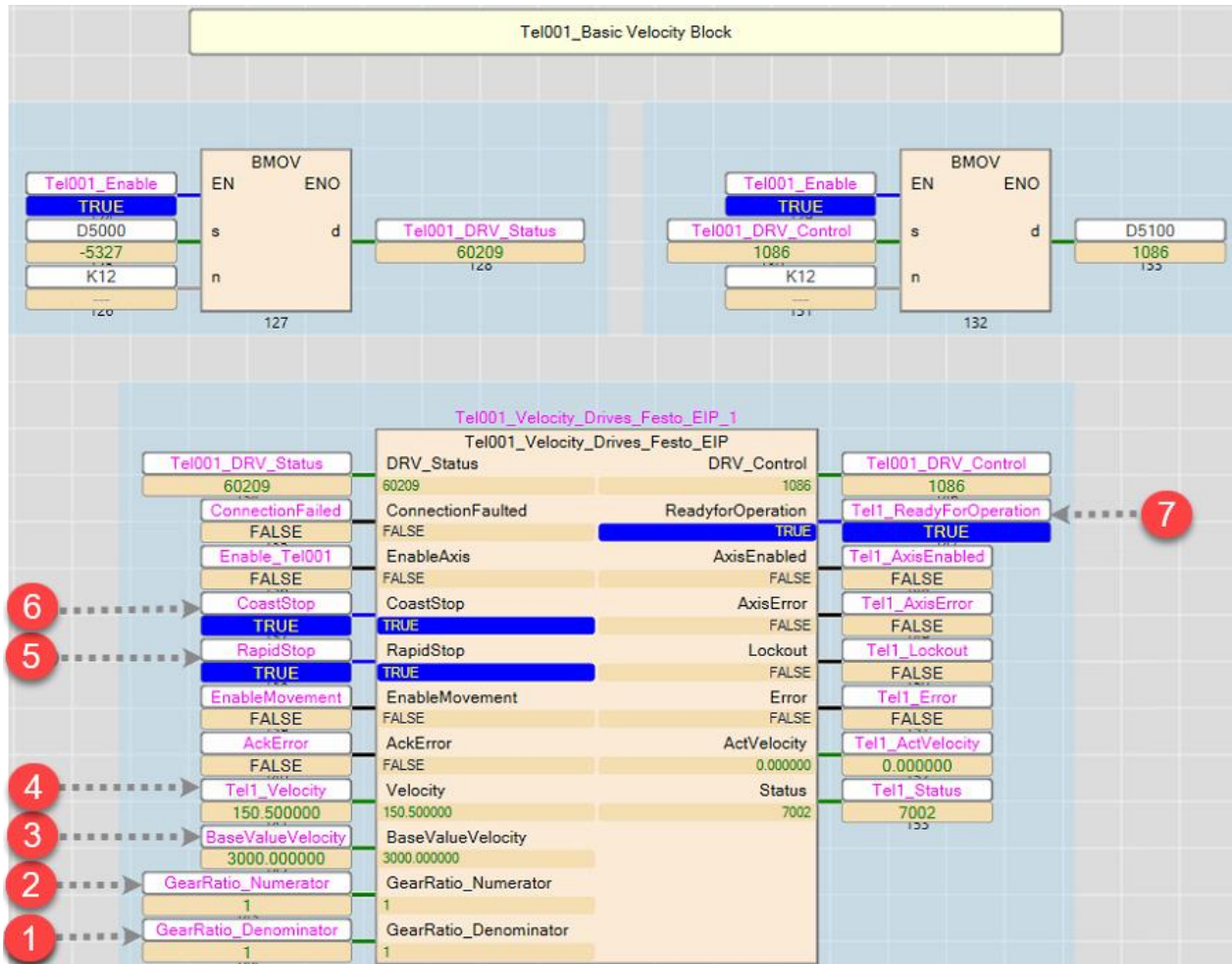
1. Enable the BMOV block. (Refer Note below).
2. Ensure that the 'RapidStop' and 'CoastStop' conditions are set to true. If either the 'RapidStop' or 'CoastStop' is in a false state, the drive will stop the axis, resulting in a **Lockout** and triggering an **Error**.



Note

- It is recommended not to call both the "Tel001_Velocity_Drives_Festo_EIP" and "Tel102_Velocity_Drives_Festo_EIP" function blocks in the program. If they are called, use the BMOV instruction to move process data to the DRV_Status and DRV_Control tags.
- Set the BMOV instruction to True for the respective "Tel001_Velocity_Drives_Festo_EIP" and "Tel102_Velocity_Drives_Festo_EIP" to assign process data.
- Do not enable both "Tel001_Velocity_Drives_Festo_EIP" and "Tel102_Velocity_Drives_Festo_EIP" at the same time, as this may result in malfunction.

Step – 2: FB View at Parameters Configuration Completed



1. Enter the value of the GearRatio_Numerator. User can find the value from the parameter P1.1242.0.0
2. Enter the value of the GearRatio_Denominator. User can find the value from the parameter P1.1243.0.0



Information

- The set velocity is always based on the motor velocity. If the user has a gear mounted, the gear factor is not included in the calculation. Therefore, the gear factor needs to be added in the FB to achieve the set velocity at the drive side.

3. Enter the value of the Base Value Velocity. User can find the value from the parameter P1.11280701.0.0
4. Enter the velocity value within a range that is twice the range of the Base value velocity.



Information

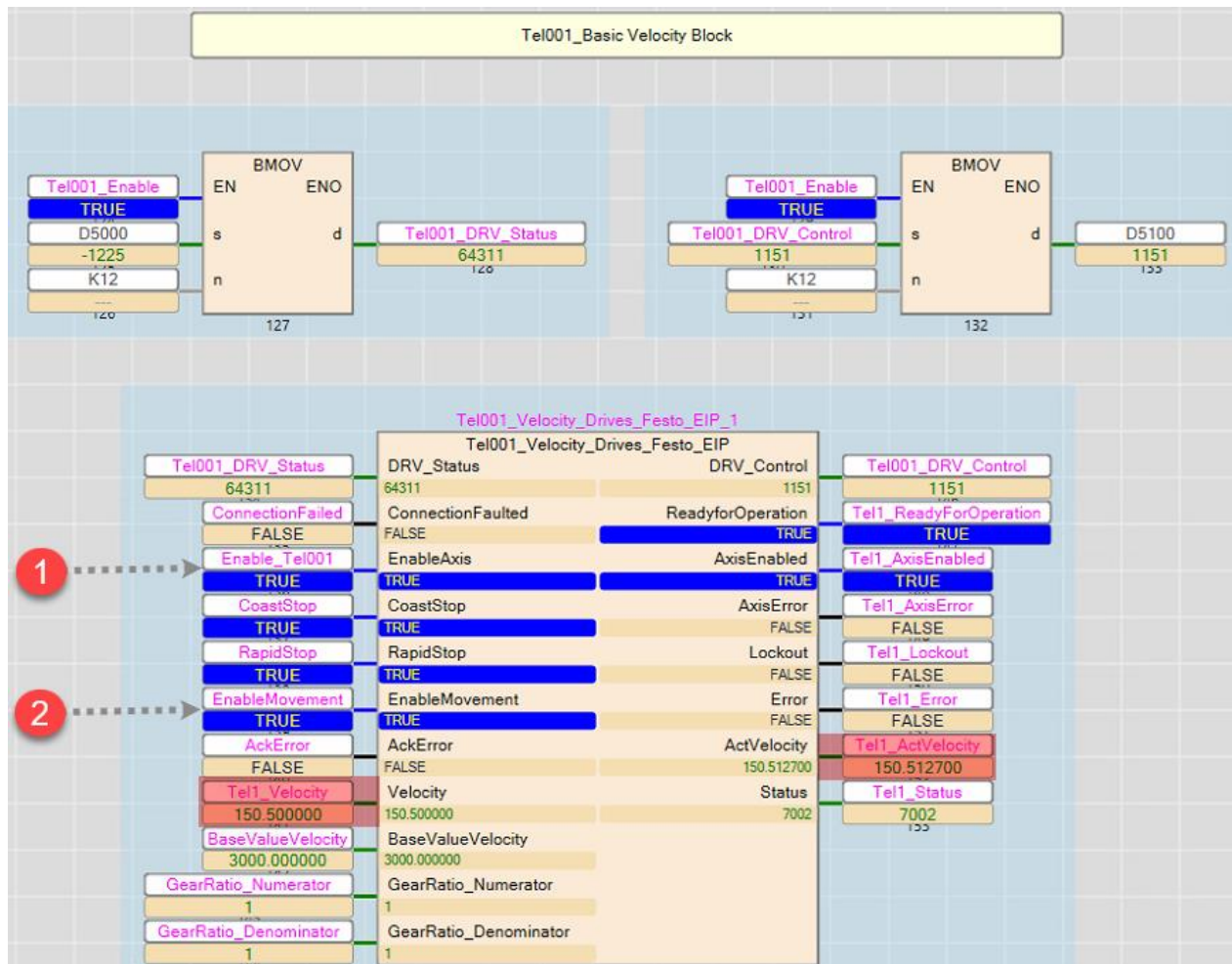
- Users can enter a velocity value that is twice the BaseValueVelocity. If a value higher than twice the BaseValueVelocity is entered, the actual velocity will not exceed twice the BaseValueVelocity.
- User can also vary the setpoint value during drive in run condition.

5. Ensure the “**RapidStop**” is in true condition if “**RapidStop**” is in false state then drive will stop the axis with deceleration time.
6. Ensure the “**CoastStop**” is in true condition if “**CoastStop**” is in false state then immediate stop (Emergency Stop) is active.
7. Once the project is ready & communication established with CMMT drive the “Tel001_Velocity_Drives_Festo_EIP” FB looks like above figure. The “ReadyforOperation” status tag indicates the FB ready state.



Note

- “ReadyforOperation” status tag indicates drive is in ready state
- Drive will run when “ReadyforOperation” status tags are true.

Step – 3: FB View After Run Command

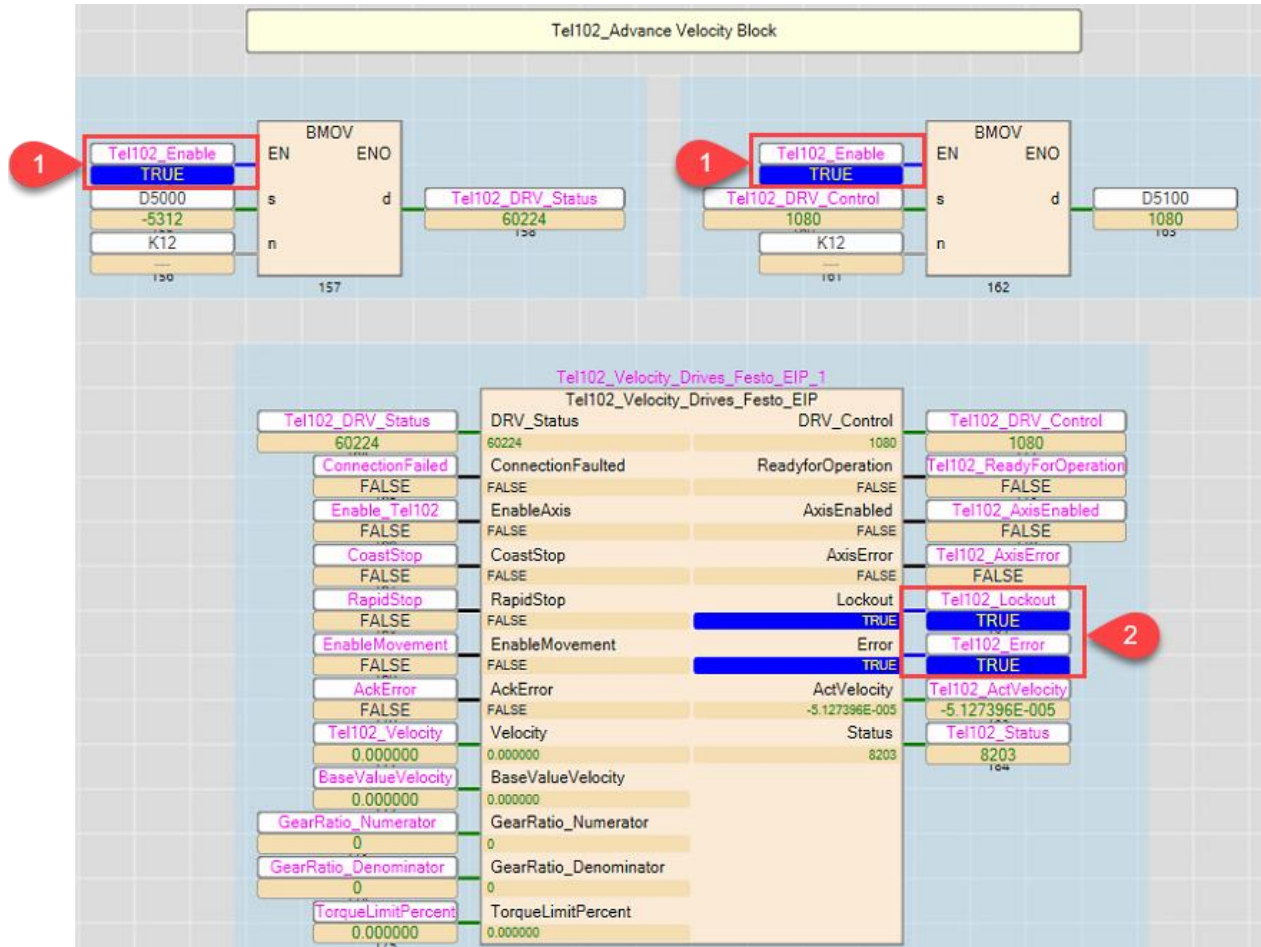
1. When “**EnableAxis**” is true then FB will enable the drive. At this stage “**AxisEnabled**” status tags will become true.
At this stage drive is ready to execute the run command provide by user.
2. Change the “**EnableMovement**” tag from false to true for provide the run command to the drive.

**Note**

- Motor actual speed will update in the “**ActVelocity**” status tag as highlighted in yellow color.

9.2.2 Tel001_Velocity_Drives_Festo_EIP Functions

Step – 1: FB View at First Time Online

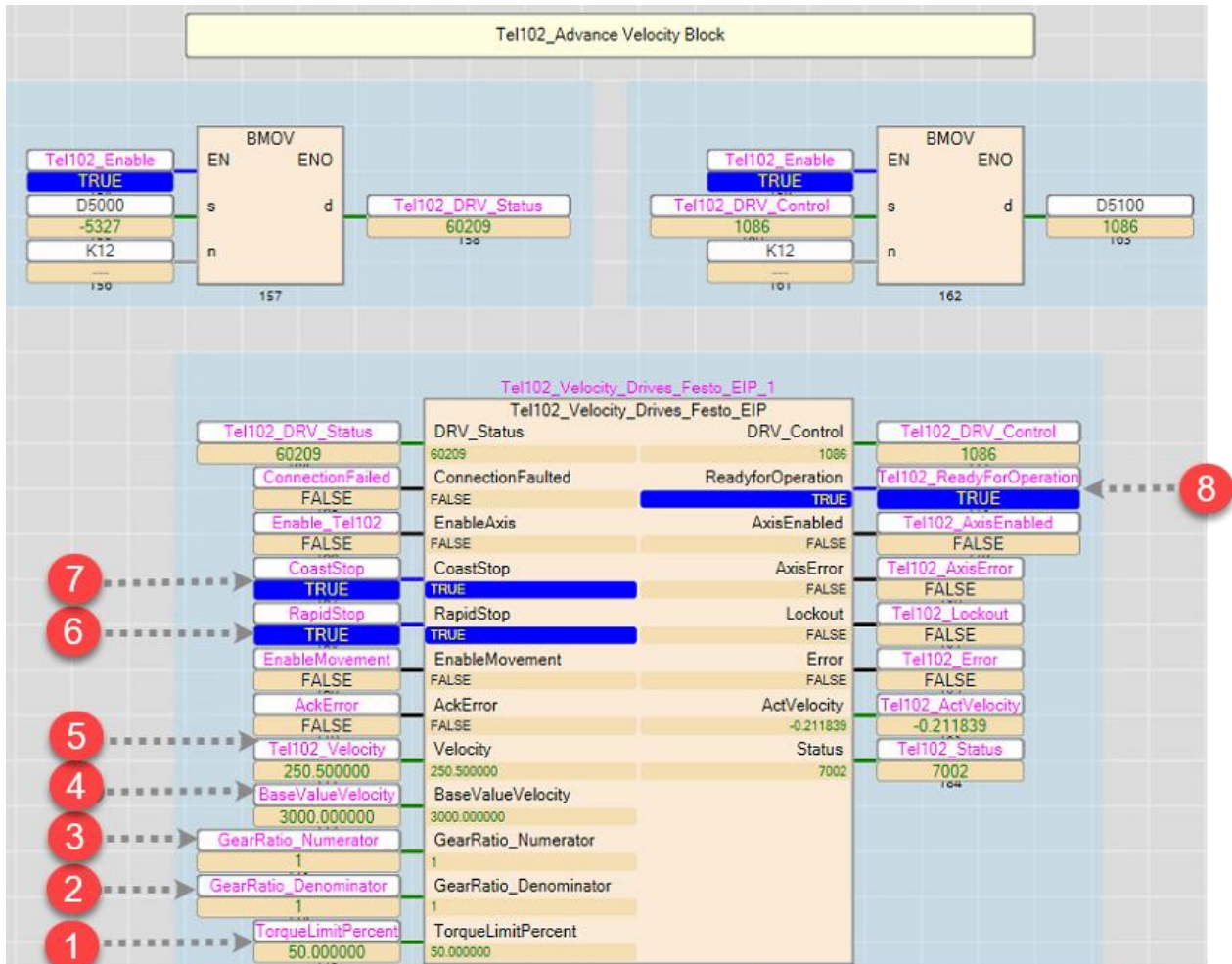


1. Enable the BMOV block. (Refer Note below).
2. Ensure that the 'RapidStop' and 'CoastStop' conditions are set to true. If either the 'RapidStop' or 'CoastStop' is in a false state, the drive will stop the axis, resulting in a **Lockout** and triggering an **Error**.



Note

- It is recommended not to call both the "Tel001_Velocity_Drives_Festo_EIP" and "Tel102_Velocity_Drives_Festo_EIP" function blocks in the program. If they are called, use the BMOV instruction to move process data to the DRV_Status and DRV_Control tags.
- Set the BMOV instruction to True for the respective "Tel001_Velocity_Drives_Festo_EIP" and "Tel102_Velocity_Drives_Festo_EIP" to assign process data.
- Do not enable both "Tel001_Velocity_Drives_Festo_EIP" and "Tel102_Velocity_Drives_Festo_EIP" at the same time, as this may result in malfunction.

Step – 2: FB View at Parameters Configuration Completed

1. Enter the value of torque limit. User can also torque limit during drive in running condition.
2. Enter the value of the GearRatio_Numerator. User can find the value from the parameter P1.1242.0.0
3. Enter the value of the GearRatio_Denominator. User can find the value from the parameter P1.1243.0.0

**Information**

- The set velocity is always based on the motor velocity. If the user has a gear mounted, the gear factor is not included in the calculation. Therefore, the gear factor needs to be added in the FB to achieve the set velocity at the drive side.

4. Enter the value of the Base Value Velocity. User can find the value from the parameter P1.11280701.0.0
5. Enter the velocity value within a range that is twice the range of the Base value velocity.

**Information**

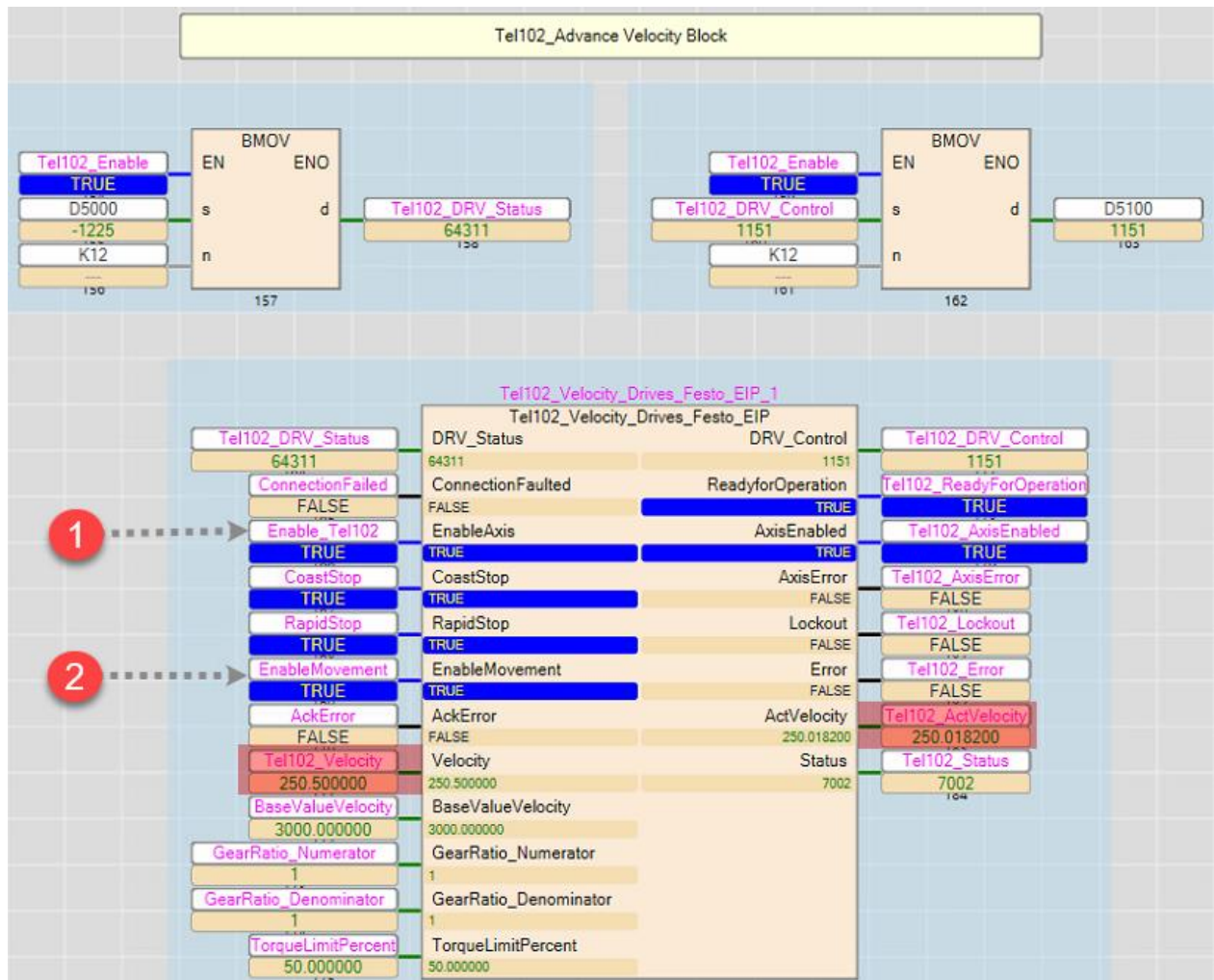
- Users can enter a velocity value that is twice the BaseValueVelocity. If a value higher than twice the BaseValueVelocity is entered, the actual velocity will not exceed twice the BaseValueVelocity.
- User can also vary the setpoint value during drive in run condition.

6. Ensure the **“RapidStop”** is in true condition if **“RapidStop”** is in false state then drive will stop the axis with deceleration time.
7. Ensure the **“CoastStop”** is in true condition if **“CoastStop”** is in false state then immediate stop (Emergency Stop) is active.
8. Once the project is ready & communication established with CMMT drive the “Tel001_Velocity_Drives_Festo_EIP” FB looks like above figure. The “ReadyforOperation” status tag indicates the FB ready state.

**Note**

- “ReadyforOperation” status tag indicates drive is in ready state
- Drive will run when “ReadyforOperation” status tags are true.

Step – 3: FB View After Run Command



1. When “**EnableAxis**” is true then FB will enable the drive. At this stage “**AxisEnabled**” status tags will become true.
At this stage drive is ready to execute the run command provide by user.
2. Change the “**EnableMovement**” tag from false to true for provide the run command to the drive.

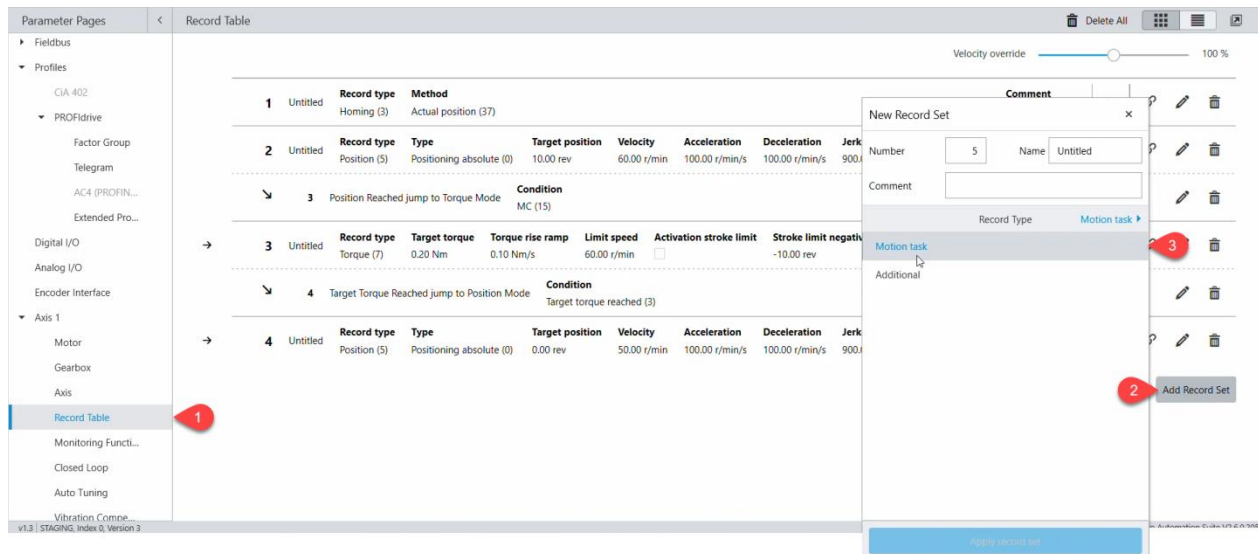
**Note**

- Motor actual speed will update in the “**ActVelocity**” status tag as highlighted in yellow color.

10 Record table

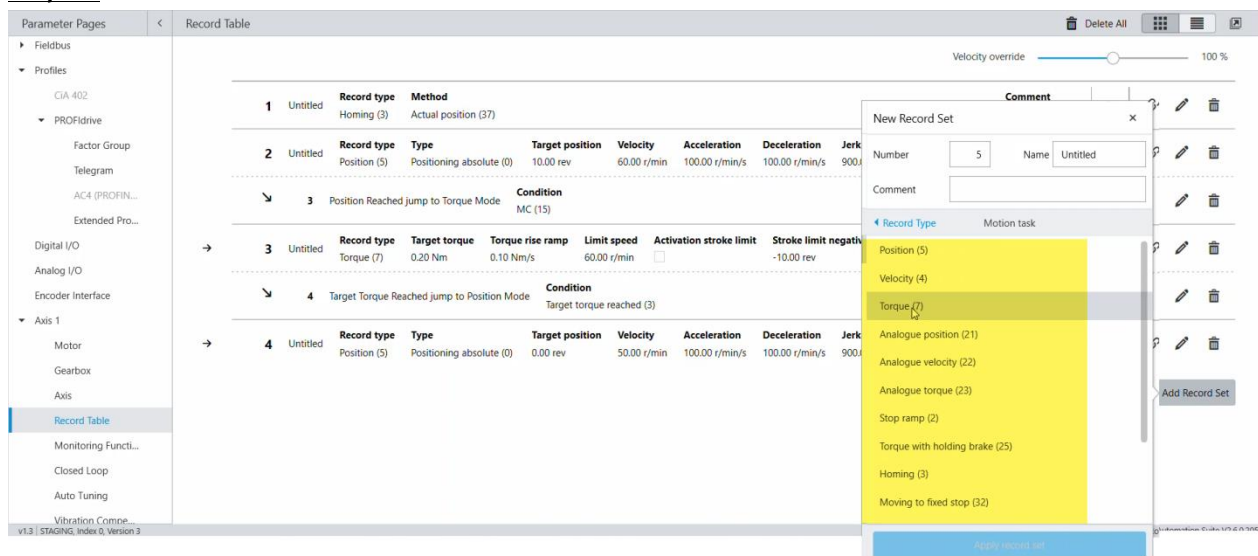
10.1 Set the Task/Sequence in Record Table

Step – 1



1. In **Parameter pages** => **Axis 1** => **Record Table**
2. Click on the **Add Record Set**
3. Select **Motion task**

Step – 2



- Select as per User requirement.

Step – 3

Edit Record Set

Number: 12 Name: Untitled

← Motion task Torque (7)

Target torque	5.00	Nm
Torque rise ramp	1.00	Nm/s
Limit speed	0.40	m/s
Activation stroke limit	☒ Active	
Stroke limit negative	-50.00	mm
Stroke limit positive	50.00	mm

Change record set

- Set the parameter value as per user requirement.
- Enable Activation Stroke limit function if required, set the stroke limit values.

Step – 4

Store on Device



Reinitialize

- Click the **“Store on device”** button to store all the parameters on drive memory.
- Click the **“Reinitialize”** the button to initialize the drive.

**Note**

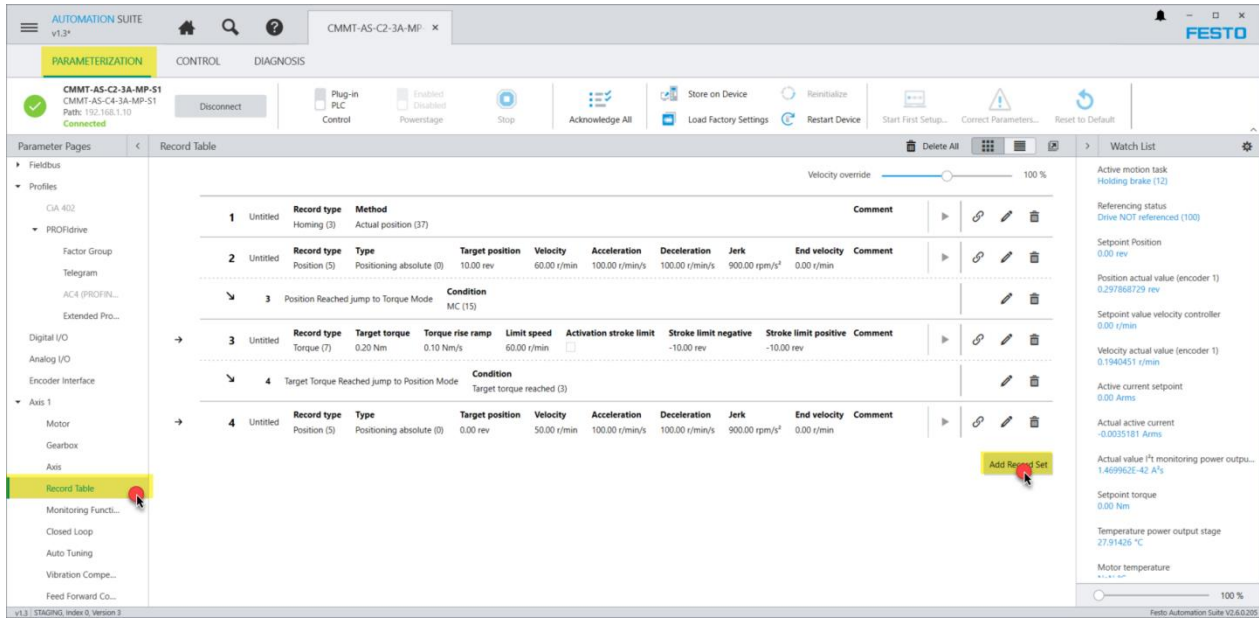
- The triangle warning symbol on both buttons indicates action required.
- All the modified parameters will automatically erase from drive during power restart If modified parameters are not stored in the drive.
- Its mandatory to reinitialize the drive for some parameter modification. If drive is not reinitialized the drive performance remains same as like previous configuration.

10.2 Testing the Task/Sequence of control through FAS

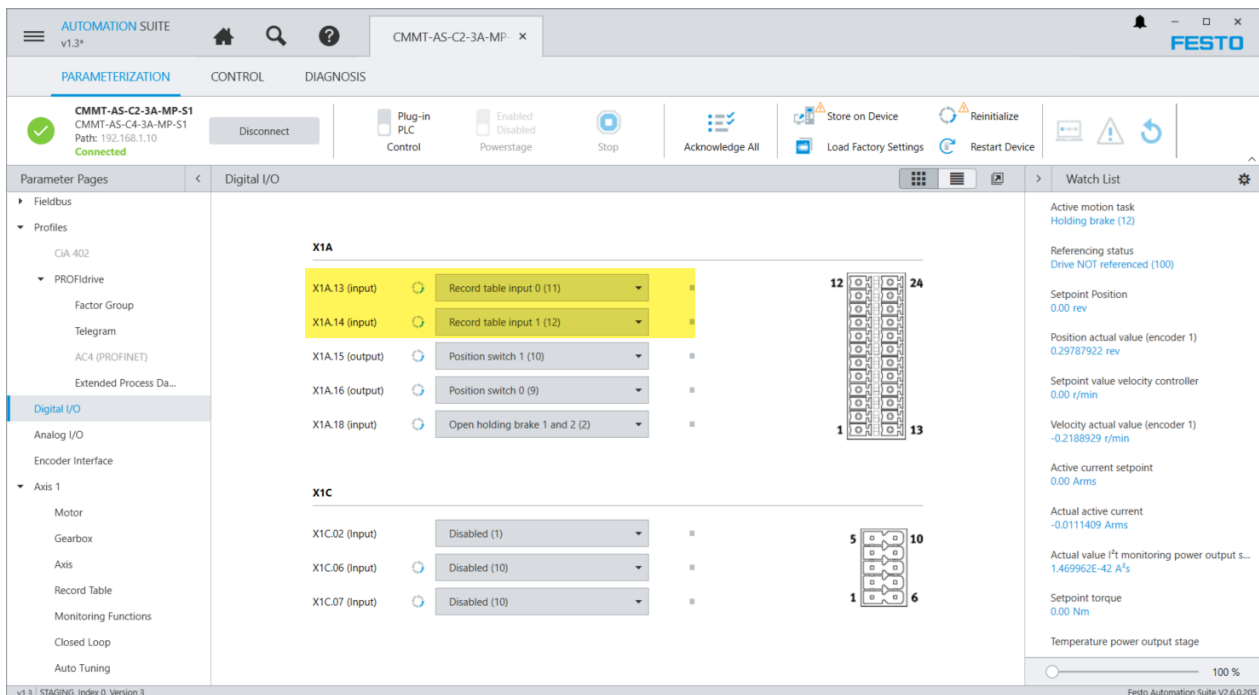
Run the CMMT-AS without PLC or Field Bus to ensure the Take/Sequence created in Record table.

Method – 1: Using I/O's running the Task/Sequence

Step – 1: Create the record list as per requirement, Example: shown in below image.



Step – 2



- assign digital input XA.13 to “record table input-0”
- assign digital input XA.14 to “record table input-1”

Step – 3

Parameter List						
ID	Name	Value	Unit			
▼ /Axis1/Controller enable group[0] (9) ↻						
P1.10231.0.0	Device status IO	Ready (1)				■
P1.10232.0.0	Controller enable selection	I/O (1)				■
P1.10234.0.0	Controller enable operating mode	Start record table (15)				■
P1.10235.0.0	Target velocity for controller enable (velocity operation)	0.00	r/min			■
P1.10236.0.0	Target torque for controller enable (torque operation)	0.00	Nm			■
P1.10237.0.0	Maximum velocity for controller enable (torque opera...	1.00	r/min			■
P1.10238.0.0	Index for controller enable	0				■
P1.112819.0.0	Error active	☐ Active				■
P1.11280018.0.0	Torque increase for controller enable	0.00	Nm/s			■
▼ /Axis1/PROFIdrive MELDW group[0] (12) ↻						
P1.11249110.0.0	MELDW.11 Controller enable	☐ Active				■
▼ /Axis1/PROFIdrive device identification group[0] (18) ↻						
P1.24126.0.0	Assignment controller enable	4				■
P1.24126.0.1	Assignment controller enable	11				■

Parameters can be found within the “Controller enable group” for axis-1

- P1.10232.0.0 – Controller enable selection (set to “ENABLE by IO” → CTRL-EN / X1A.3)
- P1.10234.0.0 – Controller enable operating mode (set to “Start record table”)
- P1.10238.0.0 – Index for controller enable (set to the record index you wish to start, here “0”)

Step – 4

Store on Device



Reinitialize

- Click the **“Store on device”** button to store all the parameters on drive memory.
- Click the **“Reinitialize”** the button to initialize the drive.

**Note**

- The triangle warning symbol on both buttons indicates action required.
- All the modified parameters will automatically erase from drive during power restart If modified parameters are not stored in the drive.
- Its mandatory to reinitialize the drive for some parameter modification. If drive is not reinitialized the drive performance remains same as like previous configuration.

Method – 2: In FAS, Bypass the PLC through enabling Plug-in PLC Control

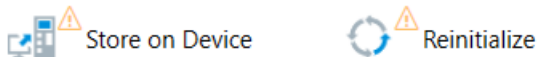
Step – 1

The screenshot shows the FESTO AUTOMATION SUITE v1.3* interface. The top bar has tabs for 'PARAMETERIZATION' (selected), 'CONTROL', and 'DIAGNOSIS'. Below the top bar, there's a status bar showing 'CMMT-AS-C2-3A-MP-S1' and 'Connected'. The left sidebar shows a tree view with 'Axis 1' expanded and 'Record Table' selected. The main area displays a 'Record Table' with four records. Red callout numbers 1 through 7 point to specific UI elements: 1. Connect button, 2. Parameterization tab, 3. Record Table in sidebar, 4. New record button, 5. Plug-in PLC Control button, 6. Play button, and 7. Store on Device button.

Record type	Type	Target position	Velocity	Acceleration	Deceleration	Jerk	End velocity	Comment
1	Untitled	Homing (3)	Actual position (37)					
2	Untitled	Position (5)	Positioning absolute (0)	10.00 rev	60.00 r/min	100.00 r/min/s	100.00 r/min/s	900.00 rpm/s²
3	Position Reached jump to Torque Mode	Condition	MC (15)					
3	Untitled	Torque (7)	0.20 Nm	0.10 Nm/s	60.00 r/min			
4	Target Torque Reached jump to Position Mode	Condition	Target torque reached (3)					
4	Untitled	Position (5)	Positioning absolute (0)	0.00 rev	50.00 r/min	100.00 r/min/s	100.00 r/min/s	900.00 rpm/s²

1. Click on **Connect** to establish the connection with CMMT-AS Drive.
2. Once the connection establishes, Go to **Parameterization**.
3. Expand Axis 1 => Click on Record Table
4. Create the record list, User defined.
5. Toggle **Control** and **Powerstage** button.
6. Click on the Play Button of the selected record list with user want to execute.
After checking the Task/Sequence, Store the Record Table in the Drive.

Step – 2



- Click the **“Store on device”** button to store all the parameters on drive memory.
- Click the **“Reinitialize”** the button to initialize the drive.

11 How to control the torque control at the position.

11.1 Application description

This document was created to be able to set-up the CMMT-AS into the “Travel to fixed stop” (application class 3) mode with using the PTP_Drives_Festo_EIP function block and the telegram 111. This instruction shows the needed and useful parameters on the CMMT-AS and on the other hand the must changes on the PTP_Drives_Festo_EIP function block.

In general, the PTP_Drives_Festo_EIP function block does not offer in- or output interfaces for a “Travel to fixed stop”. To setup this mode it needs some changing on the default settings of the control words/ConfigEPos at PTP_Drives_Festo_EIP function block (Mitsubishi) side.

Mitsubishi named the “Torque Mode” as “Travel to fixed stop”. This mode is a positioning task with a defined clamping torque.

11.1.1 Used Parameters for Torque

Name	Parameter Number
Clamping torque [Nm]	526801
Fixed stop detection monitoring window	4694
Fixed stop detection damping time	4693
Fixed stop negative stroke limit	11280409
Fixed stop positive stroke limit	11280408
Lower limit value torque	852
Upper limit value torque	853
Target position	11280604
Profile speed	11280605
Monitoring window target torque	4611
Damping time target reached	468

11.1.2 Travel top fixed stop (application class 3)

Travel to fixed stop performs a positioning with reference to a defined max. clamping torque. During travel to fixed stop a fixed stop is approached from the current position before reaching the target position (e.g. at a work piece). Then a torque is established up to the desired clamping torque. For example, the following parameters can be set:

- Position
- Speed
- Acceleration
- Deceleration
- Clamping torque
- Clamping torque offset

A current positioning task can be switched by "STW2.8 Traverse to fixed end stop". The switching runs a positioning task with clamping torque. The closed-loop limit manager limits the motion to the clamping torque. On completion of the task the original limit is restored.

The following error monitor is not active during the task and the following status bits are set:

- ZSW2.8 Travel to fixed stop active
- POS_ZSW2.14 Move to fixed stop active

The following error monitor of the motion monitor is used during the task to detect the fixed stop.

When the fixed stop is detected, "POS_ZSW2.12 Fixed stop reached" is set and the stroke limit monitor of the motion monitor is activated based on the current position.

With pending clamping torque "POS_ZSW2.13 Fixed stop Clamping torque reached" is set.

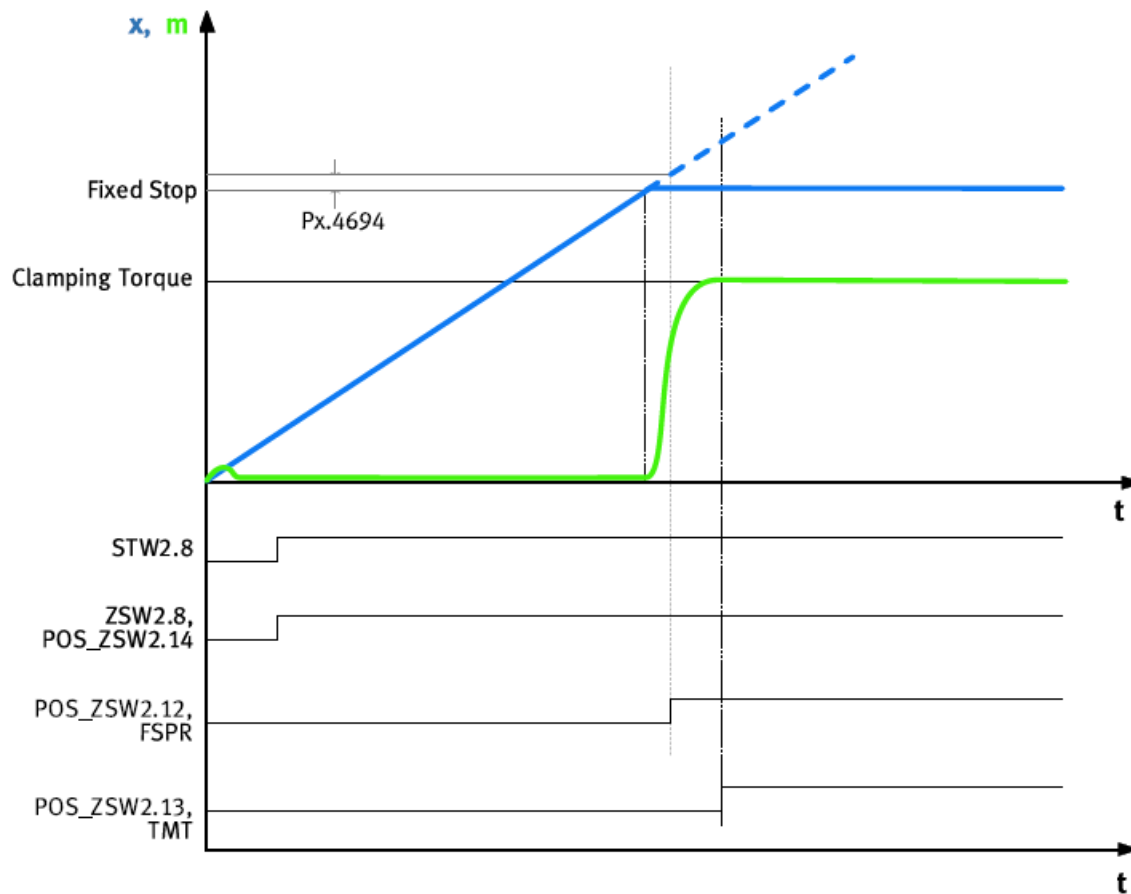
The clamping torque remains pending until a new travel command starts.

When the stroke limits for the fixed stop monitor are reached, "POS_ZSW2.12 Fixed stop reached" is reset.

How to control the torque control at the position.

Timing

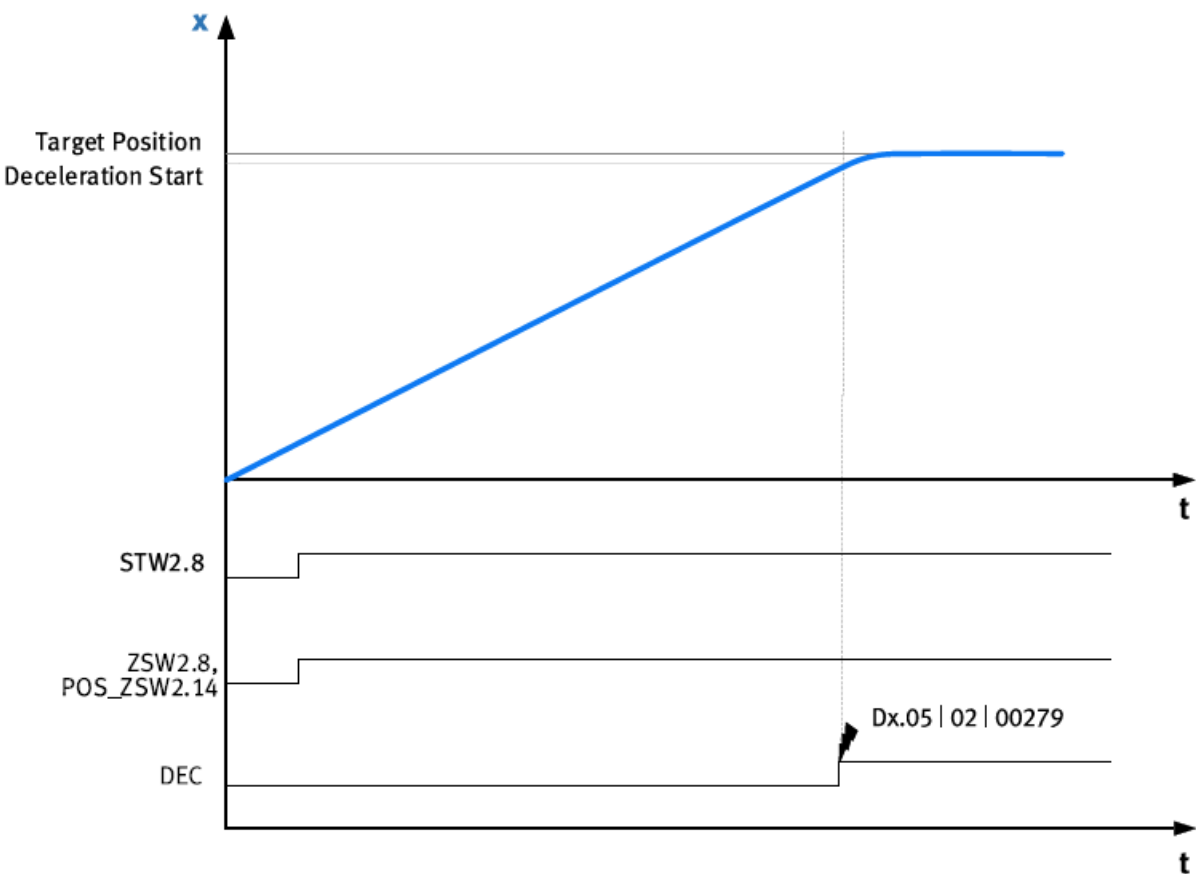
Example 1: travel to fixed stop with reaching and stopping at the fixed stop



Legend for timing diagram travel to fixed stop

Name	Description	Parameter Number
Fixed Stop	Fixed Stop	-
Clamping torque	Clamping torque	526801
FSPR	Motion monitoring function "fixed stop reached" (1 = status reached)	Px.460
TMT	Motion monitoring function "target torque range monitor" (1 = status reached)	Px.460

Example 2: travel to fixed stop without reaching the fixed stop



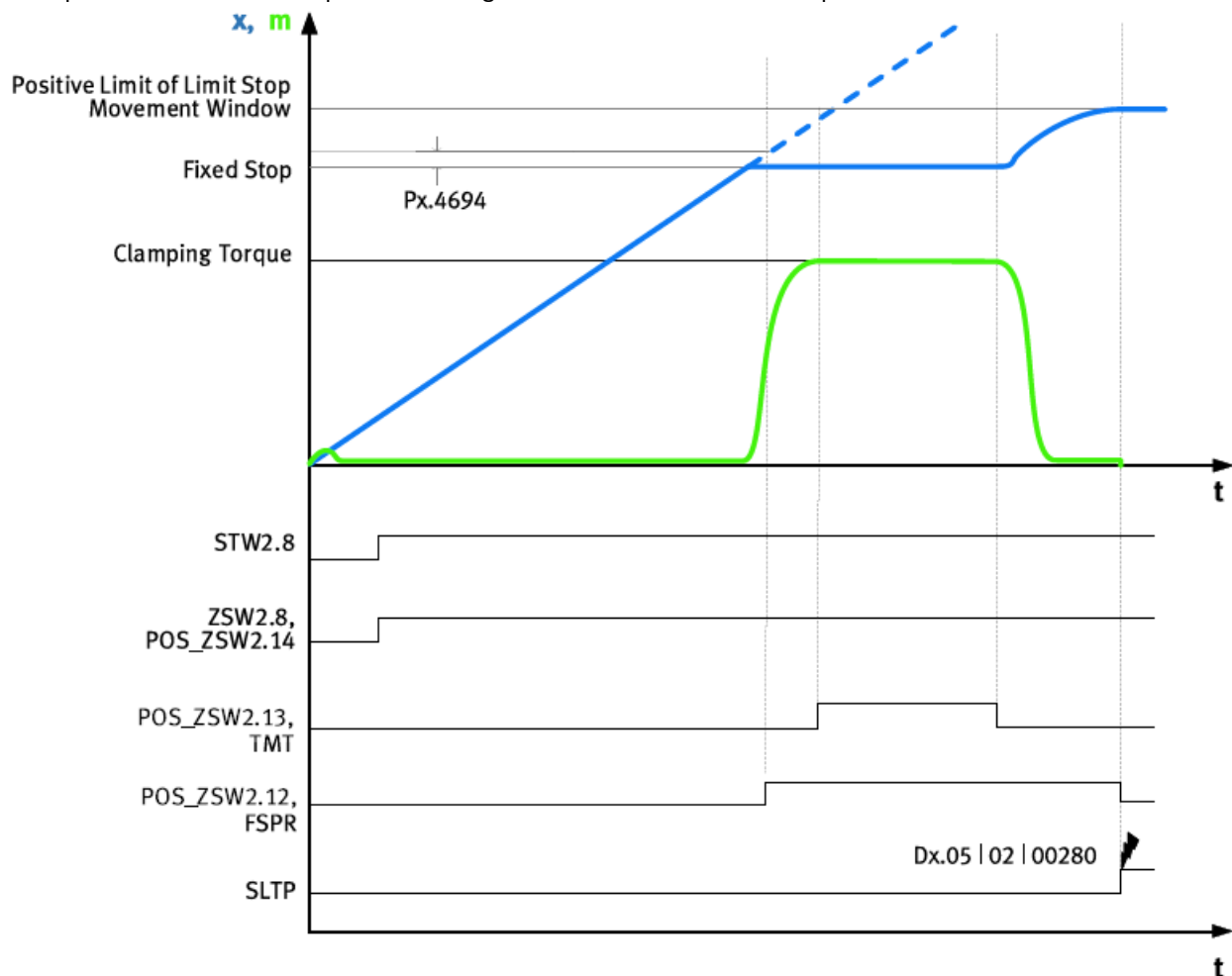
Information

The diagnostic message Dx.05.02.00279 must be activated in the Error classification within the Festo Automation Suite

Name	Description	Parameter
Target position	Target position	-
Deceleration start	Start of deceleration	-
DEC	Motion monitoring function "drive decelerated" (1 = statusreached)	Px.460
Dx.05 02 000279	Diagnostic message Fixed stop not detected	-

How to control the torque control at the position.

Example 3: travel to fixed stop with reaching and feedback at the fixed stop



Information

The diagnostic message Dx.05.02.00280 must be activated in the Error classification within the Festo Automation Suite

Legend for timing diagram fixed stop feeds back

Name	Description	Parameter
Positive Limit of Limit Stop Movement Window	Positive stroke limit of fixed stop monitor	11280408 11280409
Clamping torque	Clamping torque	526801
Fixed Stop	Fixed Stop	-
TMT	Motion monitoring function "target torque range monitor" (1 = status reached)	Px.460
FSPR	Motion monitoring function "fixed stop reached" (1 = status reached)	Px.460
STLP	Motion monitoring function "stroke limit reached" (1 = status reached)	Px.460
Dx.05 02 280	Diagnostic message Monitoring window of fixed stop left	Px.460

11.1.3 General parameters for “Travel to fixed stop”

Used parameters are described in the software documentation of the CMMT-AS in detail.

Below were the parameter numbers and the related chapters which are specified in the official documentation of CMMT-AS.

Details of the motion monitoring functions -> Motion monitoring functions.

The fixed stop detection acts like the following error monitor for position with critical limit and timing

-> Following error. The following error of the position and a damping time are used (Px.4694, Px.4693).

The detection of the pending clamping torque acts like the target range monitor for torque with critical limit and timing -> Target area monitoring.

The monitoring of the stroke limits after detected fixed stop acts like the stroke limit reached motion monitor -> Stroke limit reached.

The window limits can be set in the positive and negative directions (Px.11280408, Px.11280409 -> Tab. 266 Parameters).

If the motion leaves the monitoring window in the positive and negative direction it is detected and triggers the following diagnostic message:

– Monitoring window of fixed stop left: Dx.05 | 02 | 00280

The following parameter determines the braking behaviour on exit of the monitoring window:

– Activation of automatic stop ramp stroke limit: Px.4675

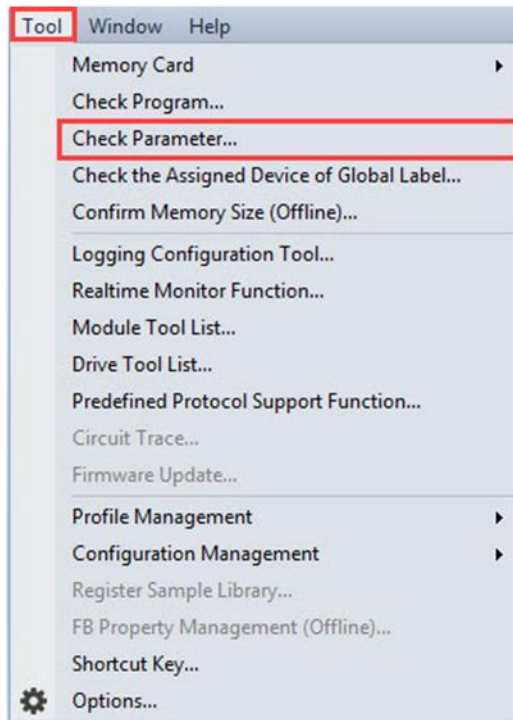
The clamping torque depends on the direction of motion. The set clamping torque is added with the offset. This means that the resulting clamping torque depends on the sign of the offset.

An asymmetrical clamping torque can be set with the offset for suspended axes (parameter Clamping torque offset, Px.11280407).

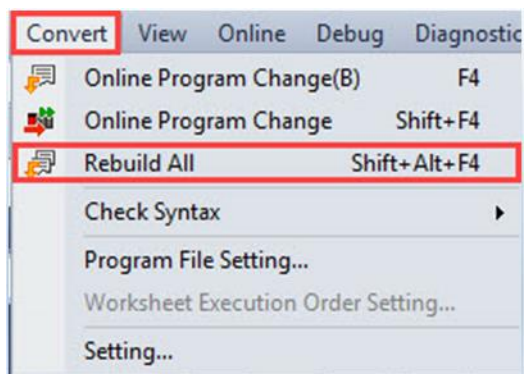
12 Downloading And Monitor Mode.

12.1 Download.

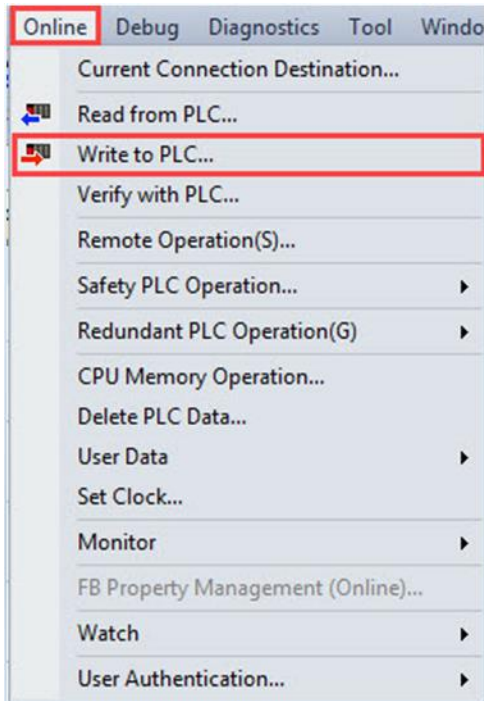
1. Click on **Tool** » Click **Check Parameter** as shown below.



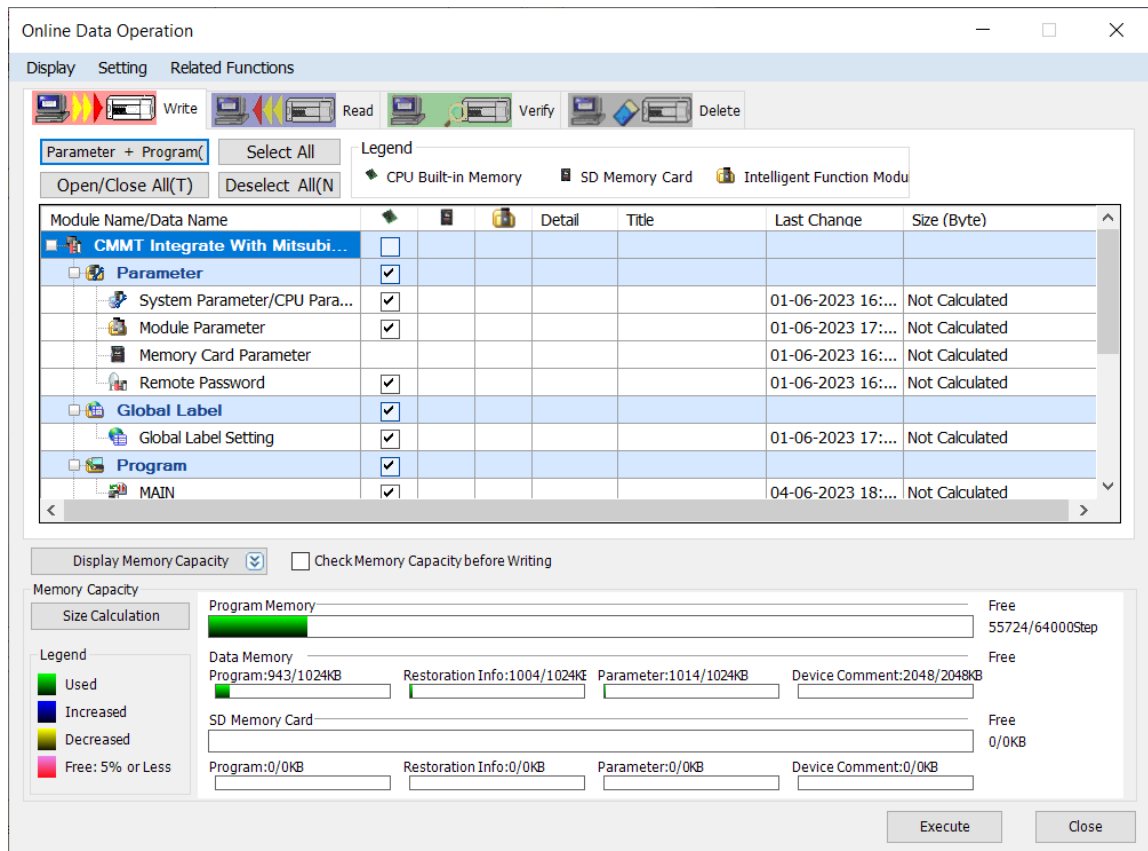
2. Click on **Convert** » Click **Rebuild All** as shown below.



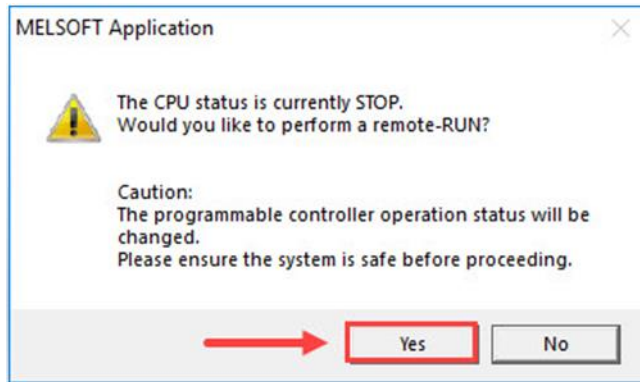
- Click on **Online** » Click **Write to PLC** as shown below.



- Click on **Select All** to download all program and parameters. Click on **Execute**.

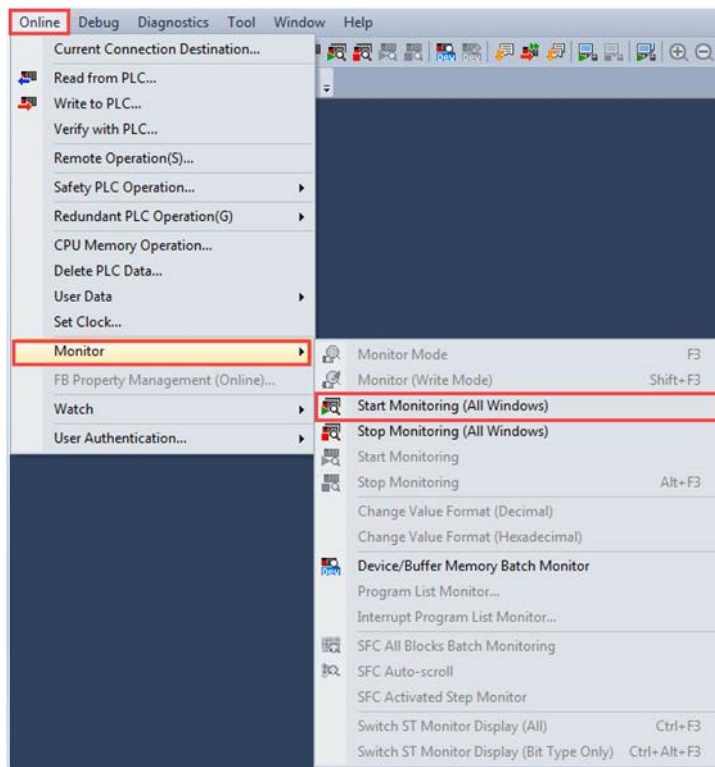


5. Click on **Yes** to Download the Project.

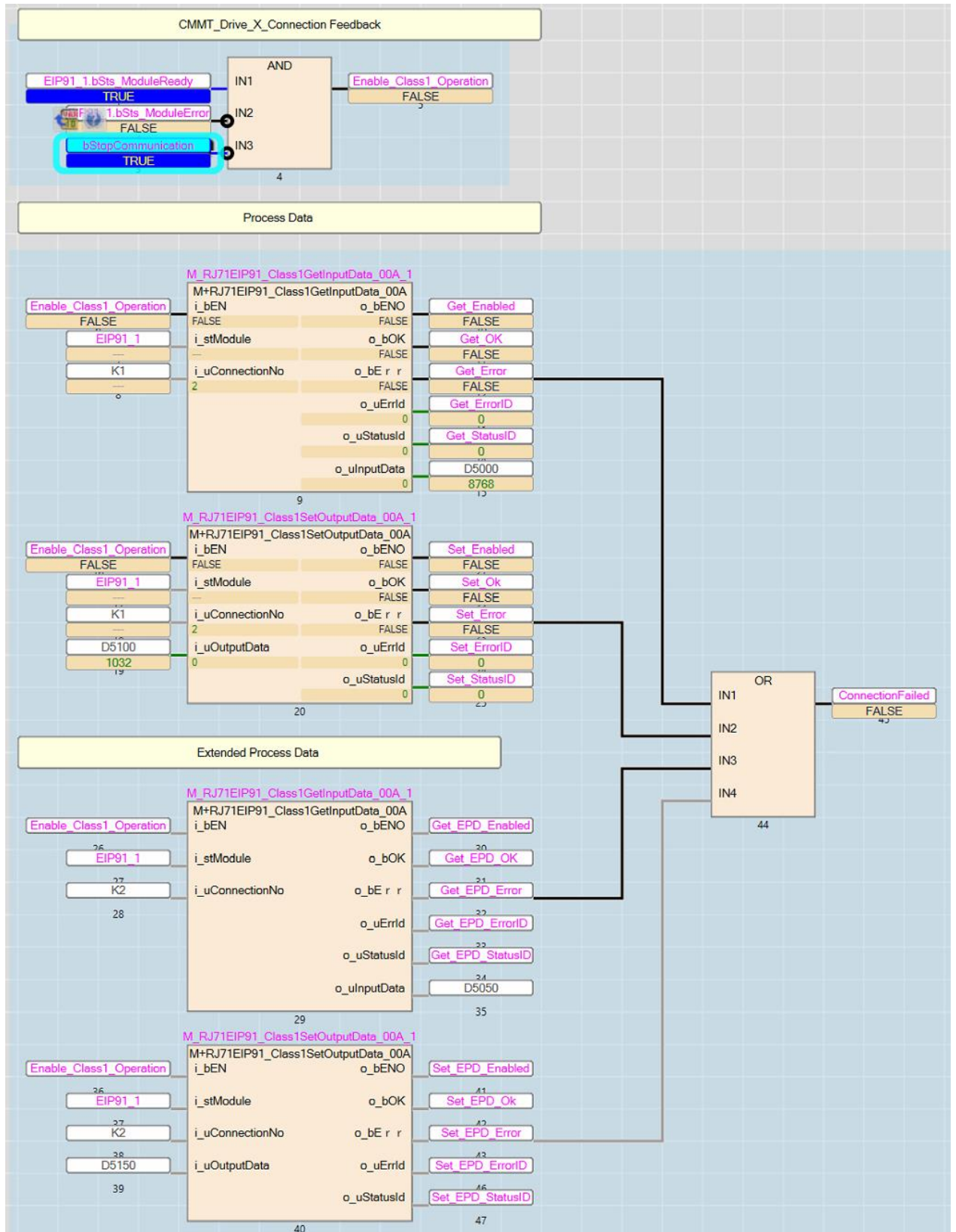


12.2 Monitor Mode.

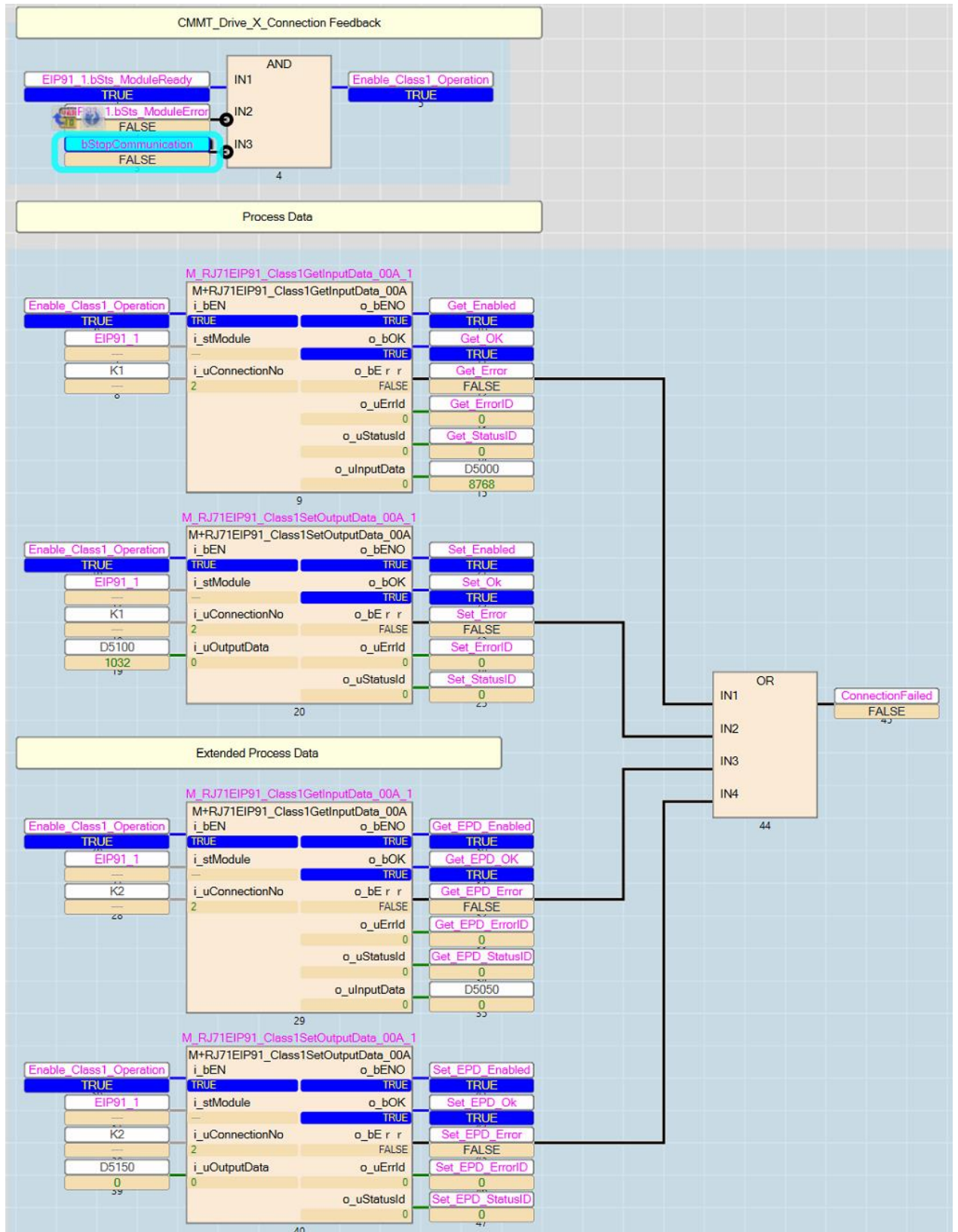
1. Click on **Online** » Click on **Monitor** » Click **Start Monitoring**.



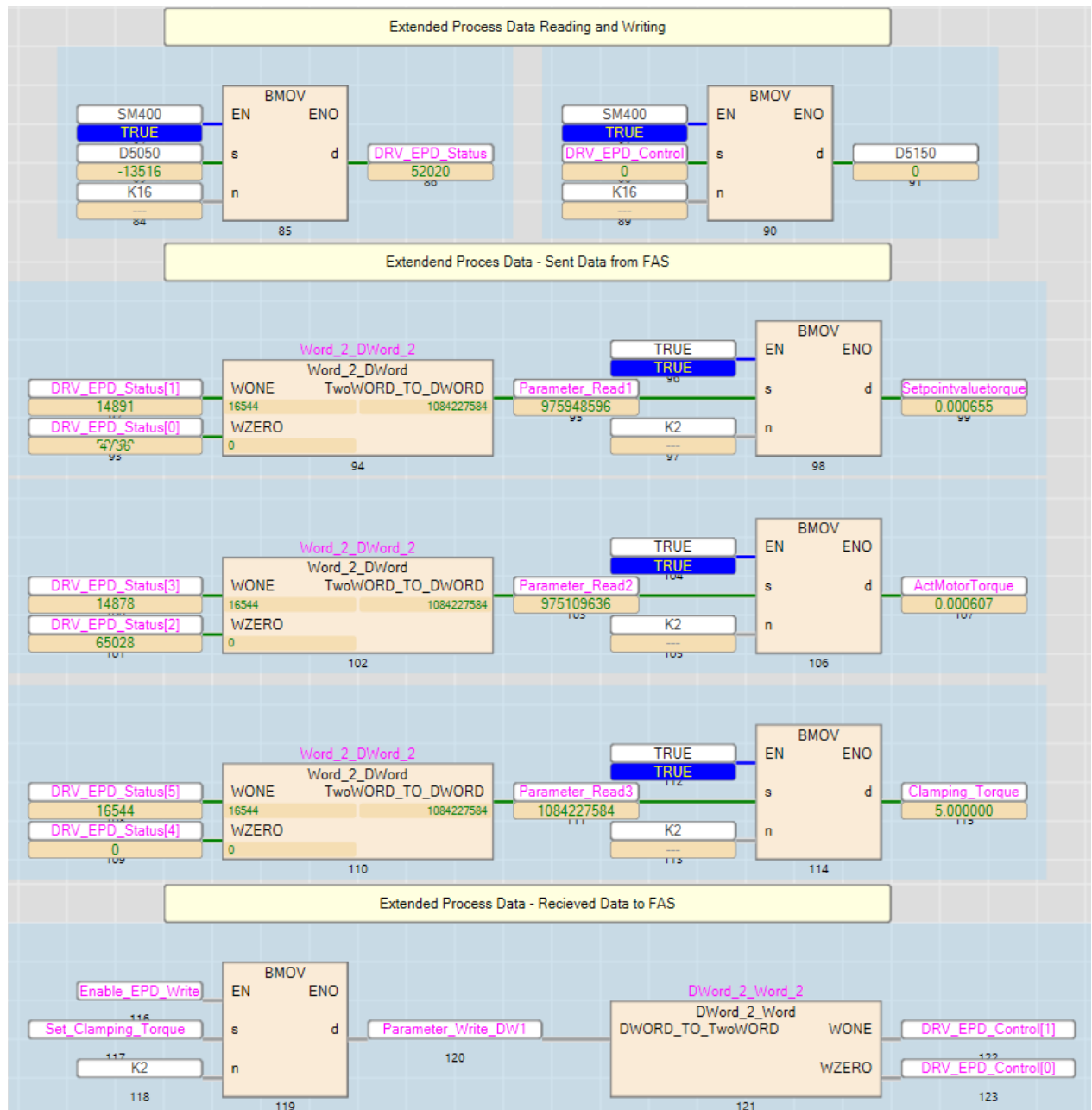
2. Now All the windows in Monitor Mode.



3. Enable the Get and Set Data block to communicate with External Device(CMMT).



4. Extended Process Data.



5. Reading the Parameter Value from CMMT.

- Enter the PARAMETER Value and INDEX Value.
- Execute the Block.
- Values are received in the Array of Word.
- Here the Received Values is **Word 1= 60815; Word 2= 16873**

P1.2851.0.0	Lower limit value power output stage temperature	-10.00 °C	
/System/Device temperature monitoring group[0] (28)			
P0.920.0.0	Temperature power output stage	29.40802 °C	
P0.921.0.0	Temperature status power output stage	No limit violated (0)	
P0.9314.0.0	Upper limit value warning threshold power output stage temperat...	80.00 °C	
P0.9315.0.0	Upper limit value power output stage temperature	90.00 °C	
P0.9316.0.0	Lower limit value warning threshold power output stage temperatu...	0.00 °C	

Temperature power output stage

P0.920.0.0 (FLOAT32)

EtherCAT: CoE 0x2128.1 (REAL)

EtherNet/IP: CIP 2246.0 (REAL)

PROFINET: PNU 2246.0 (REAL)

Specifies the actual temperature value of the power output stage.

The screenshot shows a ladder logic network for parameter reading and writing. The first network is a 'Para_Execute' coil connected to a 'PTP_Drives_Parameter_EIP_1' block. This block has several inputs and outputs related to device communication and error handling. The second network is a 'Para_Finish' coil connected to a 'Para_Finish' block. The variable table on the right lists the following variables:

Name	Current Value	Display Form
Drives_Controls/Para_Instance	2,246	Decimal
Drives_Controls/Para_Attribute	0	Decimal
Drives_Controls/Para_FB_WriteDataLength	0	Decimal
Drives_Controls/Parameter_Write	0.000000	--
Drives_Controls/Para_FB_WriteData	--	--
Drives_Controls/Para_FB_ReadData	--	--
[0]	60,815	Decimal
[1]	16,873	Decimal
[2]	0	Decimal
[3]	0	Decimal
[4]	0	Decimal
[5]	0	Decimal

The screenshot shows a ladder logic network for parameter read conversion. It starts with a 'Para_FB_ReadData[1]' coil connected to a 'Word_2_DWord' block. This block has two inputs: 'WONE' (value 16873) and 'WZERO' (value 60815). The output of this block is connected to a 'Parameter_Read' block, which outputs a 'Double Word Value' of 1105849743. This value is then connected to a 'BMOV' block, which outputs a 'Real value' of 29.241000.

- Enter the Input Array of Word in **Word_2_DWord** and get Double Word Value Output.
- Enter the same Double Word Value Output in BMOV block and get Real value Output.

6. Writing the Parameter Value to CMMT.

- Enter the PARAMETER Value and INDEX Value.
- Enable the xReadWrite to write the data
- Execute the Block.
- Values are Send in the Array of Word(USE First two words for writing the value in the array).
- Here the Sending Values is **Word 1= 0000; Word 2= 16544**

The screenshot shows the 'Clamping torque' parameter configuration window. The parameter is 'P1.526801.0.0' (Clamping torque) with a value of 0.00 Nm. The 'EtherNet/IP' section is highlighted, showing the 'CIP 12168.0 (REAL)' value. The 'PROFINET' section shows the 'PNU 12168.0 (REAL)' value. The 'Default' value is 1E+37 Nm and the 'Last saved' value is 0.00 Nm.

The screenshot shows a ladder logic network for parameter write conversion. It starts with a 'Parameter_Write' coil connected to a 'BMOV' block. This block has two inputs: 'Parameter_Write' (value 5.000000) and 'Parameter_Write' (value 2). The output of this block is connected to a 'DWord_2_Word' block. This block has two inputs: 'DWord_2_Word' (value 1084227584) and 'WZERO' (value 0). The output of this block is connected to a 'Para_FB_WriteData[1]' block, which outputs a 'Word Value' of 16544. The variable table on the right lists the following variables:

Name	Current Value	Display Form
Drives_Controls/Para_Instance	12,168	Decimal
Drives_Controls/Para_Attribute	0	Decimal
Drives_Controls/Para_FB_WriteDataLength	4	Decimal
Drives_Controls/Parameter_Write	5.000000	--
Drives_Controls/Para_FB_WriteData	--	--
[0]	0	Decimal
[1]	16,544	Decimal

- Enter the Input Real Value Output in BMOV block and get Double Word Value Output.
- Enter the DWord Output value in **DWord_2_Word** and get Array of Word Value Output.

Parameter Reading and Writing

PTP_Drives_Parameter_EIP_1

TRUE

K1

16#C0A80105

3232235781

Para_EnableWrite

TRUE

Para_Instance

12168

Para_Attribute

0

Para_FB_WriteDataLength

4

Para_FB_WriteData

0

xExecute

TRUE

wTargetDeviceNo

1

dwTargetIP

3232235781

xReadorWrite

TRUE

wInstance

12168

wAttribute

0

wWriteDataLength

4

awWriteData

0

xDone

TRUE

xBusy

FALSE

xError

FALSE

wErrorID

0

awReadData

0

0

0

0

Para_FB_Done

TRUE

Para_FB_Busy

FALSE

Para_FB_Error

FALSE

Para_FB_ErrorID

0

Para_FB_ReadData

0

ON

OFF

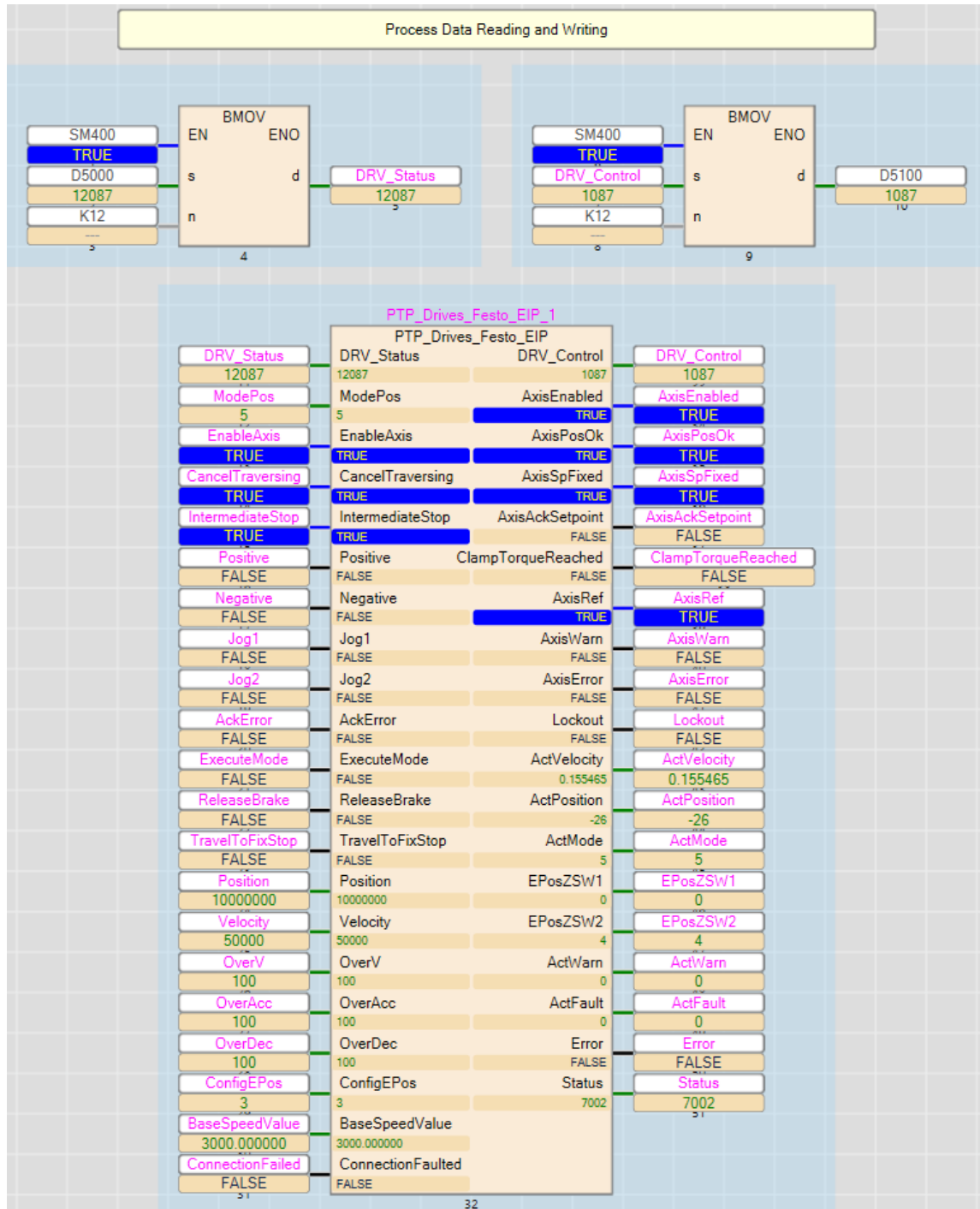
ON/OFF toggle

Update

Name	Current Value	Display Form
Drives_Controls/Para_Instance	12,168	Decimal
Drives_Controls/Para_Attribute	0	Decimal
Drives_Controls/Para_FB_WriteDataLength	4	Decimal
Drives_Controls/Parameter_Write	5.000000	--
Drives_Controls/Para_FB_WriteData	--	--
[0]	0	Decimal
[1]	16,544	Decimal
[2]	0	Decimal
[3]	0	Decimal
[4]	0	Decimal
[5]	0	Decimal
[6]	0	Decimal

ID	Name	Value	Unit
/Axis1/Control limitation group[0] (20)			
P1.526801.0.0	Clamping torque	5.00	Nm
P1.11280407.0.0	Clamping torque offset	0.00	Nm
/Axis1/PROFIdrive POS_ZSW2 group[0] (16)			
P1.112413130.0.0	POS_ZSW2.13 Fixed stop Clamping torque reached	☐ Active	

7. Homing Done through PTP_Drives_Festo_EIP FB.



13 Sample Code Description

For instance let's take an sequence as described below.

control sequence

1. Move to a specific location by position control.
2. Press with torque control at the position.
 - Stroke limit function applied in torque control
(For example, if the stroke limit is set to 100mm, the torque control stops when it reaches 100mm during torque control)
3. Return to initial position by position control after press.

Above Sequence can be executed in two ways,

1. Writing the logic in PLC using PTP_Drives_Festo_EIP Modes of Operation
2. Using Record Table, instead writing the entire logic in PLC and then use PTP_Drives_Festo_EIP FB to execute Record Table.

13.1 Method 1 - Using PTP_Drives_Festo_EIP Modes of Operation



Note

- Refer the [Chapter – 4](#) for Integrating the FB's in PLC program.



Information

PTP_Drives_Festo_EIP FB is written in Structure Text language, it can be called in Ladder and FBD/LD Program Language

13.1.1 Set EPD and Dynamics Parameters in FAS

Step – 1

1. To execute the sequence, user has to add the EPD, given below as in FAS.

Extended process data (EtherNet/IP)

Extended process data active

☒ Active
☒ Active

Sent Data

ID	Parameter	Type	Byte position	
0 P1.2220.0.0	Setpoint value torque	FLOAT32	0	
1 P1.150.0.0	Actual torque value motor shaft	FLOAT32	4	
2 P1.526801.0.0	Clamping torque	FLOAT32	8	

Add process channel

Number of bytes Tx

20

Number of bytes Tx (Offline)

12

Received Data

ID	Parameter	Type	Byte position	
0 P1.526801.0.0	Clamping torque	FLOAT32	0	

Add process channel

Number of bytes Rx

4

Number of bytes Rx (Offline)

4

Extended Process Data :- Sent Data from FAS (Read Data)

ID	Parameter	Type	Byte Position	Word
P1.2220.0.0	Setpoint value torque	FLOAT32	0	2
P1.150.0.0	Actual torque value motor shaft	FLOAT32	4	2

Extended Process Data :- Received Data to FAS from PLC (Receive Data)

ID	Parameter	Type	Byte Position	Word
P1.526821.0.0	Clamping Torque	FLOAT32	0	2

**Note**

- In this method –
- Axis won't stop even after achieving the Clamp torque. Axis continues to move till the set Position is reached.
 - To stop axis when clamp torque reached is True, following parameters to be set. "CancelTraversing" and "IntermediateStop" input tags must be False. Dynamics – such as Acceleration & Deceleration must be maximum to stop the axis.

PARAMETERIZATION CONTROL DIAGNOSIS

CMMT-AS-C2-3A-MP-S1
CMMT-AS-C4-3A-MP-S1
Path: 192.168.1.10
Connected

Disconnect Plug-in PLC Control Enabled Disabled Powerstage Stop Acknowledge All Store on Device Reinitialize Load Factory Settings Restart Device

Parameter Pages < Telegram Telegram selection Telegram (111) Actual application class Application class 3 (3)

PROFIdrive
Factor Group
Telegram

AC4 (PROFINET)
Extended Process Da...
Digital I/O
Analog I/O
Encoder Interface
Axis 1
Motor
Gearbox
Axis
Record Table
Monitoring Functions
Closed Loop
Auto Tuning
Vibration Compensation
Feed Forward Control
Modulo Mode
Position Trigger

Dynamic Values

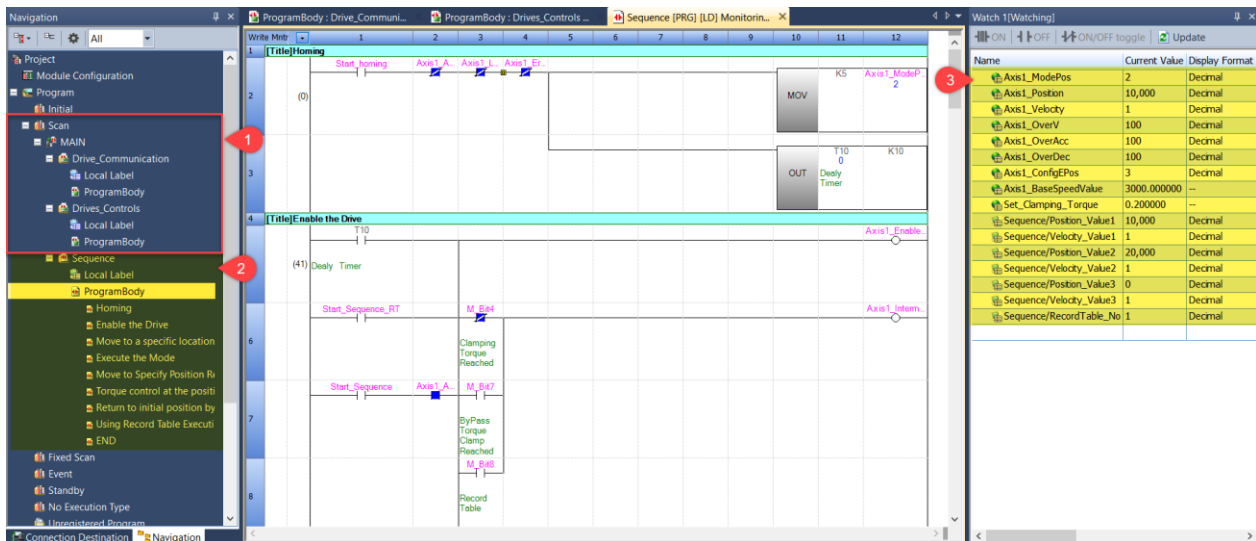
Coast stop

Parameter	Value	Unit
Acceleration	3000.00	r/min/s
Deceleration	3000.00	r/min/s
Jerk	300000.00	rpm/s ²
Deceleration (system stop AC1/AC3)	600.00	r/min/s
Jerk (system stop AC1/AC3)	600000.00	rpm/s ²
Deceleration (stop ramp)	12000.00	r/min/s
Jerk (stop ramp)	600000.00	rpm/s ²
Base value acceleration	600.00	r/min/s
Base value deceleration	600.00	r/min/s
Base value velocity	3000.00	r/min
Base value velocity (PLC, motorside)	3000.00	r/min

13.1.2 Execute the Logic in PLC using EPD

Step – 1:

Few customer uses Ladder language predominantly. (Sample Code program is written in ladder language)



Refer Sample Code program for viewing the logic.

1. Under Scan Program, [Program block] => [Customer] defined by name “Sequence”.
2. List of Device Label to modify values in Watch 1.
3. Follow the instruction given in [chapter 7](#) to add EPD

13.1.3 Procedure to control in Torque Mode using PTP_Drive_Festo_EIP function block (Refer [Chapter 6.4.9](#))

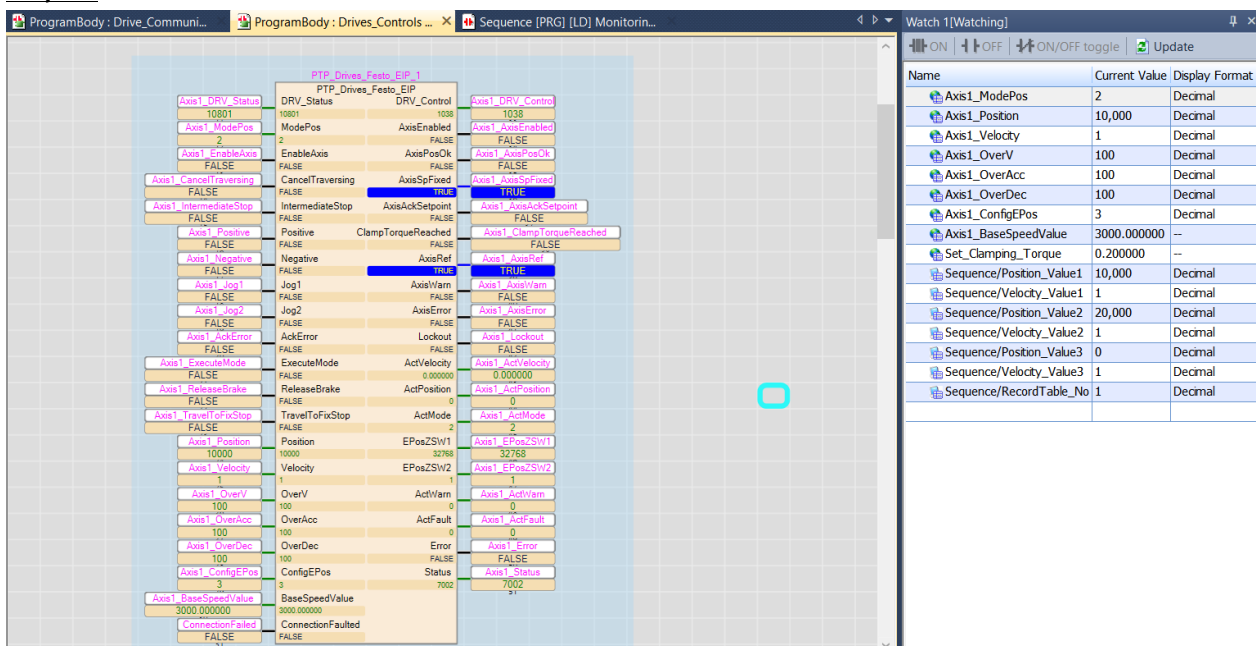
- Using a positioning mode absolute or relative (ModePos = 1 or 2 (positioning absolute/relative))



Note

- Sample Code is for your reference. It's not recommended to follow exact sequence.
- User can take this as reference to write own logic.
- Verify the Position and Velocity value based on the Liner/Rotative Axis. Correct Position and Velocity to be entered in user defined logic or even while using sample code.

Step – 1



At first scan of PLC, the PtP_Drives_Festo_EIP FB shows an “Error” and “Lockout”.

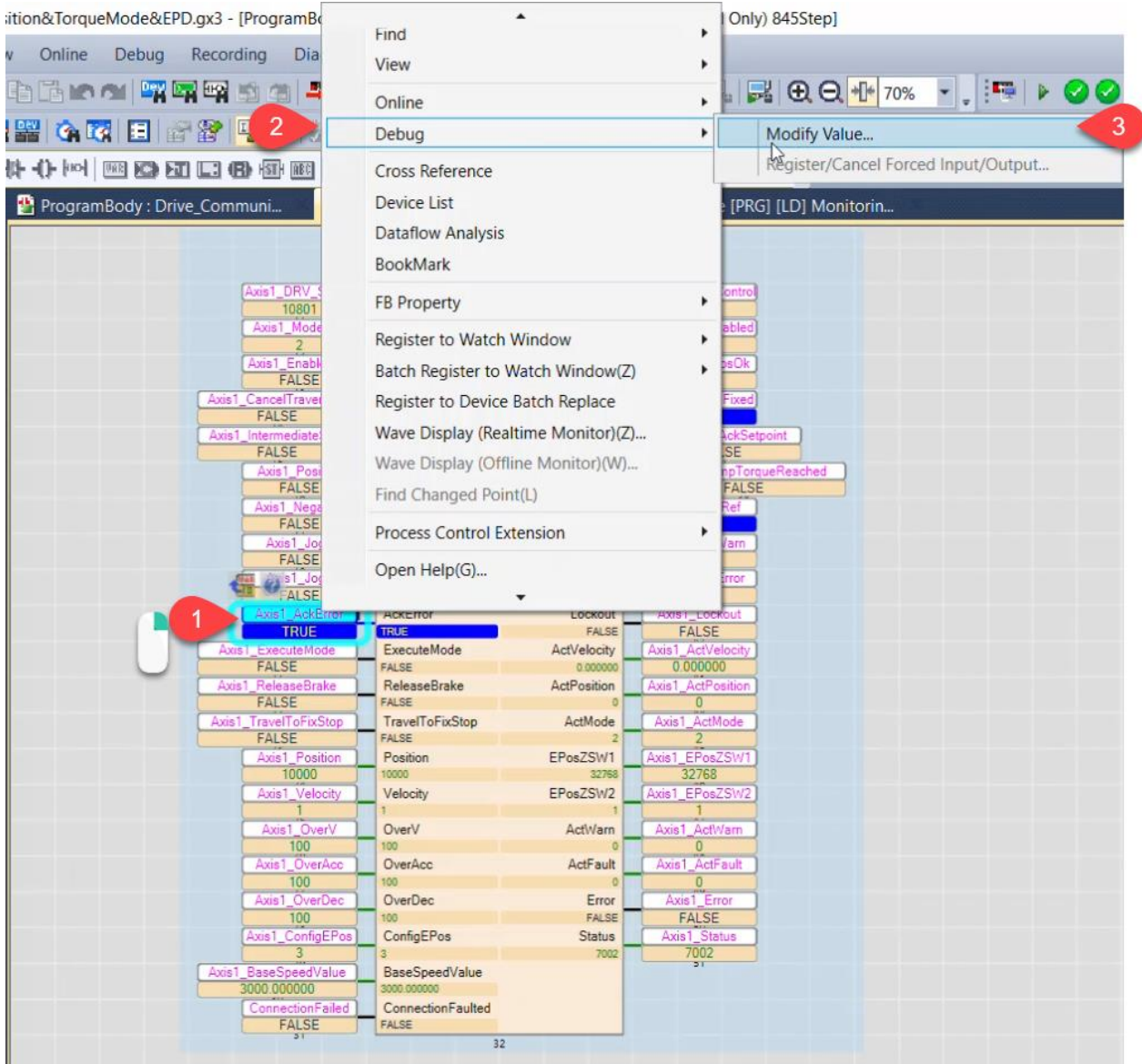
- To clear the Error and Lockout follow the steps below.

Enter the value in

- Axis1_ModePos := 5 [Home to Current Position [ModePos – 5]]
- OverV := 100
- OverAcc := 100
- OverDec := 100
- Axis1_ConfigEPos := 3 or 15. (Refer [Chapter 5.1.2](#))
- Axis1_BaseSpeedValue := 3000.0

After entering the value PtP_Drives_Festo FB does not shows an “Error” and “Lockout”.

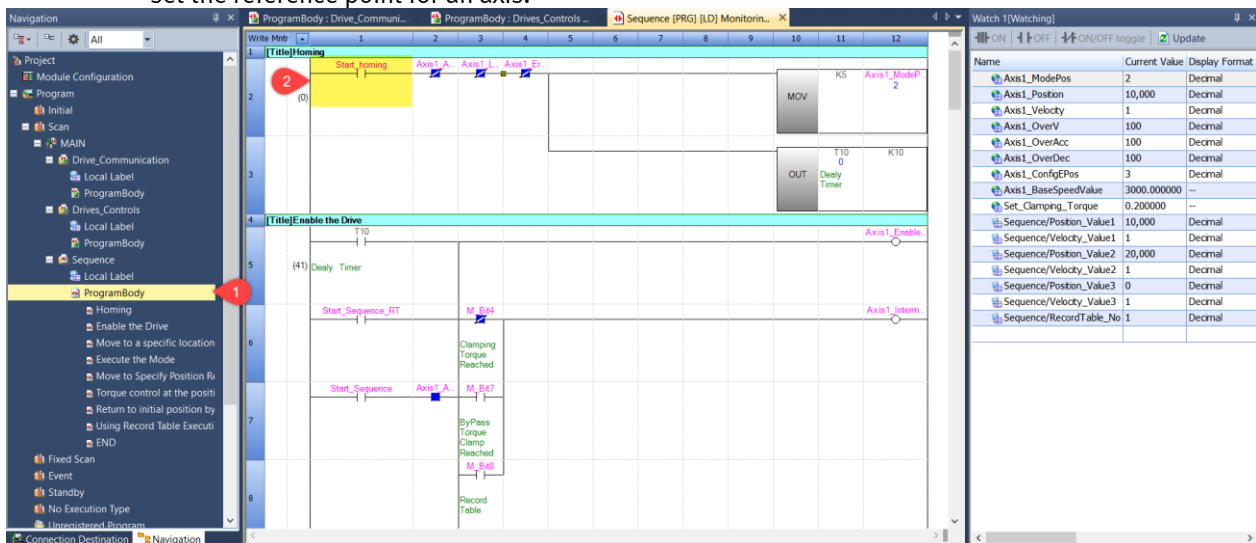
In case if Error Bit is True, see the below given image to enable “Axis1_AckError” bit.



- Right-Click [Axis1_AckError] => [Debug] => [Modify Value]
A new window “Modify Value” will be opened.
Or
Press “Shift+Enter” button to modify value.
- Go to Debug to modify value.
- Click “Modify value” button to change the Input state, or to change the Input Value.

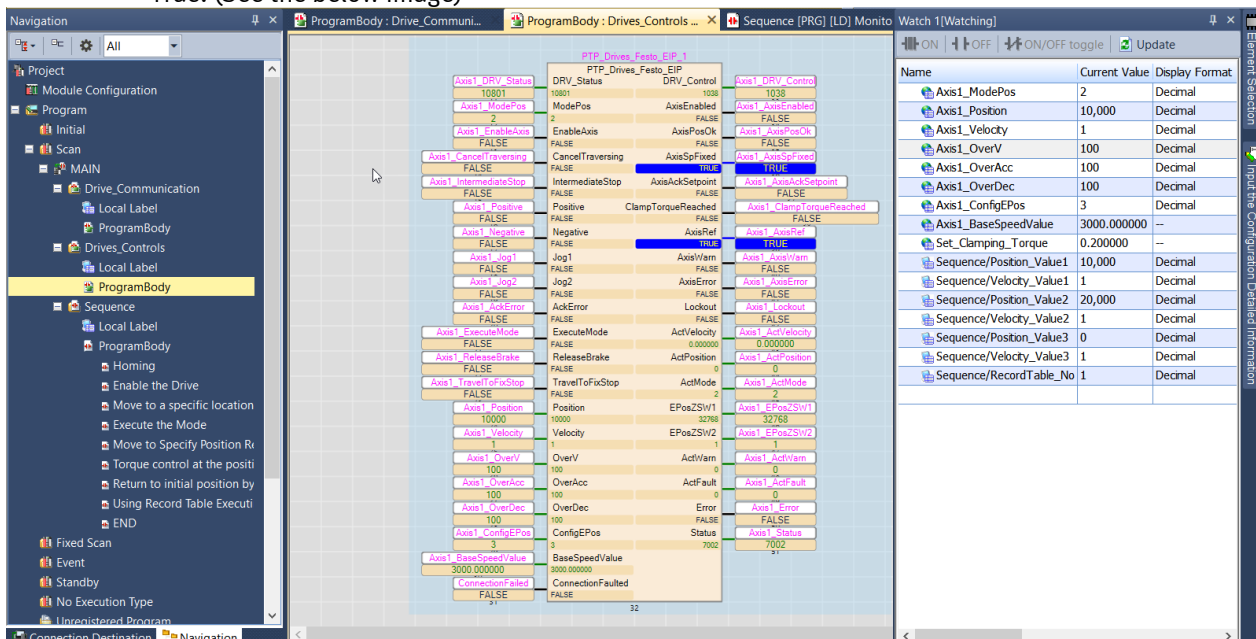
Step – 2:

- Check “**AxisRef**” status tag is True. If not true user could also use other modes like mode 4 or 5 to set the reference point for an axis.



- Expand [Scan Program] => [Main] => [Sequence] => [ProgramBody]
A program will be opened in new window.
- Select the “**Start_Homing**” button to “**ON**” it.
- Either use short cut “**Shift+Enter**” key will toggle the state of Start_Homing Button or
Right-Click [Axis1_AckError] => [Debug] => [Modify Value]
- Click “**ON**”/”**OFF**” button to modify value using Modify Value pop-up window.

- After Toggling the “**Start_Homing**” the axis will be referenced. Check “**AxisRef**” status tag will be True. (See the below image)



- Select the “**Start_Homing**” button to “**OFF**” it.

Step – 3 :

- Set the **“Clamping torque”** as per user defined, this is taken from the EPD.

**Note**

- If Clamping Torque is not set then, Travel to fixed stop won't work. As soon sequence is executed the **“ClampTorqueReached”** Bit will be True.
- Axis position won't change and Drive will go to Error mode.

- Expand [Scan Program] => [Main] => [Sequence] => [ProgramBody]
A program will be opened in new window.
- Select the **“Start_Sequence”** button to **“ON”** it.
- Either use short cut **“Shift+Enter”** key will toggle the state of Start_sequence Button or

Right-Click [Start_Sequence] => [Debug] => [Modify Value]

- Click **“ON”/“OFF”** button to modify value using Modify Value pop-up window.

The screenshot displays the Mitsubishi GX Developer interface. The main window shows the ladder logic for the PTP_Drives_Festo_EIP_1 program. The Watch window on the right shows the current values of various variables. The 'Sequence/Start_Sequence' variable is highlighted in blue and set to TRUE.

Name	Current Value	Display Format	Data Type
Axis1_ModePos	2	Decimal	Do
Axis1_Position	10,000	Decimal	Do
Axis1_Velocity	1	Decimal	Do
Axis1_OverV	100	Decimal	Do
Axis1_OverAcc	100	Decimal	Do
Axis1_OverDec	100	Decimal	Do
Axis1_ConfigEPos	3	Decimal	Do
Axis1_BaseSpeedValue	3000.000000	--	FL
Set_Clamping_Torque	0.200000	--	FL
Sequence/Position_Value1	10,000	Decimal	Do
Sequence/Velocoty_Value1	1	Decimal	Do
Sequence/Position_Value2	20,000	Decimal	Do
Sequence/Velocoty_Value2	1	Decimal	Do
Sequence/Position_Value3	0	Decimal	Do
Sequence/Velocoty_Value3	1	Decimal	Do
Sequence/RecordTable_No	1	Decimal	Do
Sequence/Start_Sequence_RT	FALSE	BIN	Bit
Sequence/Start_Sequence	TRUE	BIN	Bit

Below sequence will be executed after **“Start_Sequence”** button to **“ON”**

- Move to a specific location by position control.
 - Set the **“ModePos”** value to 2 & check the output tag **“ActMode”** has the same value as ModePos.
 - Enable the axis by toggling **“EnableAxis”** input tag.
 - “CancelTraversing”** and **“IntermediateStop”** input tags must be **TRUE**.
 - Enter the Position command value in **“Position”** tag & Velocity command value in **“Velocity”** tag. Give the absolute position value. The drive will decide forward & reverse direction based on its current position.
 - Toggle the **“ExecuteMode”** command tag (Rising Edge) to start a motion task using the configured parameters (Every new position task would require a toggle of the **“ExecuteMode”** command tag).
 - The **“AxisAckSetpoint”** status tag will be true to acknowledge the task has been started.
 - If the task is successfully completed, it is acknowledge by rising edge of **“AxisPosOk”** status variable. Also if task is unsuccessful **“AxisError”** output tag will be **TRUE**.
 - “AxisSpFixed”** Status variable indicates the axis mobility status . If the axis is standby the **“Axis-SpFixed”** status variable will be high.
- Move with torque control at the position.
 - Stroke limit function applied in torque control - (Optional as per user requirement)
(For example, if the stroke limit is set to 10mm, the torque control stops when it reaches 10mm during torque control)

Travel to fixed Stop using Absolute Positioning Mode

- Enable the axis by toggling “**TravelToFixStop**” input tag.
 - Set the “**ModePos**” value to 2 & check the output tag “**ActMode**” has the same value as ModePos.
 - Enable the axis by toggling “**EnableAxis**” input tag.
 - “**CancelTraversing**” and “**IntermediateStop**” input tags must be **TRUE**.
 - Enter the Position command value in “**Position**” tag & Velocity command value in “**Velocity**” tag. Give the absolute position value. The drive will decide forward & reverse direction based on its current position.
 - Toggle the “**ExecuteMode**” command tag (Rising Edge) to start a motion task using the configured parameters (Every new position task would require a toggle of the “**ExecuteMode**” command tag).
 - The “**AxisAckSetpoint**” status tag will be true to acknowledge the task has been started.
 - If the task is successfully completed, it is acknowledge by rising edge of “**AxisPosOk**” status variable. Also if task is unsuccessful “**AxisError**” output tag will be **TRUE**.
 - “**AxisSpFixed**” Status variable indicates the axis mobility status . If the axis is standby the “**Axis-SpFixed**” status variable will be high.
 - The “**ClampTorqueReached**” status tag will be true to acknowledge the “**Clamping torque**” has been reached.
3. Return to initial position by position control after press.
- Set the “**ModePos**” value to 2 & check the output tag “**ActMode**” has the same value as ModePos.
 - Enable the axis by toggling “**EnableAxis**” input tag.
 - “**CancelTraversing**” and “**IntermediateStop**” input tags must be **TRUE**.
 - Enter the Position command value in “**Position**” tag & Velocity command value in “**Velocity**” tag. Give the absolute position value. The drive will decide forward & reverse direction based on its current position.
 - Toggle the “**ExecuteMode**” command tag (Rising Edge) to start a motion task using the configured parameters (Every new position task would require a toggle of the “**ExecuteMode**” command tag).
 - The “**AxisAckSetpoint**” status tag will be true to acknowledge the task has been started.
 - If the task is successfully completed, it is acknowledge by rising edge of “**AxisPosOk**” status variable. Also if task is unsuccessful “**AxisError**” output tag will be **TRUE**.
 - “**AxisSpFixed**” Status variable indicates the axis mobility status . If the axis is standby the “**Axis-SpFixed**” status variable will be high.

Name	Current Value	Display Format	Da
Axis1_ModePos	2	Decimal	Wc
Axis1_Position	0	Decimal	Do
Axis1_Velocity	2	Decimal	Do
Axis1_OverV	100	Decimal	Wc
Axis1_OverAcc	100	Decimal	Wc
Axis1_OverDec	100	Decimal	Wc
Axis1_ConfigEPos	3	Decimal	Do
Axis1_BaseSpeedValue	3000.000000	--	FL
Set_Clamping_Torque	0.200000	--	FL
Sequence/Position_Value1	10,000	Decimal	Do
Sequence/Velocoty_Value1	1	Decimal	Do
Sequence/Position_Value2	20,000	Decimal	Do
Sequence/Velocoty_Value2	1	Decimal	Do
Sequence/Position_Value3	0	Decimal	Do
Sequence/Velocoty_Value3	2	Decimal	Do
Sequence/RecordTable_No	1	Decimal	Do
Sequence/Start_Sequence_RT	FALSE	BIN	Bit
Sequence/Start_Sequence	TRUE	BIN	Bit

13.2 Method 2 - Using FAS Record Table and PTP_Drives_Festo Modes of Operation



Note

- Refer the [Chapter – 4](#) for Integrating the FB's in PLC program.



Information

PTP_Drives_Festo_EIP FB is written in Structure Text language, it can be called in Ladder and FBD/LD Program Language

13.2.1 Set the Task/Sequence in Record Table

Step – 1:

Using Record Table, instead writing the entire logic in PLC and then use PTP_Drives_Festo_EIP FB to execute Record Table.

- Using a Traversing/Record Selection Mode [ModePos – 6]



Note

- Sample Code is for your reference. It's not recommended to follow exact sequence.
- User can take this as reference to write own logic.
- Verify the Position and Velocity value based on the Liner/Rotative Axis. Correct Position and Velocity to be entered in user defined FAS Record Table

Before downloading the program in PLC, first define sequence in FAS under Record Table

Step – 2

Refer [chapter 10.1](#) for setting the Record Table in FAS.

Record Table										
Delete All										
1	1	Homing	Record type Homing (3)	Method Current position (37)						
2	2	Move to Position	Record type Position (5)	Type Positioning absolute (0)	Target position 100.00 mm	Velocity 0.02 m/s	Acceleration 0.05 m/s ²	Deceleration 0.05 m/s ²	Jerk 1.50 m/s ³	End velocity 0.00 m/s
3	3	Position Reached jump to Torque Mode	Condition MC (15)							
4	3	Torque Mode	Record type Torque (7)	Target torque 0.30 Nm	Torque rise ramp 0.10 Nm/s	Limit speed 0.01 m/s	Activation stroke limit ☐	Stroke limit negative -120.00 mm	Stroke limit positive 120.00 mm	
5	4	Target Torque Reached jump to Position Mode	Condition Target torque reached (3)							
6	4	Initial Position	Record type Position (5)	Type Positioning absolute (0)	Target position 0.00 mm	Velocity 0.05 m/s	Acceleration 5.00 m/s ²	Deceleration 5.00 m/s ²	Jerk 150.00 m/s ³	End velocity 0.00 m/s
Add Record Set										

1. First sequence is to do Homing with current position(37)
2. Second Sequence is to Move to a specific location by Position Absolute (0)
Enter parameter such as
 - Target Position to be reached
 - Velocity at which position to be reached,
 - Acceleration
 - Deceleration
 - Jerk
 - End Velocity
3. Position Reached jump to Torque Mode - (Optional for user to define if required).
4. Third Sequence is to move in Torque Mode.
 - Stroke limit function applied in torque control
(For example, if the stroke limit is set to 120mm, the torque control stops when it reaches 120mm during torque control)
Enter parameter such as
 - Target Torque (User needs to enter the Torque considering Dynamic Value to achieve Target Torque – (Refer [Chapter 13.1, Step – 1](#), Dynamic Value image))
 - Target rise ramp should be less than Target Ramp.
 - Limit Speed – Speed at which axis should move to achieve the target torque.
 - Select Activation stroke limit (User requirement - If target torque not reached under the stroke limit then “AxisError” output tag will be TRUE).
 - Stroke limit negative
 - Stroke limit positive
5. Target Torque reached move from Torque Mode to Position Mode - (Optional for user to define if required).
 - User should add “Stop Ramp” record table after Target Torque reached, to stop at that sequence, else can directly jump to next sequence.
6. Fourth Sequence is to Move to a specific location by Position Absolute (0)
Enter parameter such as
 - Target Position to be reached
 - Velocity at which position to be reached,
 - Acceleration
 - Deceleration
 - Jerk
 - End Velocity

**Note**

In this method –

- Axis won't stop even after achieving the Target torque if Activation stroke limit is not enabled.
- To stop axis when Target torque reached is True, following parameters to be added in record table. “Stop Ramp” task must be added in record table.

13.2.2 Procedure to control in Traversing/Record Table using PTP_Drive_Festo_EIP function block (Refer Chapter 6.4.9)

Step – 1

The screenshot displays the Mitsubishi GX Developer interface. The main window shows the configuration for the PTP_Drive_Festo_EIP function block. The configuration is organized into three columns: Inputs, Function Block, and Outputs. The Watch window on the right shows the current values of the variables used in the function block.

Name	Current Value	Display Format
Axis1_ModePos	2	Decimal
Axis1_Position	10,000	Decimal
Axis1_Velocity	1	Decimal
Axis1_OverV	100	Decimal
Axis1_OverAcc	100	Decimal
Axis1_OverDec	100	Decimal
Axis1_ConfigEPos	3	Decimal
Axis1_BaseSpeedValue	3000.000000	--
Set_Clamping_Torque	0.200000	--
Sequence/Position_Value1	10,000	Decimal
Sequence/Velocoty_Value1	1	Decimal
Sequence/Position_Value2	20,000	Decimal
Sequence/Velocoty_Value2	1	Decimal
Sequence/Position_Value3	0	Decimal
Sequence/Velocoty_Value3	1	Decimal
Sequence/RecordTable_No	1	Decimal

At first scan of PLC, the PTP_Drive_Festo_EIP FB shows an “Error” and “Lockout”.

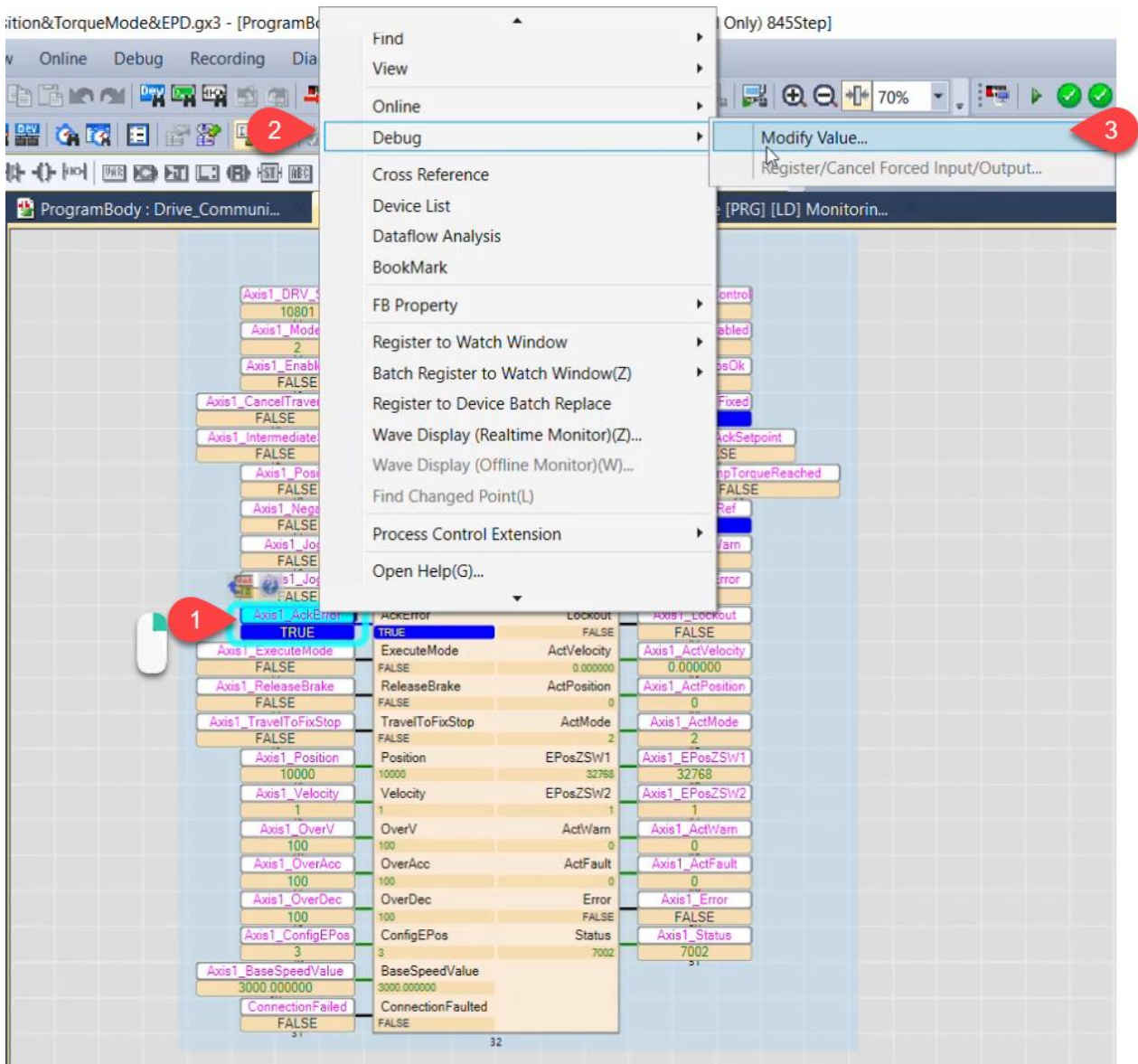
- To clear the Error and Lockout follow the steps below.

Enter the value in

- Axis1_ModePos := 5 [Home to Current Position [ModePos – 5]]
- OverV := 100
- OverAcc := 100
- OverDec := 100
- Axis1_ConfigEPos := 3 or 15. (Refer Chapter 5.1.2)
- Axis1_BaseSpeedValue := 3000.0

After entering the value PTP_Drive_Festo_EIP FB does not shows an “Error” and “Lockout”.

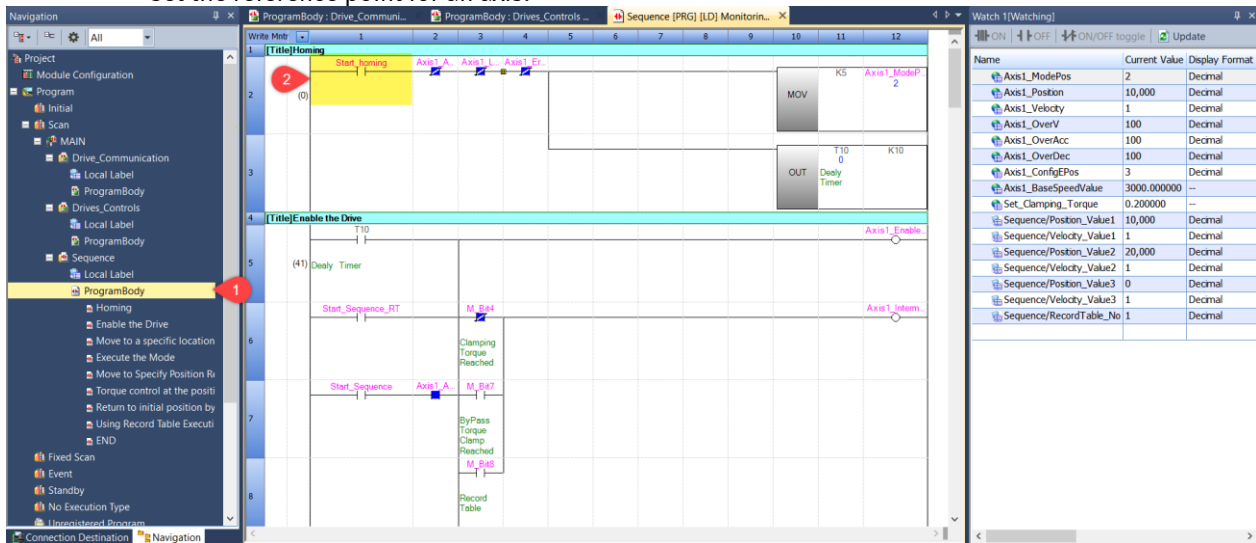
In case if Error Bit is True, see the below given image to enable “Axis1_AckError” bit.



1. Right-Click  [Axis1_AckError] => [Debug] => [Modify Value]
A new window **"Modify Value"** will be opened.
Or
Press **"Shift+Enter"** button to modify value.
2. Go to Debug to modify value.
3. Click **"Modify value"** button to change the Input state, or to change the Input Value.

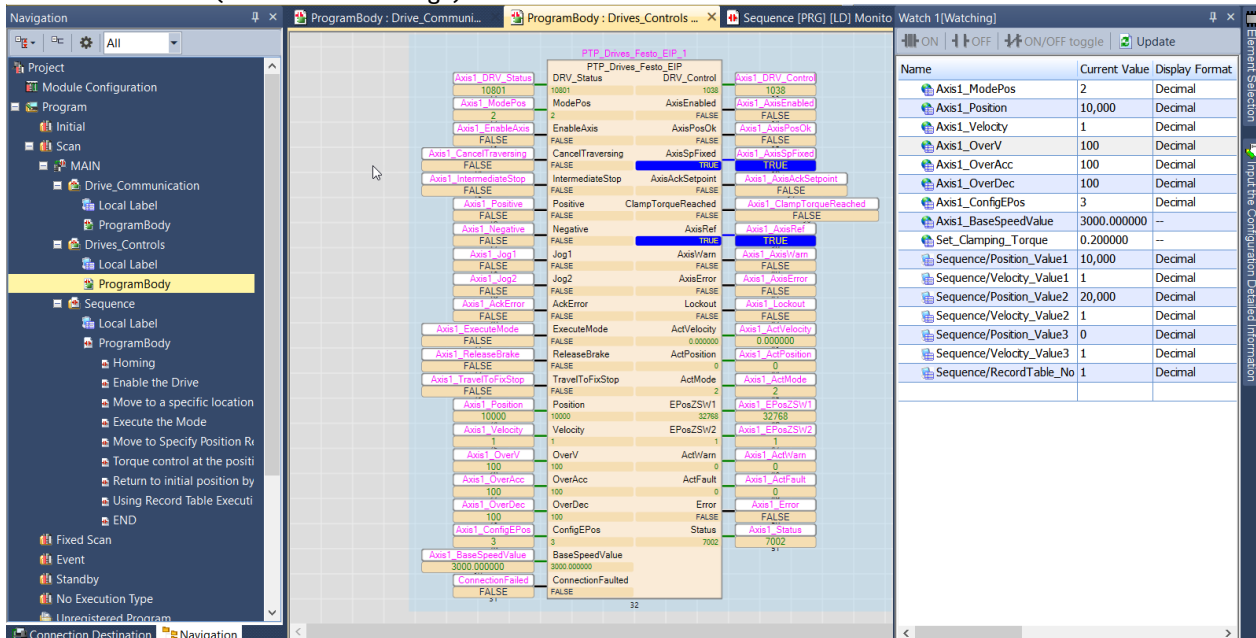
Step – 2:

- Check **“AxisRef”** status tag is True. If not true user could also use other modes like mode 4 or 5 to set the reference point for an axis.



- Expand [Scan Program] => [Main] => [Sequence] => [ProgramBody]
A program will be opened in new window.
- Select the **“Start_Homing”** button to **“ON”** it.
- Either use short cut **“Shift+Enter”** key will toggle the state of Start_Homing Button or
Right-Click [Axis1_AckError] => [Debug] => [Modify Value]
- Click **“ON”/“OFF”** button to modify value using Modify Value pop-up window.

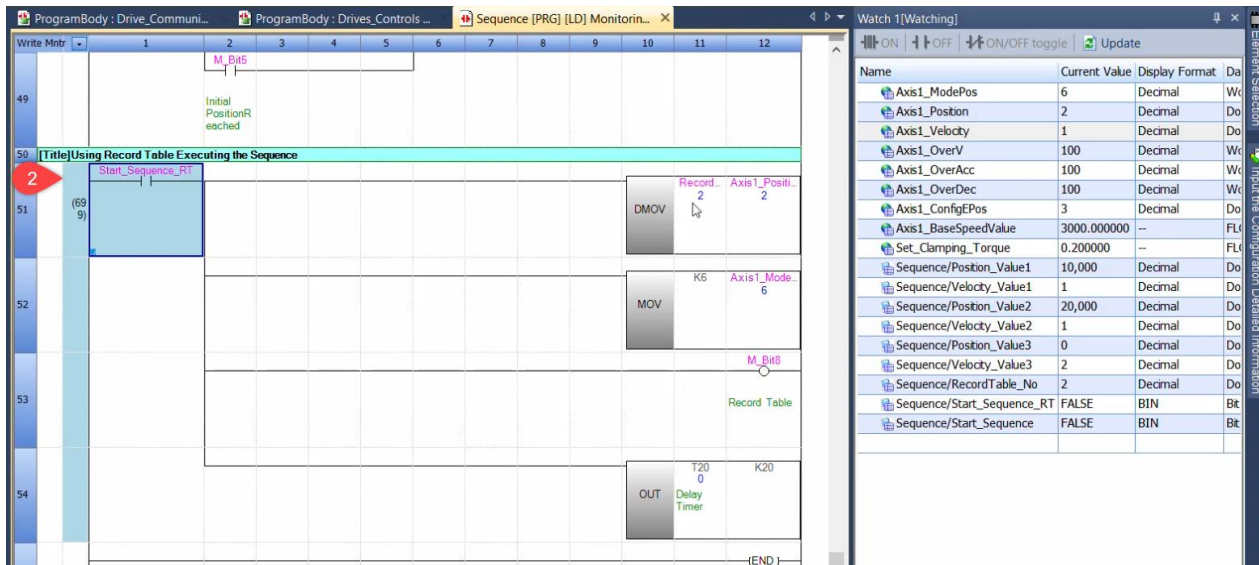
- After Toggling the **“Start_Homing”** the axis will be referenced. Check **“AxisRef”** status tag will be True. (See the below image)



- Select the **“Start_Homing”** button to **“OFF”** it.

Step – 3:

Below given logic is for executing the task in Record Table.



1. Select the record table by entering the table number (highlighted in below image) in “**Position**” command tag. The value must be between 0 to 127. (Record Table Number)

Record Table									
Delete All									
1	Homing	Record type	Method						
		Homing (3)	Current position (37)						
2	Move to Position	Record type	Type	Target position	Velocity	Acceleration	Deceleration	Jerk	End velocity
		Position (5)	Positioning absolute (0)	100.00 mm	0.02 m/s	0.05 m/s ²	0.05 m/s ²	1.50 m/s ³	0.00 m/s
3	Position Reached jump to Torque Mode	Condition							
		MC (15)							
3	Torque Mode	Record type	Target torque	Torque rise ramp	Limit speed	Activation stroke limit	Stroke limit negative	Stroke limit positive	
		Torque (7)	0.30 Nm	0.10 Nm/s	0.01 m/s	☐	-120.00 mm	120.00 mm	
4	Target Torque Reached jump to Position Mode	Condition							
		Target torque reached (3)							
4	Initial Position	Record type	Type	Target position	Velocity	Acceleration	Deceleration	Jerk	End velocity
		Position (5)	Positioning absolute (0)	0.00 mm	0.05 m/s	5.00 m/s ²	5.00 m/s ²	150.00 m/s ³	0.00 m/s
Add Record Set									

2. Select the “**Start_Sequence_RT**” button to “**ON**” it.
 - a. Either use short cut “**Shift+Enter**” key will toggle the state of Start_Sequence_RT Button or Right-Click  [Start_Sequence_RT] => [Debug] => [Modify Value]
 - b. Click “**ON**”/”**OFF**” button to modify value using Modify Value pop-up window.

Below sequence will be executed after “**Start_Sequence_RT**” button to “**ON**”

- Set the “**ModePos**” value to 6 & check the output tag “**ActMode**” has the same value as ModePos.
- Enable axis by toggling “**EnableAxis**” command tag.
- Check “**AxisRef**” status tag is True. If not true user could also use other modes like mode 4 or 5 to set the reference point for an axis.
- “**CancelTraversing**” and “**IntermediateStop**” command tags must be in **TRUE** state.
- Enter the percentage of Override acceleration, Override deceleration & Override velocity in “**Over-Acc**”, “**OverDec**” & “**OverV**” respective input tags.
- The status tags “**AxisEnabled**” & “**AxisRef**” must be **TRUE**. Also status tag “**Error**” must be false.
 1. Toggle the “**ExecuteMode**” command tag (Rising Edge) to start a motion task using the configured parameters (Every new position task would require a toggle of the “**ExecuteMode**” command tag).
 2. The “**AxisAckSetpoint**” status tag will be true to acknowledge the task has been started.

3. If task is successfully completed, it is acknowledge by rising edge of “**AxisPosOk**” OUTPUT variable. Also if the is task unsuccessful “**AxisError**” output variable will be **TRUE**.
4. “**AxisSpFixed**” Status variable indicates the axis mobility status . If the axis is standby the “**AxisSpFixed**” status variable will be high.

ProgramBody : Drive_Communi... ProgramBody : Drives_Controls ... Sequence [PRG] [LD] Monitorin... Watch 1[Watching]

PTP_Drives_Festo_EIP_1

Axis1_DRV_Status 47922
Axis1_ModePos 6
Axis1_EnableAxis TRUE
Axis1_CancelTraversing TRUE
Axis1_IntermediateStop TRUE
Axis1_Positive FALSE
Axis1_Negative FALSE
Axis1_Jog1 FALSE
Axis1_Jog2 FALSE
Axis1_AckError FALSE
Axis1_ExecuteMode TRUE
Axis1_ReleaseBrake FALSE
Axis1_TravelToFixStop FALSE
Axis1_Position 2
Axis1_Velocity 1
Axis1_OverV 100
Axis1_OverAcc 100
Axis1_OverDec 100
Axis1_ConfigEPos 3
Axis1_BaseSpeedValue 3000.000000
ConnectionFailed FALSE

PTP_Drives_Festo_EIP_1
DRV_Status 47922
ModePos 6
EnableAxis TRUE
CancelTraversing TRUE
IntermediateStop TRUE
Positive FALSE
Negative FALSE
Jog1 FALSE
Jog2 FALSE
AckError FALSE
ExecuteMode TRUE
ReleaseBrake FALSE
TravelToFixStop FALSE
Position 2
Velocity 1
OverV 100
OverAcc 100
OverDec 100
ConfigEPos 3
BaseSpeedValue 3000.000000
ConnectionFaulted FALSE

Axis1_DRV_Control 1151
Axis1_AxisEnabled TRUE
Axis1_AxisPosOk TRUE
Axis1_AxisSpFixed TRUE
Axis1_AxisAckSetpoint TRUE
Axis1_ClampTorqueReached FALSE
Axis1_AxisRef TRUE
Axis1_AxisWarn FALSE
Axis1_AxisError FALSE
Axis1_Lockout FALSE
Axis1_ActVelocity 61.230000
Axis1_ActPosition 99528
Axis1_ActMode 6
Axis1_EPosZSW1 8194
Axis1_EPosZSW2 32784
Axis1_ActWarn 0
Axis1_ActFault 0
Axis1_Error FALSE
Axis1_Status 7002

Watch 1[Watching]

Name	Current Value	Display Format	Da
Axis1_ModePos	6	Decimal	Wc
Axis1_Position	2	Decimal	Do
Axis1_Velocity	1	Decimal	Do
Axis1_OverV	100	Decimal	Wc
Axis1_OverAcc	100	Decimal	Wc
Axis1_OverDec	100	Decimal	Wc
Axis1_ConfigEPos	3	Decimal	Do
Axis1_BaseSpeedValue	3000.000000	--	FL
Set_Clamping_Torque	0.200000	--	FL
Sequence/Position_Value1	10,000	Decimal	Do
Sequence/Velocit_Value1	1	Decimal	Do
Sequence/Position_Value2	20,000	Decimal	Do
Sequence/Velocit_Value2	1	Decimal	Do
Sequence/Position_Value3	0	Decimal	Do
Sequence/Velocit_Value3	2	Decimal	Do
Sequence/RecordTable_No	2	Decimal	Do
Sequence/Start_Sequence_RT	TRUE	BIN	Bit
Sequence/Start_Sequence	FALSE	BIN	Bit