AC500 PLC

Application Note

Getting started with AC500 v3 Motion.

AN500

Contents

1 INTRODUCTION

Introduction

This application notes details in Brief how to use Automation Builder for v3 AC500 PLCs (AB v2.7.0 onwards) to define the hardware setup suitable for motion control of a single ABB Servo Drive and how to then write a simple AC500 v3 PLC program to perform motion on these drives.

A more detailed document exists here which goes into more detail on all of the steps mentioned as well as some other adjacent functions and motion related guides not covered here.

Pre-requisites

- The latest Software Tool versions can be downloaded here
- The latest Drive firmware can be downloaded here
- The PLC program was written using **Automation Builder v2.7** or later which can be downloaded here
- An installed (and licensed) copy of the ABB PLCopen v3 motion control library
- One of the following AC500 PLC processors:

AC500 Motion Kit: PM5630-2ETH, PM5650-2ETH, PM5670-2ETH, PM5675-2ETH

AC500eco Motion Kit: PM5032-ETH, PM5052-ETH, PM5072-2ETH, PM5082-2ETH

- PLC processors should be running firmware version 3.2 or later – use an SD card initially if the PLC has been shipped with v3.0 firmware and then the 'Update Firmware' facility within Automation Builder to further update the processor as required.
- Ethernet cable to connect the PLC EtherCAT coupler to the drive

The PLC Hardware required to run this example is the **ABB AC500 V3 PLC range**. These come in two main types the AC500 and AC500eco. All AC500 V3 CPU's have a type of designation with 4 numbers after the PM part of the code e.g. PM**5072**-T-2ETH, unlike the previous generation CPU's which have only 3 e.g. PM564-ETH.

2 SERVO DRIVES PREPERATION

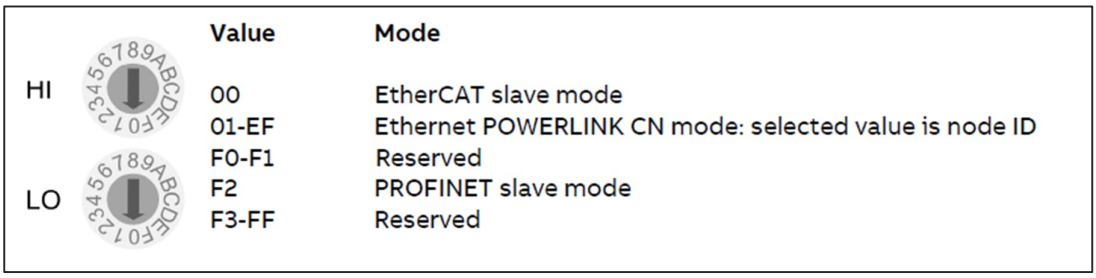
2.1 Setting up ABB Servo Drives for use with EtherCAT Master

This section assumes that you have already commissioned the drive. i.e., You have been through the Commissioning process and have tuned the control loops for the drive.

	Details on commissioning the e180 and e190 drives can be found in the application note AN250 which can be downloaded here. A similar Application Note is under development for e530 drive and will be available soon here.

2.1.1 EtherCAT Set up Process for e180 and e190

1. The first step here is to check that the rotary switches are set correctly. As shown below these can define multiple modes of operation and should be set to 00H for EtherCAT Mode



2. Next, we must define the correct ControlRefSourceStartUp, this defines the initial operating mode when the drive is powered up or restarted. The control reference source Start Up can be set using several methods.

a. Parameter viewer – Parameter Family = 'Configuration' > Parameter = 'CONTROL REF SOURCE STARTUP > Setting = RT_ETHERNET_(CiA402)

b. Command Line - using command = 'CONTROLREFSOURCESTARTUP(0)= crsRT_ETHERNET_402'

c. Wizard – From main view select '**Operating Mode**' Then set ControlRefSource = RT Ethernet (CiA 402):

press the 'Finish' Button

3. The final step is always to Save the parameters using the save ICON in the tool bar.

2.1.2 EtherCAT Set up Process for e530 EC Drives

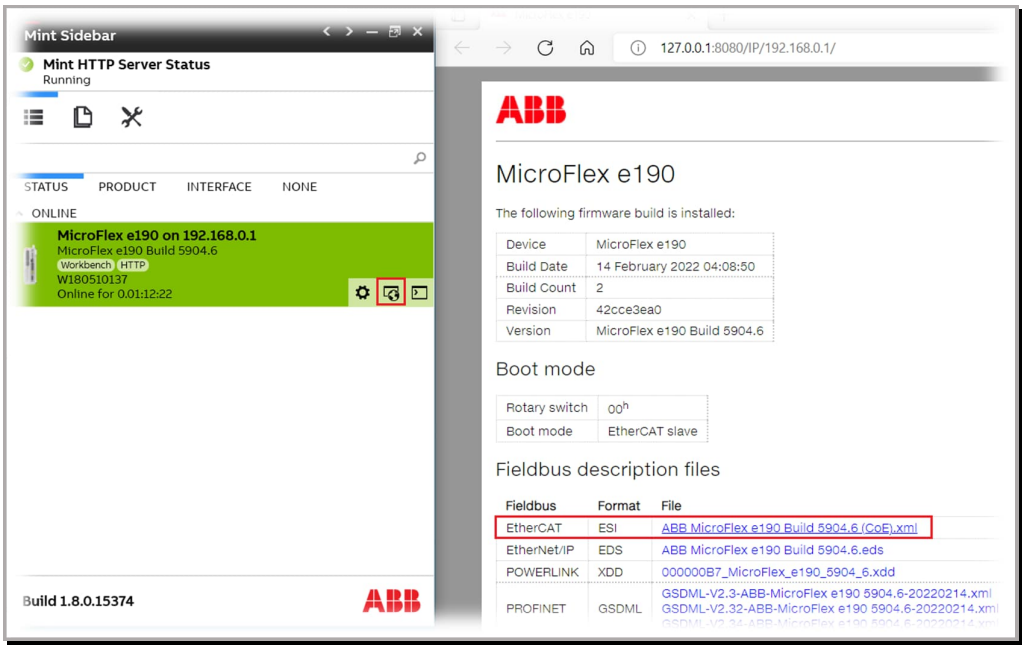
For the e530 EC drives they are ready to run out of the box with no special commissioning steps to prepare for EtherCAT control.

2.2 Getting the xml file

2.2.1 Process for e180 and e190 - Exporting the xml file from the drive.

Before starting a new PLC, project and going through the navigation of initial configuration we need to make sure that the corresponding .xml file of E190 that we need has been installed into the 'Device Repository', if not, please follow below process:

1. Open the Mint Sidebar
2. Connect to the E190 which will make it appear in the list with a green color
3. then click the 'web' icon. This will open the web server
4. Shown at the bottom of the main page a series of files associated with the installed firmware version
5. Select the EtherCAT file to initiate the .xml file download to your hard drive:



2.2.2 Process for getting the e530 EC xml file from the website.

Go to the ABB Website (Servo Products - Powering machine innovations - Low voltage AC | ABB) and download the latest version.

3 MOTION SOLUTION PROJECT

3.1 Items to consider before beginning your project.

3.1.1 Selecting the correct PLC - Introduction to the PLC capabilities

Before selecting starting your project you should select the correct PLC based on its capabilities. Below is a short introduction to AC500 and AC500eco offering.

3.1.1.1 PTO Operation

Due to differences in the performance of CPU types, there are different limits on the minimum cycle times configurable in each PLC type.

PLC Type	PM5032	PM5052	PM5072	PM5082
PTO Profiling Cycle time limits	5 ms	2 ms	1 ms*	1 ms

3.1.1.2 EtherCAT Operational limits

There are different limits for EtherCAT cycle time settings due to differences in the performance of CPU types. Minimum EtherCAT cycle time configurable in each PLC type.

PLC Type	PM5072	PM5082	PM5630	PM5650	PM5670
Min. EtherCAT master cycle time configurable (ms)	2 ms	1 ms	2 ms	1 ms	1 ms

3.1.1.3 Overall maximum configurable number of axes in each PLC type.

Apart from the limit on the minimum cycle time for each PLC type, there is a linked configuration limit on the number of axes which can be connected for each PLC type - based on the cycle time configured. Please refer to the help file for more details (PLC Automation with V3 CPUs > Libraries and solutions > Motion control library > Preconditions for the use of the libraries) and adapt the cycle time or PLC Type accordingly.

PLC Type	PM5072	PM5082	PM5630	PM5650	PM5670
Number of synchr. axis in 0.5 ms	-	-	-	-	2
Number of synchr. axis in 1 ms	-	4*	-	8	16
Number of synchr. axis in 2 ms	4	8	4	16	32
Number of synchr. axis in 4 ms	8	16	8	32	64

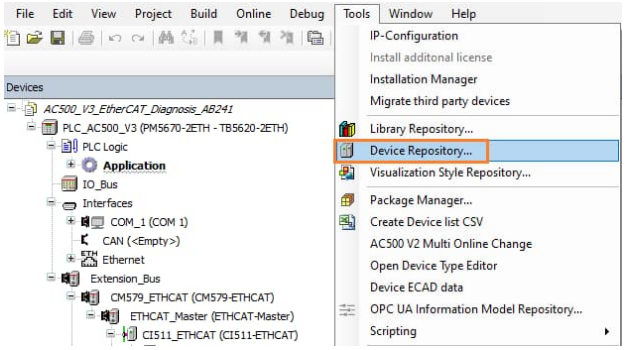
The “Motion Solution” window can be used to see how many axes are supported and how many are already used for the particular PLC type at the EtherCAT master cycle time configured by checking the slider at the bottom of the Motion Solution Wizard overview page:

3.1.2 Updating Device Description Files in the Device repository

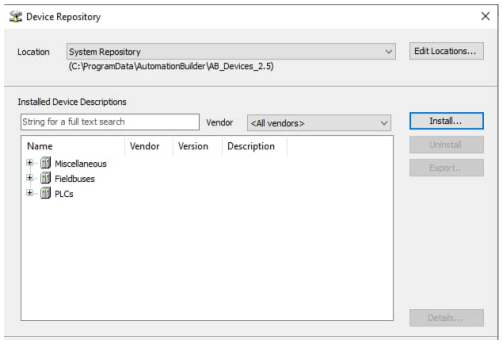
The device description files within your project should be considered, if they do not match your firmware version you might find some missing mapping or functionality. The process for adding ABB or 3rd party drives to Automation Builder device repository is the same.

To install the device description files follow these steps:

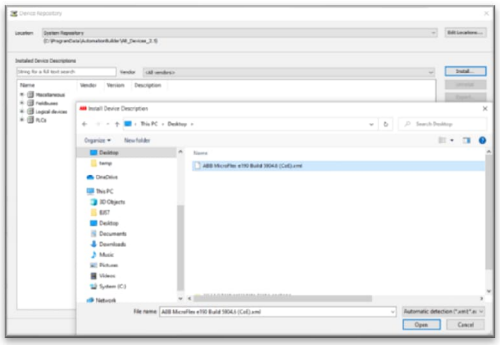
1. Click on the Device Repository under Tools menu.



2. This will open the Device Repository window, next click on 'Install' and select the device description file which needs to be installed.



3. Click 'Install⋯' and then find the location where the desired files are stored. Once selected please click 'Open'.



4. The status of the file installation will show at the bottom of the window. If successful the device will be immediately on the list of installed devices and the After successful installation, user can now add the installed device under the respective protocol configured in Automation Builder.

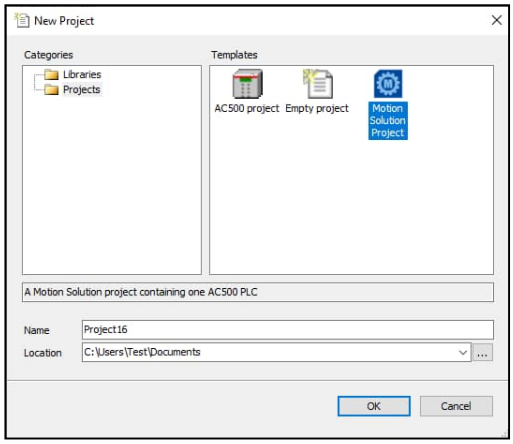
5. After this step is complete, we can use this version in new projects.

3.2 Motion Solution Project configuration

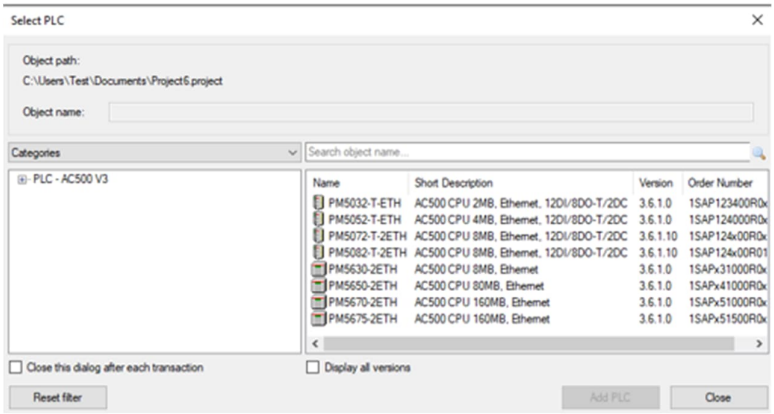
3.2.1 Creating new project

1. To Start an Automation Builder project open AB and select “New Project”.

2. Select the “Motion Solution Project” icon as shown below. Click “OK” button and a new project will be created in the specified location with the specified name.



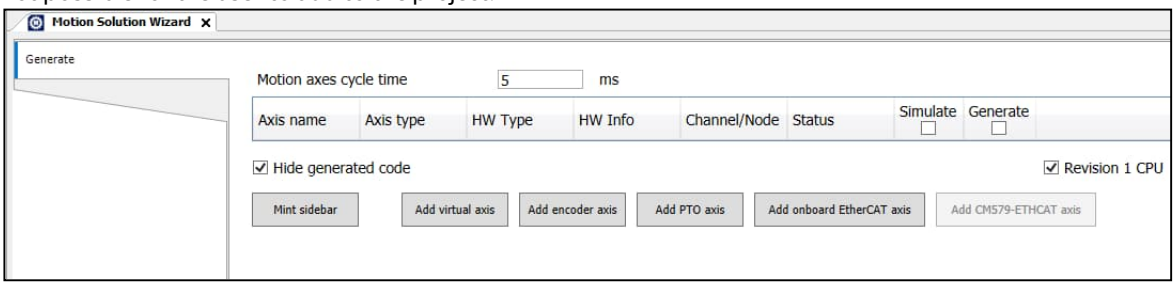
3. Add the PLC - Automation Builder will now pop up a “Select PLC” window and from here user can select one of the V3 PLC’s. For the correct selection please see section: Selecting the correct AC500 PLC type



4. In this case we will add a **‘PM5072-T-2ETH’**

5. Next click “Add PLC” button to get it added to Automation Builder hardware tree.

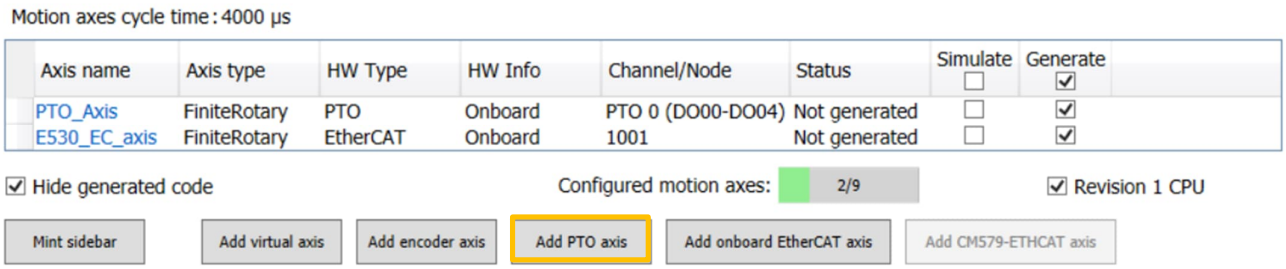
6. After creating the hardware tree, Automation Builder will launch the motion solution wizard overview page. From here user can add the axis to the application. Depending on the PLC type selected some of the axis types are not possible for the user to add to the project.



3.2.2 Adding Axes to your Project

3.2.2.1 Add a PTO axes

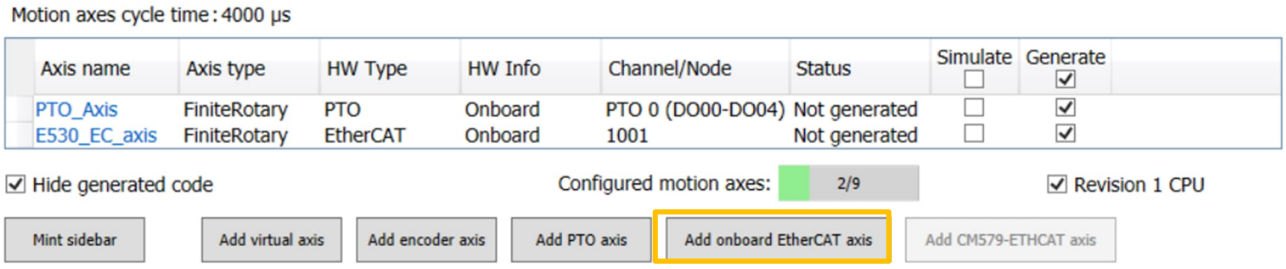
All AC500 eCo PLC's are supported with a maximum of 4 PTO axis via onboard IO channels.



To add a PTO axis you can use the button above. Its also possible by clicking on the "OnBoard_IO" object and selecting the "Add Object" option or by using the Add PTO axis button from "Motion Solution Wizard" overview page.

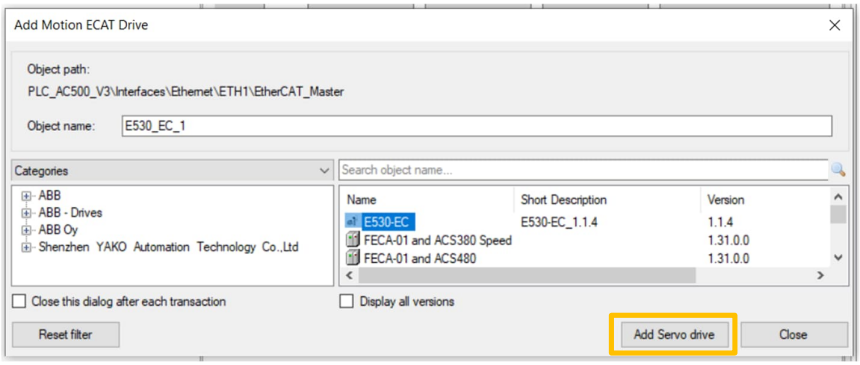
3.2.2.2 Add an EtherCAT axes

From AB2.7.0, users will have two options to realize an EtherCAT network; Using the 'AC500 and CM579-ETH' solution and the option of use AC500 eCo PM5072 and PM5082 PLC's (EtherCAT master using onboard ETH1 port.)



To add an ECAT axis you can use the button highlighted above. It is also possible by clicking on the "EtherCAT_Master" object and selecting the "Add Object" option or by using the Add PTO axis button from "Motion Solution Wizard" overview page.

After you click this, you will get to choose a drive type from the Device Repository. Here I will choose an e530-EC and then click 'Add Servo Drive'



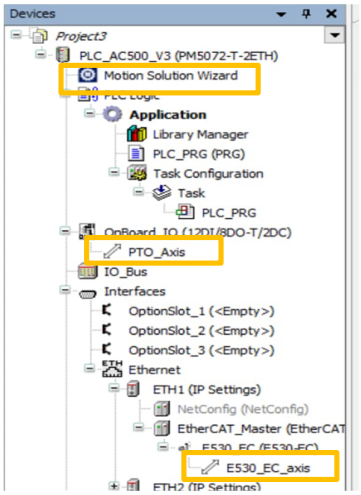
3.2.2.3 Axis Objects

Axis Objects will be shown in the menu of the Motion Solution Wizard as a list which the user can use to jump to their configuration pages and they will also be shown in the project under the hardware they are ascociated with.

Nested underneath the 'OnBoard_IO' object is the PTO Axis object.

Similarly nested underneath the 'EtherCAT master object is the e530 EtherCAT drives Axis object.

Here user can update settings as per the application requirement. Later when downloading the application, the wizard will use these settings to define the operation and scaling of this axis object.



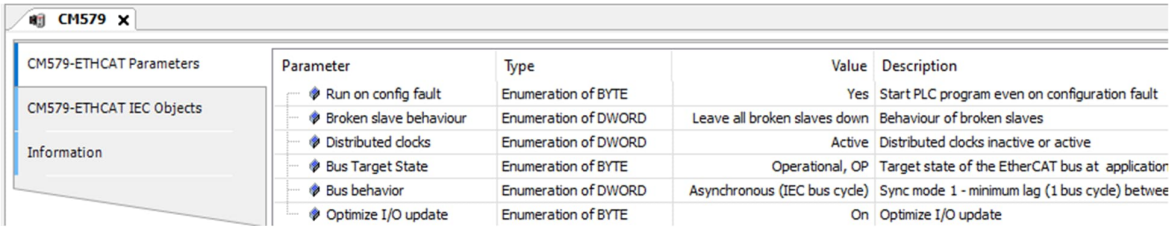
3.2.3 Configuring the EtherCAT master

EtherCAT master can be configured in AC500 CPU via CM579-ETHCAT coupler or in AC500 eCo Pro PLC's via onboard ETH1 port.

In general, all EtherCAT settings are similar in both solutions but in CM579-ETHCAT master configurator there will be more tabs than for onboard EtherCAT. Based on which EtherCAT solution is used, user needs to also use the relevant libraries for diagnosis / SDO handling etc.

CM579-ETHCAT

These are the settings specific to CM579-ETHCAT master and not valid for onboard EtherCAT. The label will normally 'CM579_ETHCAT' this can be changed by the user but is normally left as default. If so, the name will be CM579_ETHCAT'(CM579-ETHCAT)



In most cases these settings can be left at default but occasionally the user might need to change these to fit the application requirements.

- *Run on Config Fault* – As default it is set to "yes" meaning the CM579-ETHCAT will not go into error and the PLC program will not go into stop if a slave is lost, this is beneficial as the PLC is then still available to handle other follow up actions. This can be changed if the user wants a configuration where a missing slave will prevent the PLC running.
- *Distributed clocks* - As default it is set to "Active" which is a must if you want to use syncronised motion over EtherCAT.
- *Optimize I/O update* - As default it is set to "On" meaning that, consecutive I/O's are merged in one block to optimize the performance.

ETHCAT_Master (EtherCAT-Master)

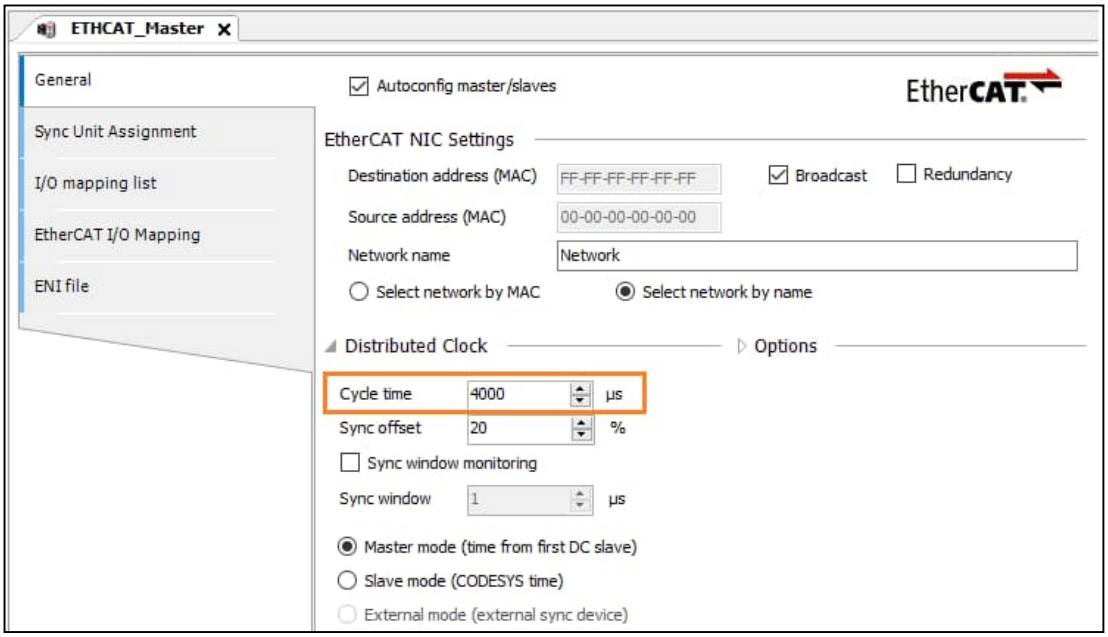
This second part of the EtherCAT master defines the specific settings that define the behavior of the EtherCAT operation and from here settings are similar in both CM579-ETHCAT and in onboard EtherCAT.

	Note: Onboard EtherCAT is only possible to configure it on PM5072 and PM5082 ETH1 port. It is recommended not to configure any other protocols on the ETH1; EtherCAT port.

The label will normally be in the format of '[name](ETHCAT-Master)' this can be changed by the user but is normally left as default. If so, the name will be **ETHCAT_Master** (ETHCAT-Master). The most important settings for this are mentioned below:

- *Auto-config master/slaves*" – As Default this is activated, this means the parameters are set automatically here in accordance with the default settings. This setting is recommended unless the user is very familiar with the setup of EtherCAT networks.

- *Cycle time* - The Default EtherCAT cycle time is 4000 µs but based on the application requirement, user can adapt the EtherCAT cycle time as as shown below.



	Please note, EtherCAT cycle time will directly influence the PLC load. If your PLC load is higher than desired, please increase the cycle time or upgrade the PLC type.

3.2.4 Configuring the EtherCAT Slave & Axis

This section explains how to configure the behavior of the Comms Slave module. From here we can define how the hardware will behave. It's important to understand the EtherCAT Slave Axis once added is split into two parts in the project tree. These two parts are described below.

3.2.4.1 EtherCAT Slave Drive Object settings

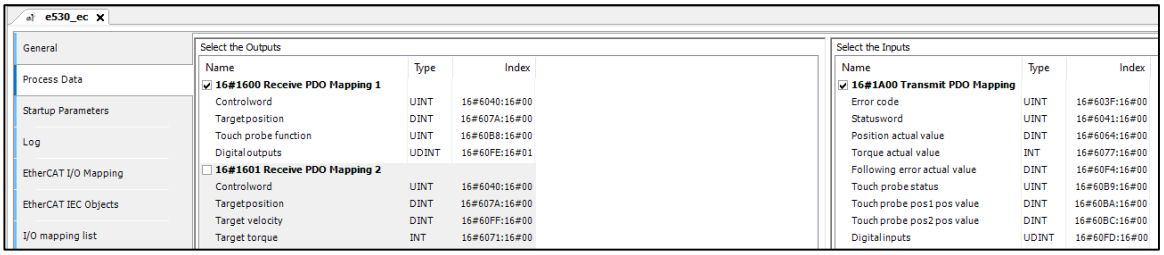
The label will normally be in the format of '[Drive name](Drive Type)' this can be changed by the user but is normally left as default. If so, the name could be **MicroFlex_e190** (MicroFlex-e190)

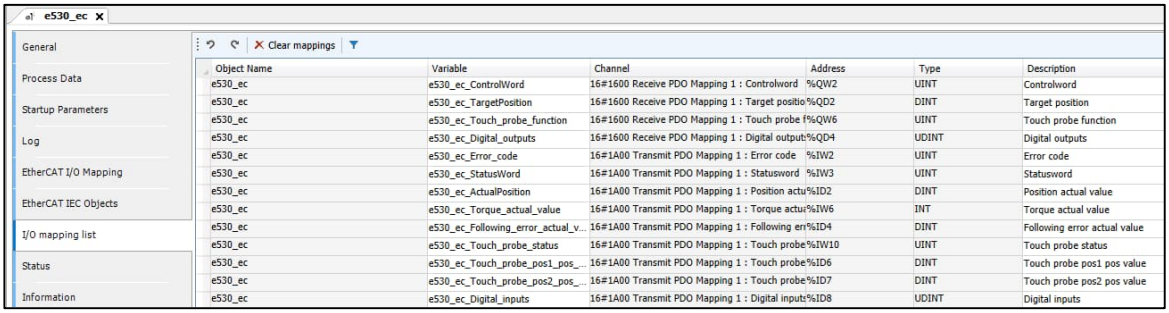
In most cases these settings can be left at default but occasionally the user might need to change these to fit the application requirements. Useful settings may be to enable Expert Settings to add additional PDO mappings, to check and set mappings or to check the status of the device once online.

3.2.4.2 PDO Mappings

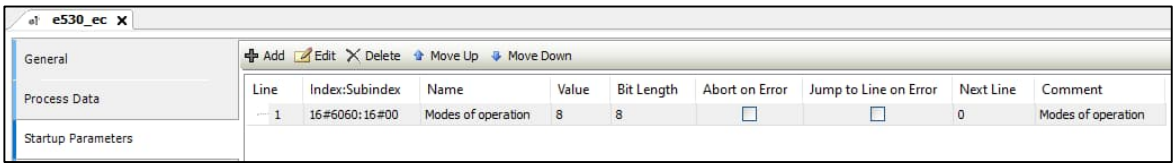
3.2.4.2.1PDO and Startup Parameters (SDO)

Based on the selections under [motion axis] > mapping of Control Type and PDO mapping the wizard will update the Process Data tab from the slave axis object and assign the autogenerated name to each object which is added.





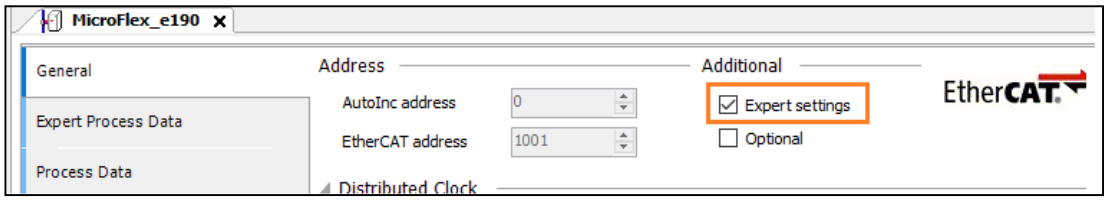
Like PDO mapping, SDO startup will be updated based on the control type and the mapping selected in the wizard.



	Note: Based on the application requirement, user can add more PDO / startup parameters manually.

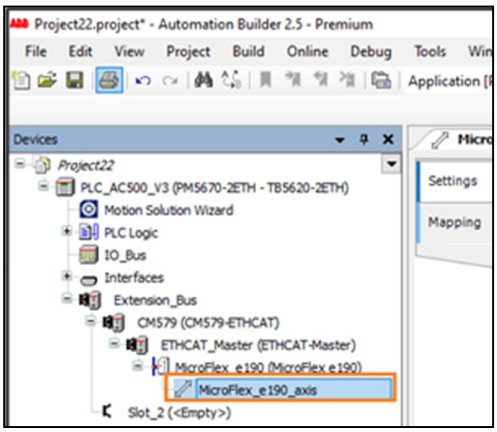
3.2.4.2.2User Defined PDO Mappings

If the user wants to add additional PDO mapped objects (which are not selectable from [motion axis] > mapping), First enable the expert settings, then the user will get an additional tab called “Expert Process Data” below the general tab and here user can add / edit / delete the mapping.



3.2.5 Axis settings

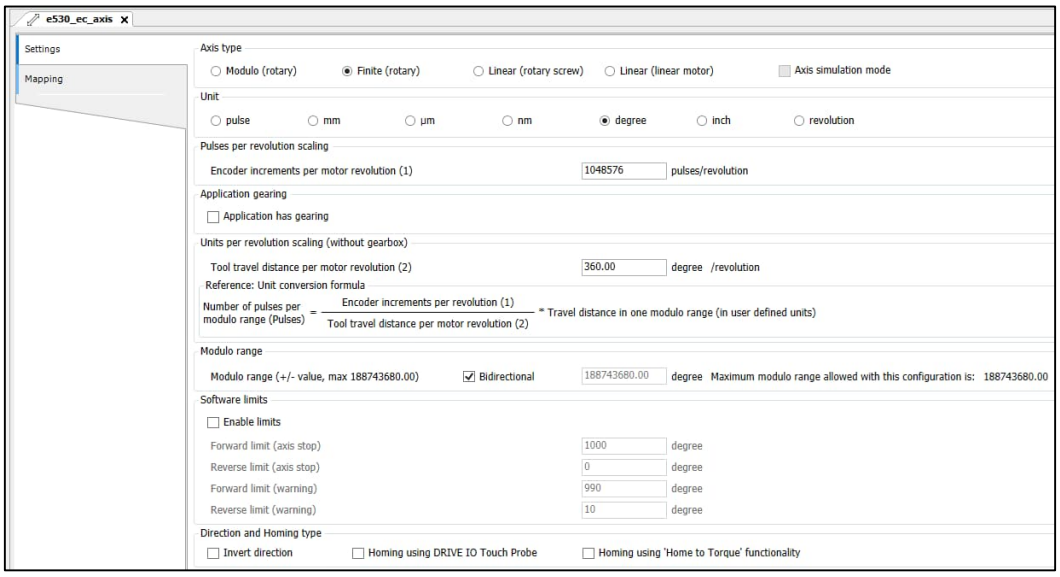
Nested underneath the EtherCAT slave Drive object is the Axis object, from here users can configure each axis as per the application requirement.



To open this, double-click on the object to open the configuration page.

3.2.5.1 Axis Settings Common to All Types

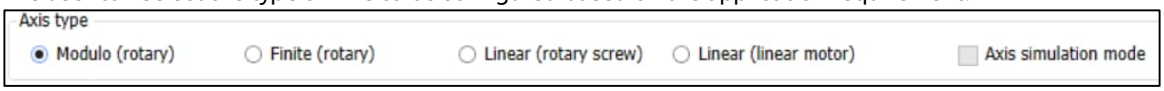
The main view for the Axis object is the 'Settings' tab and in the case of an EtherCAT axis will also show the "Mapping" tab. The user can update both the pages as per the application requirement.



Later during the "Generate application" process, the wizard will use these settings to define the operation and scaling of this axis object.

Axis Type

The user can select the type of Axis to be configured based on the application requirement.

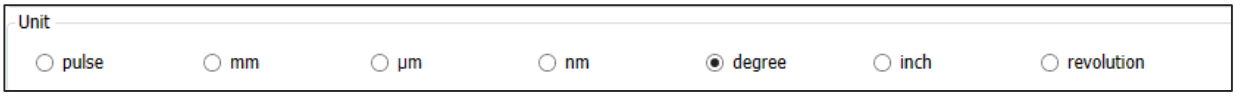


Below is the list of settings which user can configure from the Motion Control wizard along with their meanings.

Axis Type	Detail	Max Limit (Counts)
Modulo (rotary)	Default setting in the wizard. By selecting the Modulo (Rotary) your axis will be configured as a roll-over axis and the desired modulo range can be configured later.	2^31-1
Finite (rotary)	Your axis will be configured as a roll-over axis where in modulo range is non editable by the user and calculated based on the “Unit” selection, Inc_Per_R, U_Per_Rev_Nominator and U_Per_Rev_Denominator setting.	2^30 (Uni direction) 2^39 (Bidirection)
Linear (Rotary screw)	This needs to be configured when the user having a rotary motor with linear movements (linear axis).	2^31-1
Linear (Linear motor)	This needs to be configured when the user axis is a linear motor.	2^31-1
Axis simulation mode	This option is read only from here and needed when the user configured the axis but not have the real hardware yet. This can be selected from the motion solution wizard overview page. For virtual axis configuration please refer to the chapter “Add and configure virtual axis”.	NA

Unit

Based on the application requirement user can select the desired unit in the wizard and the wizard will update the subsequent parameters to the selected user unit. From the below picture user can find the currently supported unit formats.

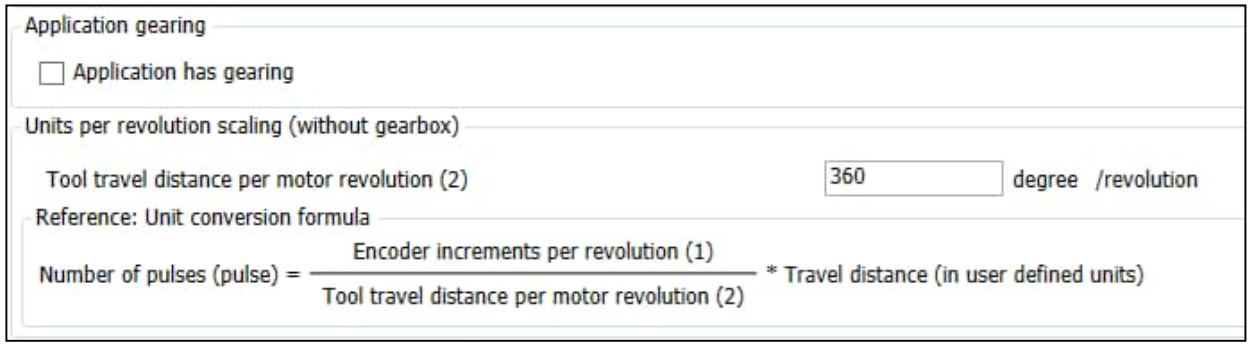


As an example, when the user selects the axis type as Modulo(rotary) and unit as degree, the wizard will update the subsequent parameters to the selected user unit and fill with default values, ex: modulo range = 360 degree (default). However please make sure the user updates the subsequent parameters as per the actual application requirement.

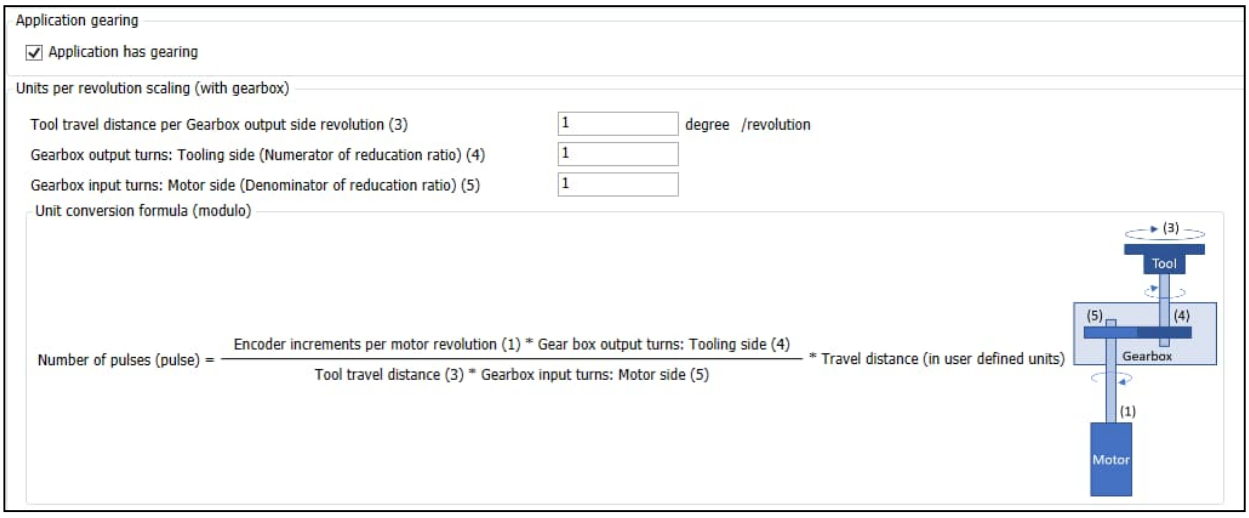
Application gearing / Units Per Revolution.

Here the user can set the tool travel distance per motor revolution and define the gearing relationship with the mechanical system.

As default the “Application has gearing” box is unchecked, this means the application has no gearing and the user can update the “tool travel distance per motor revolution” as per the application requirement:



If the application has gearing the user can check the “Application has gearing” box and they will be prompted to provide the gearing details. During the generate application, the wizard will update the same accordingly.



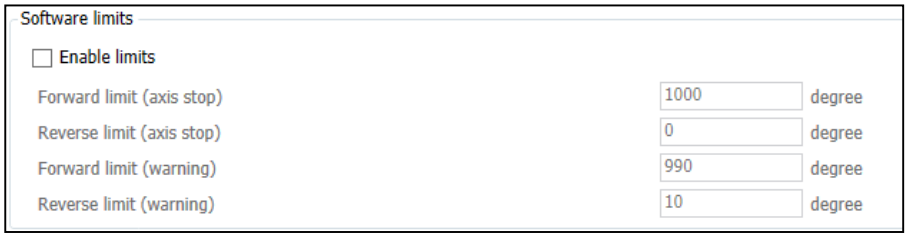
Modulo range

The user can provide the modulo range here. This is the value at which the axis position will wrap back to zero. This window will be active only when the user selects the axis type as any of the “rotary” axis.

Software Limits

The user can configure some of the common “Software Limits” from the wizard itself. Below is the list of software limits which user can configure from the wizard in the selected application units.

By default, software limits in wizard are not enabled and user need to enable the same by enabling the check box “Enable Limits”.



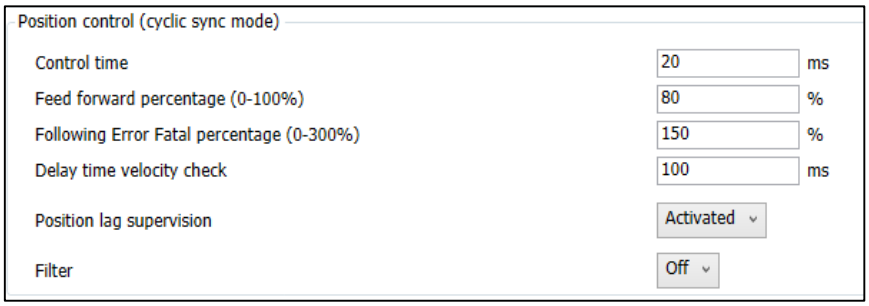
Direction and Homing Type

Invert Direction

This needs to be set for the application which needs to change the axis relationship between its real direction and that within the PLC program. By default, the check box will be unchecked, and the direction will be normal. By selecting the check box “Invert direction” both actual and reference position will be inverted, and the axis will move in opposite direction.

Position control (cyclic sync mode)

Here the user can configure the parameters related to position control and supervision. Details of each parameter is explained well in system technology chapter “Position control loop” and “Supervision” under Axis parameter chapter.

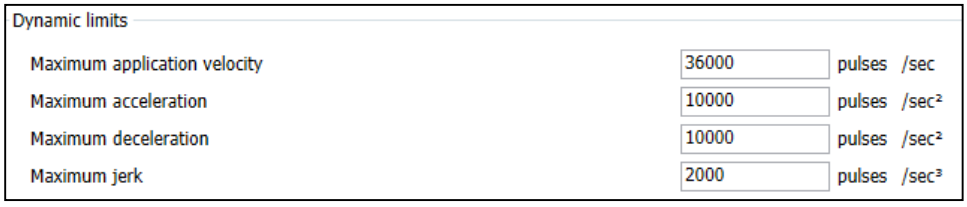


- Filter off - No filtering is applied.
- Velocity filter – several samples are used to smooth out the velocity calculated in the Kernel
- Position Filter – A linear Regression calculation is used to smooth out the position values read by the Kernel.

Dynamic limits

Users can update the maximum limits here. Some parameters depend on the drive settings and needs to be set correctly to get the desired result.

The user can set the maximum application velocity to a desired value to limit the maximum application speed.



Example:

if user is using an ABB Servo drive with, 131072 Encoder increments per revolution and a Maximum speed is 6000 RPM

Maximum application velocity

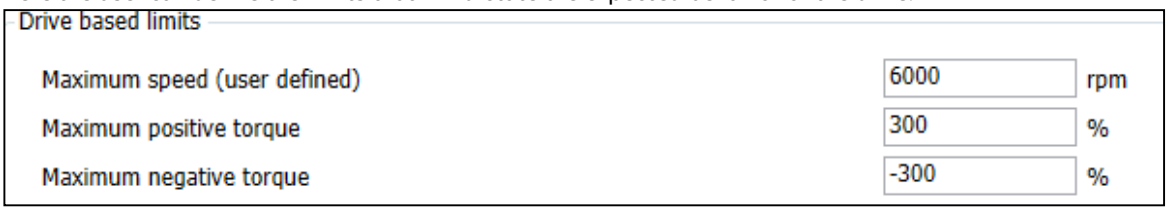
= Max application velocity in RPM * Tool travel distance per revolution / 60 * Gearbox nominator / gearbox denominator

= 6000 * 360 / 60 * 1 / 1 = 36000 degree / sec

	*For easy calculation of parameters user can use the excel "AC500_V3_MotionControl_Startup guide for MC parameterization.xlsx" from example program folder at C:\Users\Public\Documents\AutomationBuilder\Examples\PS5611-Motion\Documentation.*

Drive based limits.

Here the user can define the limits that will dictate the expected behavior of the drive.



It is recommended to keep the same Maximum speed at the drive and at the PLC parameter.

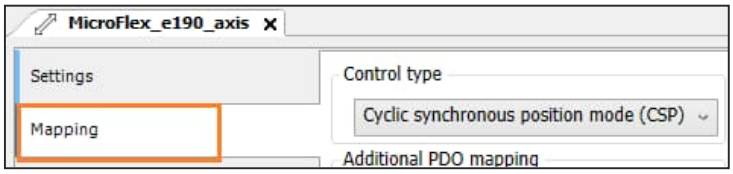
Results (calculated)

Based on the inputs provided, wizard will calculate the results and can be viewed immediately at the end of the configuration page.

3.2.5.2 Axis Object Settings Specific to EtherCAT Axes

Mapping

The Mapping and Control Type tab can be selected if the user wants to set a Control Type other than CSP (Default) and mappings other than default (Control Word, Set Position, Status Word, Actual Position). It can be found under the axis object here:



Control type

By default, wizard is selected for cyclic synchronous position mode (CSP). User can change the same based on the application requirement. The supported control modes are:

- Cyclic synchronous position mode (CSP).
- Cyclic synchronous velocity mode (CSV).
- Cyclic synchronous velocity mode for load control (CSVL).
- Cyclic synchronous torque mode(Limited CST).

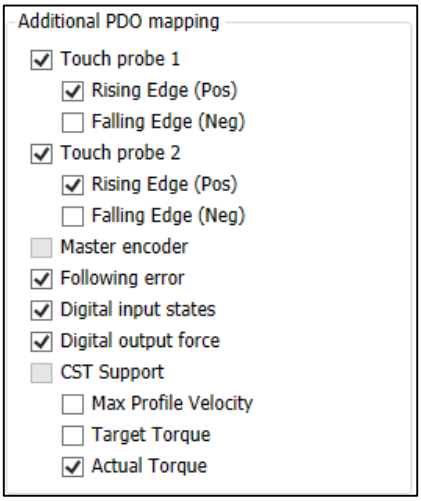
	Note: CSVL is an ABB specific mode to achieve load control / profiling. By using this mode, the user can use the Motion Control Load library which is implemented based on the "PLCOpen Motion Part 6 – Fluid Power Extensions".

Additional PDO mapping

If the application needs additional PDO mapping the wizard helps the user to add most used PDO mapping just by selecting them here.

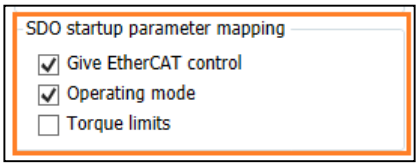
Based on the control type selected, a few of the mandatory PDO mapping are generated automatically and from additional mapping area in wizard user can find the most common PDO mapping and user can add the same based on the application requirement.

Please note, the user can add additional PDO mapping which are not listed here manually by enabling the expert settings from the slave device general configuration page (as described earlier).



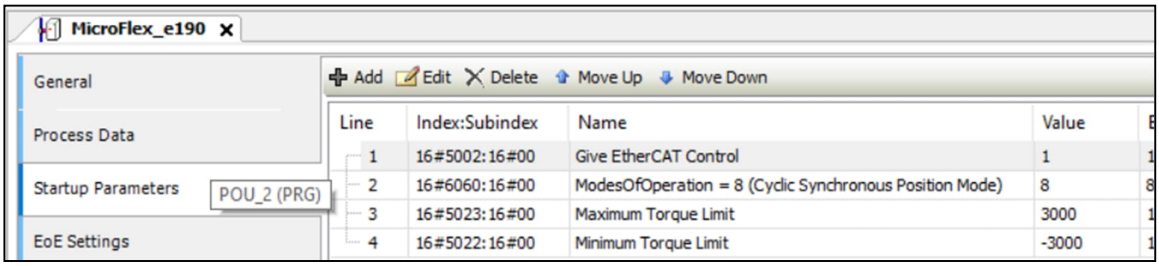
SDO start-up parameter mapping

By default, two of the SDO startup parameters are selected and it is recommended not to change these unless the user has expert level knowledge of DS 402 control modes or intends to do nonstandard start up coding as it will change the expected operation of the axis at start up.



The user can select the Torque limits and the torque values set from the “settings” page will be written to the respective slave drives startup parameters list.

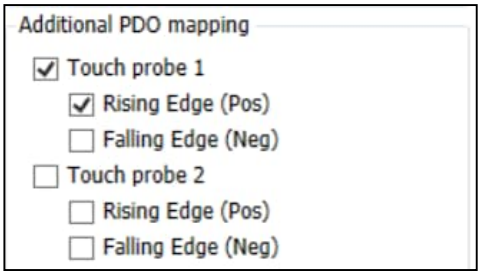
Once these settings are made and the Generate code is executed you can see that these settings have change the drives EtherCAT Slave configuration as shown in the picture below:



Settings - Homing using Drive IO Touch probe.

Selecting this will allow the axis to home to a drive-based Touch Probe using EtherCAT.

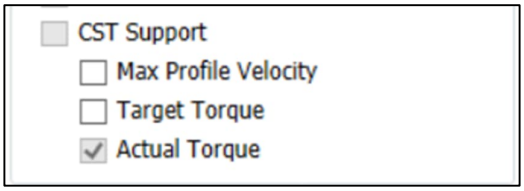
Selecting this also by default make the PDO mapping preselected for the user, but this can be later changed based on the application need.



Settings - Homing using 'Home to Torque' Functionality

Selecting this will allow the axis to home to an end stop or a torque setpoint.

This will also require mapping the Torque actual and Setpoint parameters to the axis reference.



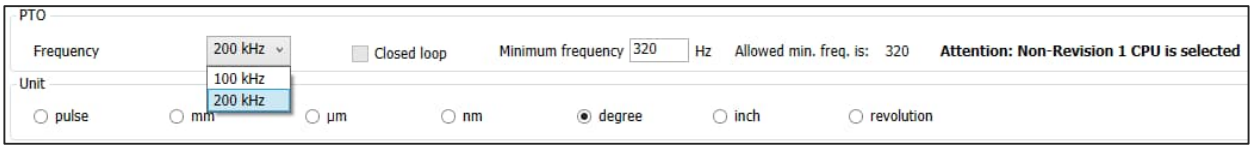
3.2.5.3 Axis Object Setting Sepecific to PTO Axes

All settings related to the application and axis specific will be done here and need to be carefully updated for each axis. Based on the inputs provided here, wizard will compile and generate the code.

PTO (Hardware settings)

With revision 1 CPU or higher, by default 4*200kHz PTO axis is supported, there is no special settings needed in the axis configuration page.

If a revision 0 CPU is used, user can set the PTO axis frequency to 100kHz or 200kHz using the drop down as shown below. Based on the PTO axis frequency selection, the onboard IO configuration will be updated.

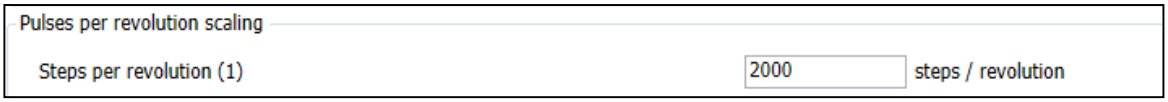


Closed loop

By default, the PTO axis is open loop and however user can also configure a maximum of two PTO axis as closed loop. To configure the axis as closed loop, user need to configure an encoder axis -> encoder source -> encoder channel -> encoder purpose as the PTO axis which needs to be a closed loop axis. This tick box will show the result of that setting!

Pulses per revolution scaling

When the axis is *open loop*, user can update the "Pulses per revolution scaling" with the steps per revolution.



	Note: Please note, when the PTO axis is open loop, it is important that the user set the Steps per revolution and Maximum Rpm by keeping the PTO frequency limits into consideration. For example, for a 100Khz PTO axis if the steps per revolution is 2000 , the maximum RPM the axis can support is 3000 RPM. (= 2000 * 3000/60 = 100Khz)

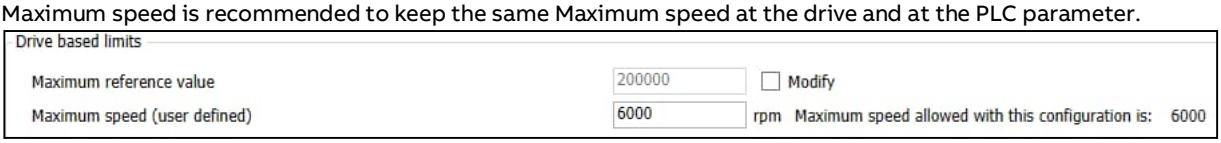
When the axis is *closed loop*, user can update the “Pulses per revolution scaling” with the actual encoder increments per motor revolution.

Drive based limits.

Maximum reference value is the maximum frequency of the drive in which it reaches the maximum RPM. In some drives, the calculated maximum reference value may need to be modified, and user can select “Modify” box and change the value.

Please note, we have the hardware limitation as 100kHz or 200kHz based on the configuration and user must make sure this limitation is not crossed.

Maximum speed is recommended to keep the same Maximum speed at the drive and at the PLC parameter.



Currently the applications torque limits in the wizard are not valid for PTO axis and this is ignored.

Note:

When the PTO axis is open loop, it is important that the user set the Steps per revolution and Maximum Rpm by keeping the PTO frequency limits into consideration. For example, for a 100Khz PTO axis if the steps per revolution is 2000 , the maximum RPM the axis can support is 3000 RPM. (= 2000 * 3000/60 = 100Khz)

When in *closed loop*, please set the Maximum RPM for the axis to reach when the maximum PTO Axis frequency is provided. For example, if the axis is configured as 200Khz and closed loop, set the Maximum RPM for the axis to reach when the 200Khz frequency is achieved.

3.2.6 Important General PLC settings

Compared with a standard AC500 project, when the user is using motion solution wizard, some default settings are changed **during code Generation**. These settings are always overwritten if there is a change in axis configuration or added a new axis. Below are the settings which are updated by motion solution wizard:

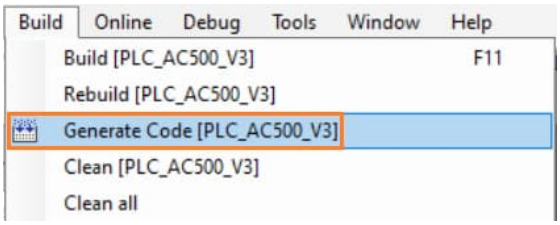
	Project type	
Setting	EtherCAT	PTO Only
IO bus – Run on congif fault	Yes	Yes
PLC > PLC Settings > Always update variables	“Disabled”	“Disabled”
PLC > PLC Settings > Bus Cycle Options	e.g. “Task” - any real slow task must be selected and it **shouldn't be set as “unspecified”** as this can lead to a negative on cyclic operation.	
CPU-Parameters > Check battery	Off	NA
CM579-ETHERCAT > Run on congif fault	Yes	NA
CM579-ETHERCAT > Bus Behaviour	Sync Mode 1	NA

It is recommended that user change the above settings manually based on the actual application requirement. To update these setting manually, user need to change the setting after all the axis changes are done and then Generate the Code.

3.2.7 Generating the Motion Configuration

Once all the configuration is completed, the user can *Generate* the application (like compile) which will then update; its settings, the PDO & SDO mapping, and the Motion task configuration automatically based on the settings and parameters provided in the wizard.

To do this, go to Build menu and click on *Generate Code*



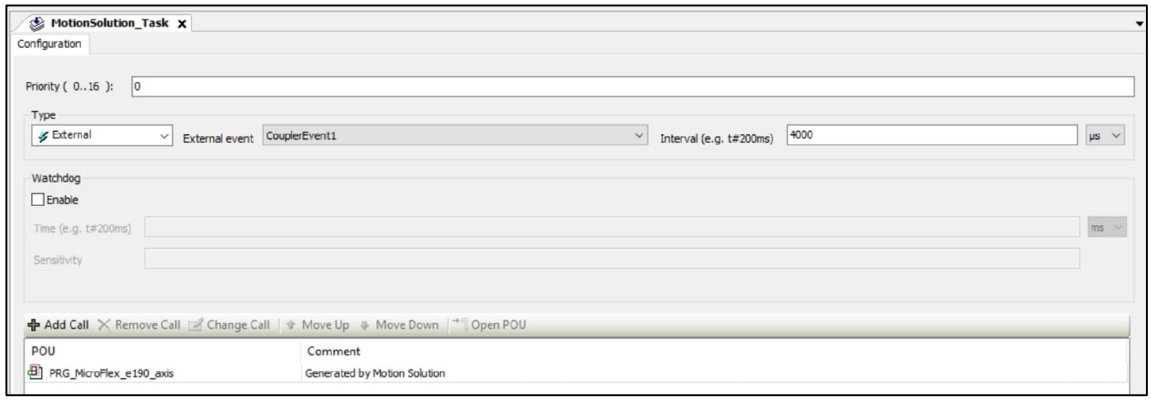
After a few seconds Automation Builder will generate the axis configuration. Once completed successfully, Automation Builder will show the message “Motion Solution Generation successful” in the message window. This can take some time based on the number of axes configured in the project and PDO mapping selection.

3.2.8 Task configuration

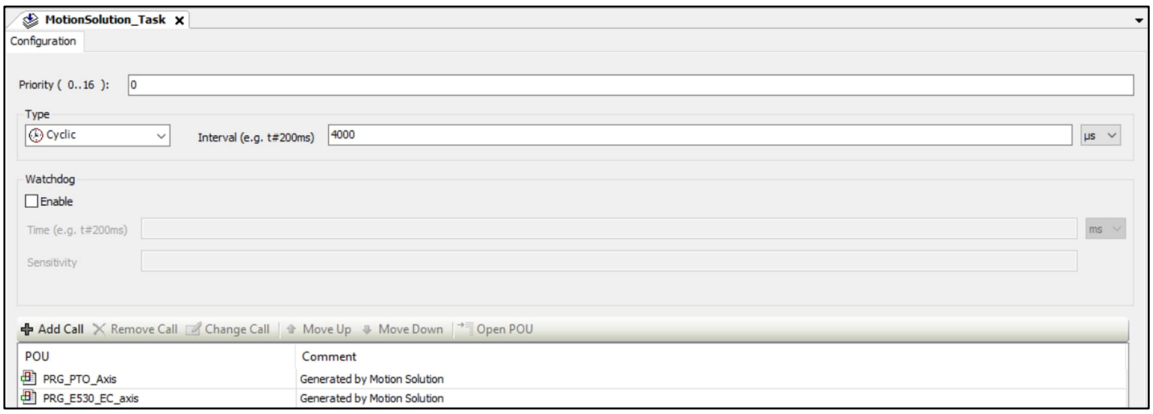
3.2.8.1 Tasks withing The Motion Wizard

The Wizard will automatically create a ‘fast task’ and a ‘slow task’ to help guide the user to correctly distribute their code. The kernel function block and the transfer of axis IO data with EtherCAT (or other hardware) should be processed in a faster task. This task should be as short and real-time (if EtherCAT) or as fast possible (if PTO) to achieve the best motion control performance.

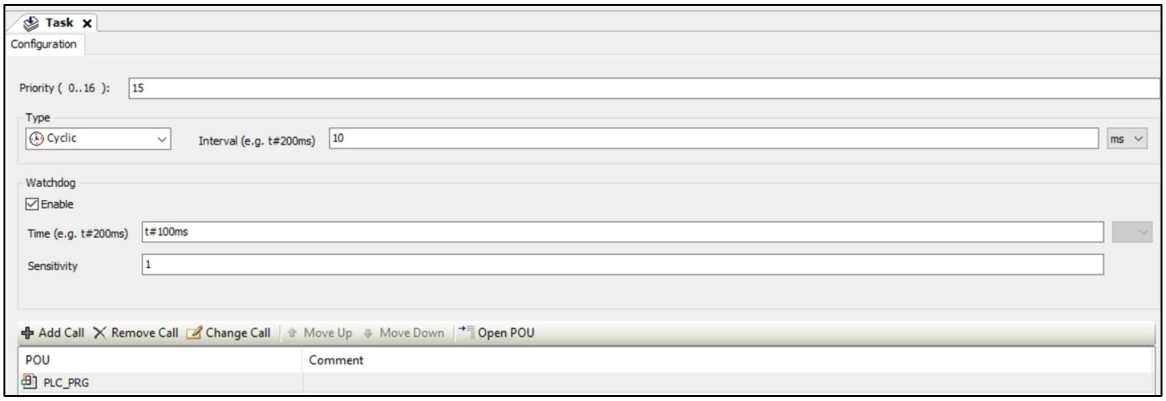
The following figure shows an example for CM579- EtherCAT ‘fast task’ - MotionSolution_Task:



The following figure shows an example for Onboard- EtherCAT ‘fast task’ - MotionSolution_Task:



The Cyclic task is shown as below:

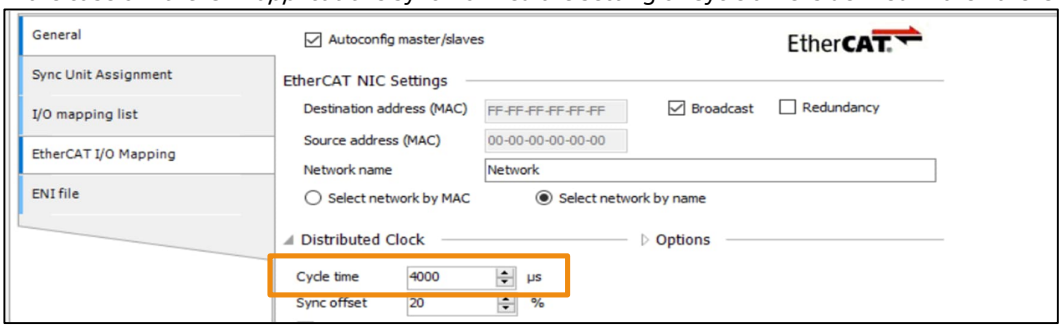


3.2.8.2 MotionSolution_Task

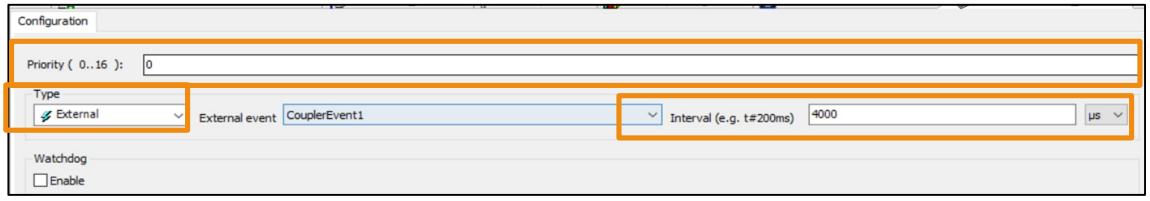
The Motion Solution Wizard will automatically create a fast task called “MotionSolution_Task”. This will call all the axis operation code, and will be configured dependently on the configuration.

3.2.8.2.1 MotionSolution_Task in EtherCAT applications

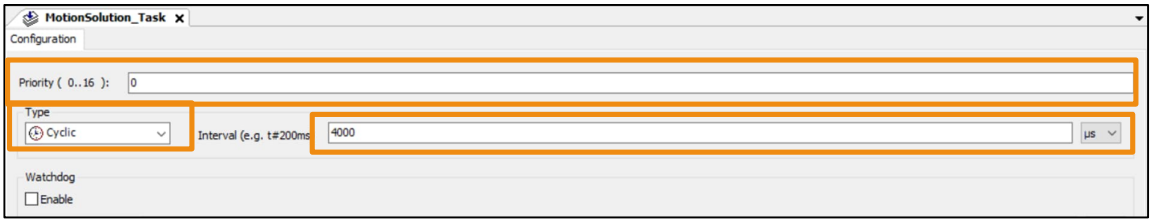
In the case of *EtherCAT applications* synchronized the setting of Cycle time is defined in the EtherCAT master;



Once the Solution is Generated this will then define a task which has a type of **External** and a **cycle time** which matches that configured in the *EtherCAT master* and the **Priority is set as “0”.**

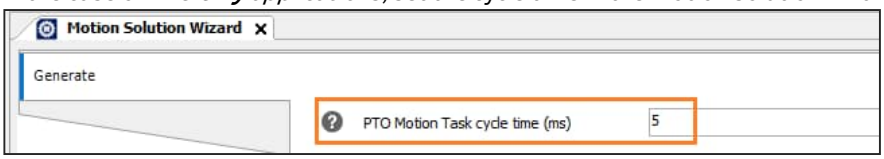


Incase of Onboard EtherCAT, the task type is **Cyclic** and a **cycle time** which matches that configured in the *EtherCAT master* and the **Priority is set as “0”.**

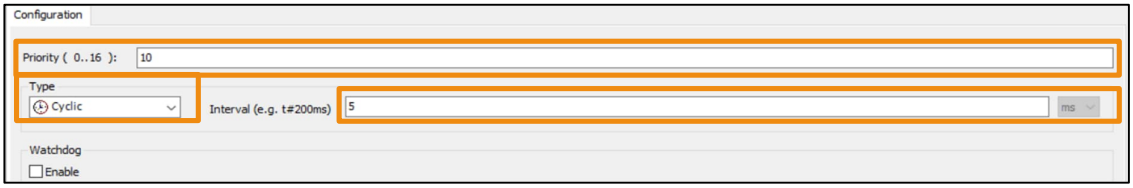


3.2.8.2.2MotionSolution_Task in PTO only applications

In the case of *PTO **only** applications*, set the cycle time in the motion solution wizard,



Once the Solution is Generated this will then define a task which has a type of Cyclic and a cycle time which matches that configured in the *Solution Wizard* and the Priority is set as “10”



3.2.8.2.3Function blocks must be called in the Motion Solution Task

As mentioned previously the ‘MotionSolution_task’ is the fastest and highest priority task in the CPU and so of course the most draining on resources. It is possible for the user to call *parts* of the motion application in this ‘MotionSolution_task’ – however excessive amounts of code or that with a poor structure in this task ***can lead to overloading of the CPU***.

Despite these limitations some function blocks must be called in the same task as the kernel function block (MotionSolution_Task). These are listed below:

- MC_CombineAxes
- MCA_MoveByExternalReference
- MCA_MoveByExtRefRelative
- MCA_DigitalCamSwitch
- MCA_GetTappetValue
- ECAT_CiA402 Touchprobe_App
- ECAT_HomingOnTouchprobe
- ECAT_CiA402ParameterHoming

	Note: For onboard EtherCAT motion axis projects, it is recommended to call all the EtherCAT FBs in one task, for coupler EtherCAT this is not relevant.

3.2.8.3 Cyclic Tasks and created task ‘Task’

As default the wizard will also create a 10ms cyclic task called ‘Task’ which the user can use for such code calling for noncritical motion Function Blocks:

Other tasks can also be added as well but to avoid complication its better not to add too many and to avoid conflicting priorities (such as making then all the same) or interceding times such as 10ms, 20ms,40ms where every x ms several tasks will be called together.

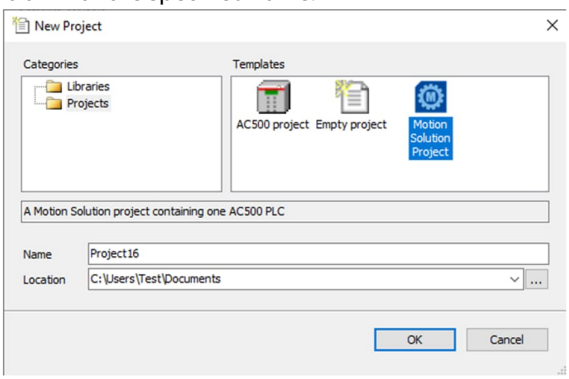
3.3 Writing Application program

Having followed section 3.2 so far, we have been through the steps to select a PLC, update the device files, set up the project and understand the important settings that will influence the behaviour. In this section we will cover the final parts of the project, programming and downloading.

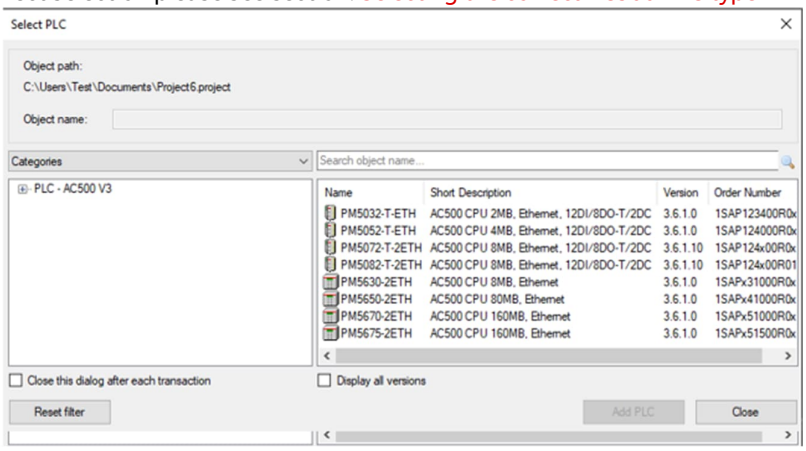
3.3.1 Setting up a project for programming introduction

Now we should understand the steps to set up a project, none the less I will introduce these steps in brief to align with the settings that we will refer to in the program.

1. The first step when creating a program is to open Automation Builder and select “New Project”.
2. Then select the “Motion Solution Project” icon as shown below. Click “OK” button and a new project will be created in the specified location with the specified name.

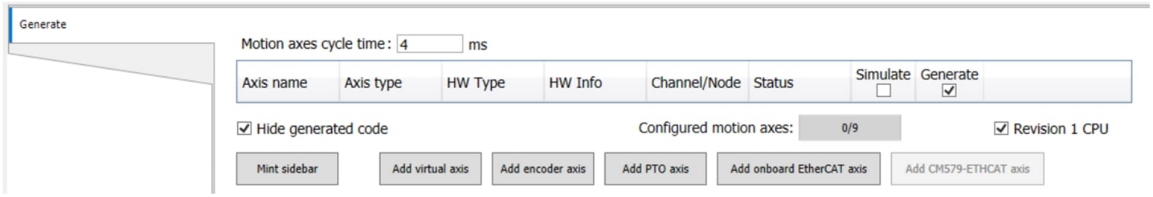


3. Add the PLC - Automation Builder will now pop up a “Select PLC” window and from here user can select one of the V3 PLC’s.
   Note: For the correct selection please see section: Selecting the correct AC500 PLC type

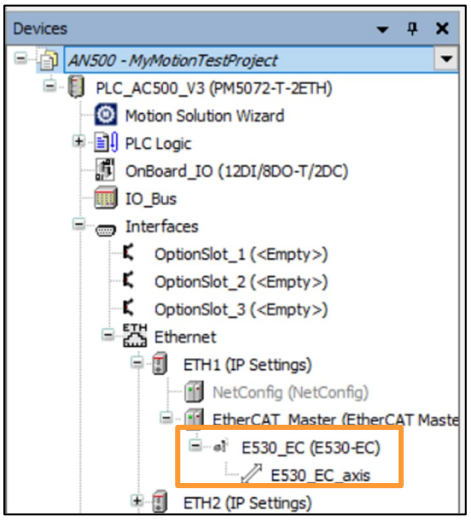


4. In this case we will add a ‘PM5072-T-2ETH’
5. Next click “Add PLC” button to get it added to Automation Builder hardware tree.
6. Then After creating the hardware tree, Automation Builder will launch the motion solution wizard overview page. From here user can add the axis to the application. Depending on the PLC type selected some of the

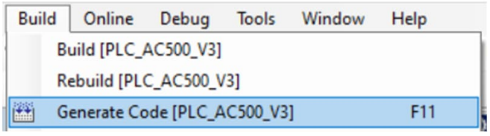
axis types are not possible for the user to add to the project:



7. Now in this case we can add one EtherCAT axis by selecting 'Add onboard EtherCAT axis' and select an 'e530-EC' from the 'Add Motion ECAT Drive' pop up, which will then create am TherCAT_Master under ETH1 and add this new axis below it:



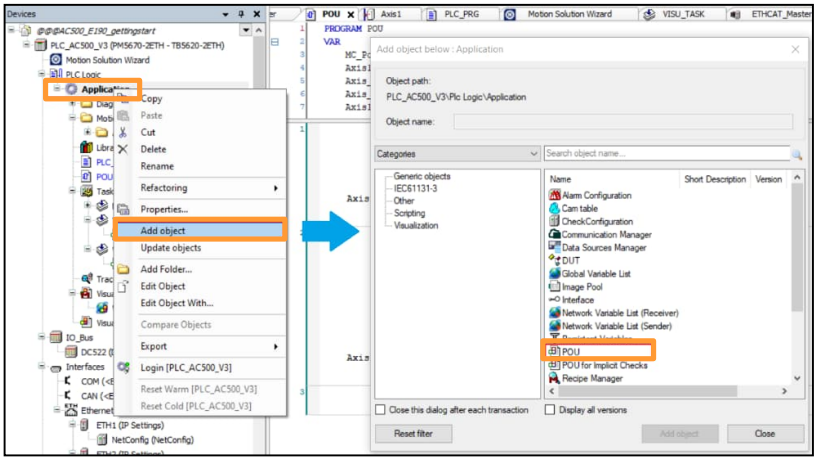
8. At this stage the user could change the Axis settings by clicking on 'E530_EC_axis' but for now we will leave these settings as default. Key settings are below:
   - Control type: CSP
   - Mapping: CWord, Sword, Apos, TPos, TP1Pos, TP2Pos
   - AxisType: Finite
   - Scaling: Degrees
9. After this we can Generate the code to build the axis_ref and other key settings for use in the program:



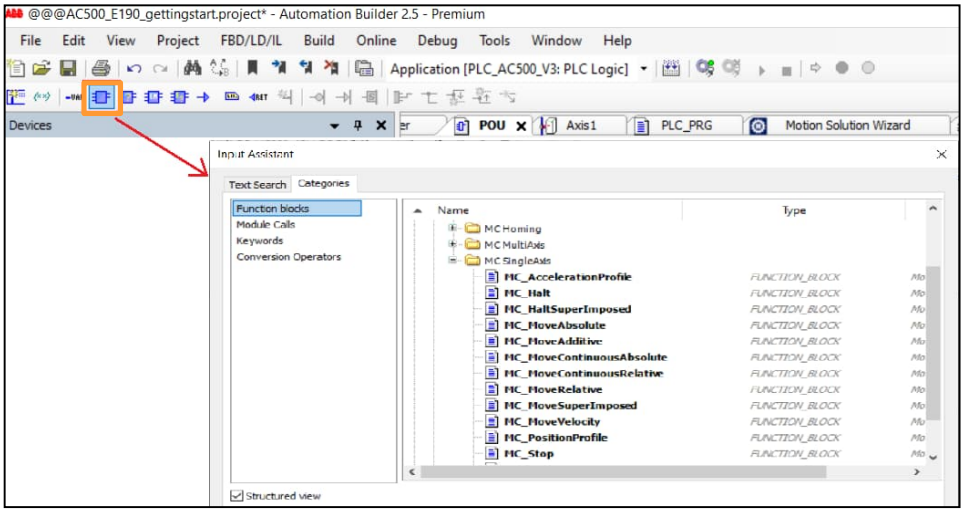
Note: If at this point there are any build errors the user must recitify before going further.

3.3.2 Program Writing

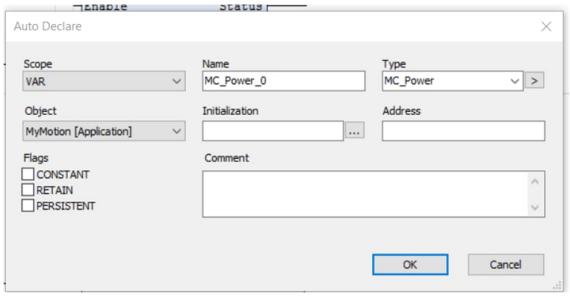
10. Next, we can add POU (Program Organization Unit) which will contain our PLCOpen Motion program. To do so Right click on Application > Add Object > POU



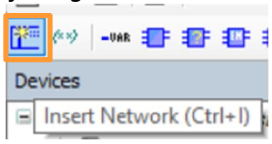
11. Next you can edit the name, we can up 'MyMotion' and as the language we can select 'FBD' then click 'Add' to add the POU to the program.
12. Now we can start inserting our Function Blocks. To do so Click this icon to insert the PLC OPEN block into POU.



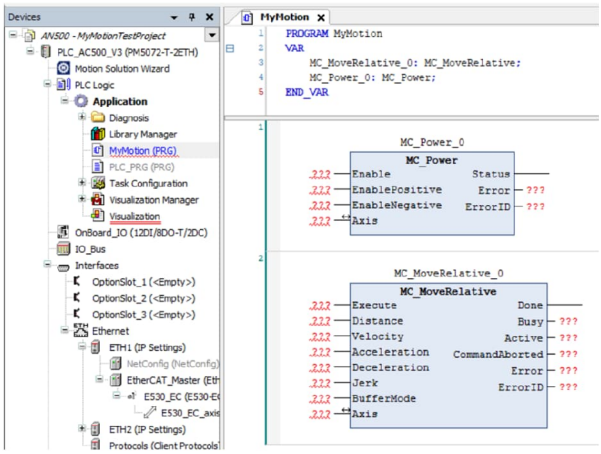
13. You will then see a pop up where you can find the libraries manually or you can also quickly find the needed MC blocks with 'Text Search'.
14. Firstly, we can select Function Block MC_Power then accept this by clicking 'OK'
15. As default the instance name will be given in the structure '[FB_name]_[Running number]' we wil simply accept this by clicking 'OK'



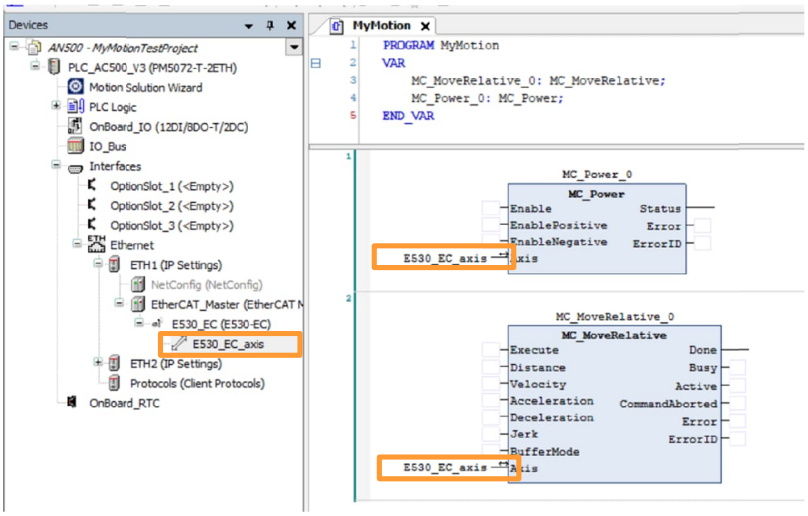
16. After this we can insert a network below by using this icon:



17. Then we can follow the previous setps to insert an instance of MC_MoveRelative. After these steps we should see the below:

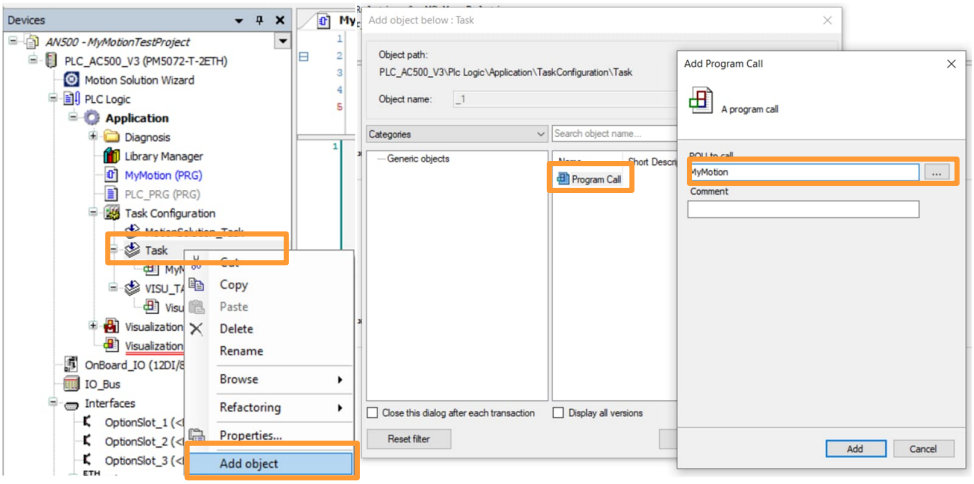


18. The only mandatory variable for every Function block interface is the 'Axis' interface. The variable name of the Axis interface must be same as the Axis name in the tree.

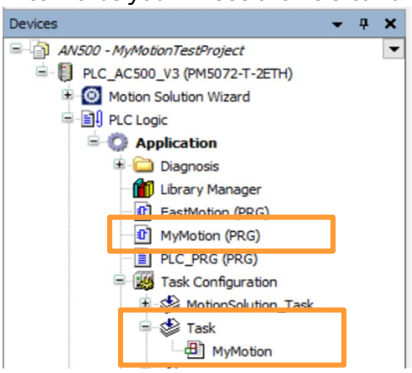


Note: All other inputs can be hard coded, attached to variables or forced on line but in this case we will use a visualization to populate then and use these function blocks.

19. Now we have the program ready to use we must ensure it is processed by calling it from a Task. As mentioned already to save CPU overload, it is recommended to consider the correct distribution of code. In this case we will put our code for the application in the **'General'** task 'Task' which has a 10ms cycle time and lower task priority than 'MotionSolution_Task'. To do so right click on 'Task' > Add Object, then select Program Call and enter the name of the POU 'MyMotion':



20. Afterwards you will see the POU call under the task

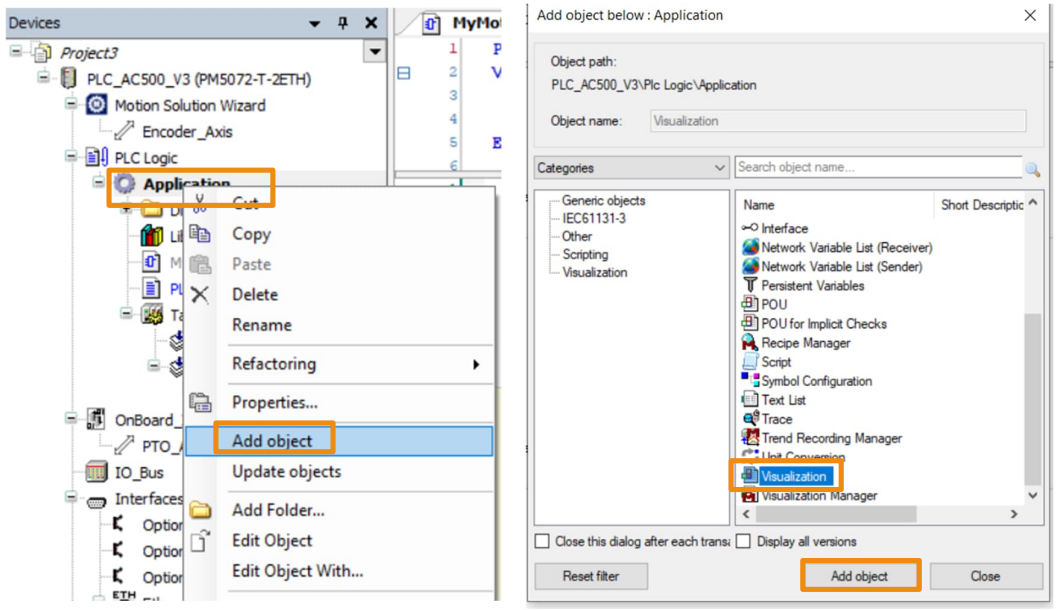


21. This program is now ready to download but there are tools available to make testing easier.

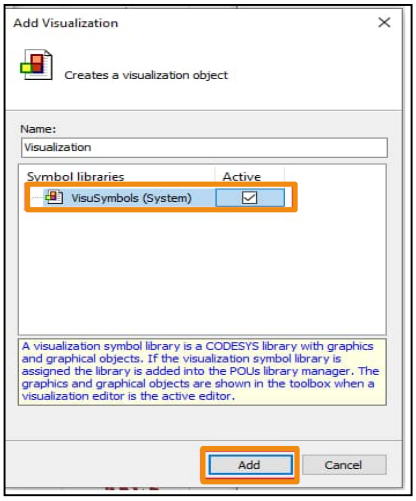
3.3.3 Visualization (Optional)

A Visualization can be used to create a quick graphical interface to test the program. There are a series of premade interfaces for the Motion Control library that are ready to be used.

1. The first step is to add the VIsualisation is to Right click on Application then ‘Add object’ and ‘Visualisation’ Then click ‘Add Object’ and ‘Add’:

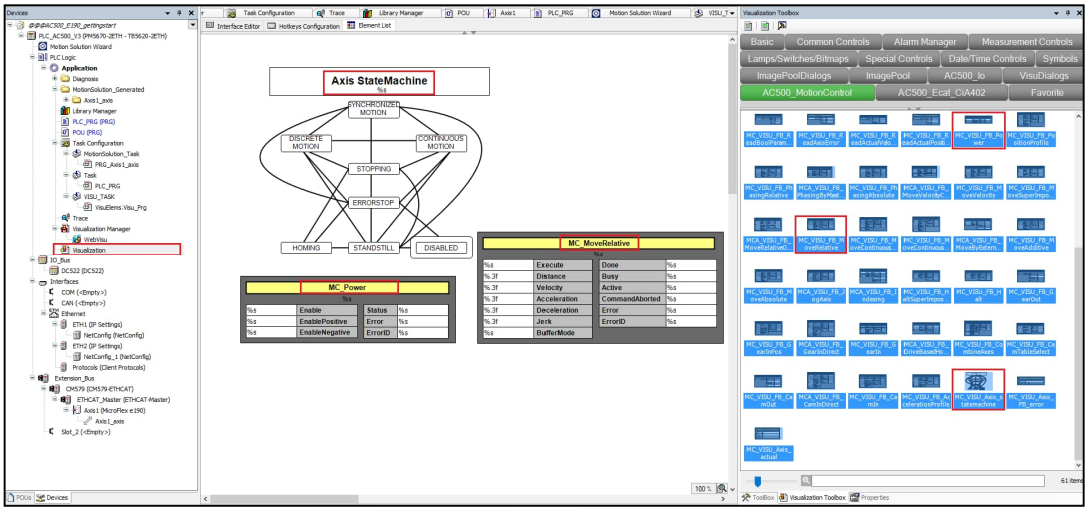


2. Then Select ‘Active’ VisuSymbols then ‘Add’



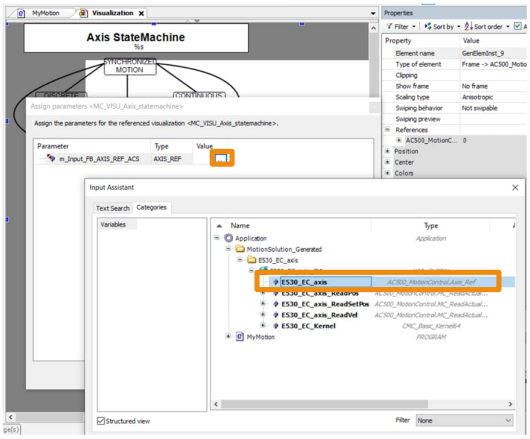
3. Next, open the Visualization Toolbox and drag and drop from the Element list.

4. Here we select *MC_VISU_FB_POWER*, *MC_VISU_FB_MoveRelative* and *MC_VISU_Axis_statemachine*:

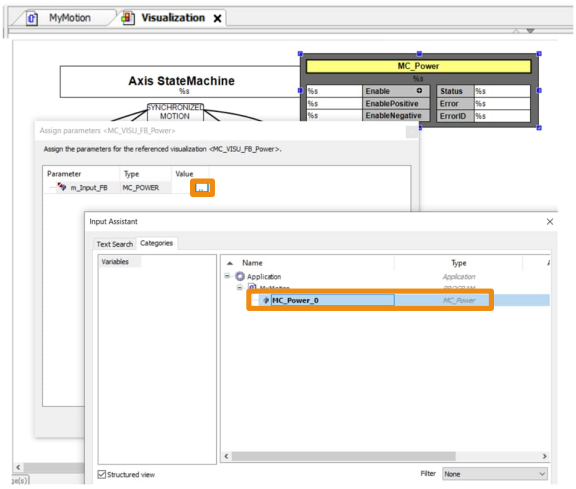


5. After dragging the element down into the page then the window will pop up to allow the associate the Visualisation with the Instance of the Function Block.

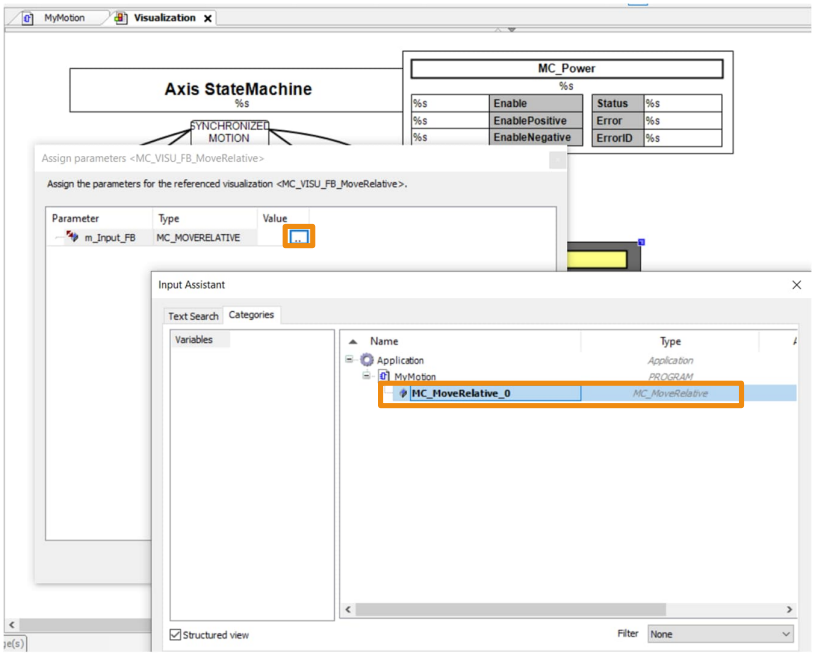
6. One by one, click the ‘Value’ item and then select the corresponding axis, Axis_Ref in this application for Axis StateMachine then click ‘OK’



7. Specify the corresponding POU, MC_Power_0 for MC_VISU_FB_Power then click ‘OK’ as below



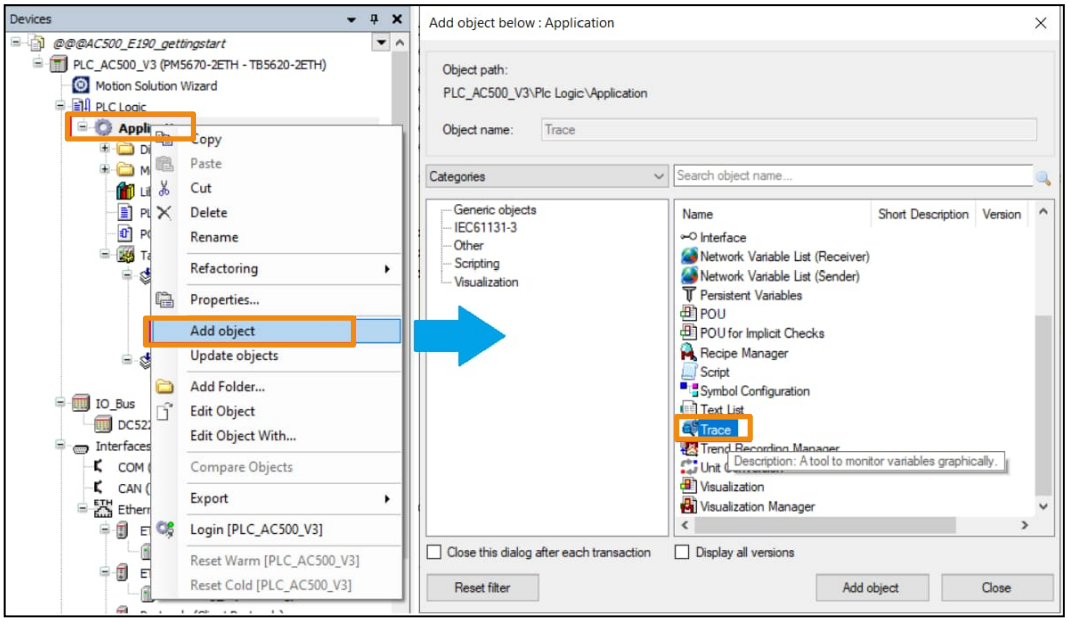
8. Specify the corresponding POU, MC_MoveRelative_0 for MC_VISU_FB_MoveRelative in the same way as below:



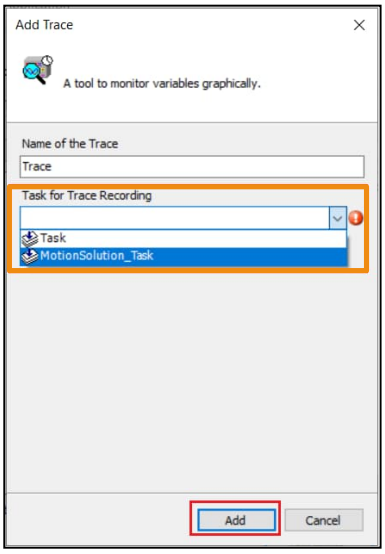
3.3.4 Trace (Optional)

Its common place that in motion applications a trace function is used to represent the motion to the user during debugging. The following steps explain how to add a trace function to the project.

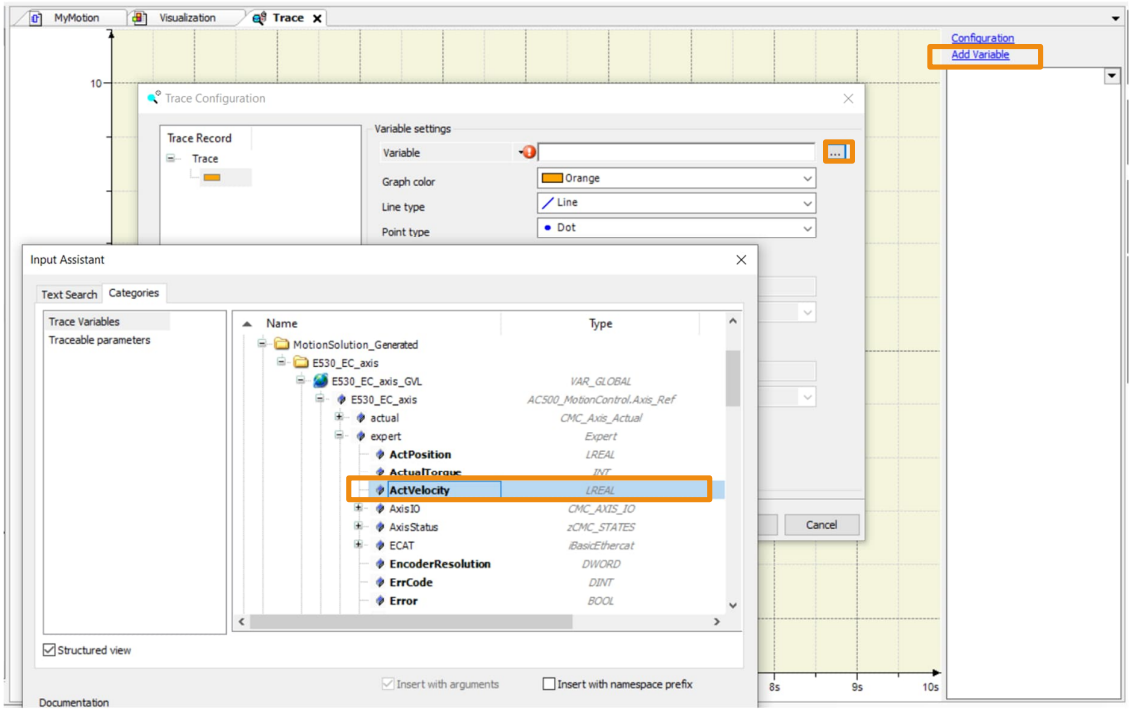
1. To add a trace to the project once again right click on 'Application'>'Add object' then select 'Trace' then Add Object.



2. The next Step is to select the Task that user wants to run the trace. Depending on the response that the user wants (in the same way as with POU calls) they can select the Motion task or the general Task. The motion task is faster but will add more CPU load, the General Task is slower but will have a lesser effect on the CPU load.



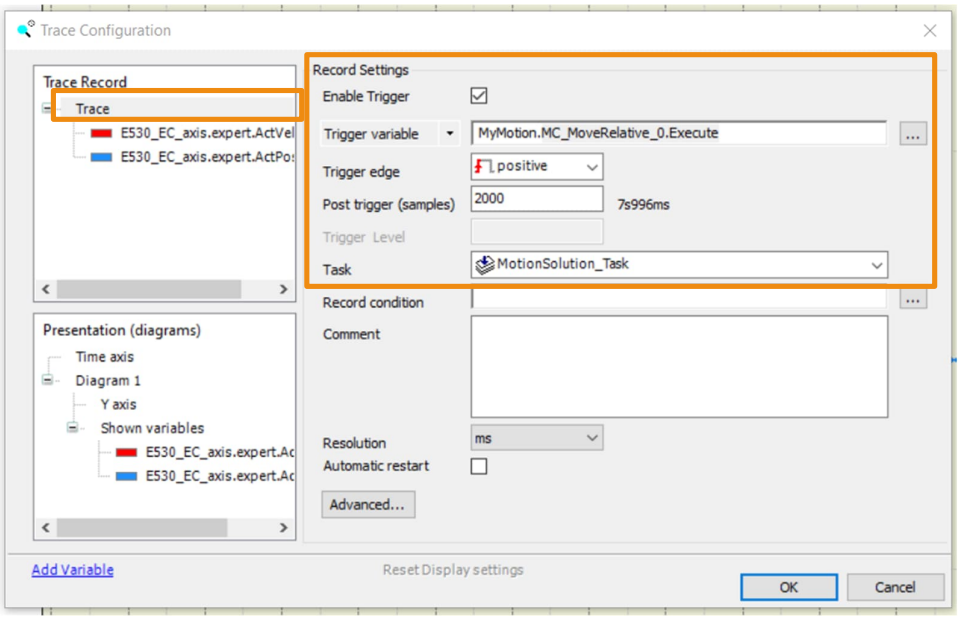
3. After the Trace is added the user can start mapping any variables they so desire to their project.



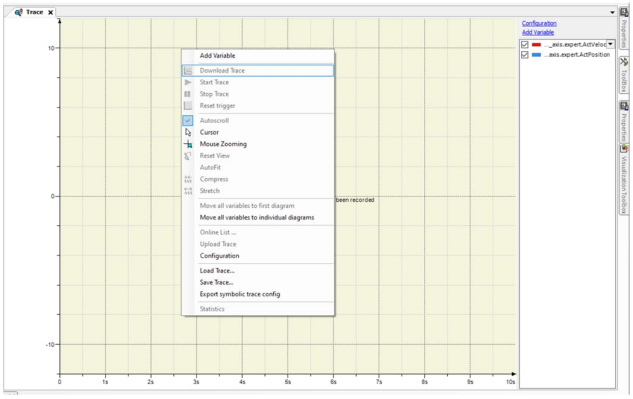
For ease of use some actual signals are available from the [axis_ref].expert section of the Axis_Ref structure. Here will we use:

- E530_EC_axis.expert.ActVelocity
- E530_EC_axis.expert.ActPosition

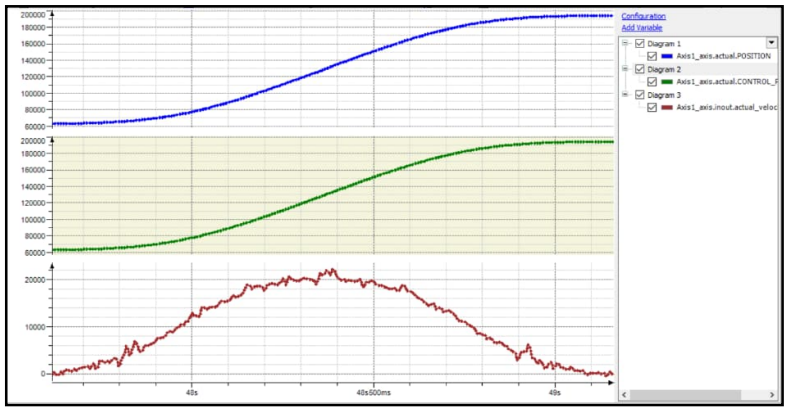
4. Its also possible to set trigger conditions which will mean than once downloaded the trace will not trigger until the condition is met. In this case that will be on the rising edge of MC_MoveRelative_0. Execute.



5. Now the trace is ready to be downloaded. To be downloaded it must be online so this step will be completed later.



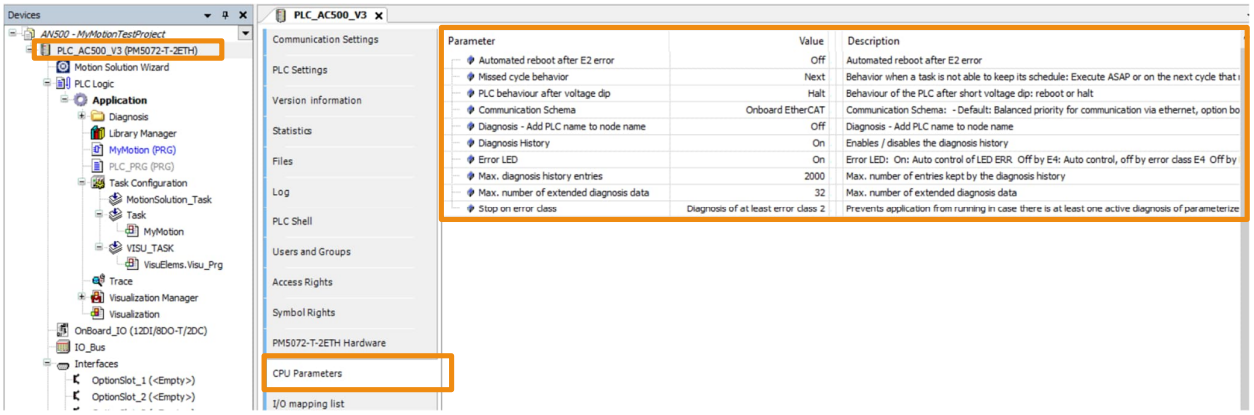
6. After its downloaded we can see a scope trace of all the mapped variables:



3.3.5 Check the CPU parameters.

CPU parameter settings should be considered when configuring the Automation Builder projects at least for demanding cases where defaults are not fitting. Please check and update the parameter settings as per your hardware setup and system requirement. To access these settings.

1. Double-click “PLC_AC500_V3”.
2. A tab opens in the editor view.
3. Select “CPU-Parameters Parameters”.

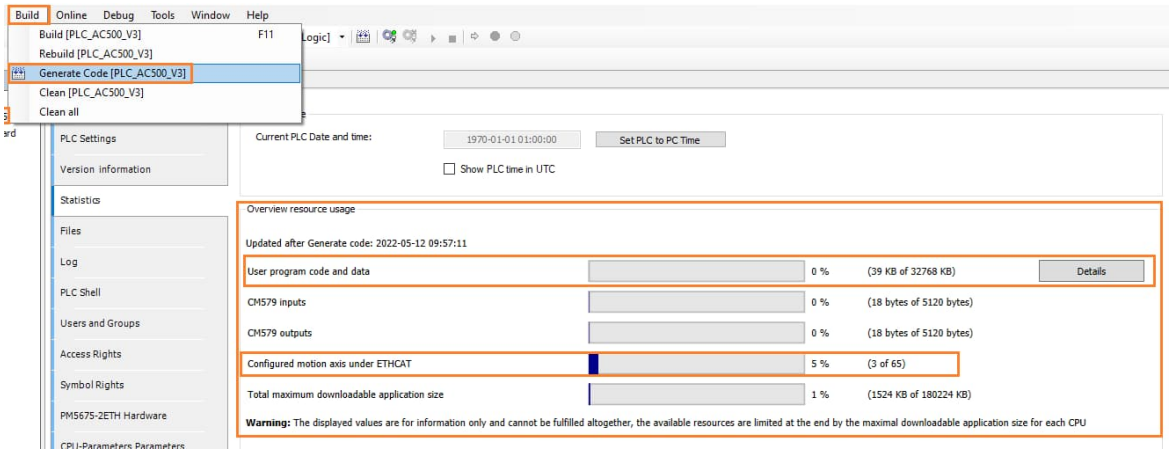


In most cases these can be left as default.

3.3.5.1 Checking program size and number of configured axes

User can check the program size and the number of configured and supported axis from *Statistics* tab. To access this page.

1. To update the Statistics to match the latest code the user first needs to perform “Generate Code” from “Build” menu
2. Double-click “PLC_AC500_V3”.
3. A tab opens in the editor view.
4. Select “Staistics”.



For the limits of “*User program code and data*” a [Details] button is available. Clicking this button will open a window showing a more detailed view of the memory usage.

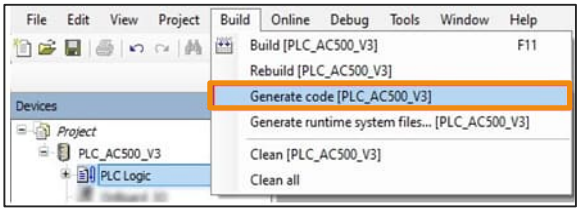
3.3.6 Logging into the CPU and download the program.

When logging into the CPU the project will be loaded into the AC500 V3 CPU. The first log-in will also load the hardware set-up.

3.3.6.1 Preparing the code

Before logging-in to the CPU, you need to compile the code without any errors.

- To do so select menu “Build -> Generate code”. The result of the compilation is shown in the “Messages” field at the bottom of the screen.

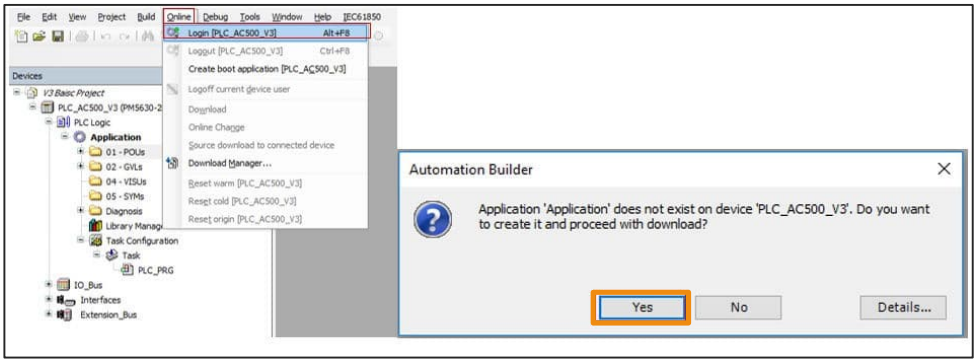


- If you skip the compiling and select “Login”, the Automation Builder will automatically trigger compiling in advance to logging-in.

3.3.6.2 Downloading to a Single CPU

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

2. A pop-up will appear, Select “Yes” to download the application to the AC500V3 CPU.



3. The PLC will be in STOP mode as shown in the status bar at the bottom of the window.

4. By default, a download generates the following actions in the CPU:

   a. The project is stored in the RAM memory.

   b. The project is stored in the flash EEPROM, if boot application was created. To force this, select menu “Online -> Create Boot Application”.

3.3.6.3 Setting the program into Run

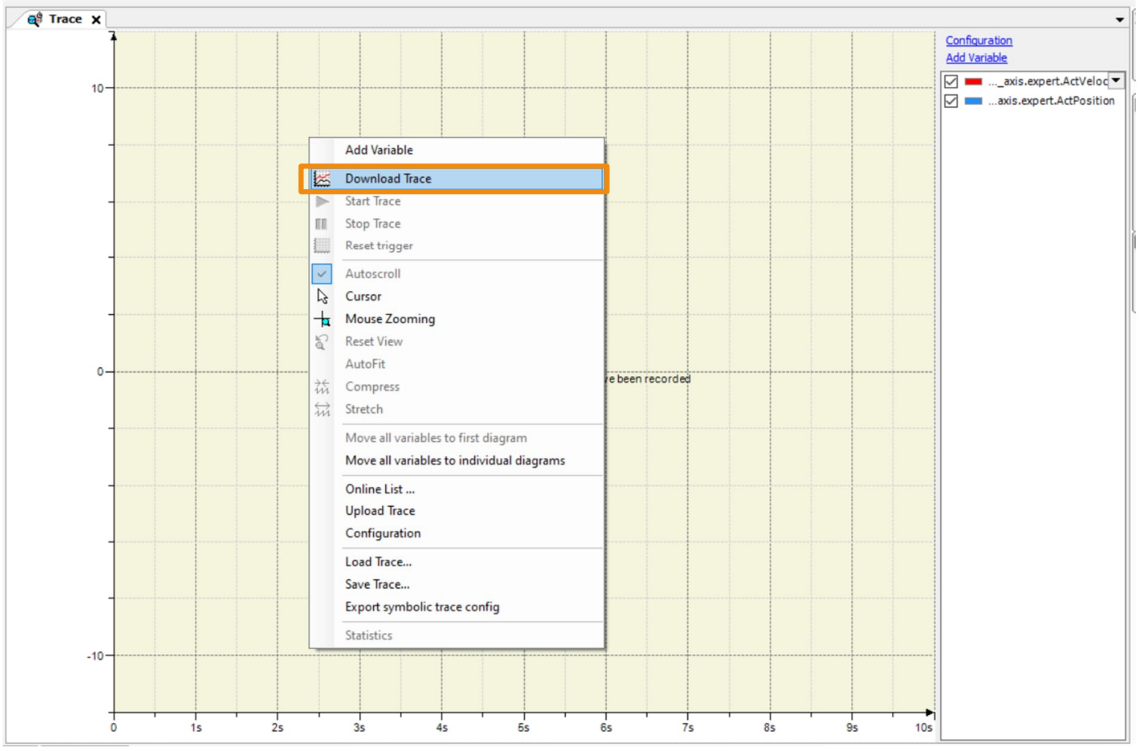
Menu select “Debug -> Start [PLC_AC500_V3]” to RUN the PLC. The status bar at the bottom of the window will show this change of status

3.3.7 Testing the program and using the program to control the Axis.

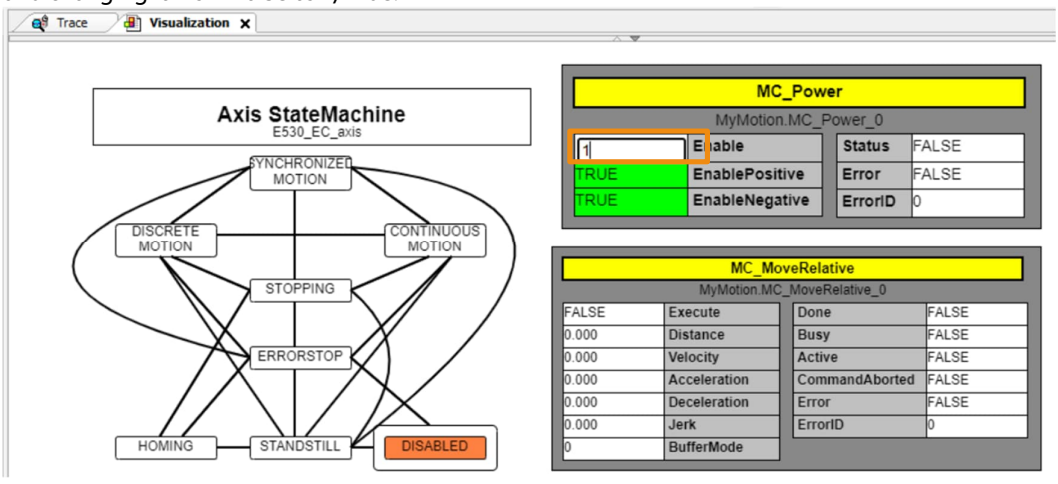
Now your project is online we can test the project you have written. To do this:

1. Navigate to the Visulaisation page. This should have the same layout as before but will now be a little more colourful showing real values in each of the boxes based on the user input but before we can use it, we must download the Trace.

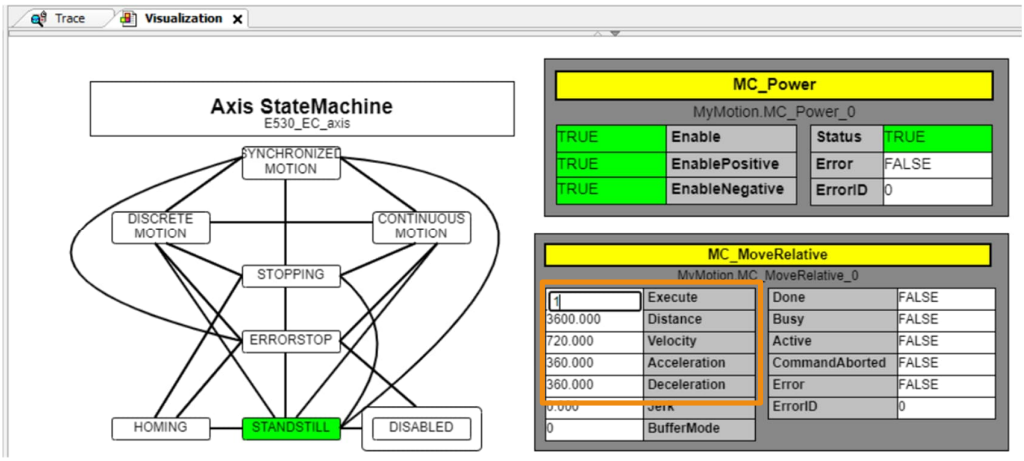
2. To do this we can naviagate to the Trace tab, Right click on the background and then select Download:



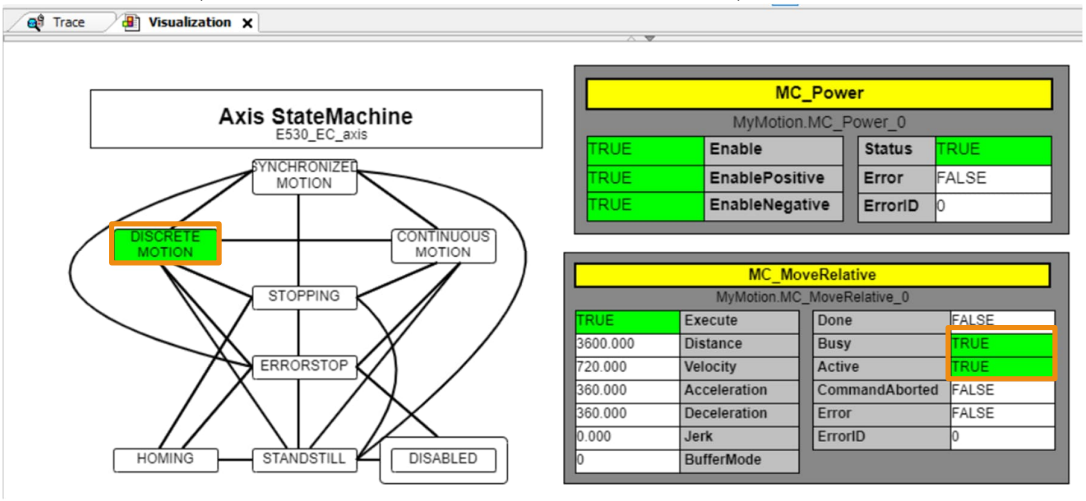
3. We can then Reopen the Visulaisation Tab and Enable the drive by clicking on the MC_Power – Enable input and changing it from False to 1/True:



4. Now the axis is enabled (MC_Power.Status = TRUE and the Axis Statemachine shows Standstill) we are ready to load the test move. As mentioned, before we have scaled our Axis Units in degrees so our settings here must reflect that:



5. When we're ready we can select Execute, for most blocks when they are processing, they will show Busy and or Active as TRUE (and the Axis Statemachine shows Discrete Motion) :

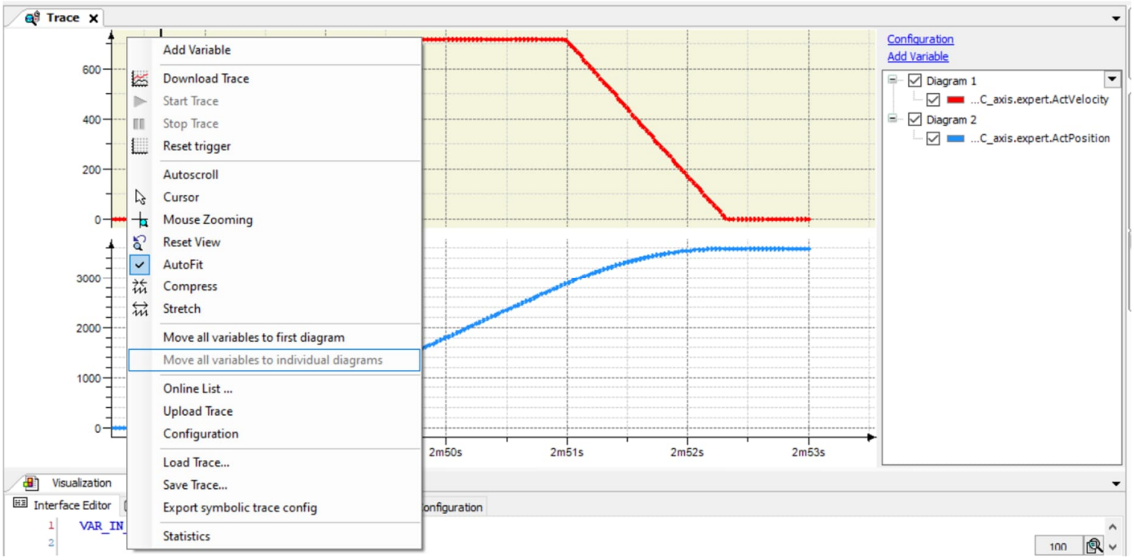


and Then Done as TRUE (and the Axis Statemachine shows Standstill) once completed.

6. Now if we go to the Trace Tab, we can see the data has been captured:

here the black line shows the trigged condition then the data after shows the defined amount of points as configured earlier (7.996 Seconds of data)

7. Right click also gives all key options to manipulate (zoom in, zoom out, stack etc), import or export the data in the Trace Tab:



8. Now the axis configuration has been completed, we can disable and log out or do further testing.

Contact us

For more information, please contact your

local ABB representative or one of the following:

new.abb.com/drives/low-voltage-ac/servo-products

new.abb.com/drives

new.abb.com/drivespartners

new.abb.com/PLC