Getting Started

A guide for using a Control Techniques drive with a CODESYS PLC over EtherCAT

Issue 1

Safety Information

Persons supervising and performing the electrical installation or maintenance of a Drive and/or an external Option Unit must be suitably qualified and competent in these duties. They should be given the opportunity to study, and if necessary, to discuss this User Guide before work is started.

The voltages present in the Drive and external Option Units are capable of inflicting a severe electric shock and may be lethal. The Stop function of the Drive does not remove dangerous voltages from the terminals of the Drive and external Option Unit. Mains supplies should be removed before any servicing work is performed.

The installation instructions should be adhered to. Any questions or doubt should be referred to the supplier of the equipment. It is the responsibility of the owner or user to ensure that the installation of the Drive and external Option Unit, and the way in which they are operated and maintained complies with the requirements of the Health and Safety at Work Act in the United Kingdom and applicable legislation and regulations and codes of practice in the UK or elsewhere.

The Drive software may incorporate an optional Auto-start facility. In order to prevent the risk of injury to personnel working on or near the motor or its driven equipment and to prevent potential damage to equipment, users and operators, all necessary precautions must be taken if operating the Drive in this mode.

The Stop and Start inputs of the Drive should not be relied upon to ensure safety of personnel. If a safety hazard could exist from unexpected starting of the Drive, an interlock should be installed to prevent the motor being inadvertently started.

General Information

The manufacturer accepts no liability for any consequences resulting from inappropriate, negligent or incorrect installation or adjustment of the optional operating parameters of the equipment or from mismatching the variable speed drive (Drive) with the motor.

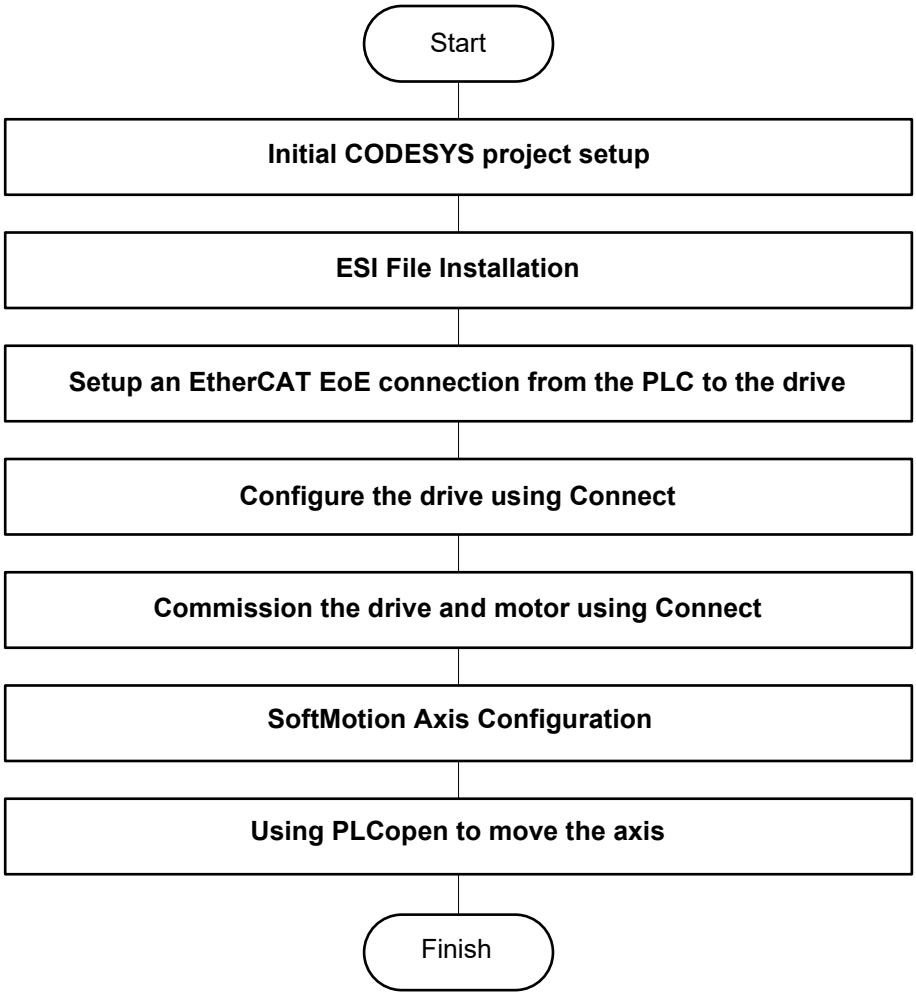
The contents of this guide are believed to be correct at the time of printing. In the interests of a commitment to a policy of continuous development and improvement, the manufacturer reserves the right to change the specification of the product or its performance, or the contents of this guide, without notice.

Table of Contents

1 Introduction

This document guides the user through the required steps, within Control Techniques Connect software and the CODESYS IDE, to get a Softmotion axis working with a Control Techniques drive over EtherCAT.

The steps required to get the axis working are as follows:



The end result is a Control Techniques drive that can be used with the Softmotion PLCopen function blocks included with CODESYS.

It is recommended to use the latest version of the CODESYS programming environment when making new projects.

HINT: The ESI files **must** match the firmware of the EtherCAT module. Section **3.9 How to upgrade the EtherCAT interface firmware using Connect**.

HINT: The information in this manual is backed up by training videos on YouTube.

Nidec Drives Support - YouTube

1.1 Minimum requirements

The minimum requirements for the PLC / IPC / Controller and programming environment are:

- A minimum of CODESYS V3.5 SP21 Patch 2 programming environment installed on the programming PC.
- The PLC / IPC / Controller must support EtherCAT communications.
- A suitable EtherCAT connection between the drive and PLC / IPC / Controller.
- An Ethernet connection between the PLC / IPC / Controller and the programming PC.
- At least 1 drive axis that has been configured using the setup described in section **2 Getting started**.
- A device description .xml file is required for the PLC / IPC / Controller.

HINT: The CT SoftMotion Interface Library may be used with any CODESYS compliant PLC / IPC / Controller, however the following Control Techniques controllers and IPCs are recommended:

MCz 201

MCe 200

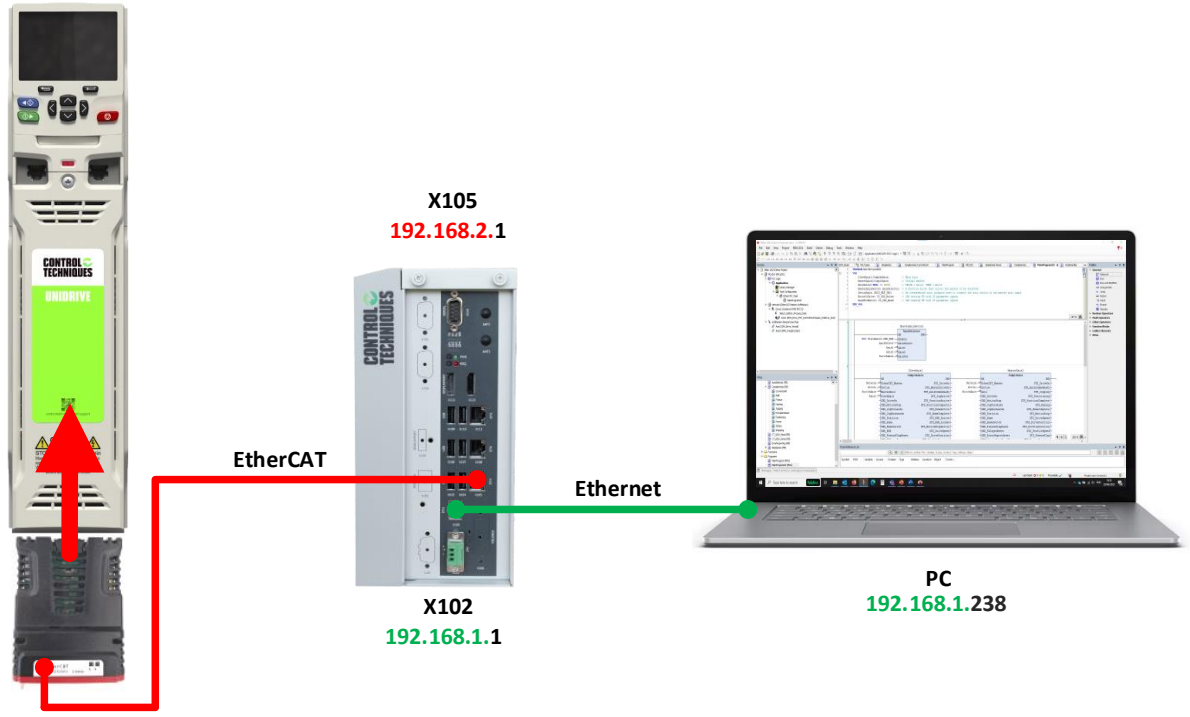
MCz 601

2 Getting started

This section describes how to commission the drive and implement a Softmotion axis in CODESYS. It is assumed that before reading this section the user has wired up the drive, in accordance with the relevant user guide, for the drive selected and the control required by the application including electrical, control and communications connections. It is further assumed that the drive has been powered up, in accordance with the relevant electrical and machine safety regulations, and that it functions correctly.

2.1 Initial CODESYS project setup

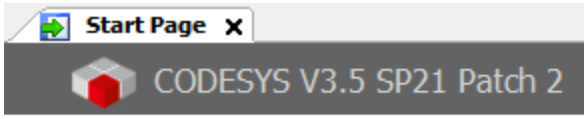
Before starting the EtherCAT network configuration, a connection must be established between the PC and the PLC / IPC / Controller. It is assumed that the PC and PLC / IPC / Controller have connected Ethernet ports and that the IP address range for the hardware is in the same range e.g. for an MCz controller from Control Techniques port X102 has the IP address 192.168.1.1 so the PC must have a unique address in this subnet such as 192.168.1.238.

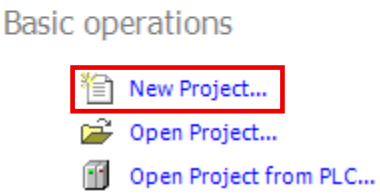


This section shows the first steps required to get connected to a CODESYS PLC / Controller / IPC.

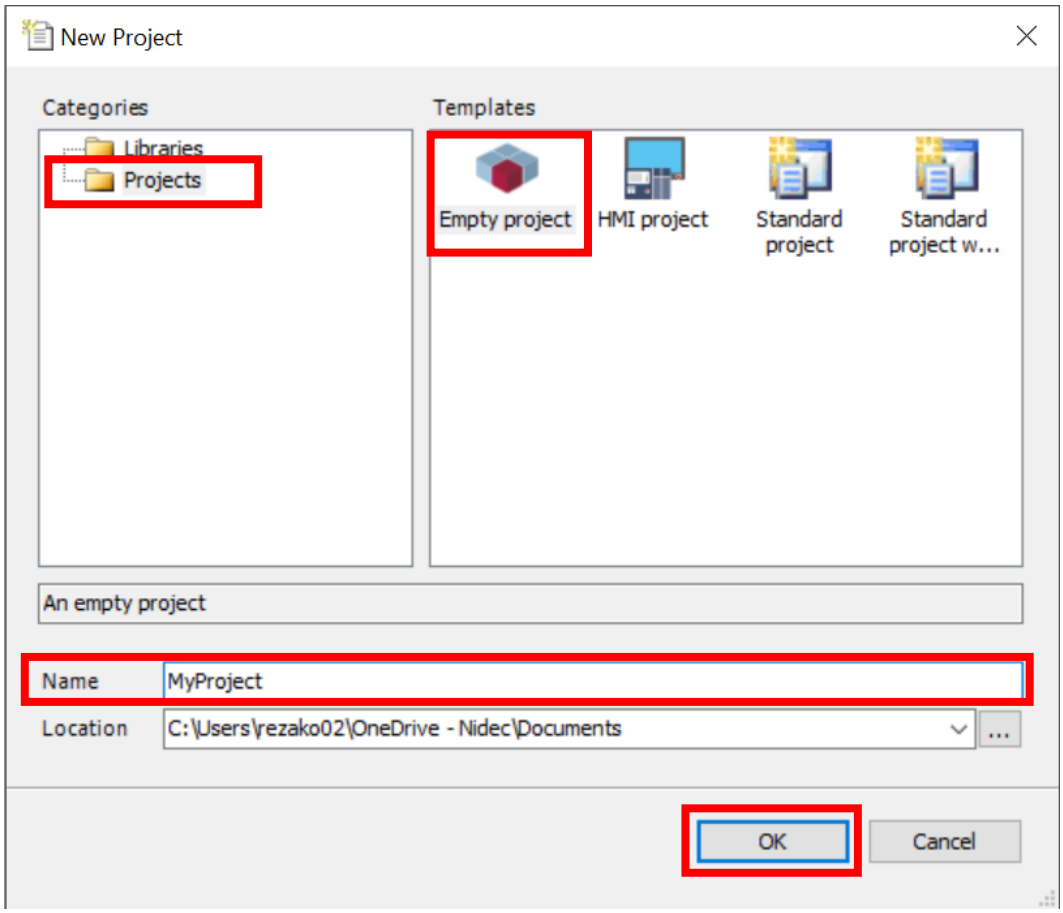
1. Open CODESYS.

2. When the start page is shown, select “New Project…”.

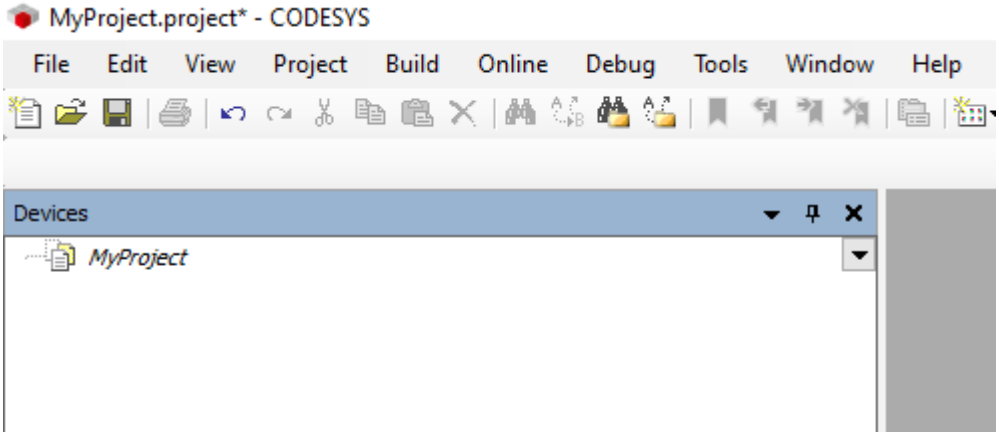




3. Select “Projects” > “Empty project”. Then name the project, and press “OK” when complete.

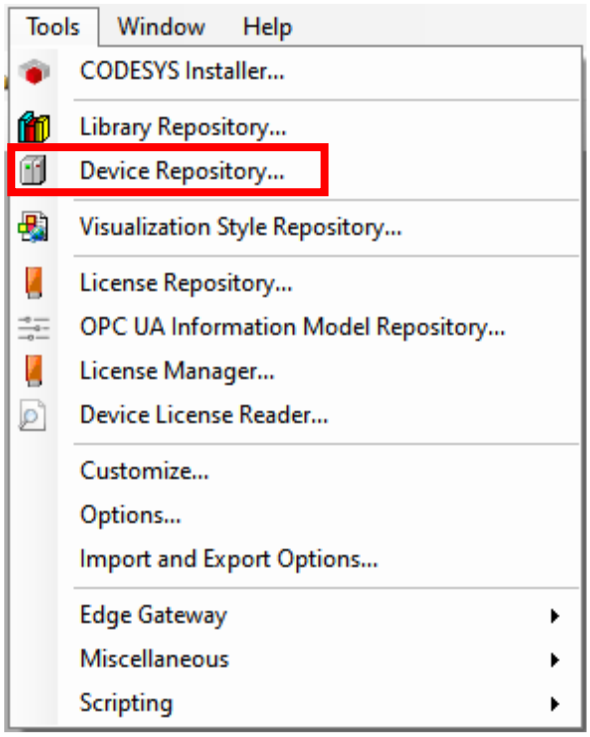


4. A blank project is created.

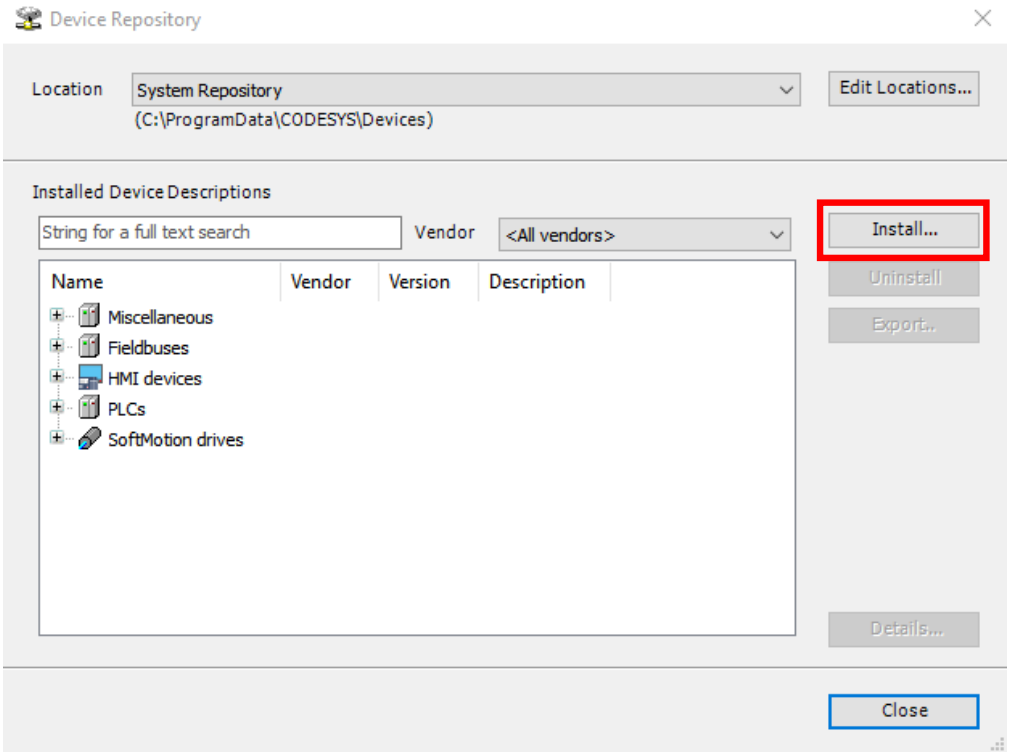


5. Jump to step 9 if the PLC / IPC / Controller to act as the EtherCAT network master has already been installed into the CODESYS device repository.

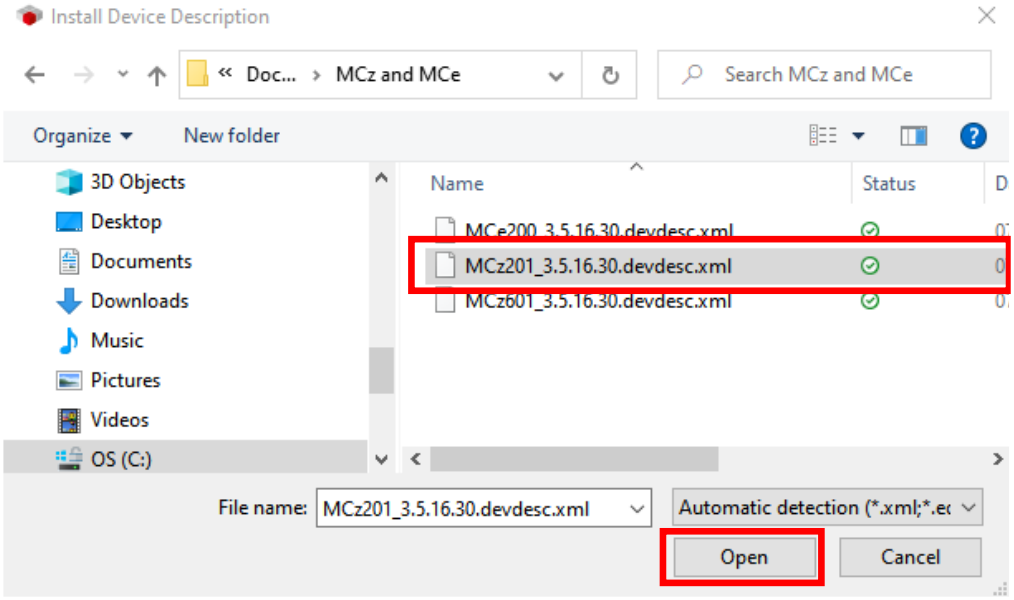
Otherwise, from the CODESYS tool bar select “Tools” > “Device Repository…”.



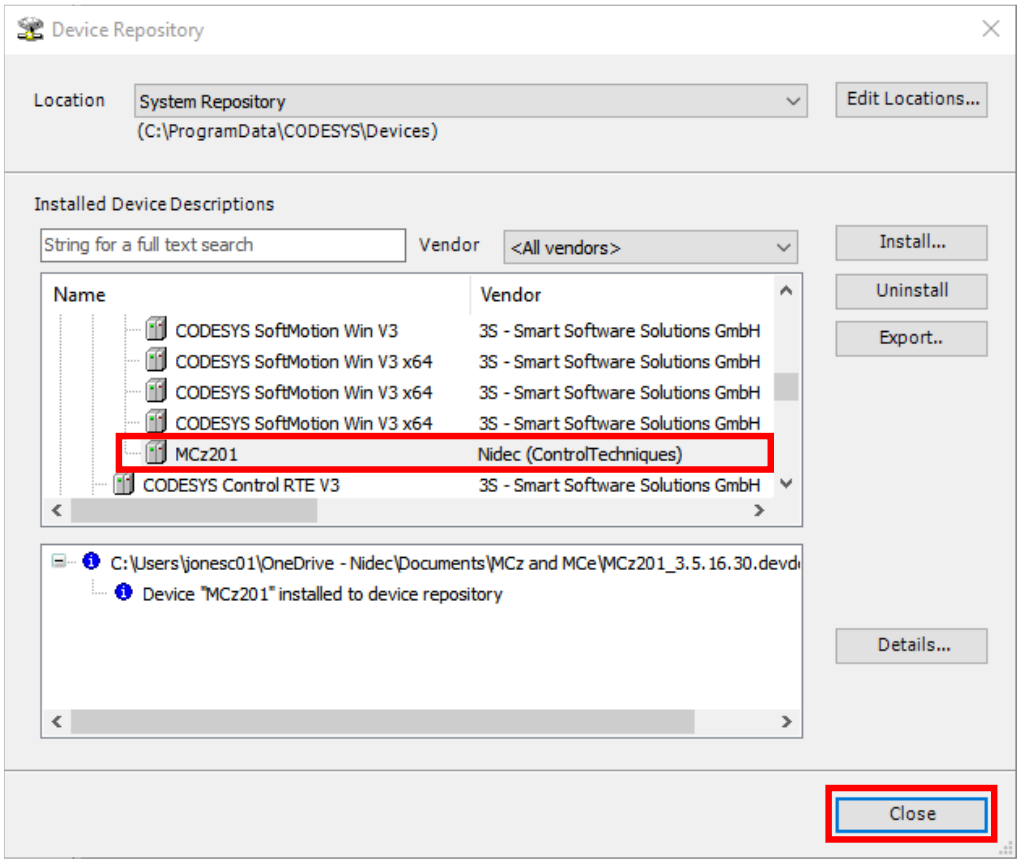
6. When the “Device Repository” dialog appears, select “Install…”



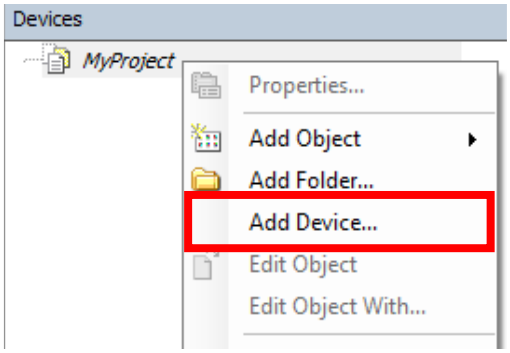
7. Browse for the location of the device description file to install, select the file, and then click “Open”.



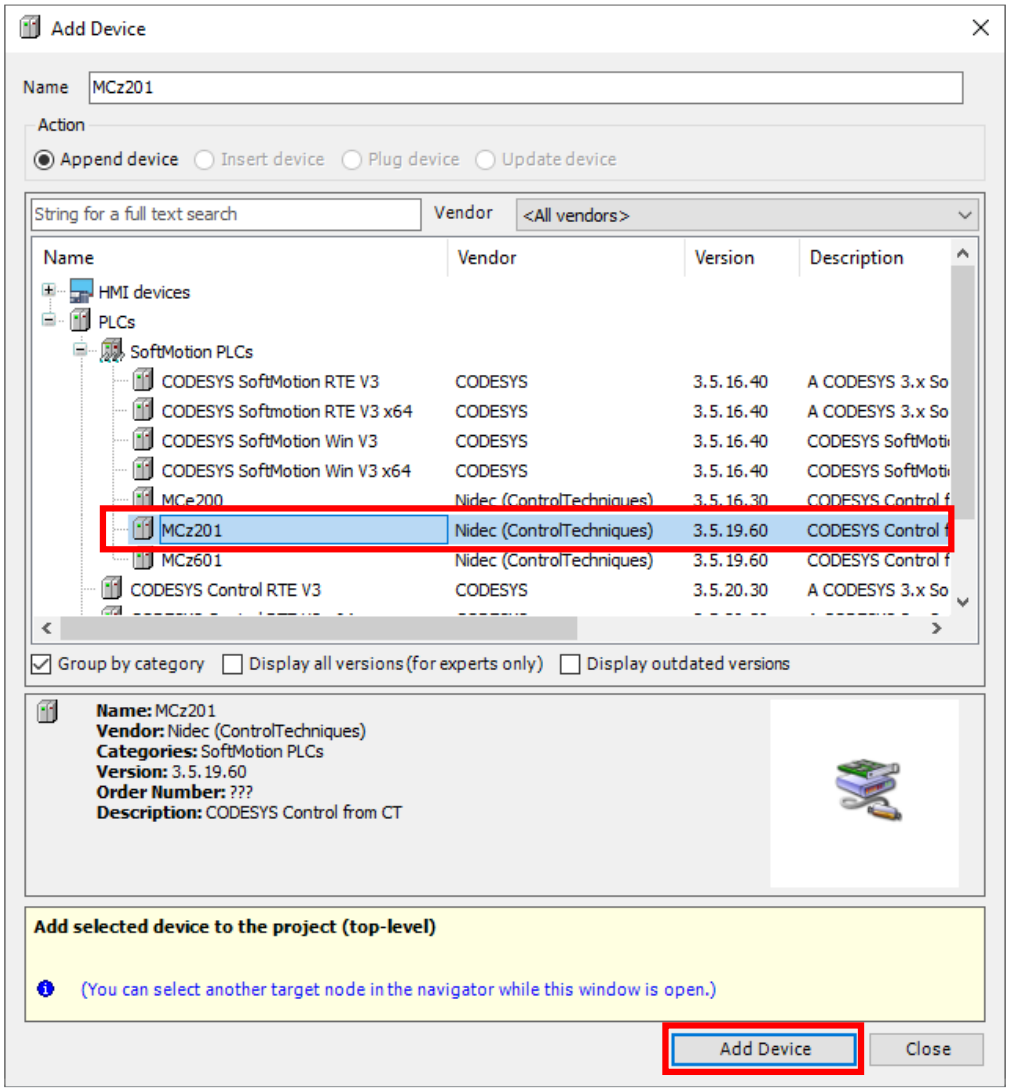
8. The device is now added to the device repository and may be added to a CODESYS project. Click close when the device is added.



9. Add the PLC / IPC / Controller to the project tree by right clicking on the project name and selecting “Add Device…”.



10. The “Add Device” dialog box will appear. Select the PLC / IPC / Controller to act as the EtherCAT network master click the “Add Device” button.

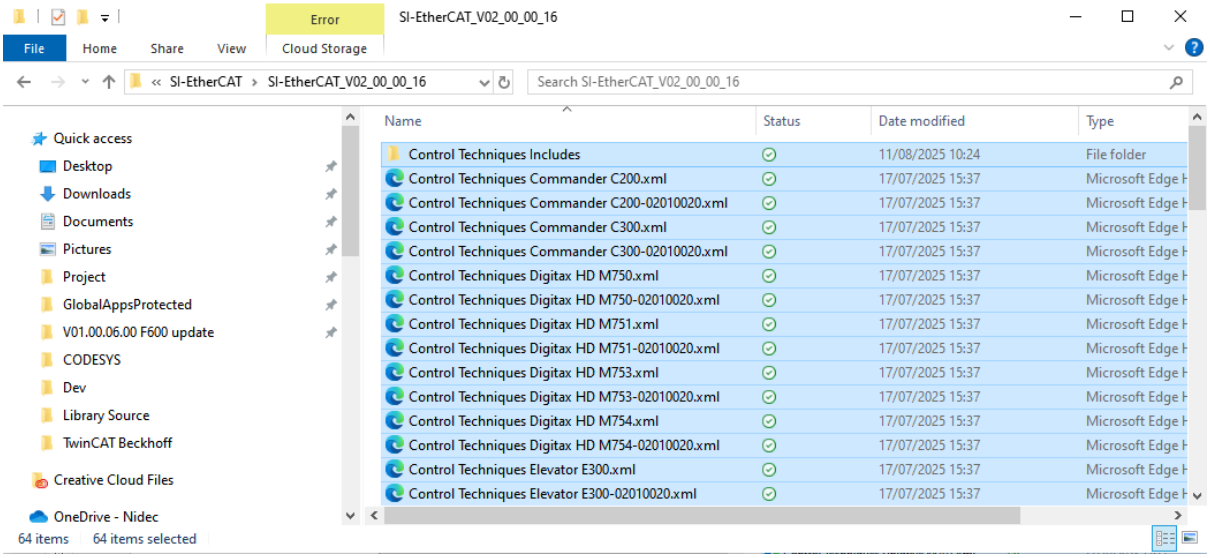


2.2 ESI File Installation

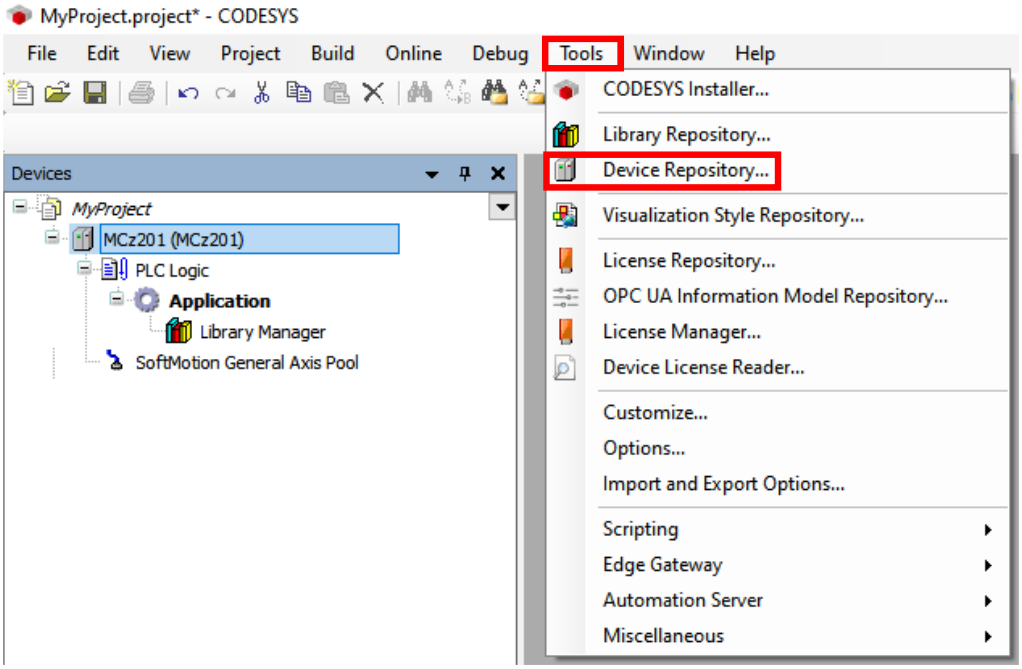
This section details how to install the ESI files into CODESYS. It is expected that section **2.1 Initial CODESYS project setup** has been completed and that CODESYS is open. If the CODESYS installation already has the ESI files for Control Techniques devices jump to section **2.3 Setup an EtherCAT EoE connection from the PLC to the drive**.

1. Get the latest ESI files from the Control Techniques section of the Nidec Drives website and extract the contents of the .zip file.

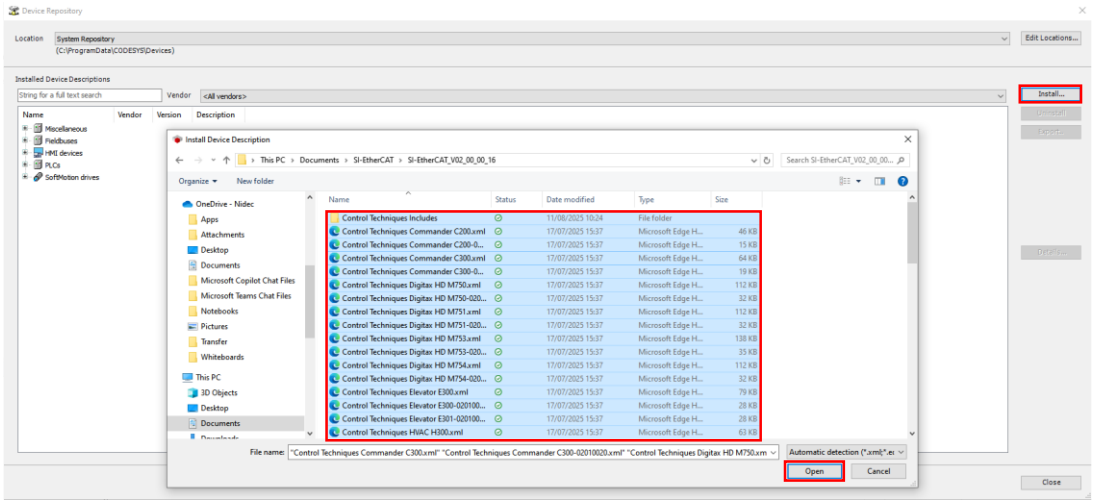
2. Extract the contents of the ESI file zip.



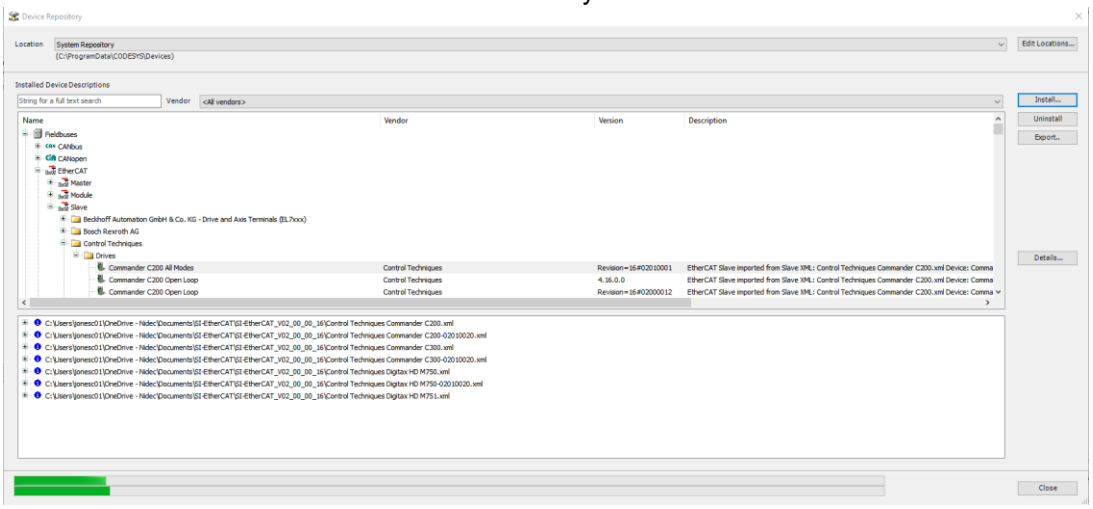
3. In CODESYS, open the “Device Repository” by selecting “Tools” > “Device Repository…”.



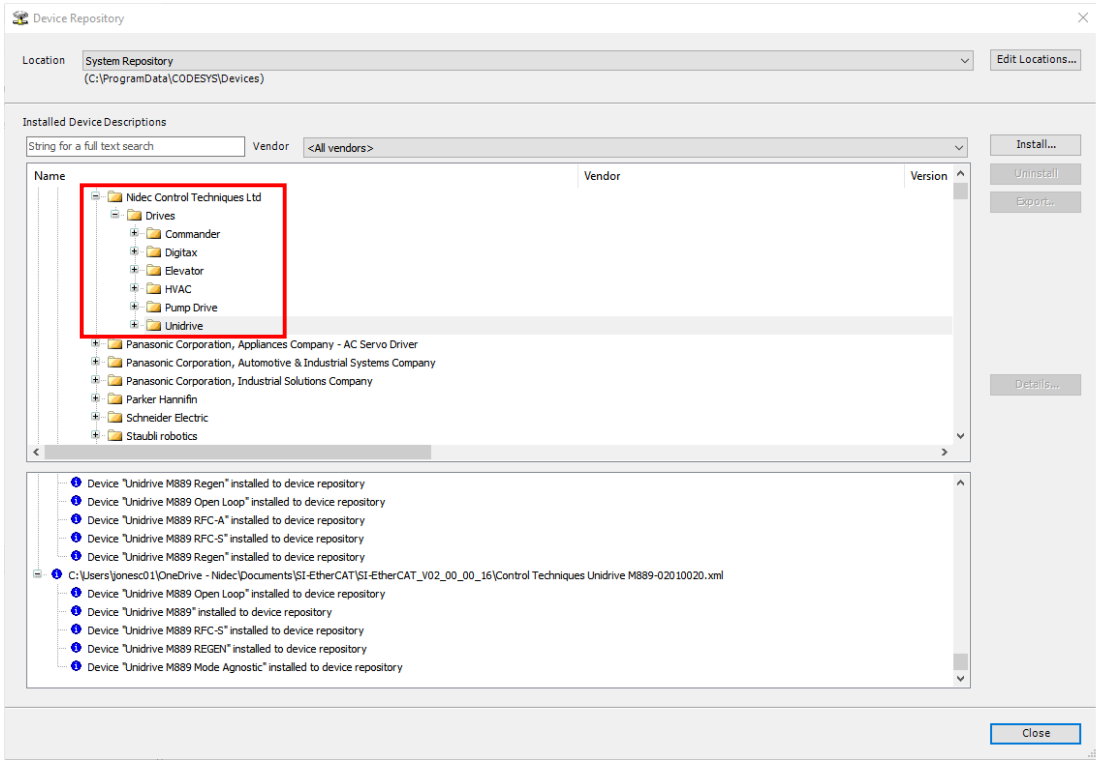
4. When the “Device Repository” opens select “Install…” and then use the file browser to find the folder where the ESI files were extracted. Highlight all of the files in folder, (Ctrl+A), and then click “Open”.



5. CODESYS will install all of the ESI files into the system.



When the installation completes the new ESI files will be found under “Fieldbuses” > “EtherCAT” > “Slave” > “Nidec Control Techniques Ltd”. For older ESI files including the CODESYS standard files for Control Techniques they will be found under “Fieldbuses” > “EtherCAT” > “Slave” > “Control Techniques”.

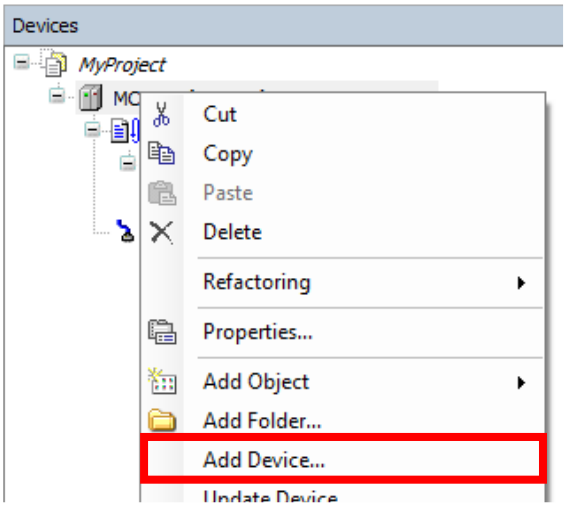


HINT: The ESI files **must** match the firmware of the EtherCAT module. Section **3.9 How to upgrade the EtherCAT interface firmware using Connect**. This section describes how to check the current SI-EtherCAT firmware version and how to perform an upgrade if necessary.

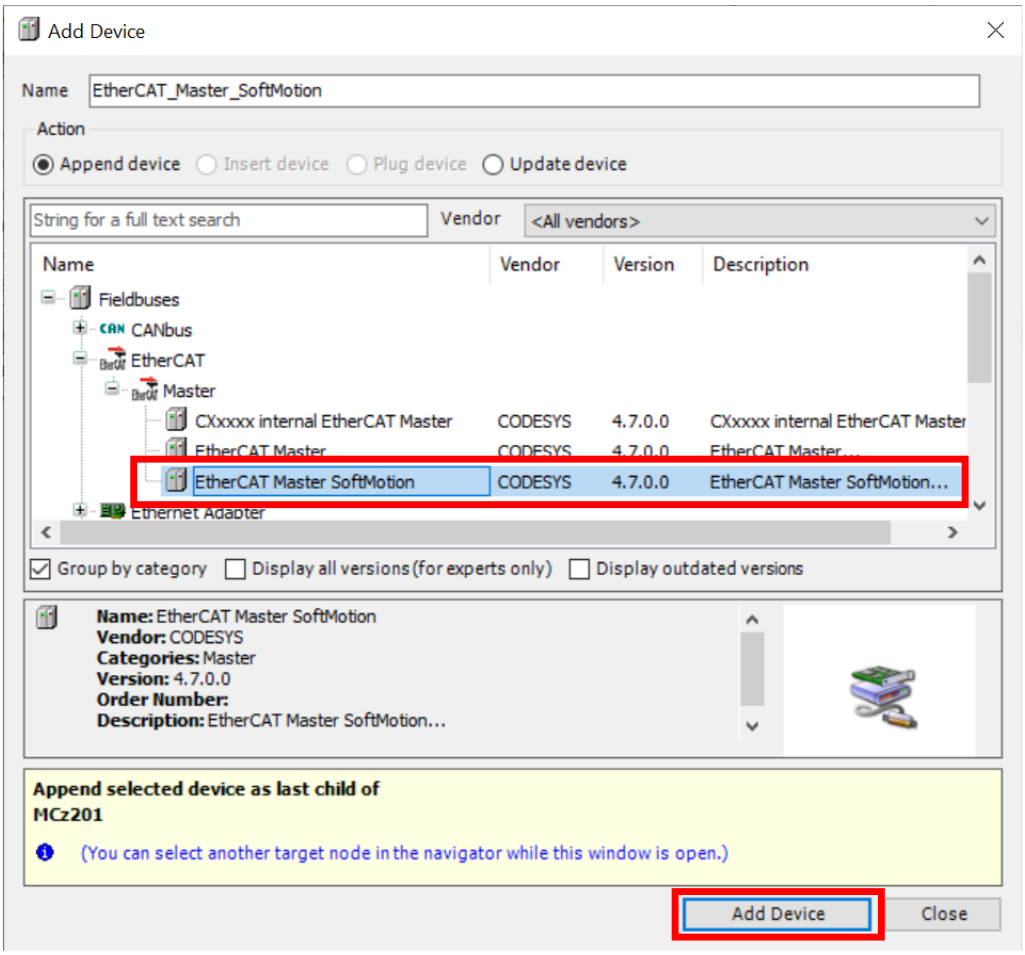
2.3 Setup an EtherCAT EoE connection from the PLC to the drive

To allow Connect or any other Ethernet based PC tool to connect to the slave drives in the EtherCAT network, an EoE (Ethernet Over EtherCAT) connection must be configured. It is assumed that sections Use the following steps to configure the EoE network:

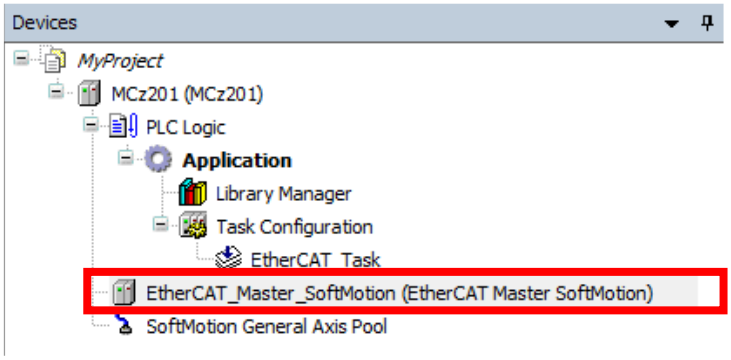
1. Add an EtherCAT master to the controller by right clicking on the controller and then select “Add device…”



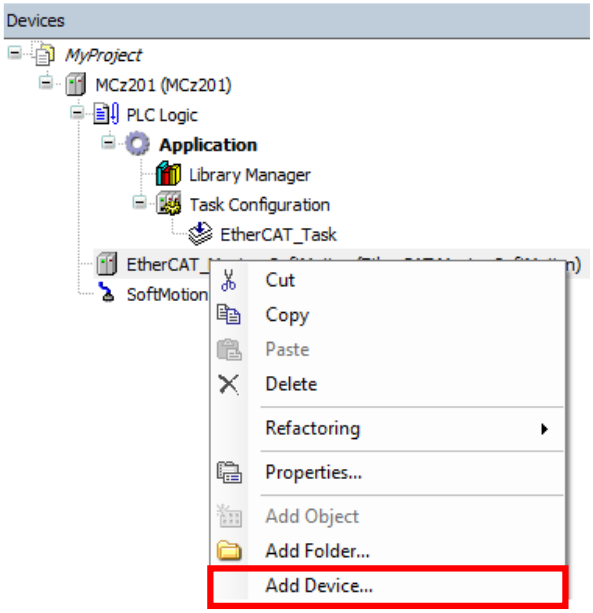
2. When the “Add Device” dialog starts, select “Fieldbuses” > “EtherCAT” > “Master” > “EtherCAT Master SoftMotion”, then click “Add Device”. Use the latest version.



The EtherCAT master has been added to the project tree:

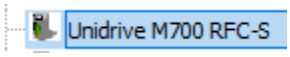


3. To add the slave device by scanning the network
4. To add the EtherCAT slave device manually, (Control Techniques drive), to the project tree by right clicking on the EtherCAT Master device and selecting “Add Device…”.

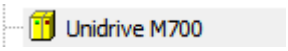


HINT: Control Techniques drives device description files are included with CODESYS but are also available from Support Suite or a Control Techniques Drive Centre or Distributor.

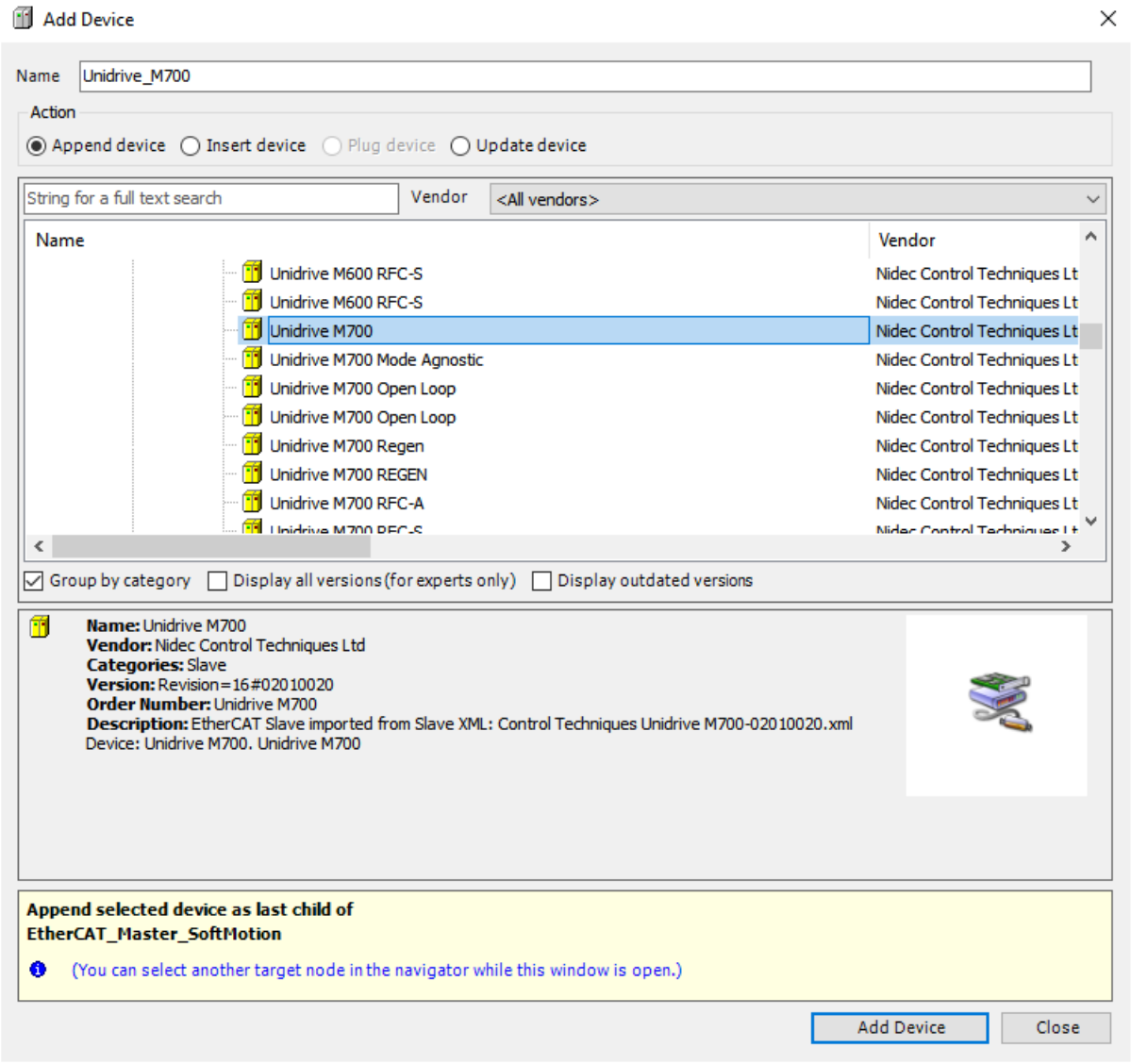
The “Add Device” dialog box will open. For older V1 SI-EtherCAT and factory fitted EtherCAT interfaces if the drive is a Unidrive M700 running in RFC-S mode for example, then the “Unidrive M700 RFC-S” device is selected using V4.16.0.0 in the Control Techniques folder.



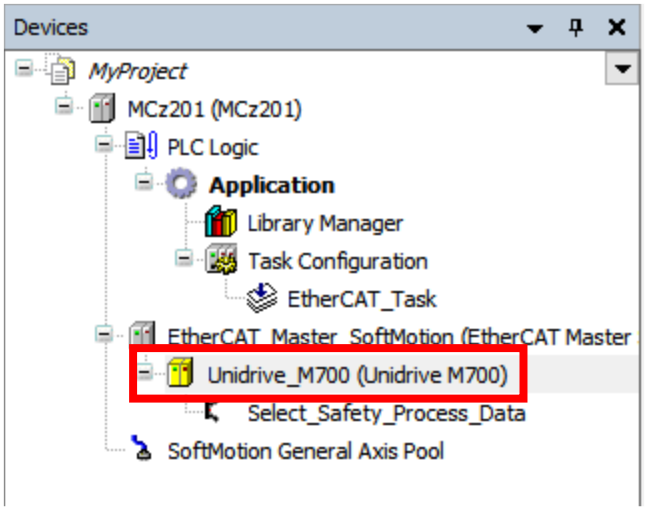
For newer V2 SI-EtherCAT and factory fitted EtherCAT interfaces if the drive is a Unidrive M700 running in any mode for example, then the “Unidrive M700” device is selected using the latest version in the Nidec Control Techniques Ltd folder.



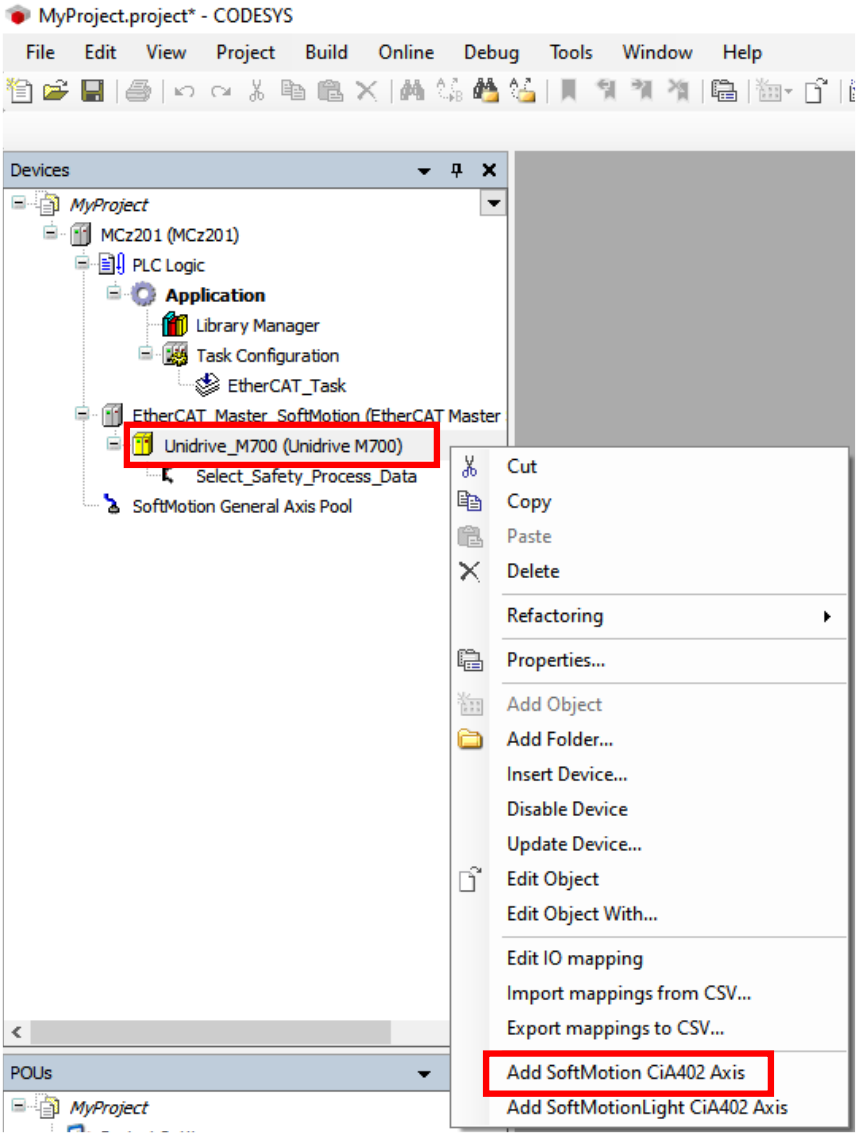
From the “Add Device” dialog, select “Fieldbuses” > “EtherCAT” > “Slave” > “Control Techniques” > “Drives” > ” Nidec Control Techniques Ltd” > “Unidrive” > “Unidrive M700”, and then select “Add Device”.



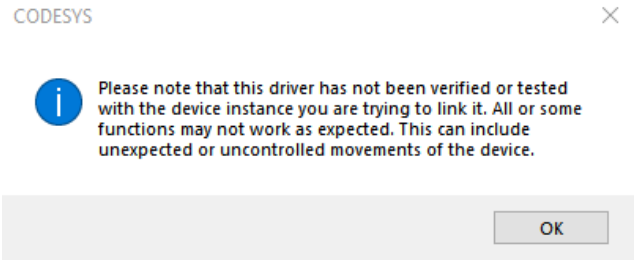
6. The device is added to the project tree:



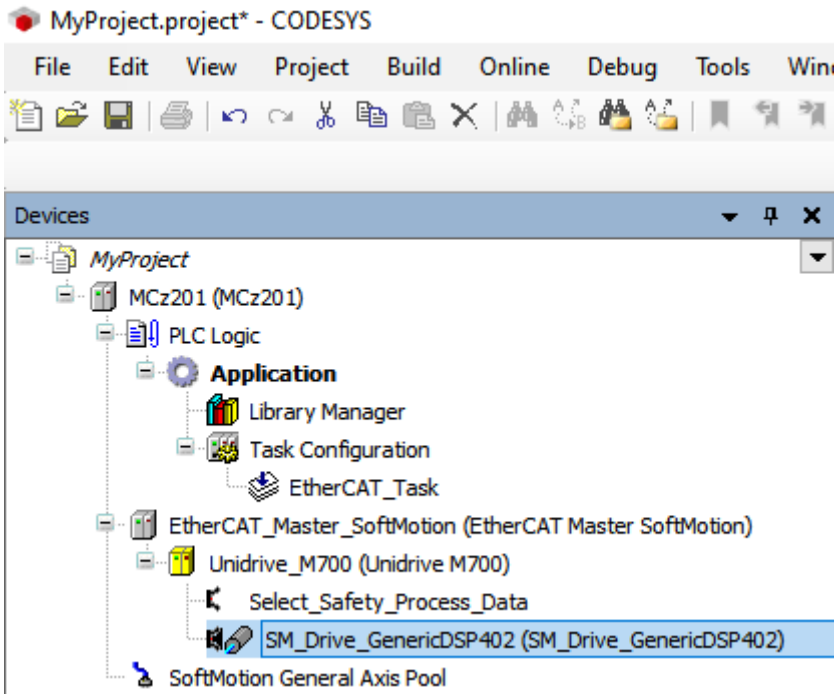
7. If the drive doesn't have SM3_Drive_ETC Softmotion axis beneath it in the tree one must be added by right clicking on the drive and then select "Add Softmotion CiA402 Axis"



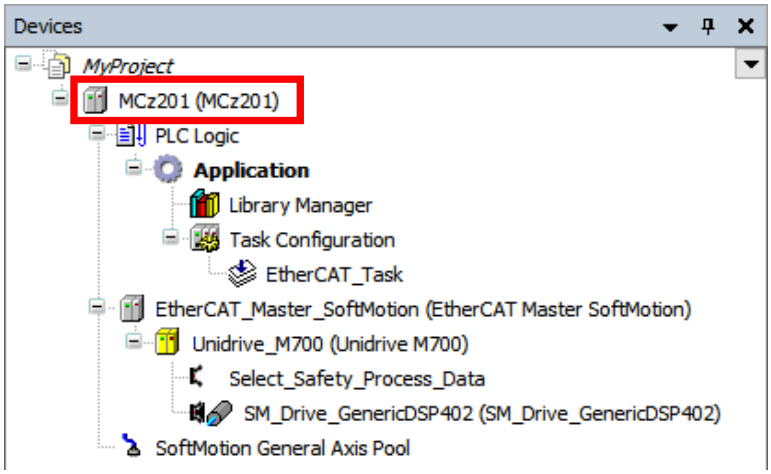
Click OK to the message that follows:



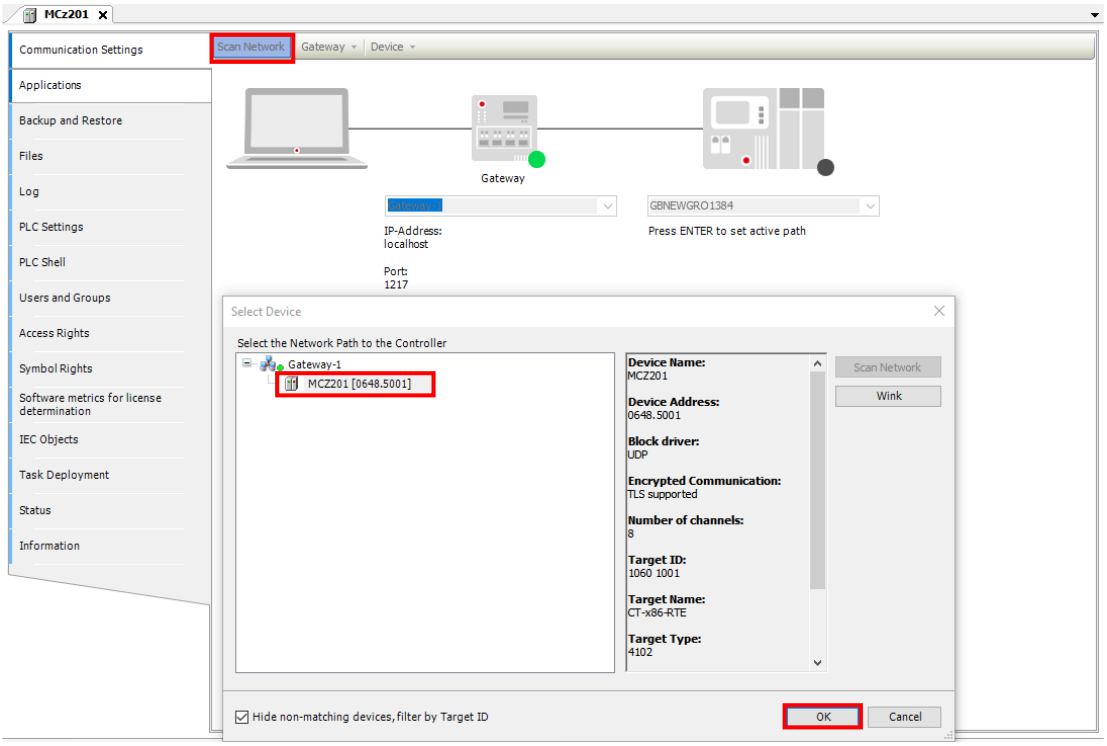
A generic axis is added to the project:


MyProject.project* - CODESYS
File
Edit
View
Project
Build
Online
Debug
Tools
Devices
MyProject
MCz201 (MCz201)
PLC Logic
Application
Library Manager
Task Configuration
EtherCAT_Task
EtherCAT_Master_SoftMotion (EtherCAT Master SoftMotion)
Unidrive_M700 (Unidrive M700)
Select_Safety_Process_Data
SM_Drive_GenericDSP402 (SM_Drive_GenericDSP402)
SoftMotion General Axis Pool

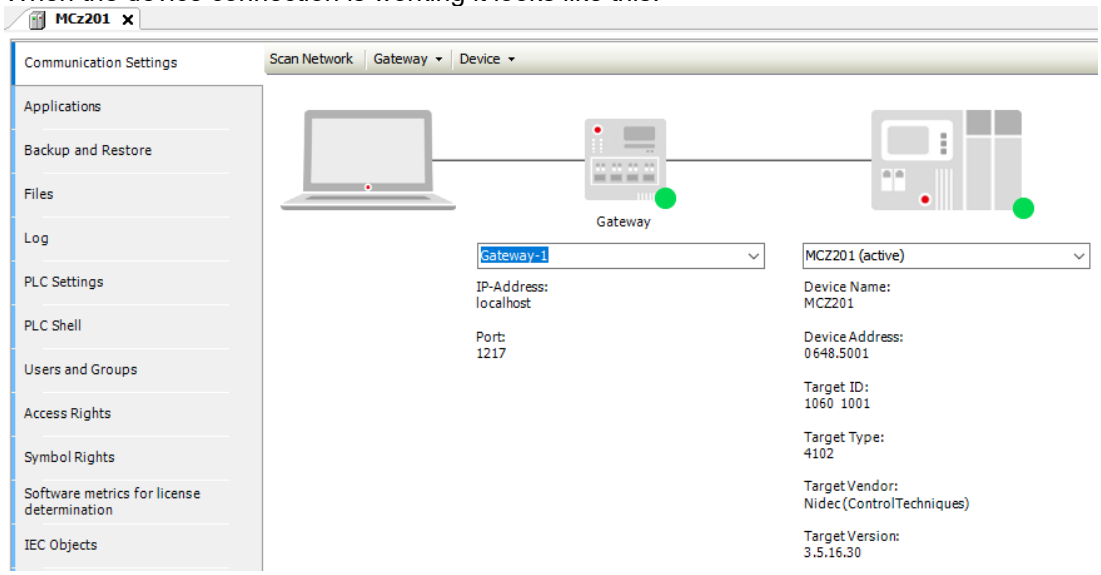
8. Connect to the target PLC / IPC / Controller by double clicking on it in the Devices pane:



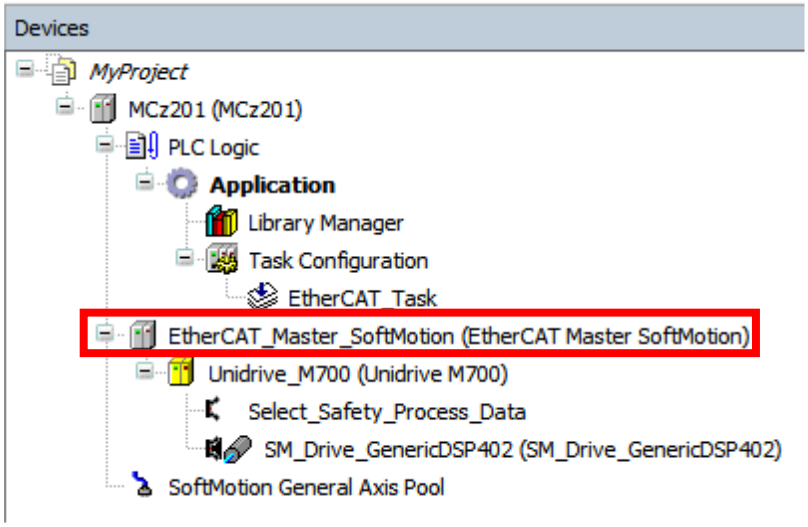
When the device tab opens, click “Scan Network…”. When the scan finishes select the device to connect to from the list and select OK.



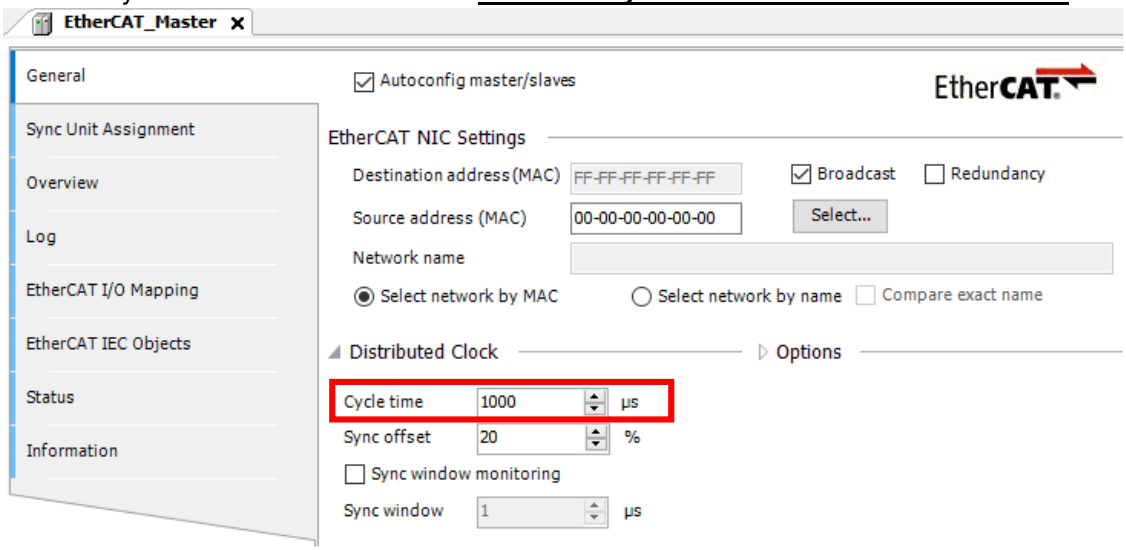
When the device connection is working it looks like this:



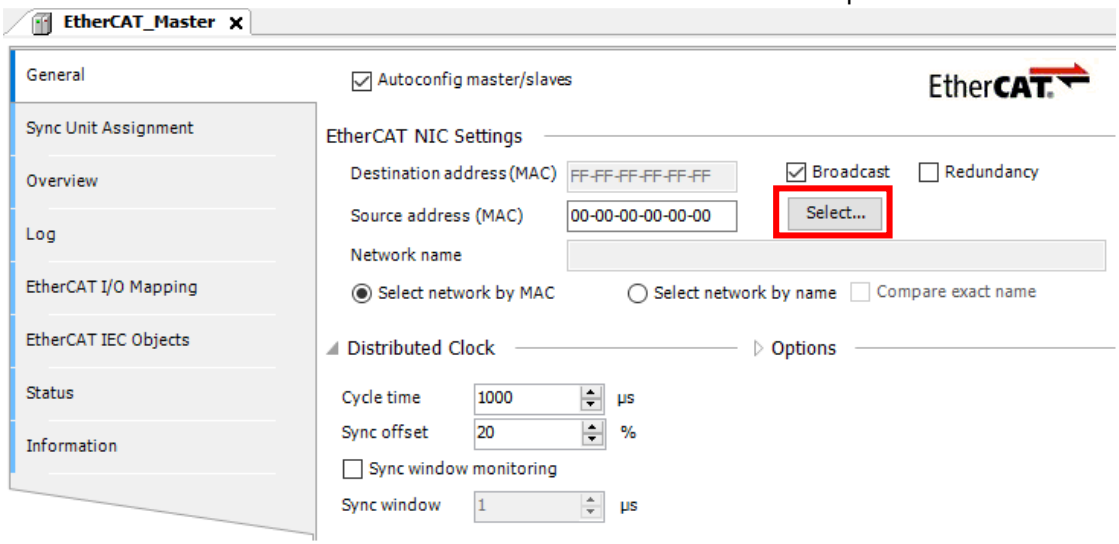
9. Configure the EtherCAT master settings. Double click on “(EtherCAT Master SoftMotion)” in the “Devices” tree:



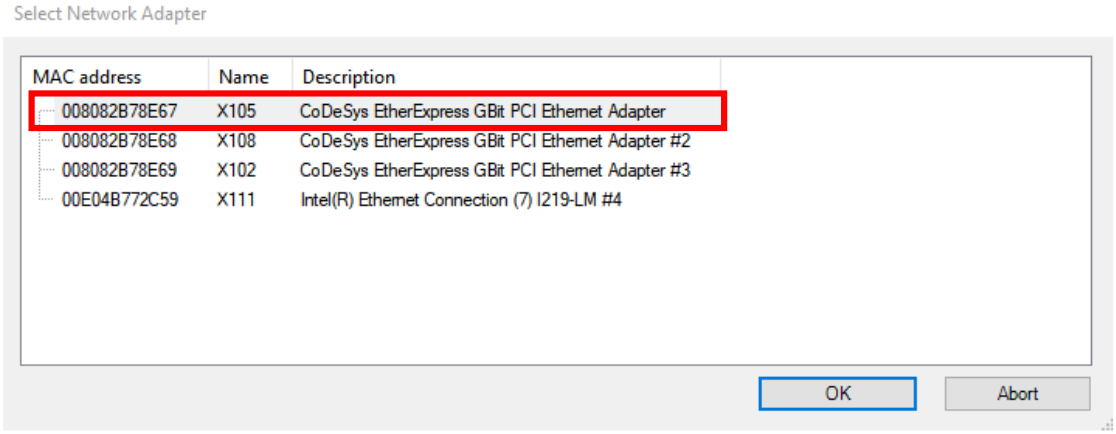
When the “EtherCAT_Master SoftMotion” tab opens, set the cycle time to 1000us – this is the maximum recommended EtherCAT cycle time for any axis taking part in an Electronic Gearbox system as a master or a slave. **The same cycle time must be used on all axes.**



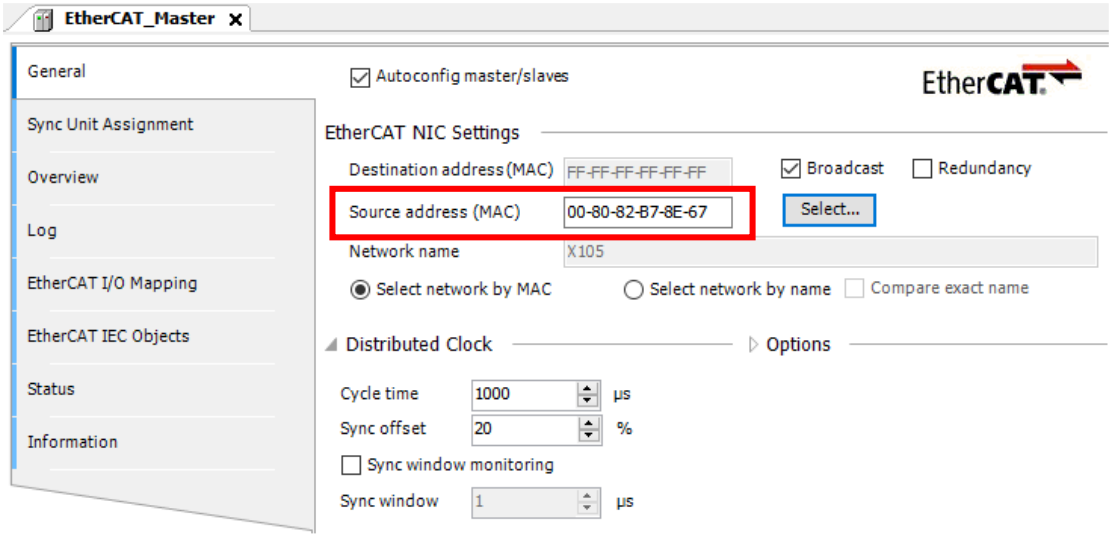
Click “Select…” to choose the PLC / IPC / Controller EtherCAT network port:



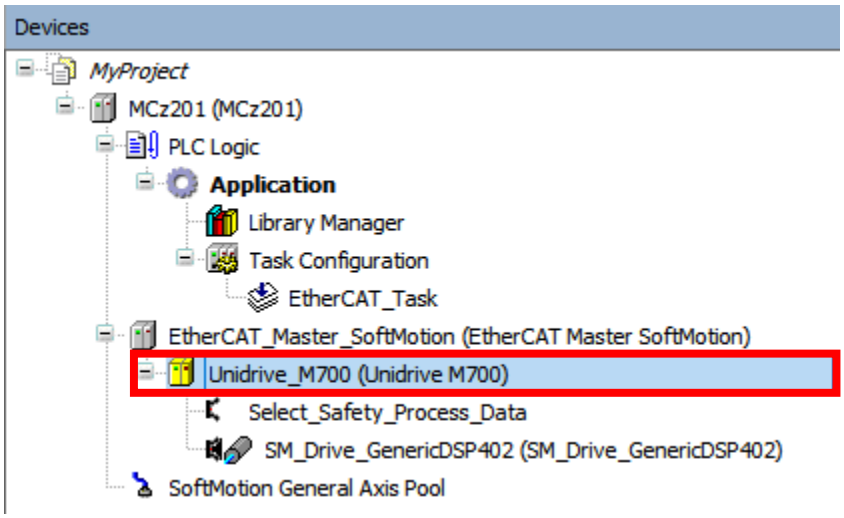
Select the port from the list and click “OK”. In the example below it is assumed that port X105 is the EtherCAT network port.



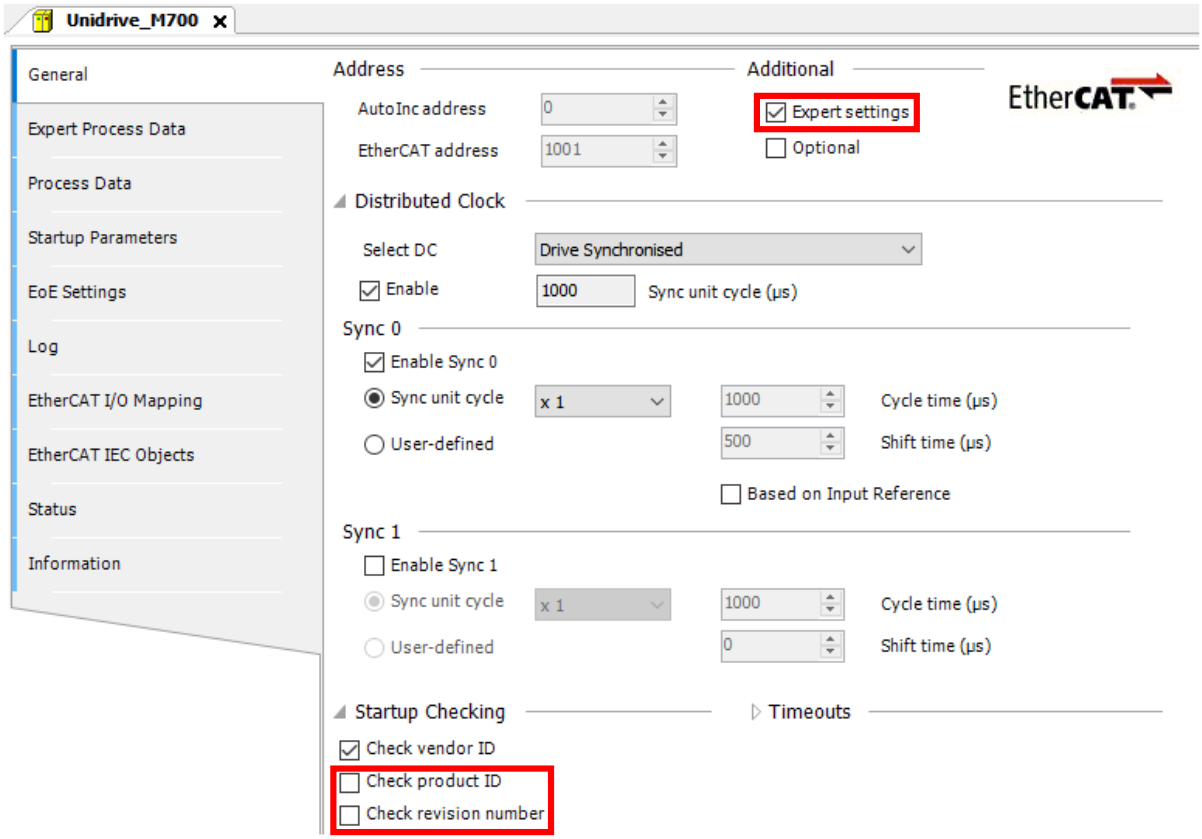
“Source address (MAC)” is now configured.



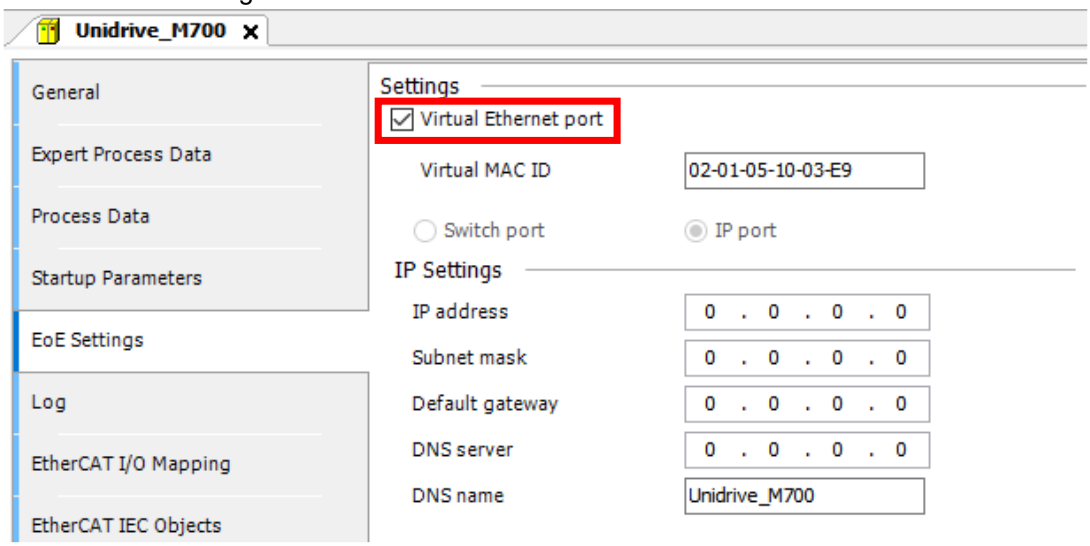
10. Setup the EtherCAT Slave EoE Connection. Double click on the slave device in the “Devices” tree:



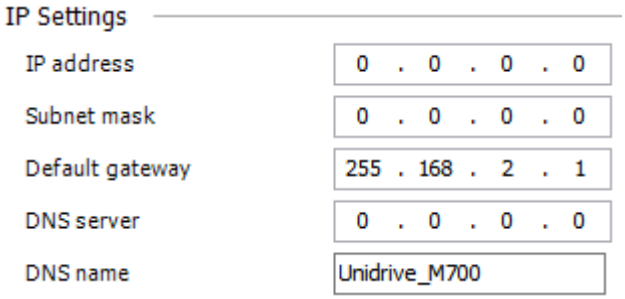
On the EtherCAT slave tab, select “Expert Settings” to reveal the EoE tab. Under the “Startup Checking” category, untick the “Check product ID and “Check revision number” boxes:



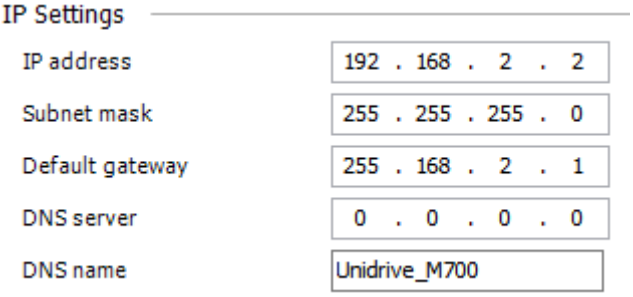
11. Select the “EoE Settings” side tab and check the “Virtual Ethernet Port” box to enable EoE and reveal the settings.



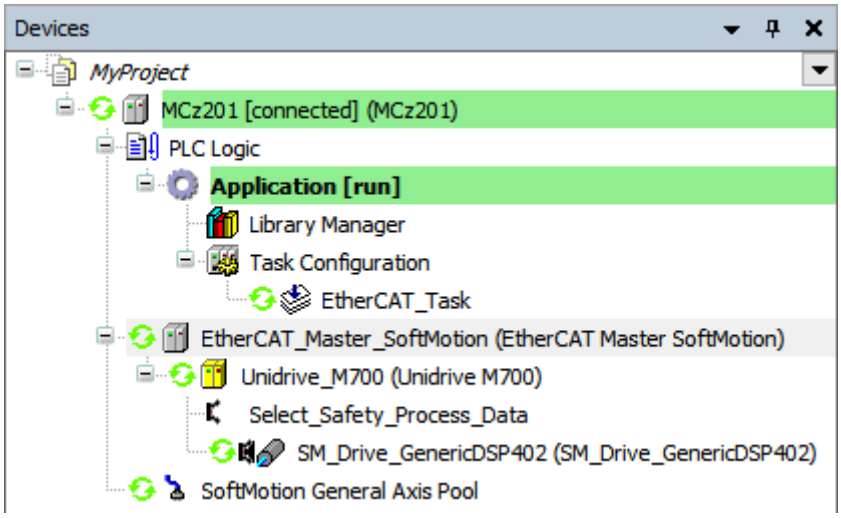
12. Find the EtherCAT master Ethernet port IP address e.g. for X105 on an MCz201 it is 192.168.2.1. Set this as the Default Gateway.



Configure the IP address of the EoE port by setting “IP Address” to the IP address that will be used by the PC tools to communicate to the slave drive e.g. 192.168.2.2. “IP Address” must be in the same subnet as the default gateway but must have a different address. Set the subnet mask to 255.255.255.0.

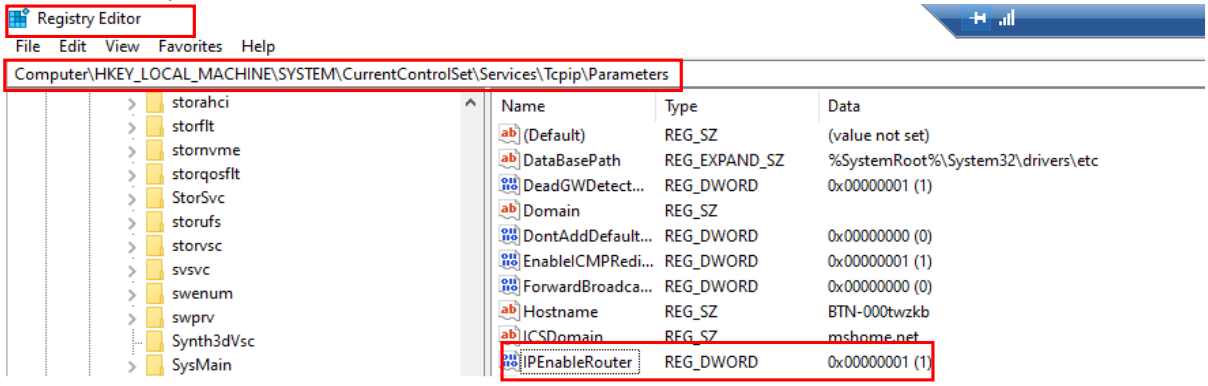


13. Compile and download the setup to the target using the login button or Ctrl+F8. When the project has downloaded and the EtherCAT has started properly it looks like this in CODESYS:



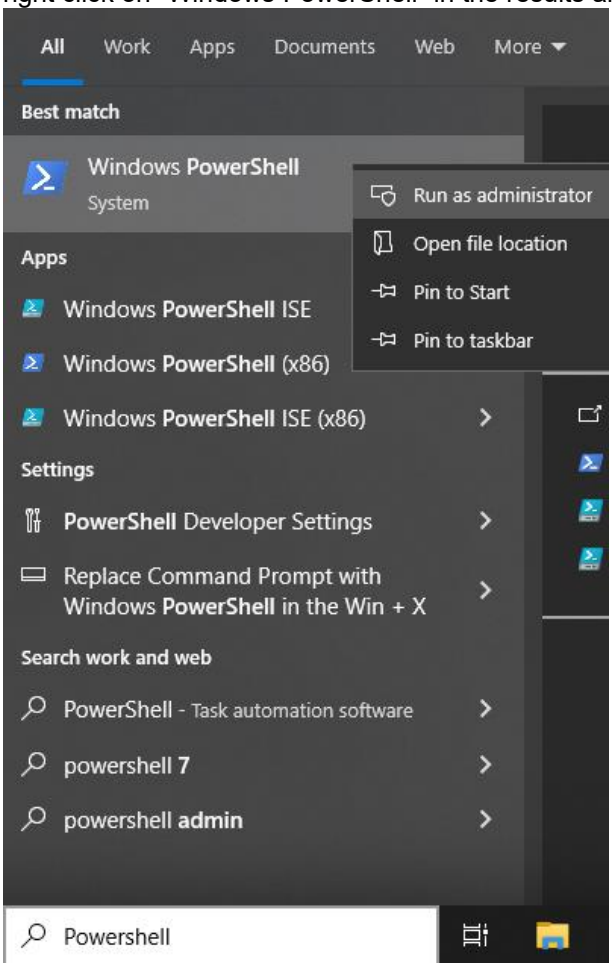
HINT: The green circling arrows next to the EtherCAT Master and Slave drive that show that the network is healthy.

14. For targets that are **not** MCz or MCe, it is worth verifying that the registry in the IPC/Controller/PLC has IPEnableRouter set to 0x00000001(1). This can be accessed using remote desktop.

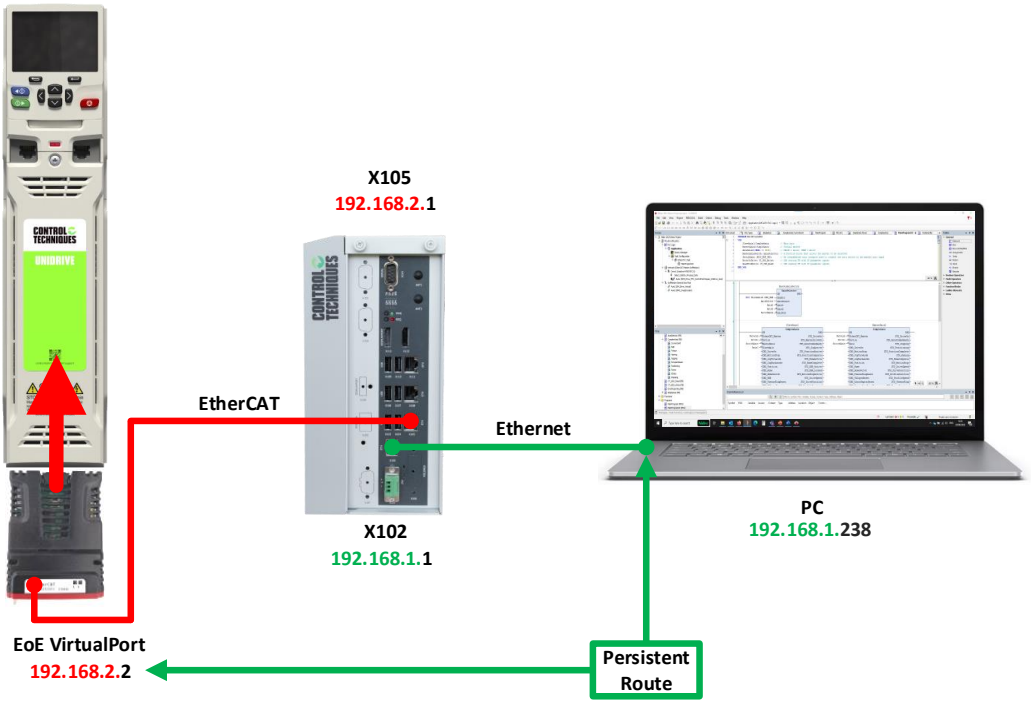


Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

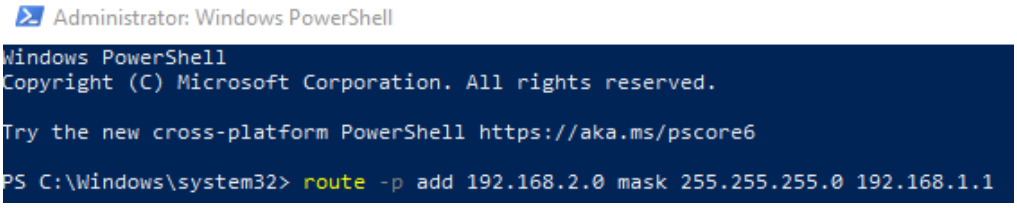
15. Create a persistent route in the PC to allow access to the EtherCAT Slave drive. Open the command prompt as an administrator by typing in “PowerShell” in the windows search, then right click on “Windows PowerShell” in the results and select “Run as administrator”.



execute the command “route -p add 192.168.2.0 mask 255.255.255.0 192.168.1.**Y**” assuming the PC is connected to the PLC / Controller / IPC on subnet 192.168.1.**Y**.

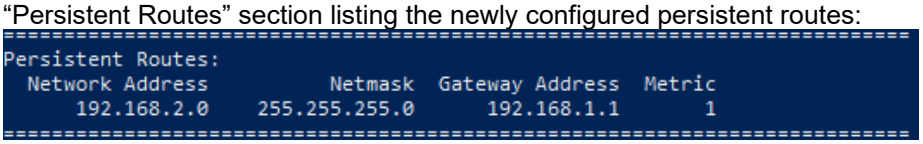


Y is the configured IP address of the PLC / Controller / IPC ethernet port being used for PC communications. In this example it is “route -p add 192.168.2.0 mask 255.255.255.0 192.168.1.1”

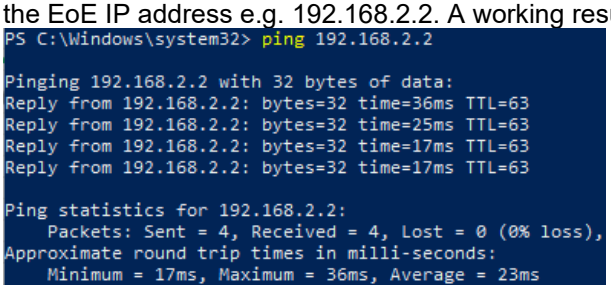


This static route allows any IP address from 192.168.2.1 to 192.168.2.255 to be accessed using EoE.

In the same command window execute the command “route print”. There will be a “Persistent Routes” section listing the newly configured persistent routes:

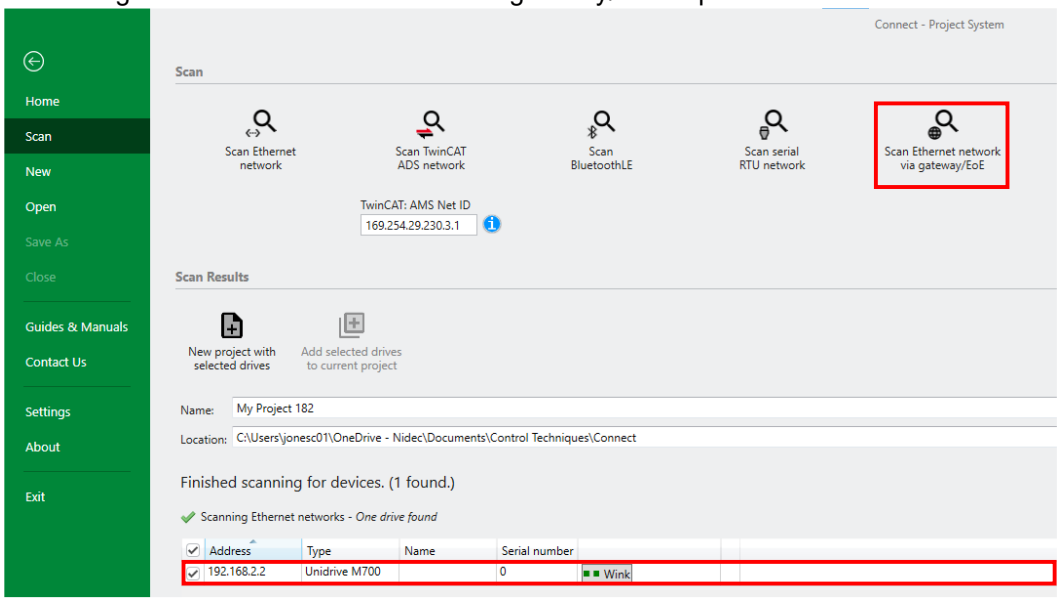


To prove the EoE IP address can be reached by PC from the same command prompt “Ping” the EoE IP address e.g. 192.168.2.2. A working result looks like this:



HINT: Once a persistent route has been configured it will remain even after the power cycling the PC. The route may be deleted using “route delete 192.168.2.0”

16. Connect can now be used to access the drive over the EoE virtual port to commission the drive using the “Scan Ethernet network via gateway/EoE” option.



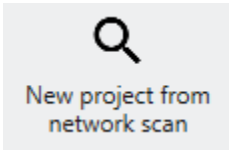
2.4 Configure the drive using Connect

Create a new Connect project to configure and commission the drive for the motion application:

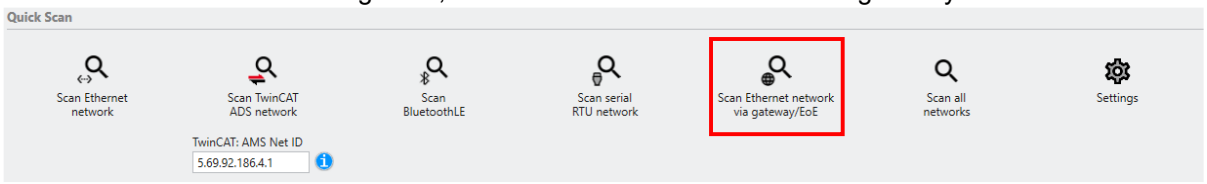
1. Open the Connect PC software by double clicking the “Connect” icon.

2. Ensure the version of Connect is a minimum V3.0.0. If an earlier version is installed please upgrade to V3.0.0; the software file may be obtained from your local Control Techniques Drive Centre / Distributor, or http://acim.nidec.com/drives/control-techniques, or Control Techniques Support Suite.

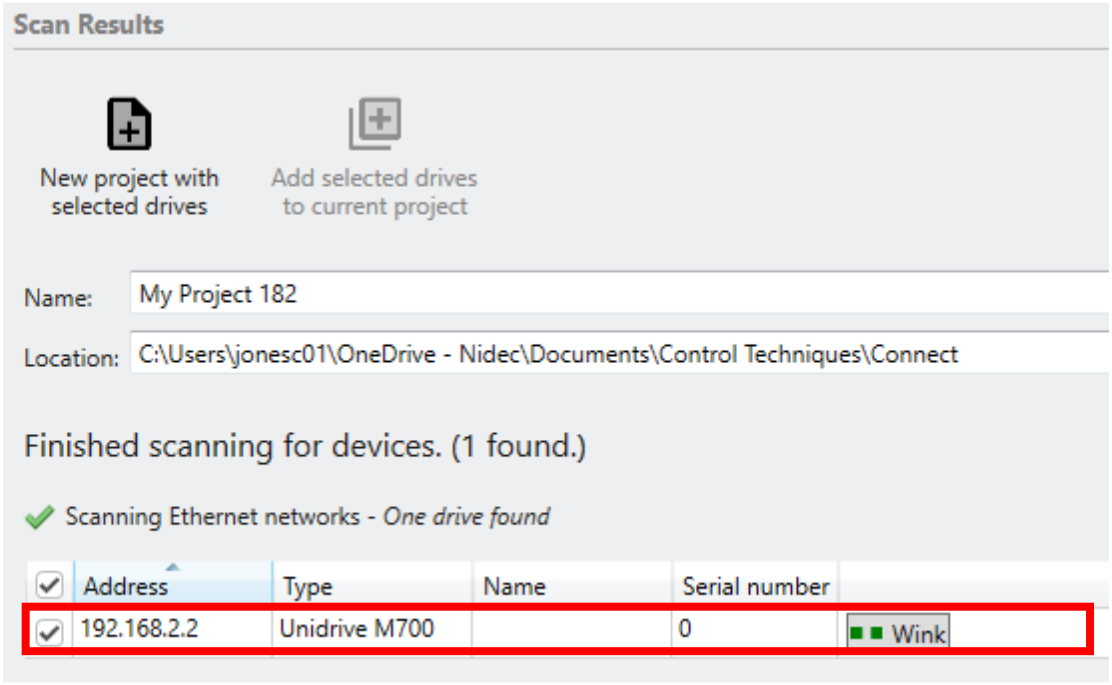
3. When Connect opens select “New project from network scan”.



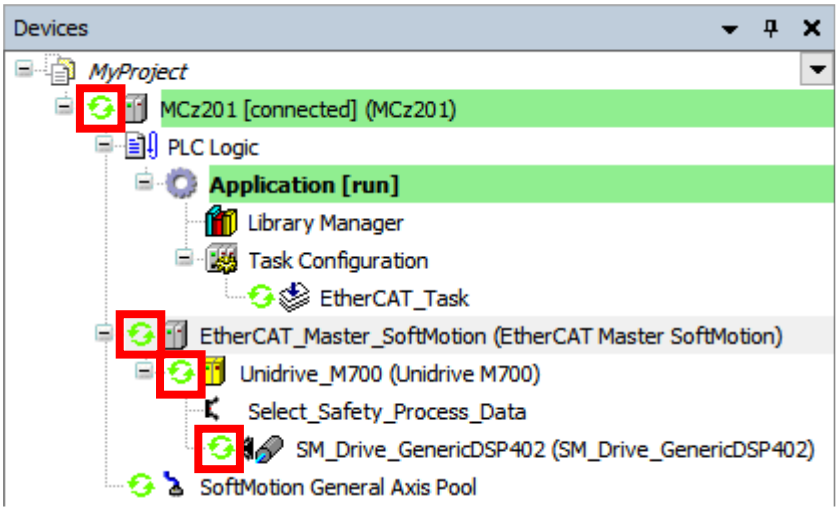
4. If EoE communications is being used, select “Scan Ethernet network via gateway/EoE”.



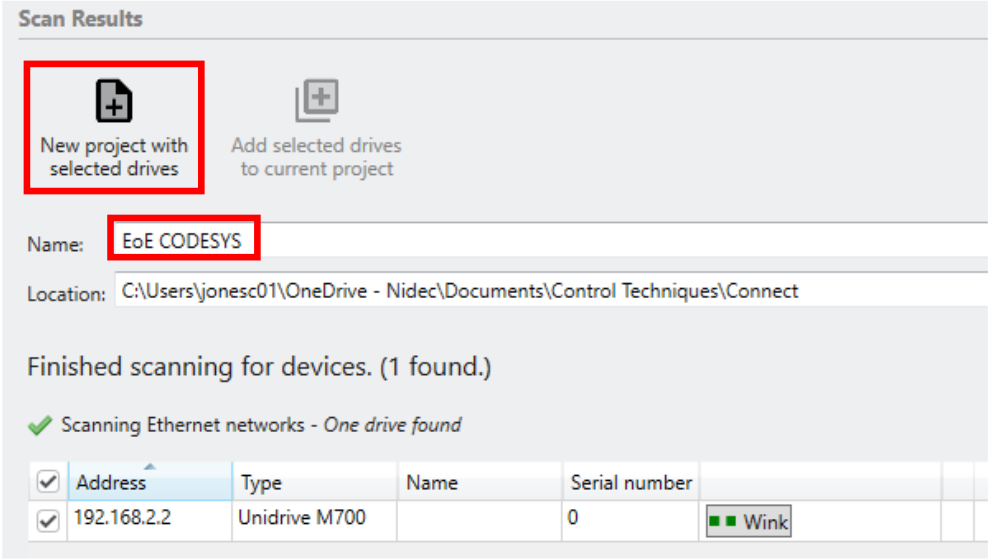
5. When the scan completes all of the available nodes on the network will be found.



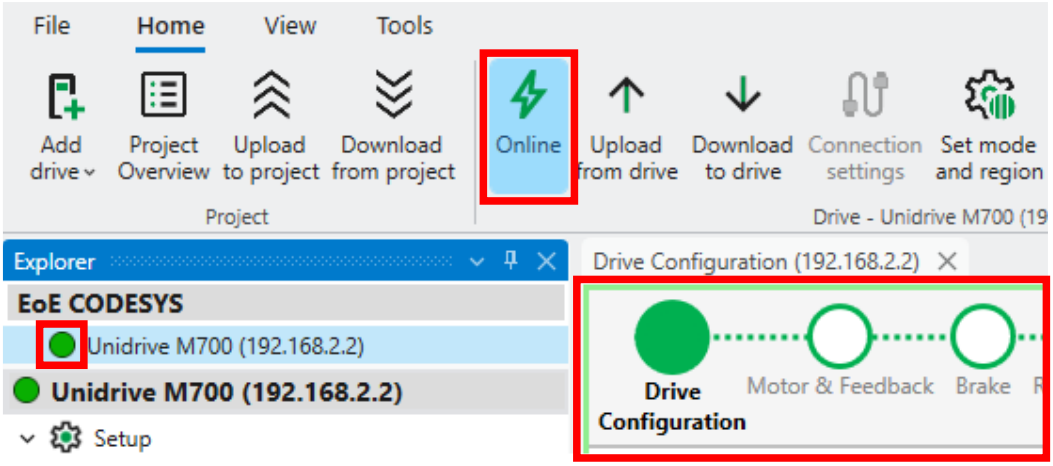
HINT: For this to be successful on an EtherCAT-based network, the CODESYS PLC/IPC and it's EtherCAT network must be running, as indicated by a green arrow symbols highlighted in CODESYS.



6. Give the project a meaningful name and then click “New project with selected drive”.

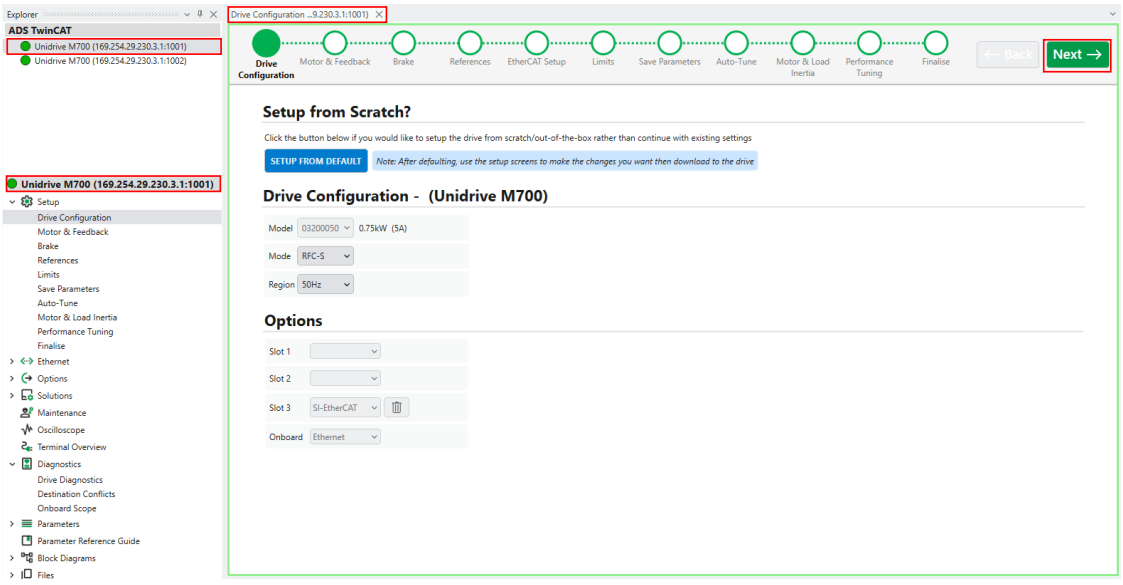


7. When the project opens, all drives on the network will be “Online” as indicated by the blue highlight on the “Online” button, the green dot next to each drive node in the Explorer tree, and the green border around the active tab page.



This means that any changes made take direct effect in the drive.

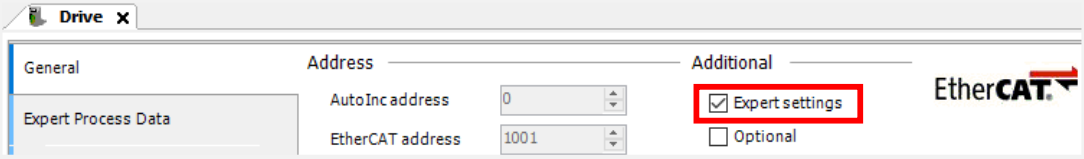
8. The first drive in the Explorer list is selected and has it's drive guided setup opened.



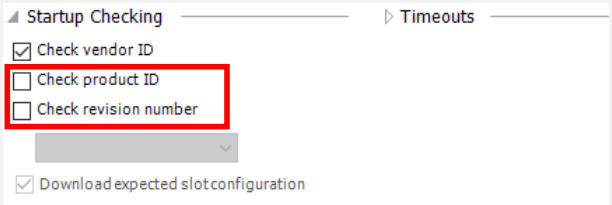
On this page the user can default the drive to remove any previous setup, change the operating mode to match the motor that has been connected to the drive, and select the region e.g. 50Hz or 60Hz.

HINT: When connecting to the drive using EtherCAT (EoE), if the EtherCAT slave was configured with an ESI file that matches the mode of the drive, e.g. "Unidrive M700 Open Loop", and the drive mode is subsequently changed, the comms connection will stop if the "Check product ID" and "Check revision number" boxes are checked. If this happens, in CODESYS, go offline, delete the EtherCAT drive nodes, right click on the EtherCAT network master and select "Scan for Devices…" to re-add the nodes in their new mode. EoE settings will need to be re-configured before going back online to download the changes.

Alternatively, navigate to the "General" tab of the EtherCAT node, ensure "Expert settings" is ticked:



Expand the "Startup Checking" menu. Untick the "Check Product ID" and "Check revision number" options before going online and downloading the application to the PLC / IPC.\

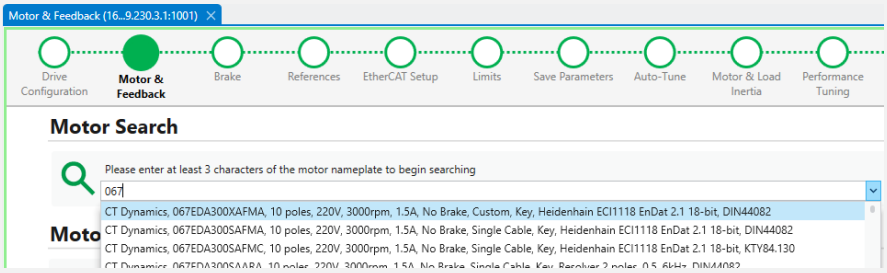


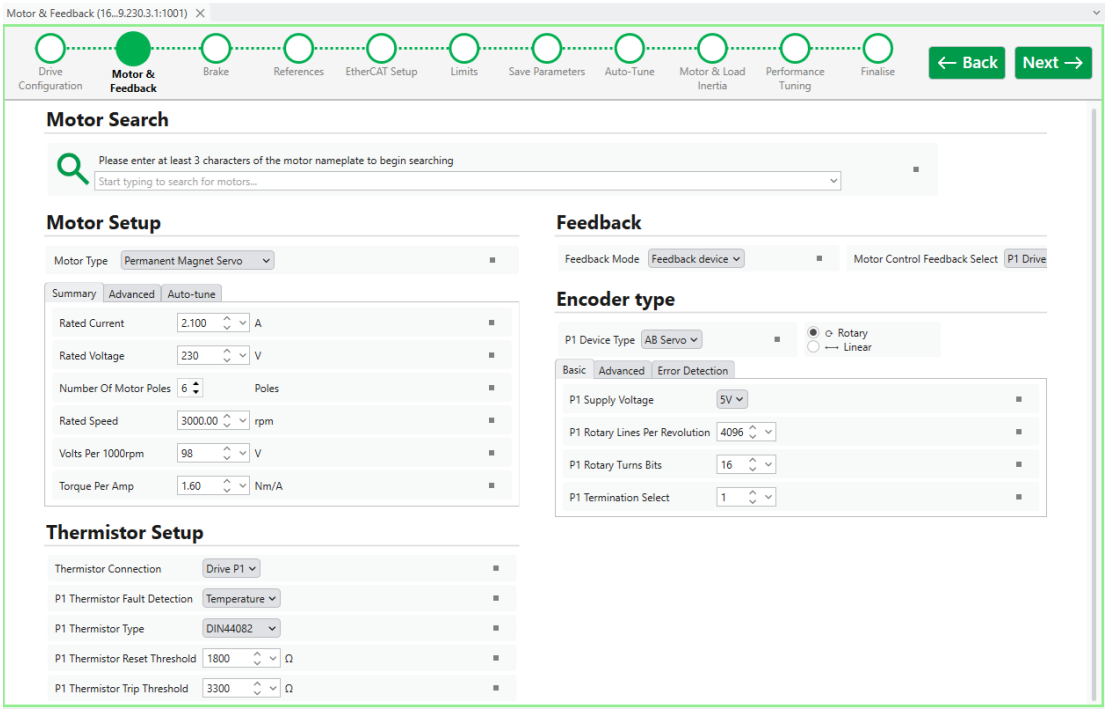
The guided setup section is selected by either the "Next" and "Back" buttons, or by directly selecting a section by clicking on the circle section markers.

When finished, click "Next" move to the motor and feedback setup.

9. Setup the motor and feedback device on the “Motor and Feedback” page. The required data for the motor and feedback device will be available on the motor name plate or the technical data sheet for the motor.

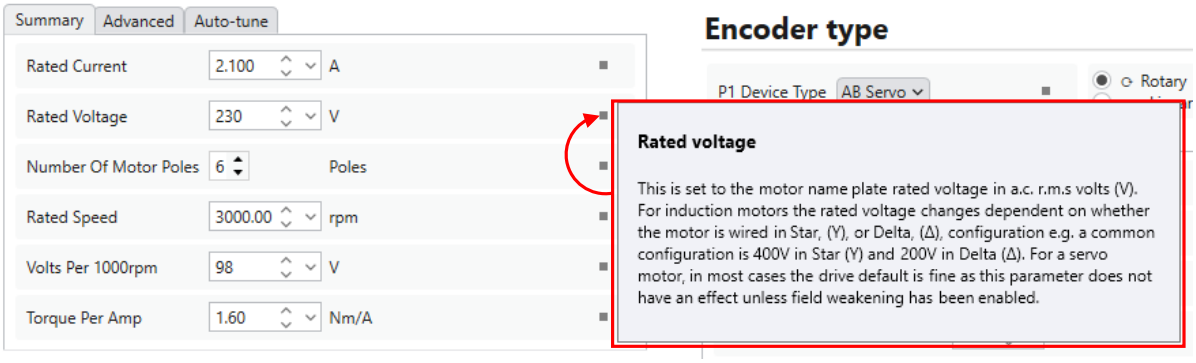
HINT: Control Techniques Dynamics permanent-magnet servo motors and Leroy Somer induction motors and permanent-magnet process motors may be found using the motor search:





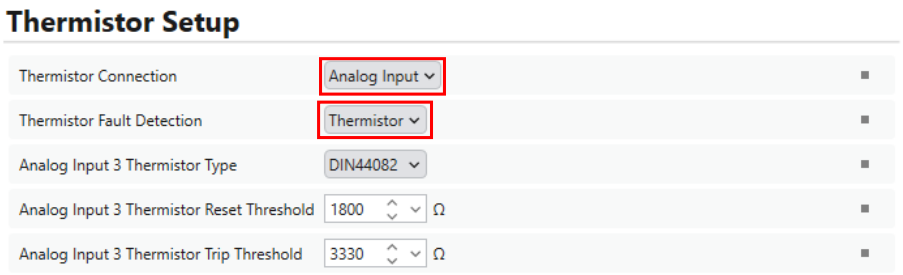
The user must configure the motor, feedback device and thermistor setup here.

Help is provided throughout Connect using small help trigger squares ■:

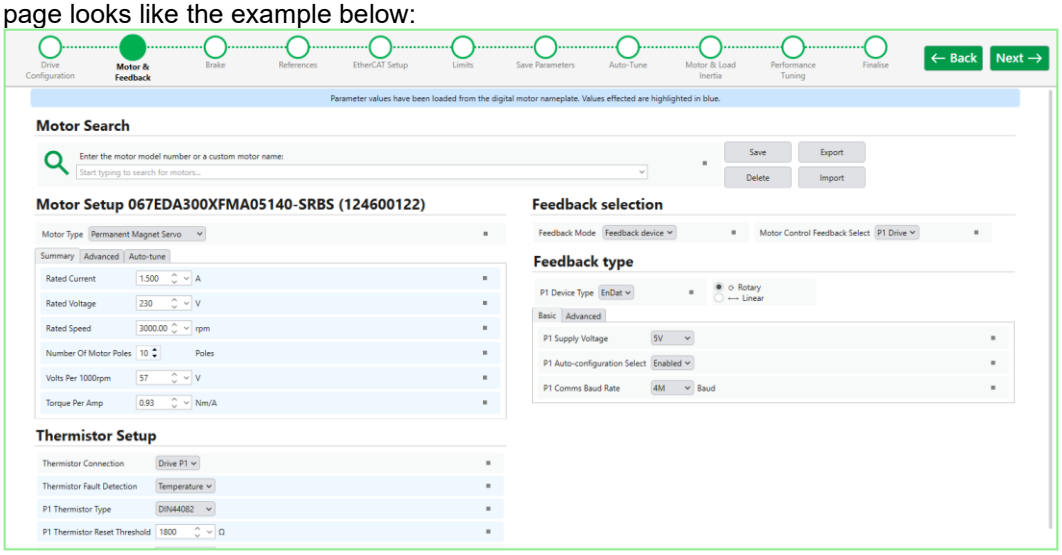


It is possible to save a motor and encoder configuration for later user; these can also be searched for.

For induction motors, where the thermistor is not typically connected to the encoder interface D Type, and instead, is connected to a terminal on the drive (e.g. Analogue input 3). Set Analogue Input 3 Mode to Thermistor.



If the drive is either a Digitax HD or a Unidrive M with firmware >=V01.61.01.00 and the motor is one from Control Techniques Dynamics that has an electronic nameplate loaded into the encoder, the motor and encoder data is setup automatically where the “Motor & Feedback” page looks like the example below:

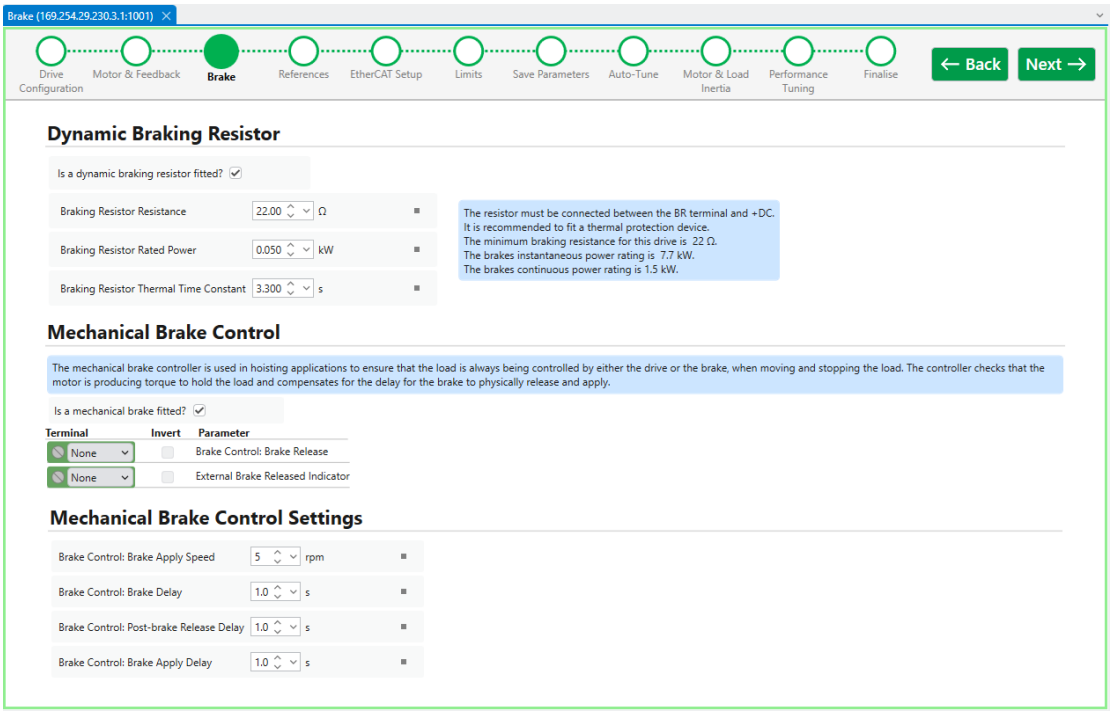


The light blue highlight shows that the electronic nameplate has been loaded. In addition, for EnDat 2.2, EnDat 3.0 and BiSS-C encoders the “Encoder type” fields are also configured automatically.

When finished, click “Next” to move to the brake setup.

10. The brake setup page allows braking resistor properties to be configured. The braking resistor is used to dissipate motor energy when slowing down. The settings provide protection for the braking resistor in addition to a thermal overload circuit which is typically provided with the resistor.

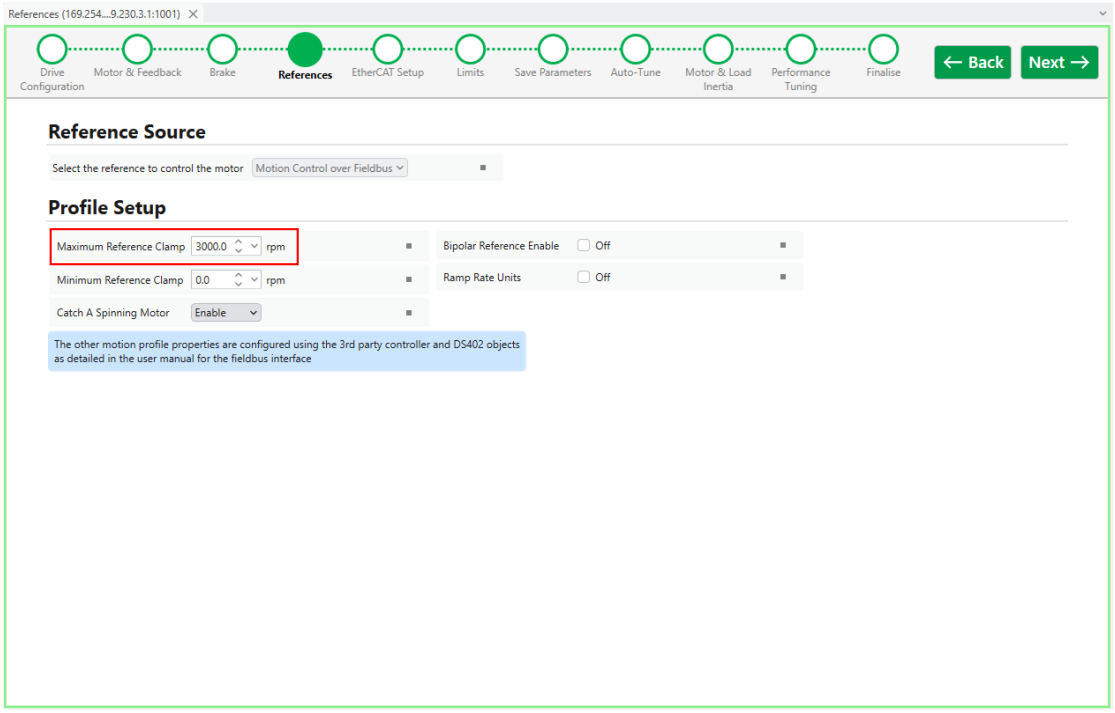
    It is also be used to set up the drives mechanical brake controller. The mechanical brake controller releases and applies the mechanical brake automatically, making sure that the load is always either in control using the drive or the mechanical brake. It is typically used in hoisting applications. For the majority of applications, assigning the “Brake Control: Brake Release” Output, setting the time for the brake to physically release the load in “Brake Control: Post-brake Release Delay”, and the time for the brake to physically apply and hold the load in “Brake Control: Brake Apply Delay” is sufficient.



    For more information on the mechanical brake controller see section **4.3 Mechanical brake controller logic**.

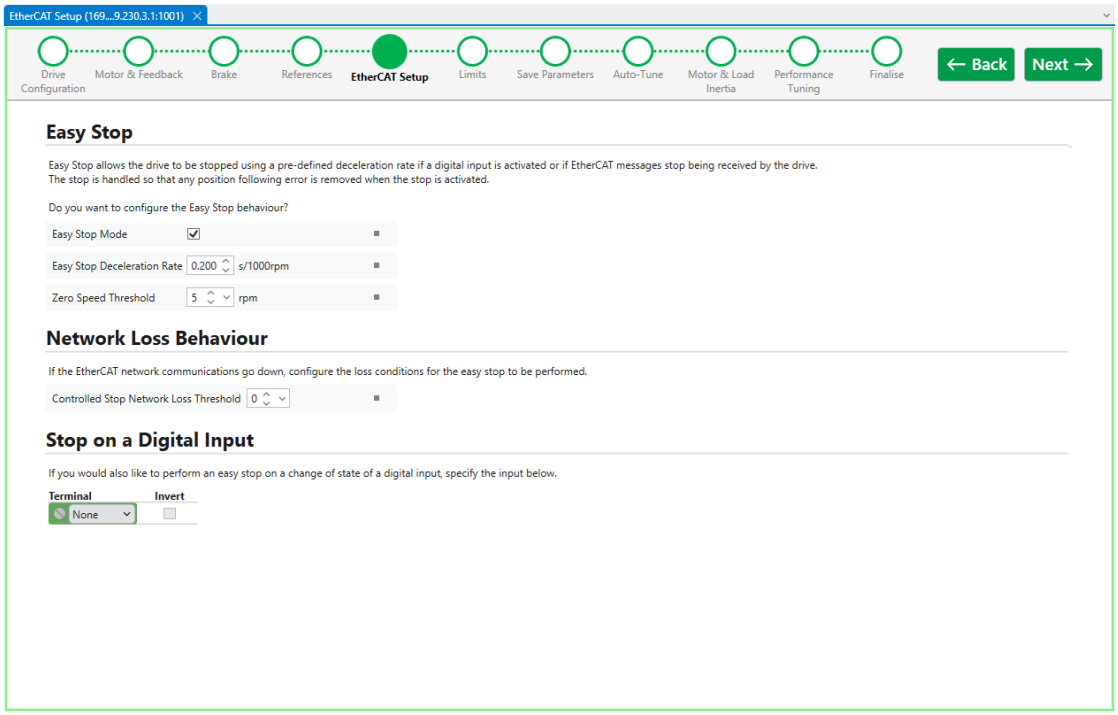
    When finished, click “Next” to move to the reference setup.

11. The reference setup page for an EtherCAT motion control application, is where the maximum reference clamp is defined. Normally, the maximum reference clamp must be set to the motor rated speed, however, in some applications this is modified and may be adjusted from this page.



When finished, click “Next” to move to the EtherCAT setup.

12. The EtherCAT Setup page allows the user to configure the stop behaviour from a digital input and from a loss of comms.



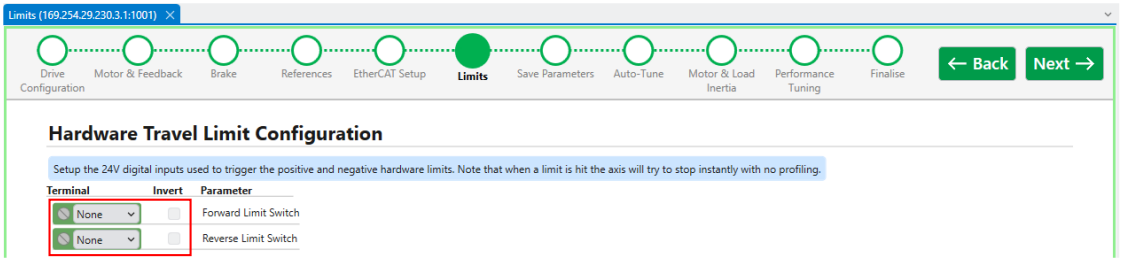
When the “Easy Stop Mode” box is checked, the user can assign a digital input that will cause the axis to stop. The axis is permitted to run when 24V is applied to it, but if 24V is removed the axis will stop, using the “Easy Stop Deceleration Rate”.

The “Easy Stop Deceleration Rate” is also used when there is a loss of EtherCAT communications. The “Controlled Stop Network Loss Threshold” is a weighted threshold that increases by 3 for every missed message and decreases by 1 for every received message. In this way, occasional missed messages don’t affect the system, but a more permanent loss of messages such as a broken EtherCAT comms cable will stop the axis.

The “Zero speed threshold” is used by the CiA402 state machine, (the CANopen motion standard used to control the axis over EtherCAT communications), to verify that the axis has actually stopped before allowing further movement. The default value is suitable for most systems except those that have a noisy or low-resolution feedback device, in which case the value must be increased to account for the additional feedback noise.

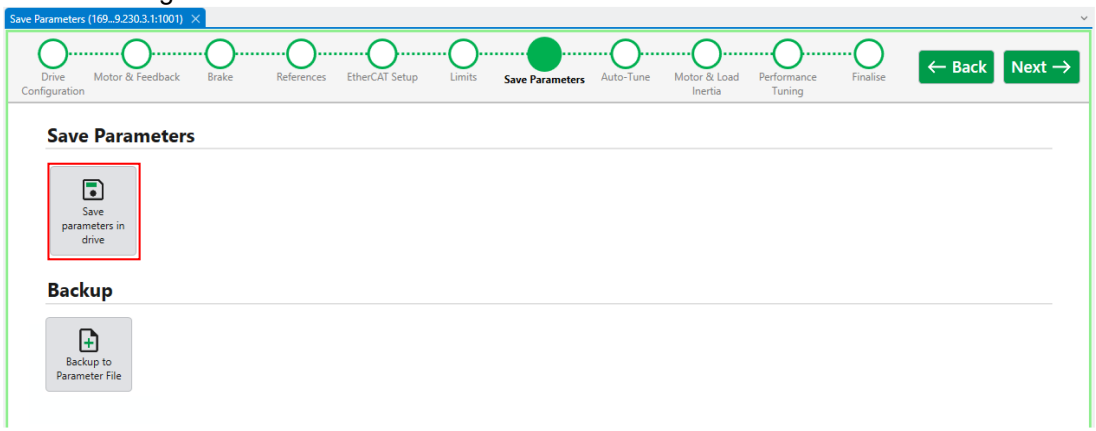
When finished, click “Next” to move to the Limits setup.

13. The limits page allows the user to assign digital inputs to become hardware limit inputs. If the application doesn’t have hardware limits, leave the digital input assignments empty and move to the next step.



When finished, click “Next” to move to the Save Parameters page.

14. The save parameters page allows the parameters configured so far to be saved. It is advised to do this prior to the commissioning activities since the power might have to be removed to correct a hardware issue as a result of a failed Autotune test, such as a reversed motor or encoder wiring.

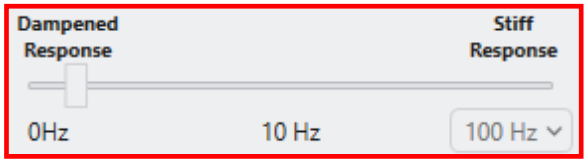


This completes the configuration section of the guided setup. The next section of the manual describes the commissioning of the drive and motor using the remaining steps in the guided setup.

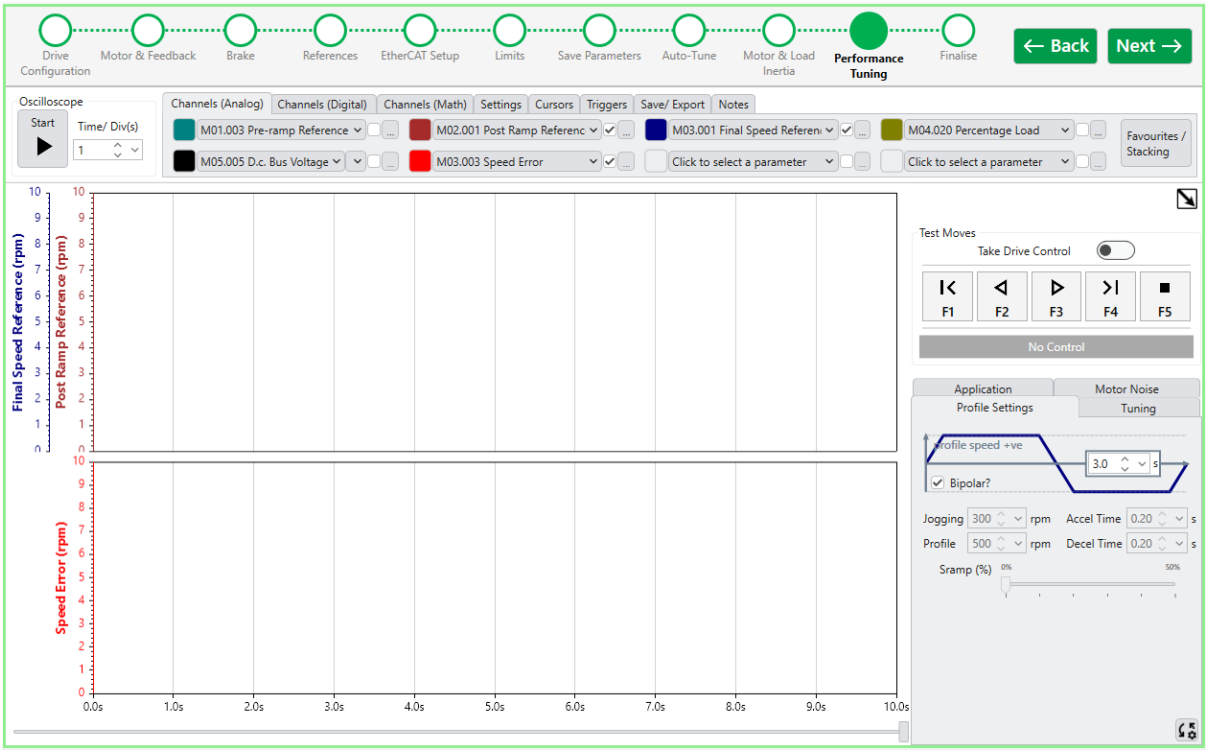
2.5 Commission the drive and motor using Connect

This section describes the process of commissioning the drive and motor in the application using the tools provided in the guided setup, which consist of:

- Auto-tune to measure the electrical properties of the motor such as resistance and inductance.
- An inertia-auto-tune to measure the inertia of the motor and load. Once this measurement is taken, tuning the axis is simplified to a single slider. It is **strongly** recommended to run this test.



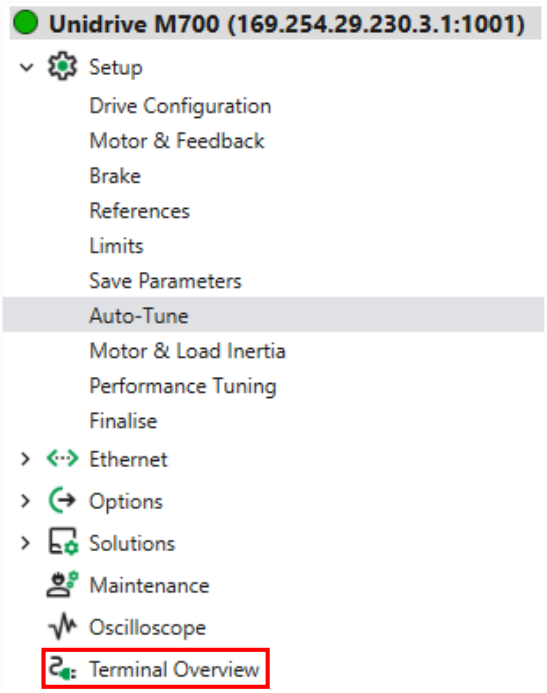
- Performance tuning and test moves. This page is a “one stop shop” with test move controls, motion profile setup, oscilloscope, and tuning controls on a single page ideal for site commissioning using a laptop.

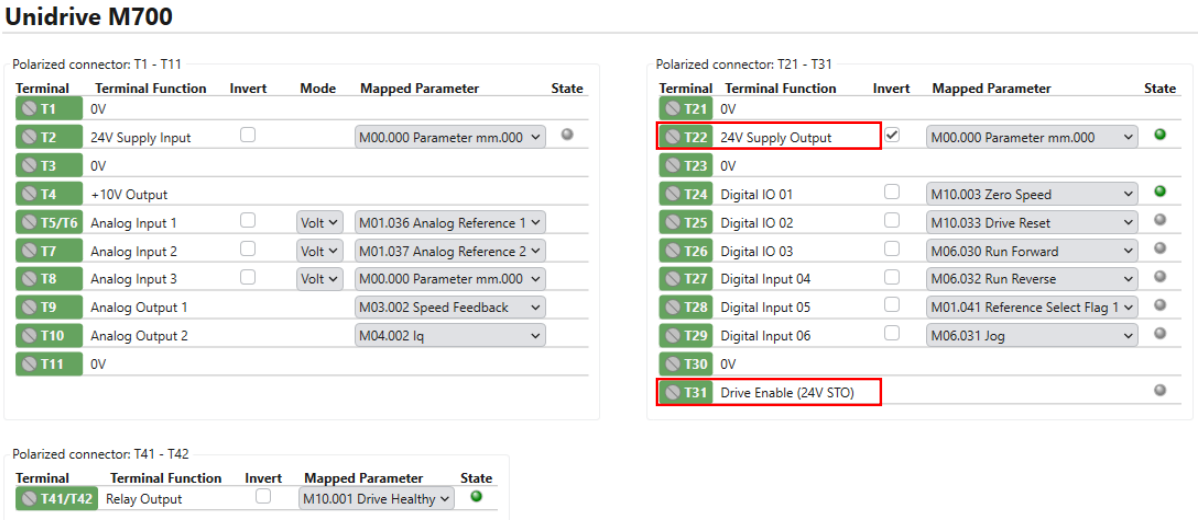


- Finalising the application by saving the tuned parameters and creating a startup list allowing the PLC / IPC to configure the drive, each time the system powers up; this is helpful if the drive must be replaced.

Use the following steps to commission the motor and drive in the application.

1. Wire a switch or similar device to the drives STO input(s) where 24V is connected to the STO input(s) to enable the drive. The drive will not be able to run the motor until such a device is fitted. The terminals that supply 24V and host the STO input(s) are shown by the “Terminal Overview” in the device explorer.



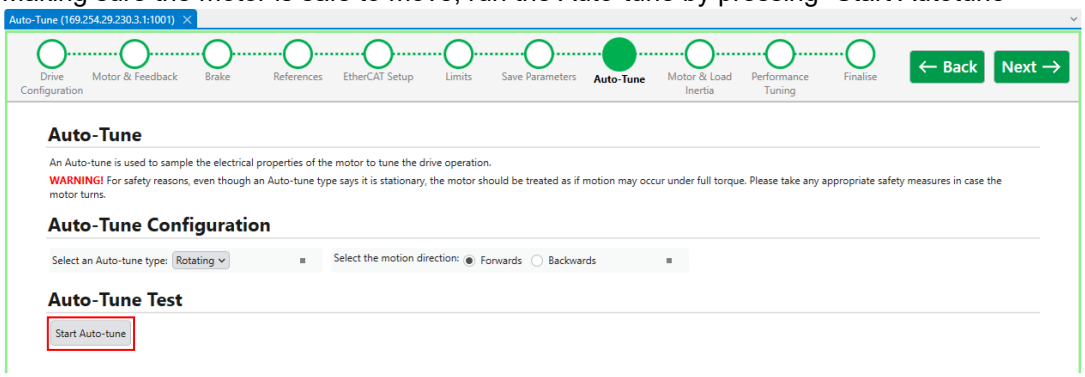


SAFETY: Remove power from the drive before attempting to modify the drive or motor electrical connections.

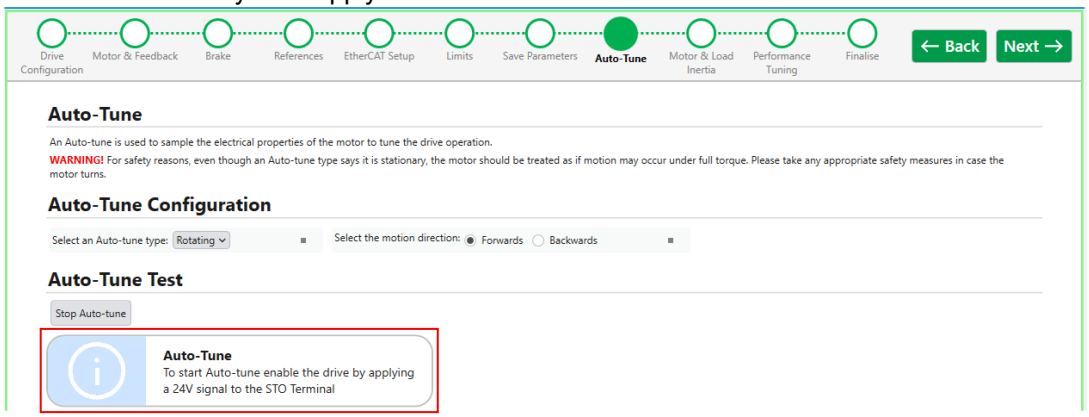
2. Before attempting to run the Auto-tune ensure the STO switch is open i.e. 24V is not connected to the STO terminal(s).

3. The default setup for the auto-tune will be suitable for most applications. Additional controls are provided for expert users.

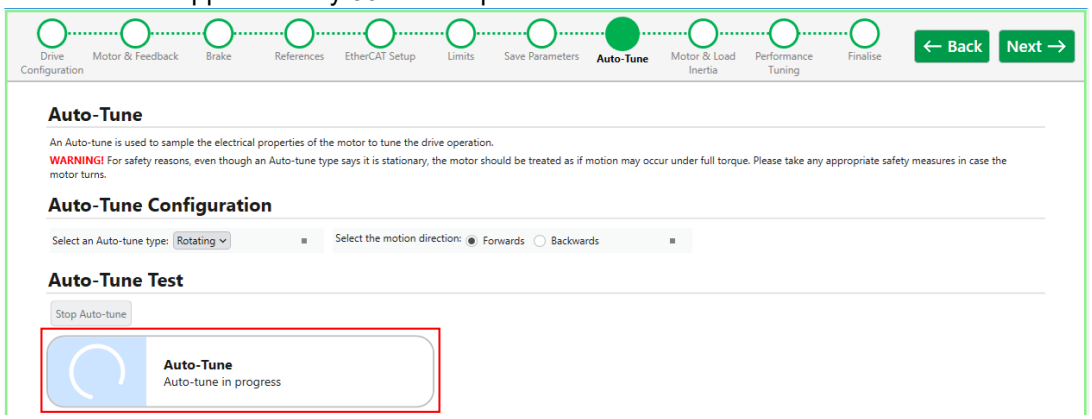
Making sure the motor is safe to move, run the Auto-tune by pressing “Start Autotune”



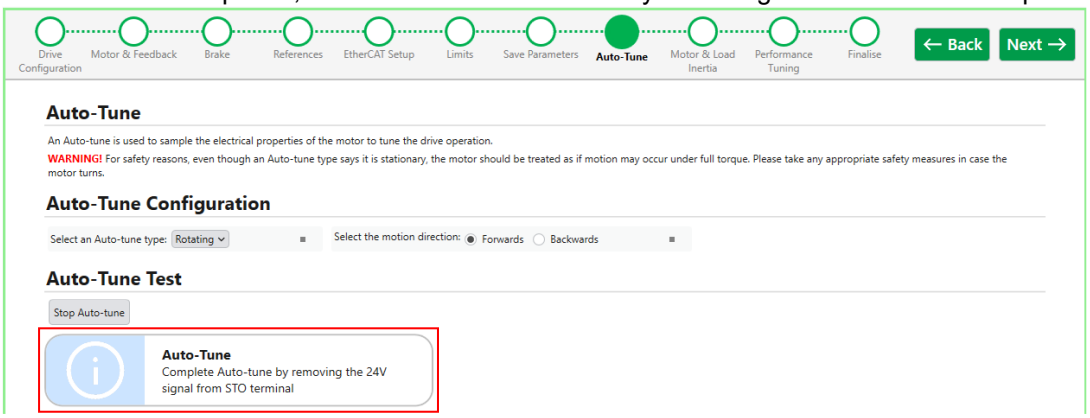
The tool will advise you to apply 24V to the STO terminal to run the test.



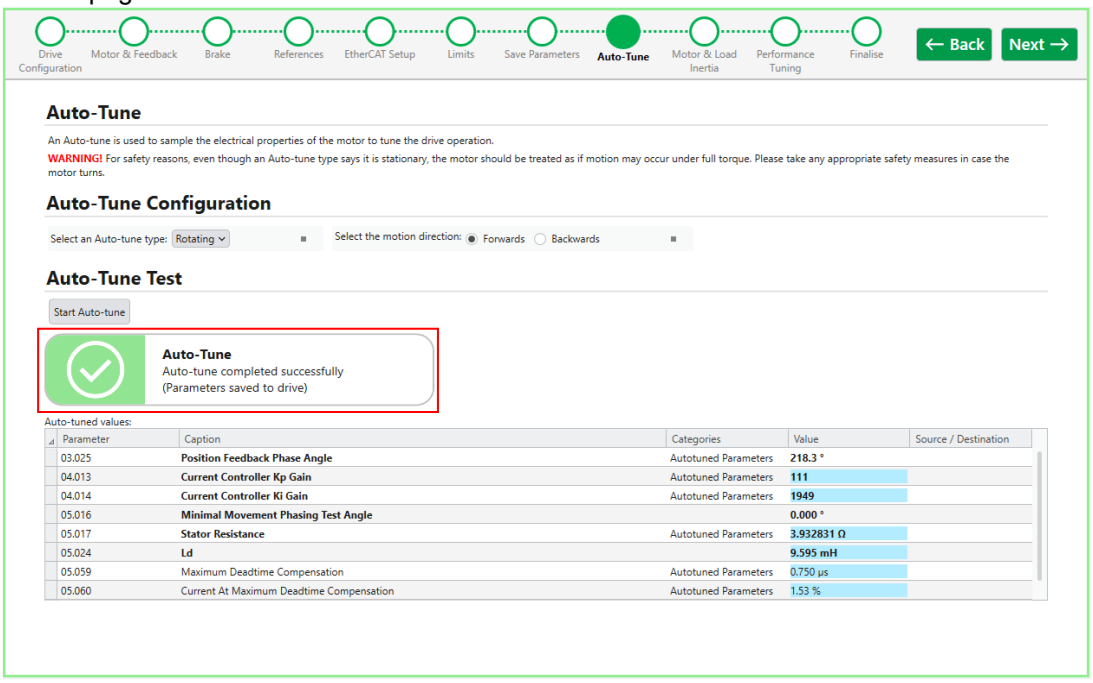
The test takes approximately 30s to complete



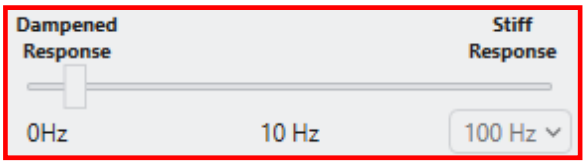
When the test completes, the drive must be disabled by removing 24V from the STO input.



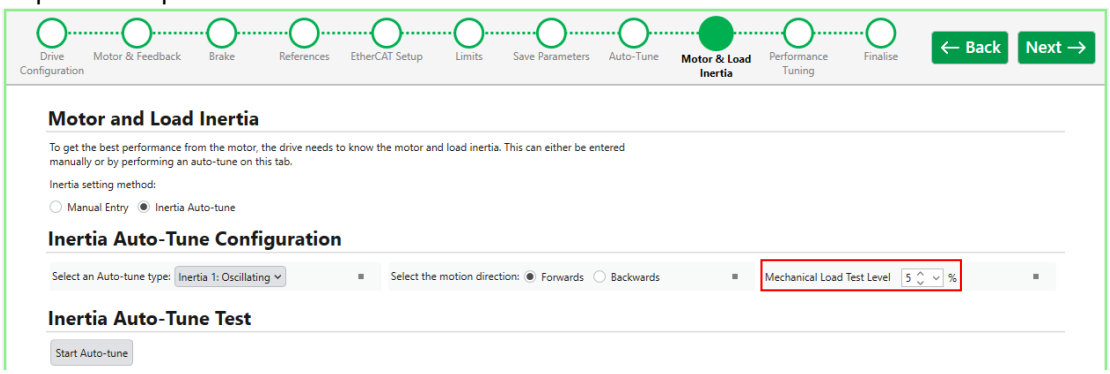
The electrical property autotune is completed. Click next to move to the “Motor and load inertia” page.



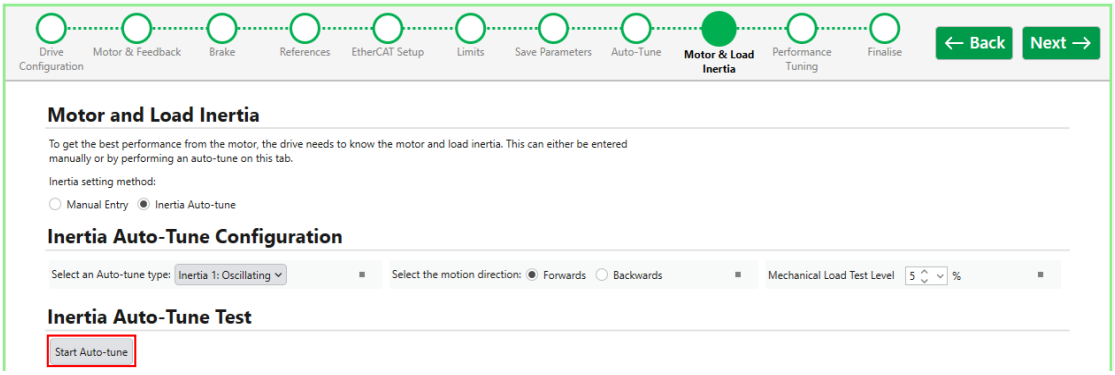
4. The “Motor and load inertia” page configures a test to measure the motor and load inertia. This test is important as it allows the speed loop gains to be configured using a single control slider:



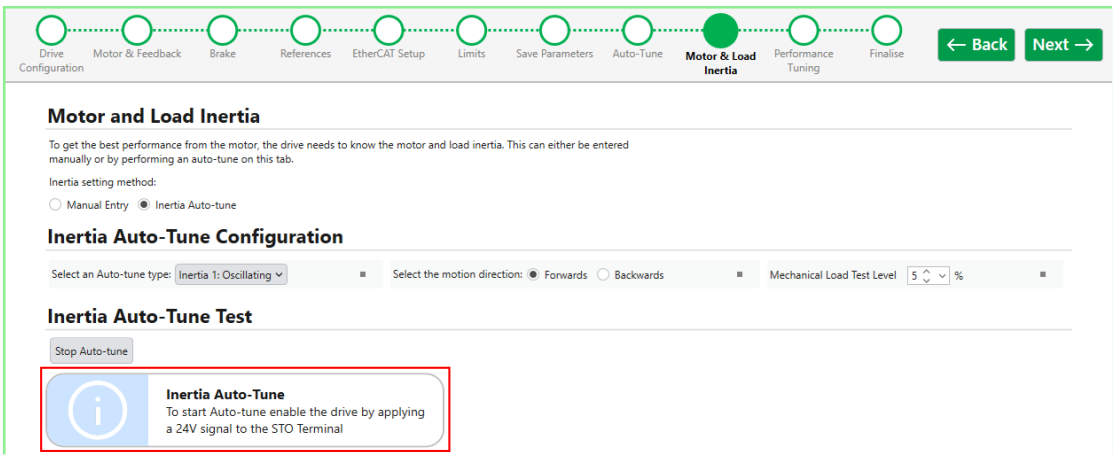
If the motor has a load connected, the default settings will be OK for most systems. Where there is no load i.e. a bare motor shaft it is recommended to increase the “Mechanical Load Test Level” to 5%. If the test fails to identify the inertia, increase the mechanical load test in steps of 5% up until 20% is reached.



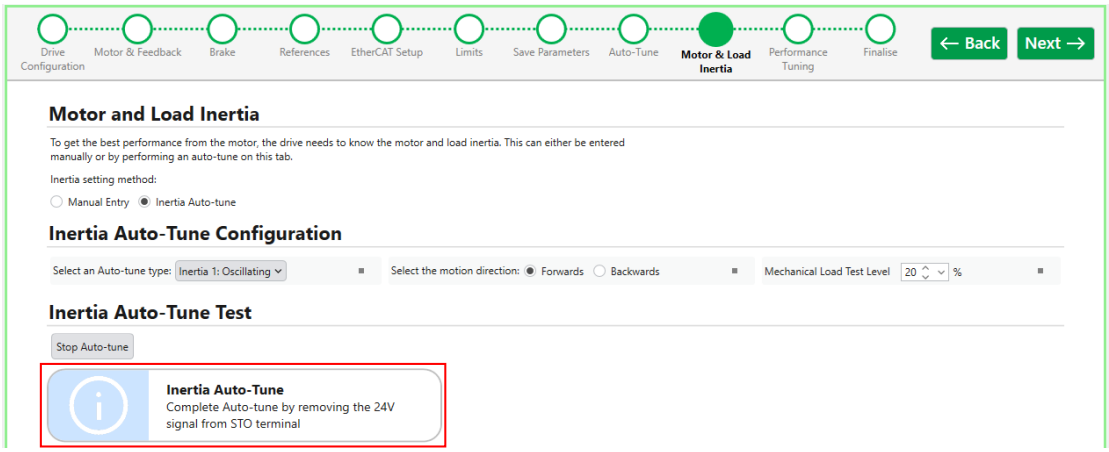
Start the inertia test by clicking “Start Auto-tune”:



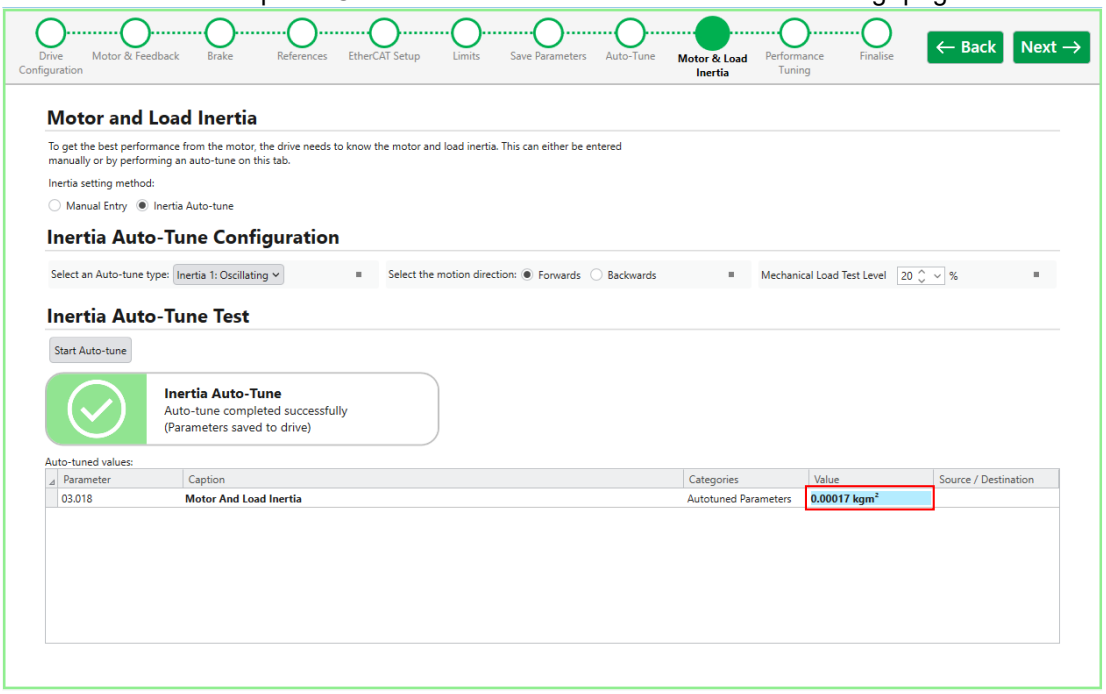
The tool will advise you to close the STO switch to apply 24V to the STO terminal and run the test.



When the test completes, remove 24V from the STO terminal.



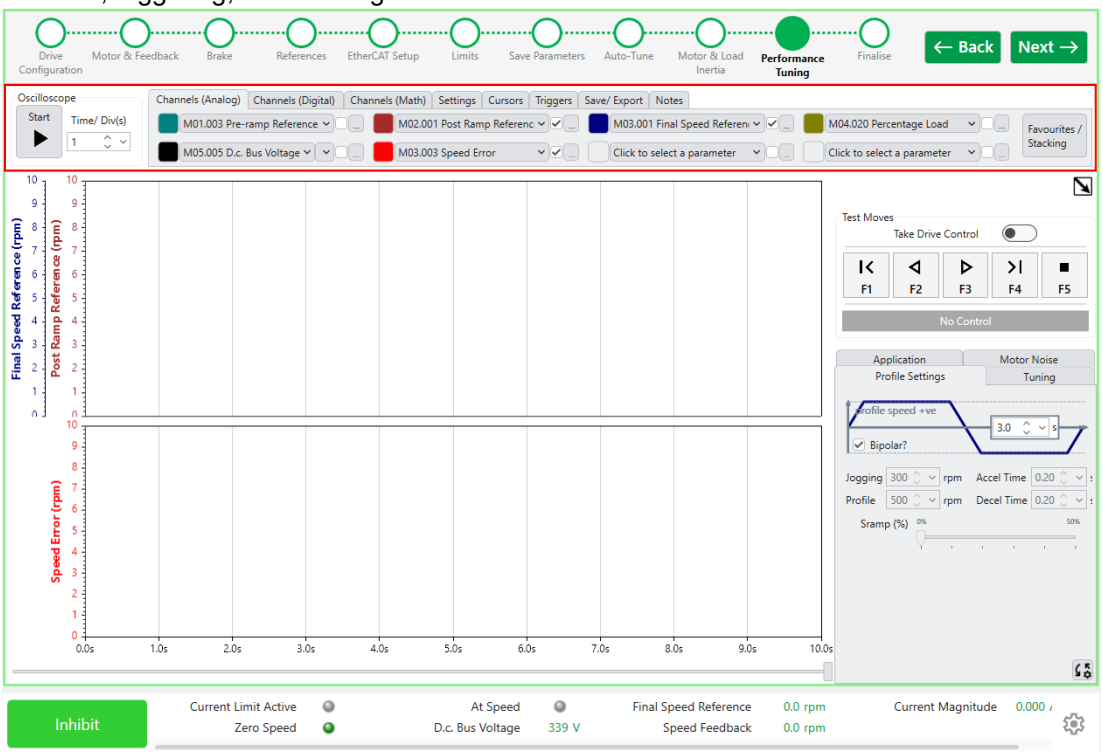
The inertia test is complete. Click Next to move to the “Performance Tuning” page.



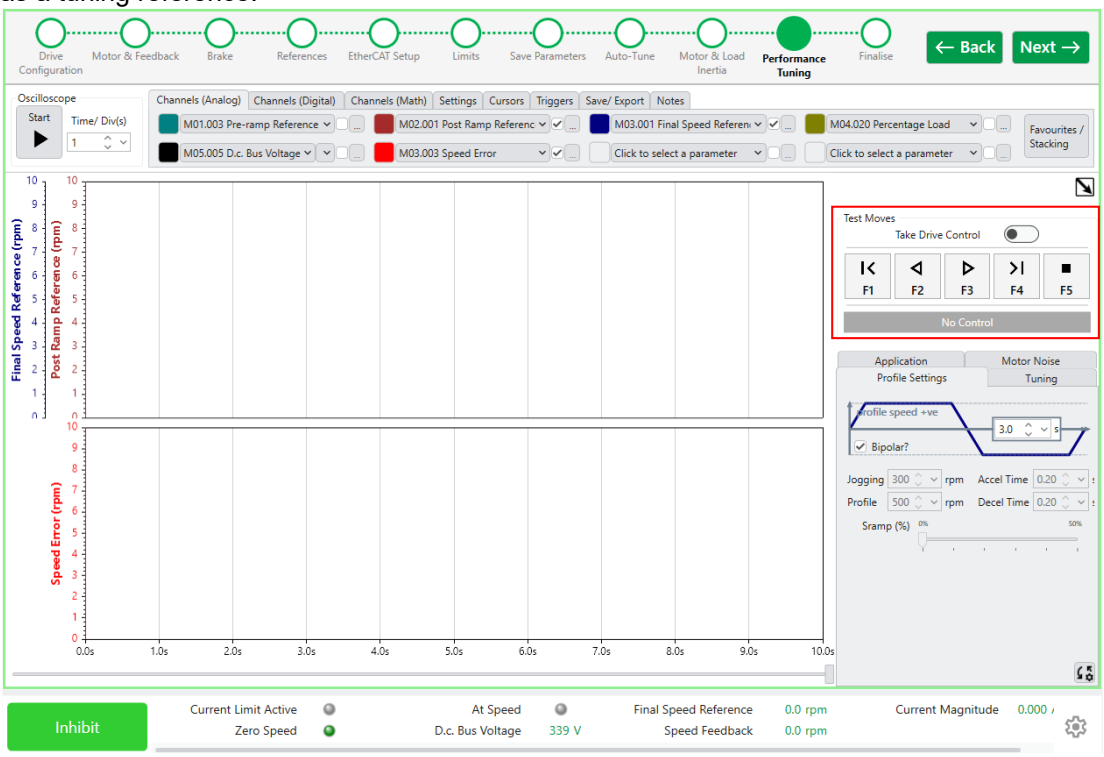
5. The “Performance Tuning” page provides a set of tools on a single page, optimised for laptop use, that are used to commission the drive and motor. The different sections of the tool are described below:

Oscilloscope – to show what is happening to the motor speed while jogging or tuning takes place:

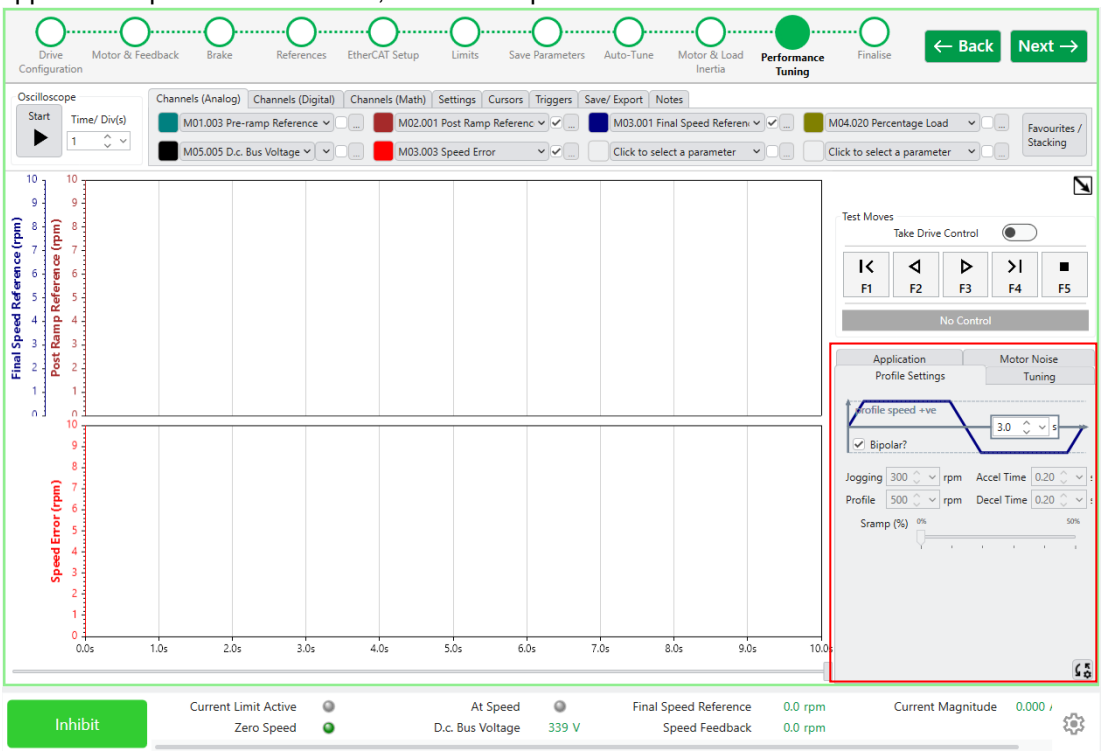
Oscilloscope controls – start and stop the trace, allow channels to be added / removed / modified, triggering, trace saving.



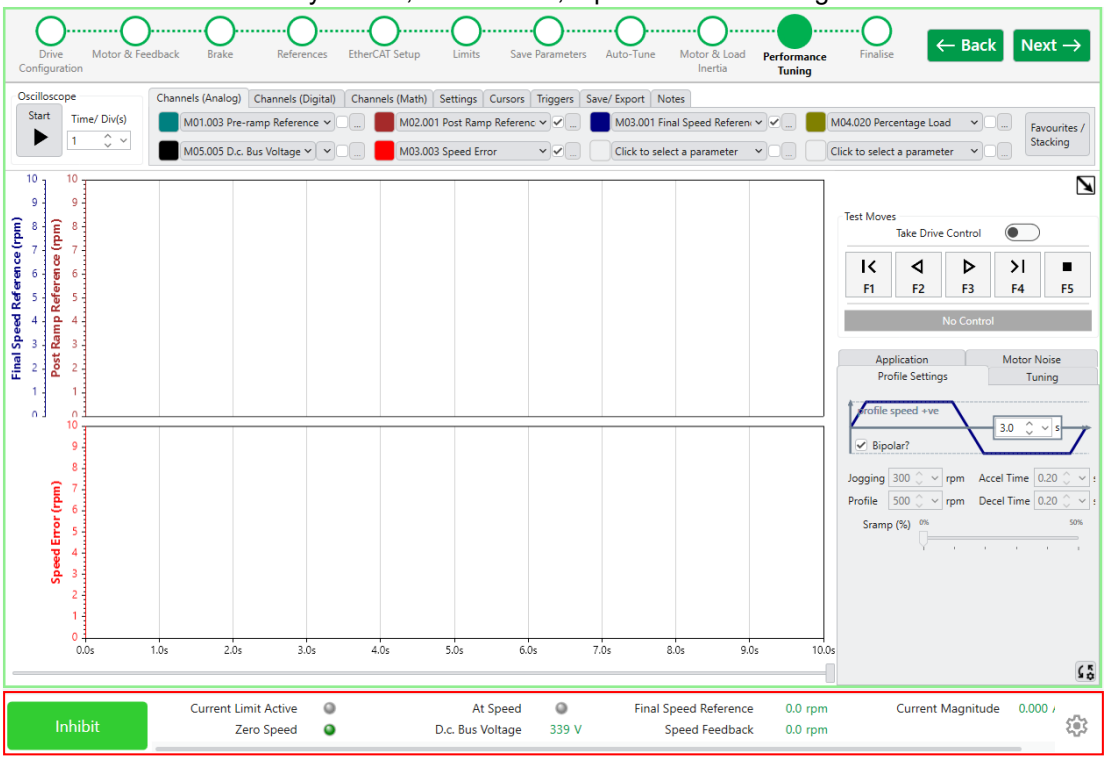
Test move commands – enable the test controls, jog forward / backwards, automatic running as a tuning reference.



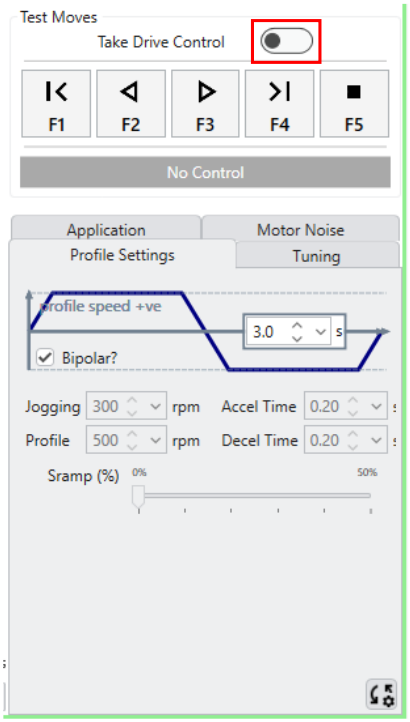
Commissioning tabs – configure the test move motion profile, speed and position loop tuning, application optimisation controls, and noise optimisation controls.



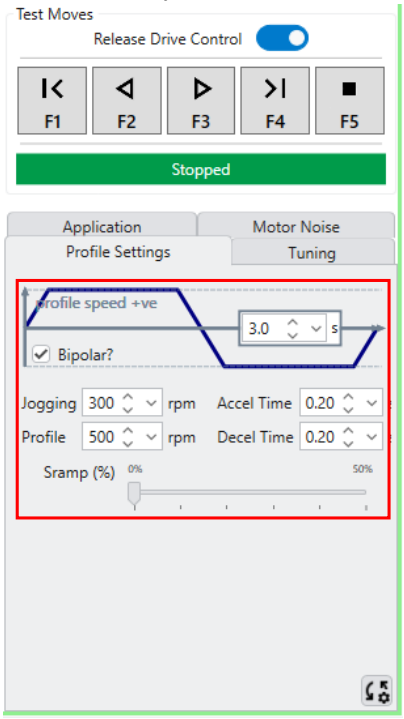
Status control – Shows key values, drive status, trip data and resetting.



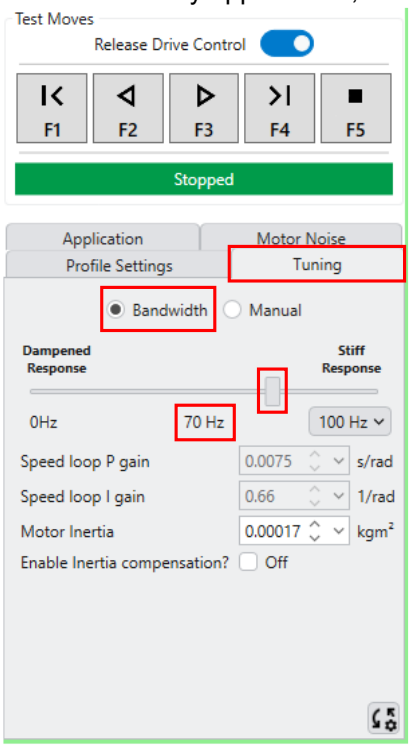
6. Begin commissioning by jogging the axis (slowly) to determine if the motor direction is correct for the application. Ensure it is safe to move the motor. Click on “Take Drive Control” and then apply 24V to the STO input.



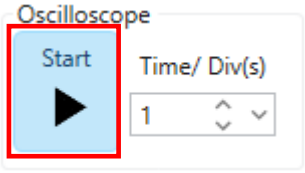
Configure the motion profile settings used when the test moves are running. This includes the jog speed, automatic running speed, acceleration and deceleration times, S ramp percentage, the overall profile time, and whether the automatic running is in one direction, or bipolar where the motor reciprocates back and forth.



Select the Tuning tab, and then select “Bandwidth” mode. This will give moderate tuning suitable for many applications, with headroom to increase the gains further if required.

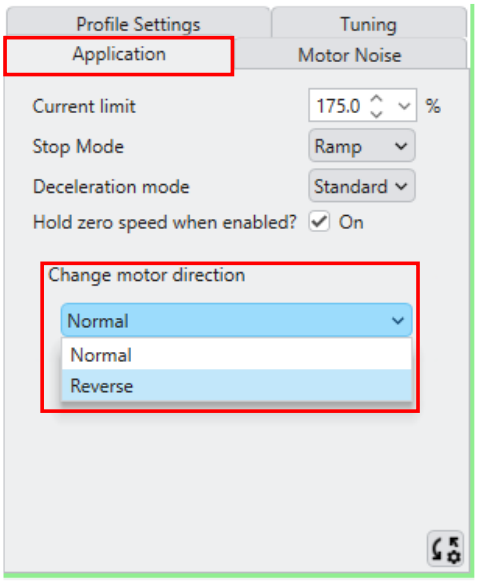


Start the oscilloscope by pressing the Oscilloscope start/stop button.



Jog the axis forward by pressing F4 or clicking on the Jog Forward button.

Verify that the movement is stable and that the forward direction of rotation is correct for the application. If the movement is unstable, modify the tuning slider until the desired performance is achieved. If the motor is turning the wrong way when jogging forward, remove 24V from the STO Input, select the Application tab, and then set the “Change Motor Direction” control to “Reverse”. Re-apply 24V to the STO terminal and verify that the motor direction is now correct when jogging forward.

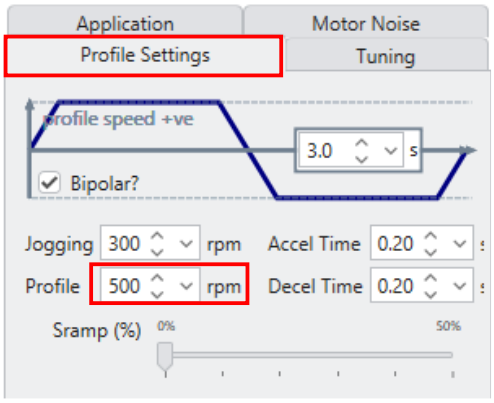


7. If the application axis has hardware limit switches connected, test them by jogging into them gently and prove that the axis stops and that the switches have been connected to the correct terminals.

8. Verify the step response of the motor and load by enabling automatic running. Before tuning using these commands make sure that if the axis has physical limits that it is placed in the centre of these limits and that Bipolar mode is selected so that the axis moves back and forth. To stop the axis press F5 or click the stop button:

Select the initial direction of travel when automatic running using F2 (backwards) or F3 (Forwards) or by clicking the Run Forwards, Run Backwards buttons:

It is recommended to start with a slow Run speed and then gradually increase to make sure the axis doesn't move too far on the first try. The speed can be increased while running but bear in mind that the higher the speed, the further the axis will travel.



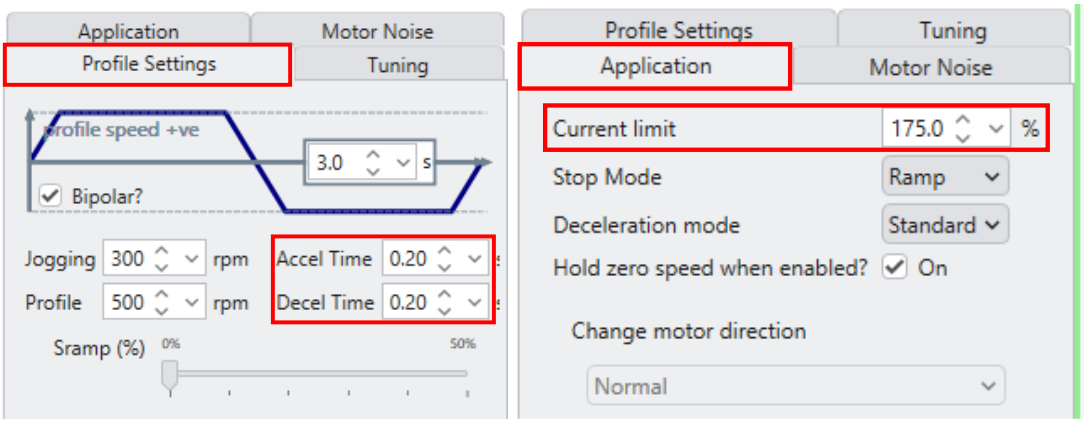
9. Once the correct automatic run profile is running, the axis may be tuned further. Observe Final Speed Reference Pr3.001 and Speed Feedback Pr3.002 overlaid with each other, and the Speed Following Error Pr3.003 on a separate trace. Ensure there is no significant overshoot when stopping or when reaching the speed reference. If there is overshoot the gains can be increased. When doing this check that the drive isn't going into current limit by observing Pr10.009; this may be placed on the same trace as the speed following error.
The example shows what the axis performance might look like prior to tuning:

In the previous example trace it can be seen that there is significant speed overshoot caused by the load inertia and default gains. The speed following error shows oscillations when accelerating and decelerating. All of these artifacts can be tuned out easily using "Bandwidth" mode and adjusting the slider control to achieve the desired performance.

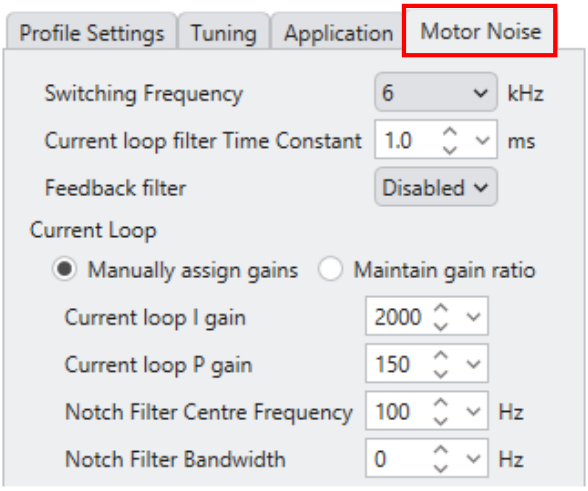
The example below shows an optimised result for the same application where Bandwidth mode tuning has been used to optimise the performance:

From the previous trace it can be seen that there is no significant speed overshoot, the speed following error isn't oscillating, and the drive isn't going into current limit.
If the drive is observed going into current limit, (Current Limit Active trace goes to 1), it is an

indication that the drive is going into constant torque due to hitting the current limit. This can be resolved by increasing the current limits on the Application tab or by decreasing the acceleration and deceleration rate. It is further recommended to verify that the drive and motor combination have been sized correctly for the application in terms of their torque and current capability.



10. If the motor is sounds noisy with the tuned values select the "Motor Noise" tab:



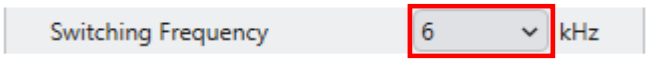
For most applications it is recommended to use a Current Loop Filter Time Constant of 1.0ms, unless the application is ultra-dynamic and needs a very fast response where a lower value is needed. A 1.0ms filter gives a 159Hz response bandwidth, where most systems will be well tuned at between 50Hz and 100Hz.

This helps in the following situations:

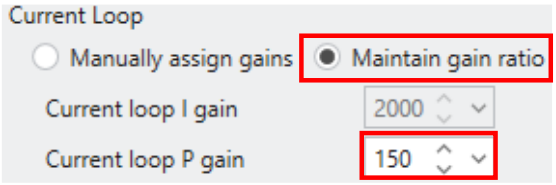
- The speed reference or speed feedback contains a noise component e.g. if the feedback device is of low resolution.
- If the speed loop gains have been raised to the point where the motor noise has become unacceptable, and the required speed loop performance hasn't been reached.
- The application often runs with a low motor speed where encoder quantisation noise is an issue.
- There is mechanical resonance affecting the feedback.

The filter frequency bandwidth with time is 1.0ms = 159Hz, 0.9ms = 177Hz, 0.8ms = 199Hz, 0.7ms = 227Hz, 0.6ms = 265Hz, 0.5ms = 318Hz, 0.4ms = 398Hz, 0.3ms = 530Hz, 0.2Hz = 795Hz, 0.1ms = 1592Hz.

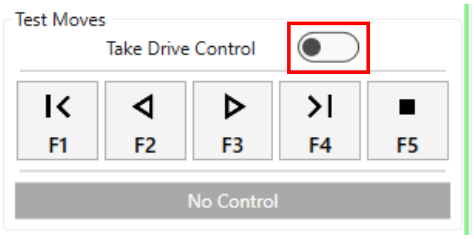
Provided the drive and motor is rated for a higher switching frequency, increasing the switching frequency can reduce audible noise from the motor.



If the motor has a continuing 1kHz tone after the adjusting the switching frequency and the current loop filter, the noise can be improved by making a small reduction in the Current loop P Gain with “Maintain Gain Ratio” selected to keep the balance between the current loop P and I gains.

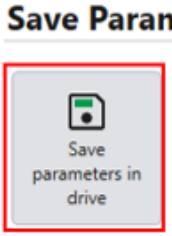


11. This stage of the tuning is complete. Further tuning may be required later to fully optimise the speed loop and position loop when the CODESYS program provides the motion reference to the drive over EtherCAT PDO. Disable the drive by removing 24V from the STO input, and then disable the test move control by clicking on the “Release Drive Control”. When drive control has been released it looks as shown below:



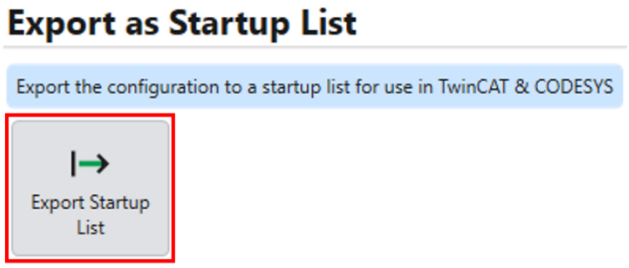
Click Next to move to the “Finalise” page.

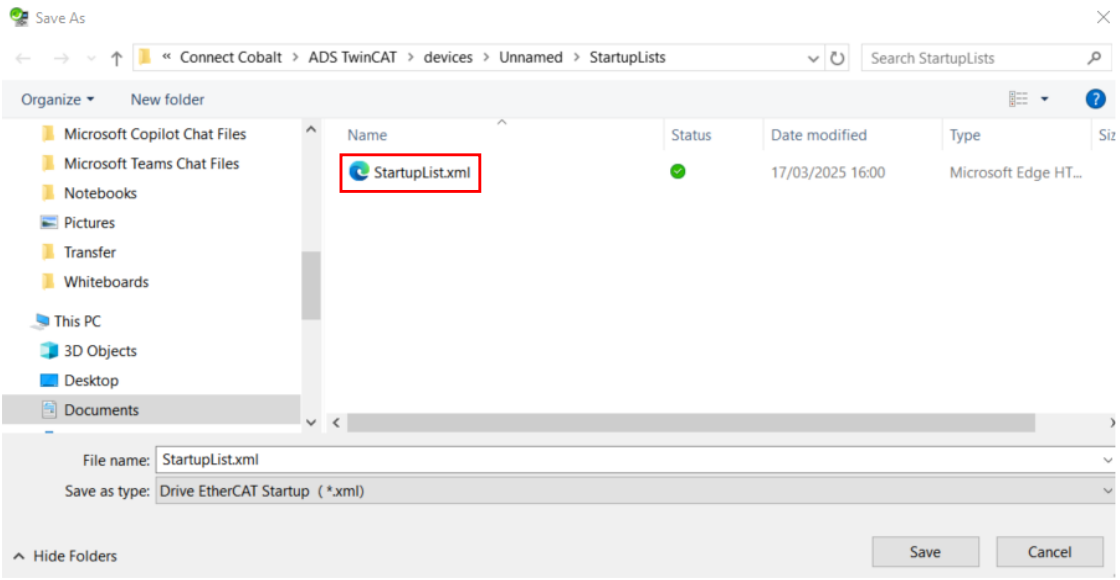
12. Save the tuned parameters in the drive by clicking the “Save Parameters” button:



13. An EtherCAT startup list may be generated to preserve the configuration in the CODESYS PLC/IPC. The startup list is in the form of an startup list.xml file that may be imported into CODESYS.

    It is recommended to perform this after the application software has been written and the position loop has been tuned.





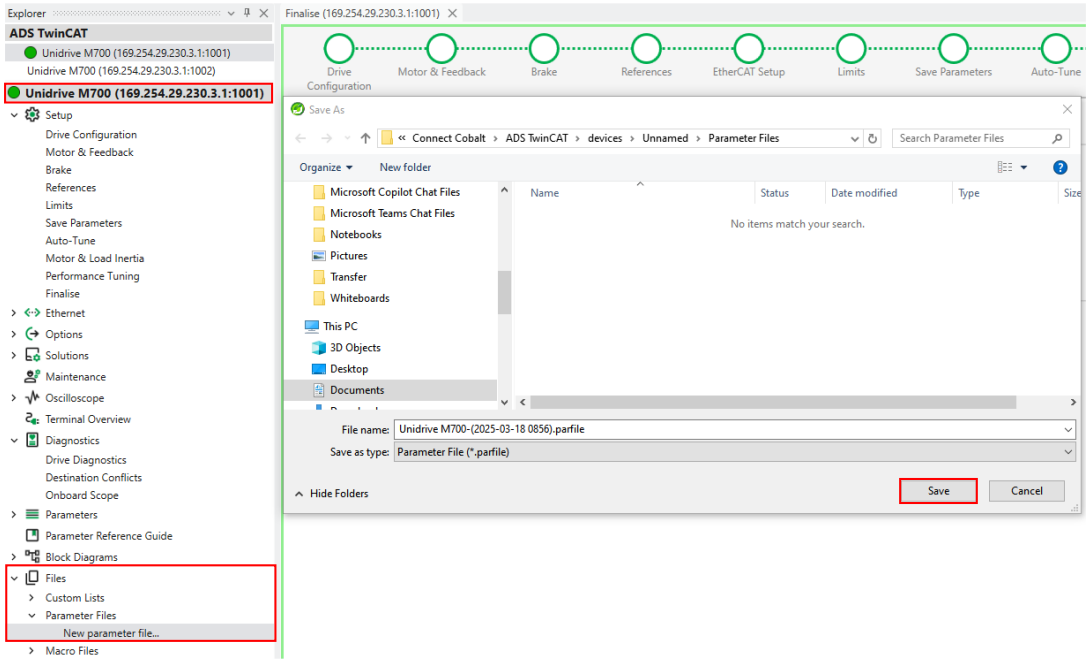
    StartupList.xml is stored in the following location: C:\Users\[USER NAME]\Documents\Control Techniques\Connect Cobalt\[PROJECT NAME]\devices\[DEVICE NAME]\StartupLists.

    See section **3.6 How to Import a startup parameter list** for more information on startup lists.

14. The startup list for an EtherCAT device is a helpful way to configure CiA402 CAN objects and drive parameters (addressed as CAN objects) when an EtherCAT node starts. There are a few standard ways in which this feature is used:

    - To configure CiA402 CAN objects that define how the system operates such as the behaviour of the drive when comms are lost via object 0x3005.
    - To apply the parameters used to setup a drive axis from scratch. This will restore the original drive configuration in case the drive parameters have been altered.
    - It can also be used to automatically configure a brand new drive (assuming the drive and EtherCAT option firmware match) in the event that a drive is replaced.

15. It is advised to make a parameter file once tuning has been completed by selecting “Files” > “Parameter Files” > “New parameter file…”. This file preserves the final setup of the drive, regardless of what happens to the Connect project later on, and forms a useful future reference of the configuration. Click “Save” to create the parameter file. The file is automatically time and date stamped, e.g. Unidrive M700-(2025-03-18 0856).parfile was created at 8:56 on 18/3/2025.



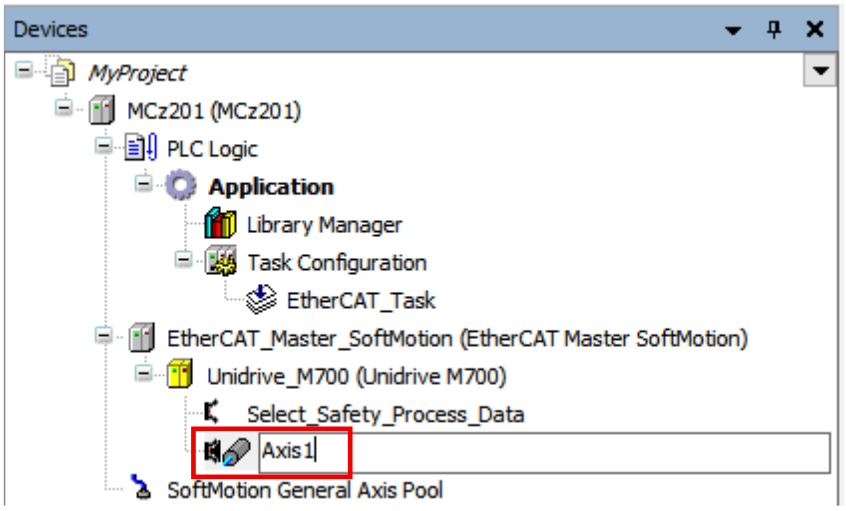
2.6 SoftMotion Axis Configuration

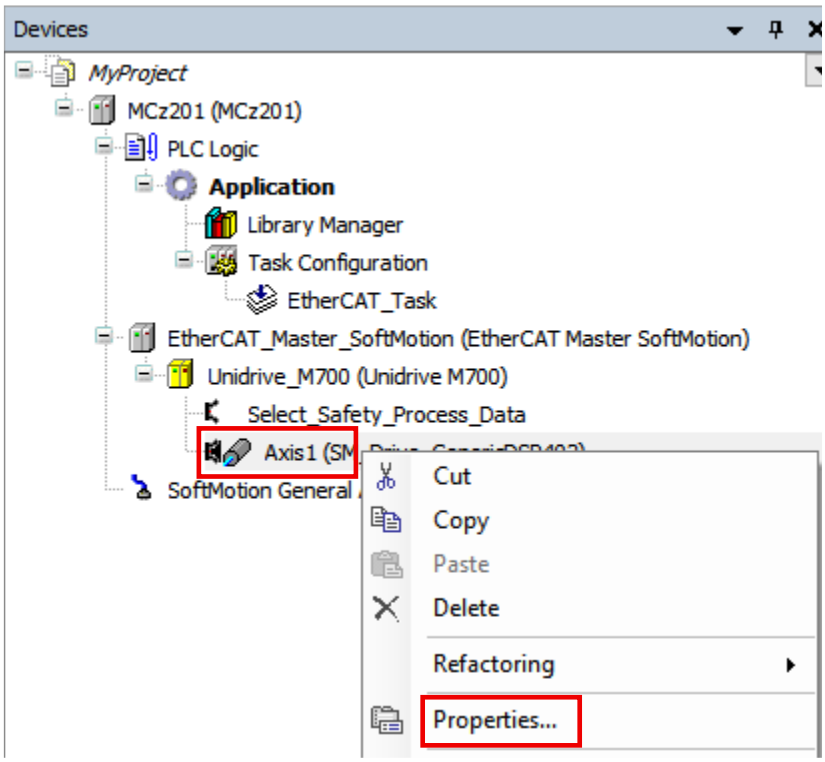
This section describes the basic setup of the SoftMotion axis that was created in the previous steps. Before beginning this section it is assumed that there is a connection from the PC to the CODESYS controller, there is a network connection to the drive and that it has been tuned according to the advice in section **2.4 Configure the drive using Connect and 2.5 Commission the drive and motor using Connect**.

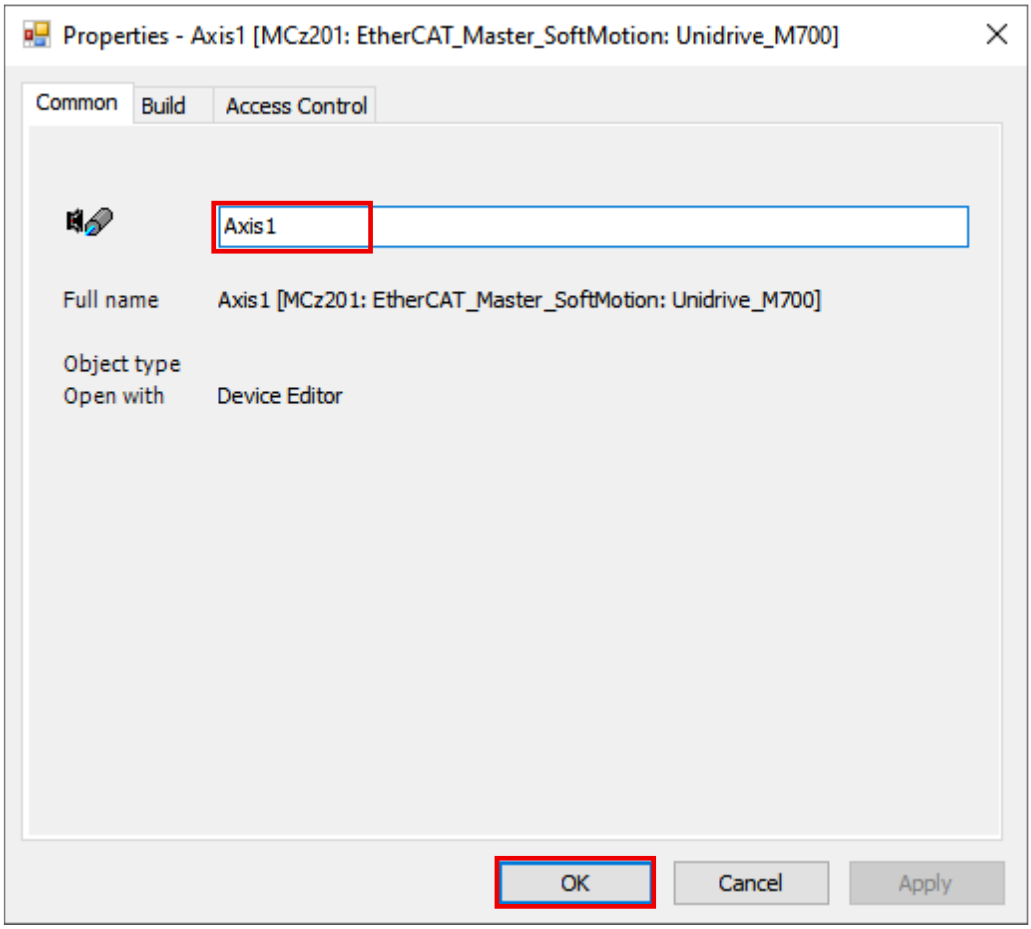
Before continuing log out of the controller by clicking the button in the tool bar or press Ctrl+F8. When the drive is logged out the button looks like this .

2.6.1 Rename the Softmotion axis

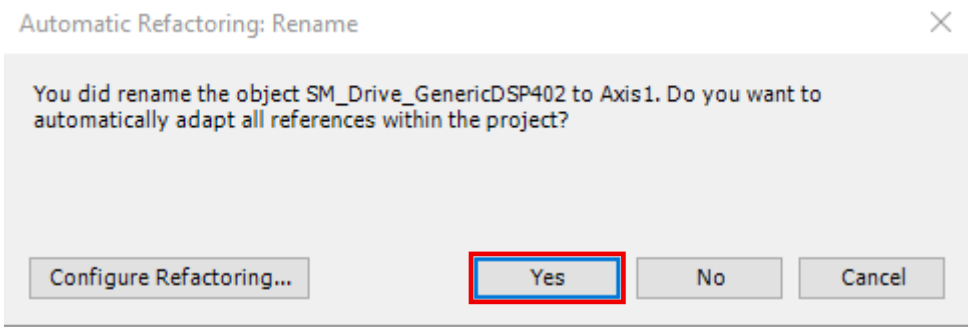
The default name for the axis is rather long and cumbersome to work with, at this stage it is recommended to give it a short but meaningful name e.g. Axis1. It can be changed directly in the tree view by clicking on the SoftMotion axis and then pause for 1s and then click again, or by right clicking on the Softmotion axis and select “Properties…”.

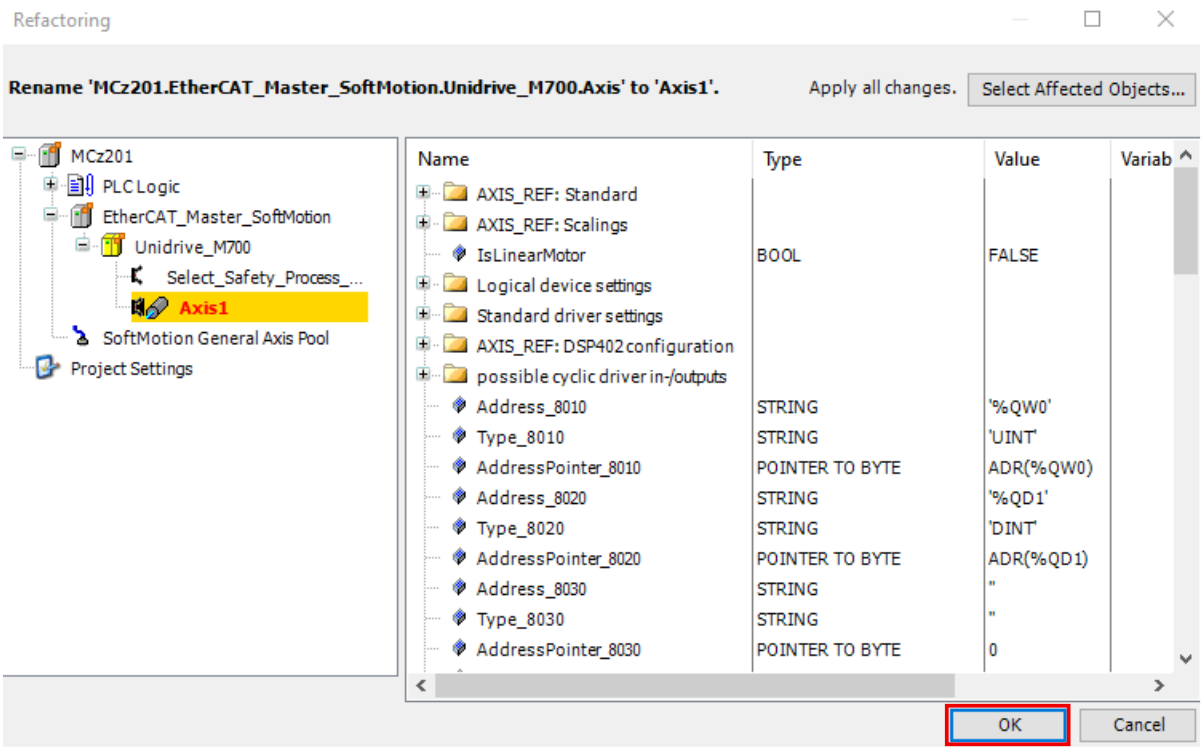





Properties - Axis1 [MCz201: EtherCAT_Master_SoftMotion: Unidrive_M700]
Common
Build
Access Control
Axis1
Full name
Axis1 [MCz201: EtherCAT_Master_SoftMotion: Unidrive_M700]
Object type
Open with
Device Editor
OK
Cancel
Apply

Click “Yes” or “OK” on the subsequent automatic refactoring pop-ups:



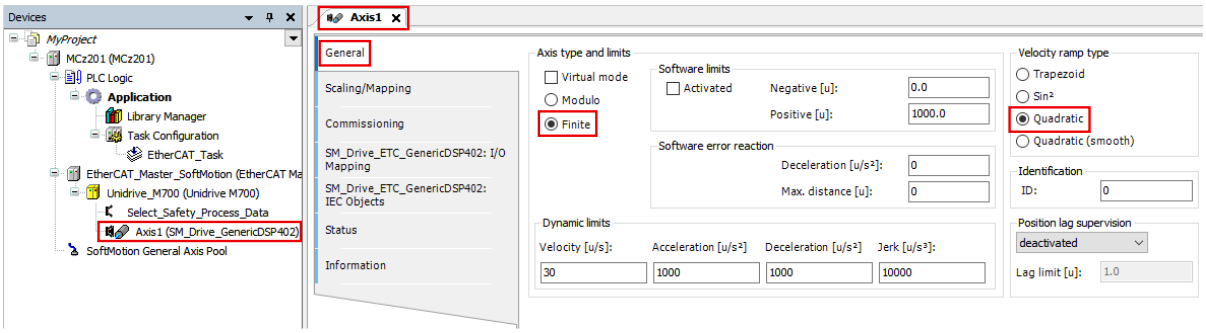


2.6.2 General setup tab

The general tab can be accessed by double clicking on the SoftMotion axis from the project tree, in this example the name of the SoftMotion axis is “Axis1”.

The general tab allows the user to configure the axis type, limits, velocity ramp type (trapezoid, Sin², quadratic, etc. and position lag supervision.

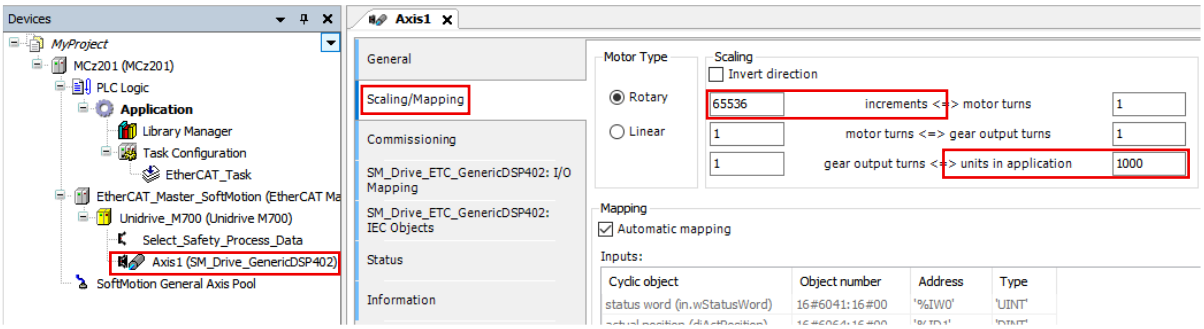
For a basic setup to get started it is recommended to set the Axis type to “Finite” and select “Quadratic” velocity ramps so that Jerk profiling is possible. Leave all other settings at their default values:



2.6.3 Scaling / Mapping setup tab

The Scaling / Mapping tab allows the user to define the type of motor in use and the relationship between technical units (for example, millimetres or degrees) and the drive units (increments).

The scaling “increments” defines the number of encoder counts for the number of “units in application”, and is used by the Softmotion system to convert the feedback values in encoder counts to technical user units. The easiest way to configure these two values is to set the “increments” to the default position feedback resolution of 65536 encoder counts per revolution of the motor, and then set the units to the number of units per revolution of the motor, in this example 1000.



See section **3.4 How to setup axis unit scaling and resolution** for more information on how to scale the units on a Softmotion axis.

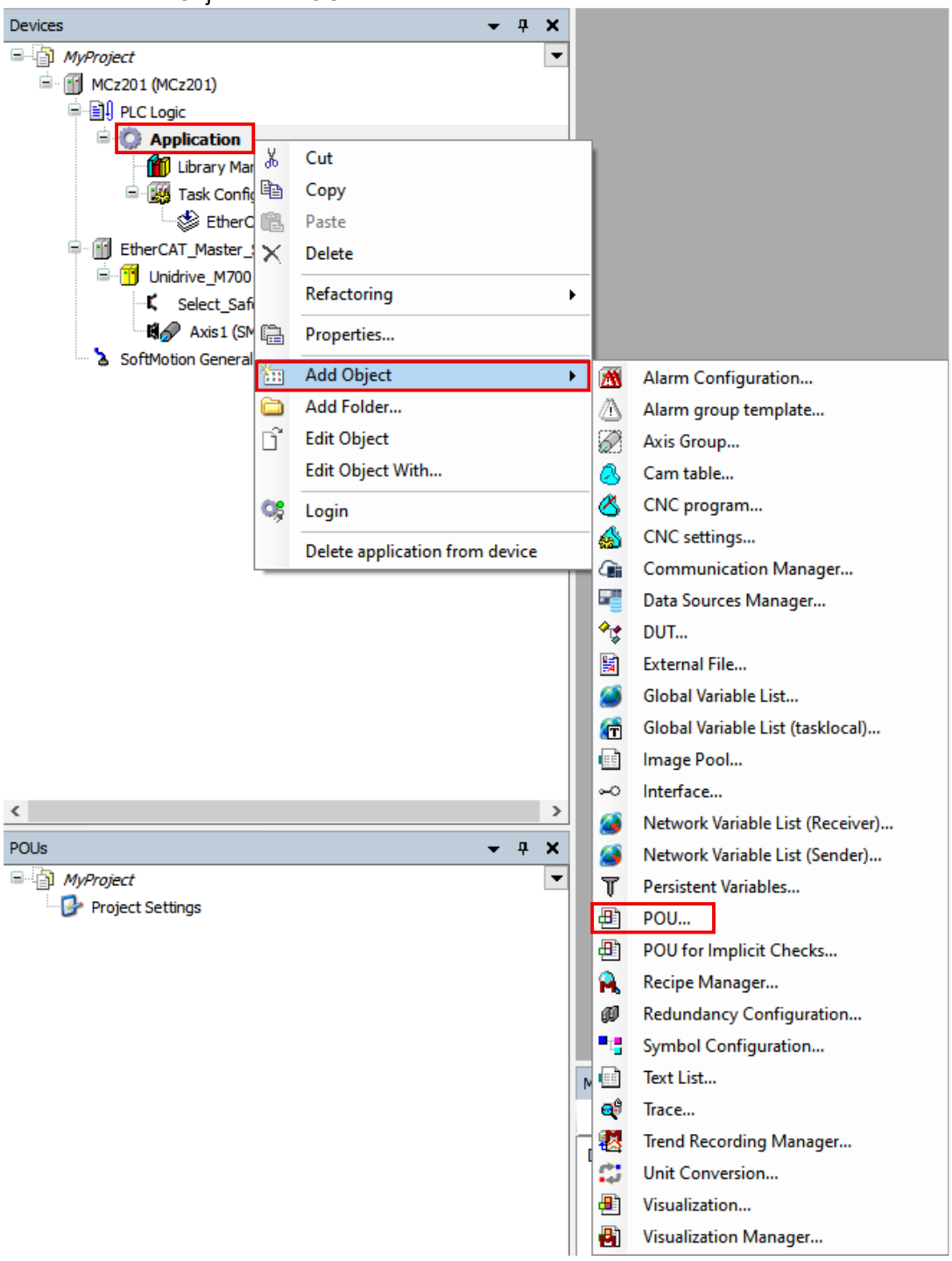
2.7 Using PLCopen to move the axis

This section gives a simple guide to setting up and testing the SoftMotion axis using PLCopen function block commands such as MC_Power, MC_Jog, and MC_Reset.

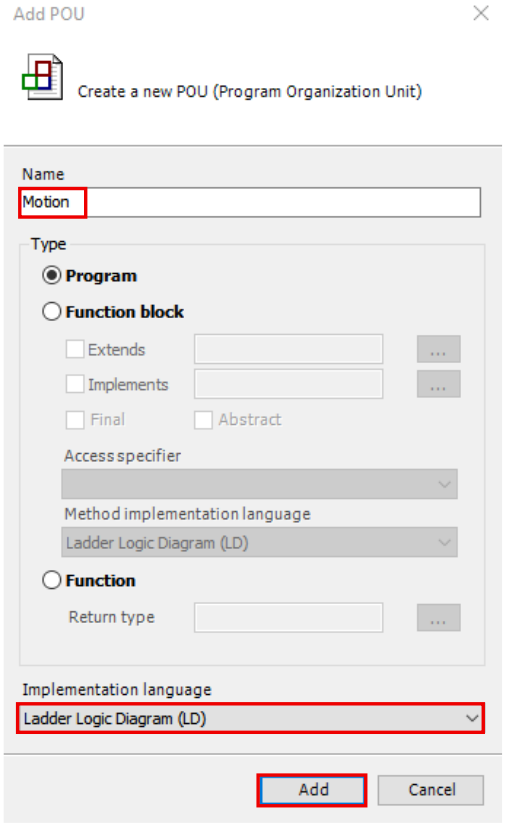
Before beginning this section it is assumed that all of the previous steps in section **2 Getting started** have been completed.

Make sure the CODESYS is logged out of the controller, i.e. the logging in button looks like this  . If it is still logged in, the logging in button looks like  click the button press Ctrl+F8.

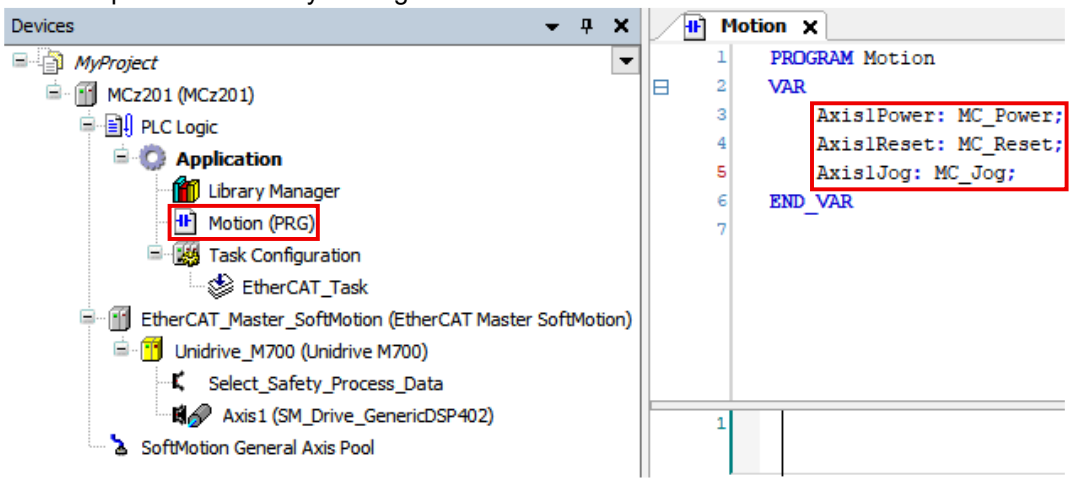
1. Add a POU to hold the Motion Software. Right click on “Application” in the Devices tree, and then select "Add Object” > “POU…”.



2. When the “Add POU” dialog opens, Set the name of the POU, e.g. “Motion”, set the implementation language, e.g. “Ladder Logic Diagram (LD)”, and then click “Add”.



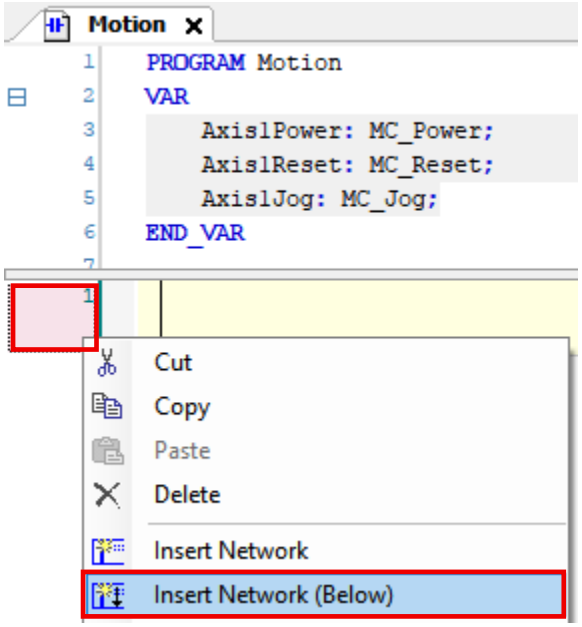
3. When the “Motion” POU is created, double click on it to open the POU. Add declarations for an MC_Power, MC_Reset and MC_Jog. When the Softmotion axis was created all of the required libraries to add the PLCopen motion blocks were automatically added so there is no need to update the “Library Manager”.



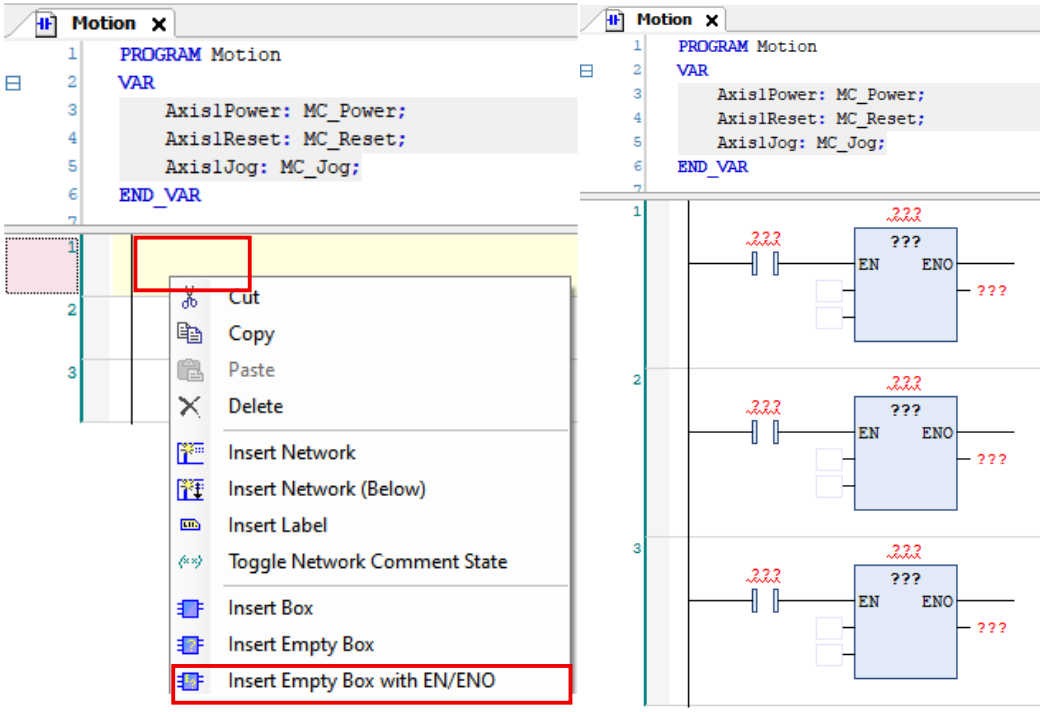
CODE: Copy the code below:

```
Axis1Power: MC_Power;
Axis1Reset: MC_Reset;
Axis1Jog: MC_Jog;
```

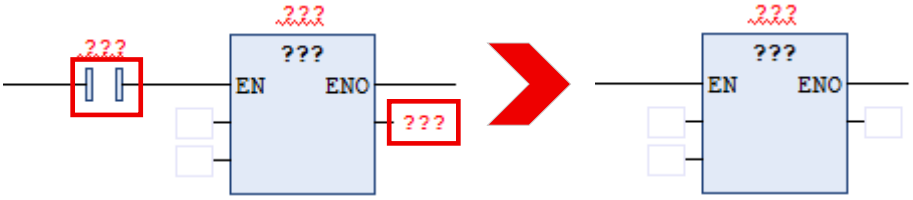
4. Add 2 extra networks, one for each PLCopen block. Right click on box to the left of the first rung and select “Insert Network (Below). Repeat this to add the 3rd network rung:



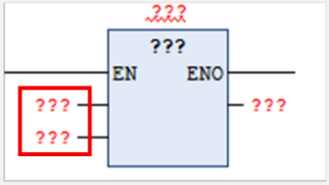
5. Add an EN/ENO empty box in each network rung, one for each PLCopen block. To do this Right click on the yellow space in a ladder rung and then select “Insert Empty Box with EN/ENO”.



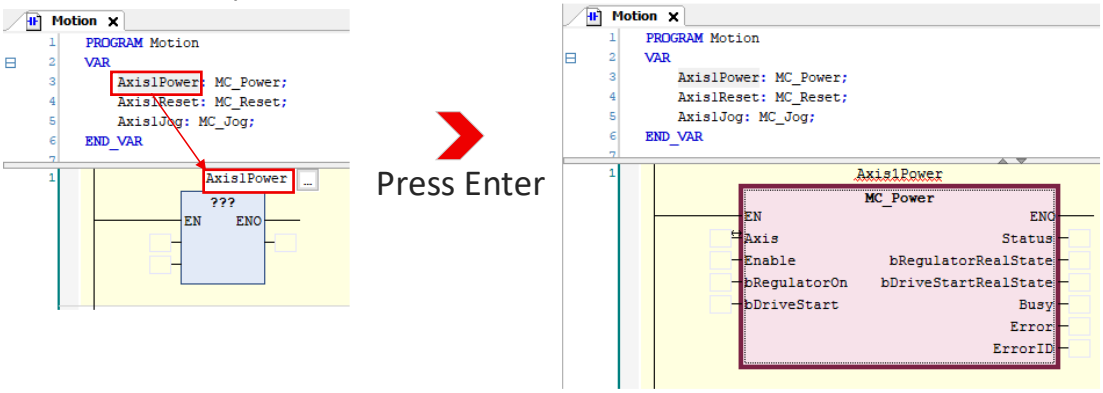
6. Select and delete the contacts to the left of the boxes and delete the “???” at the output.



HINT: If there are “???” at the inputs to the function, block follow the instructions in section **3.5 How to remove “???” from LD diagram inputs and outputs**.

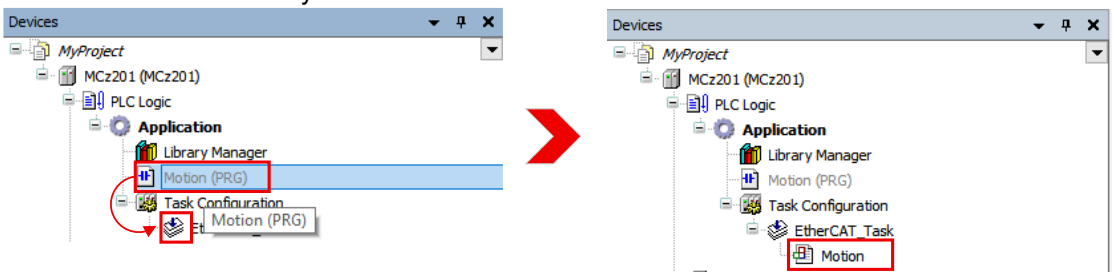


7. Assign the function block instances to all 3 boxes. This can be done in a few ways but one of the easiest is to simply copy and paste the instance name from the declaration to the box instance name and press enter:

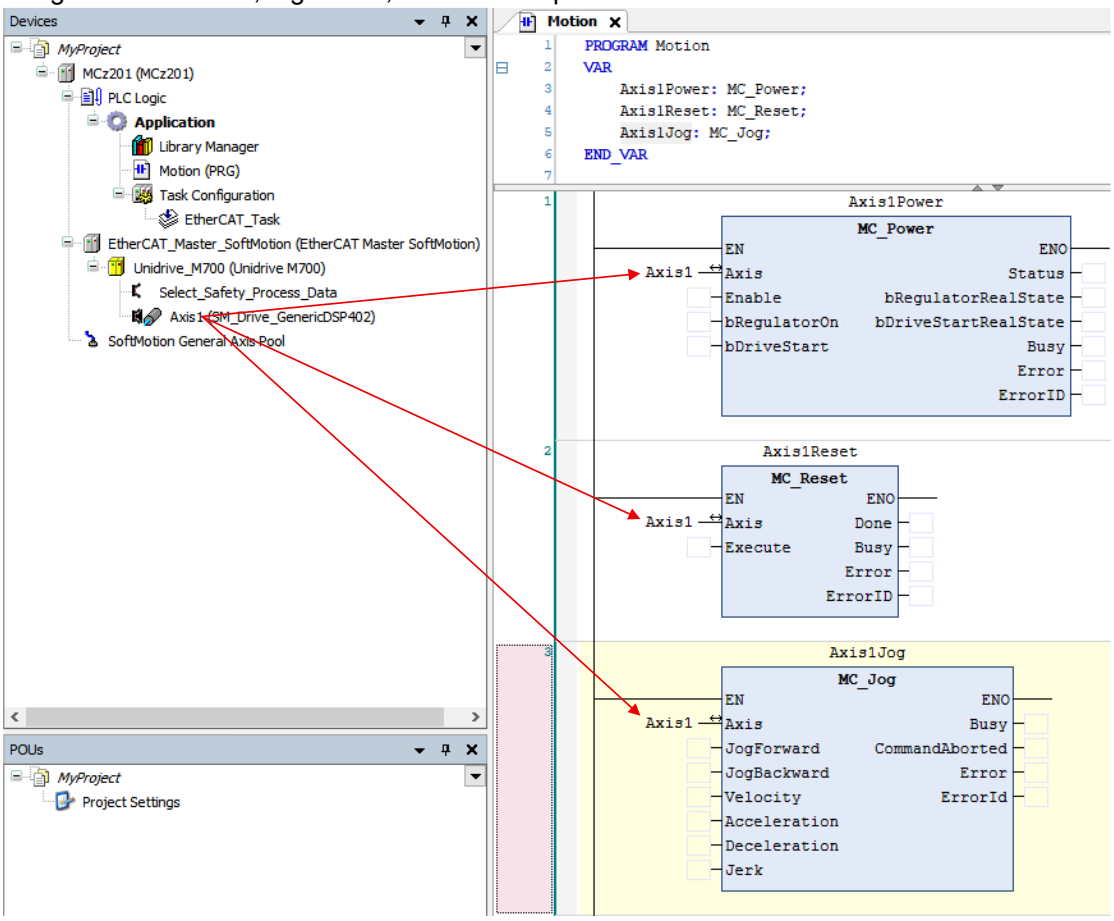


Repeat this process for the other 2 function block instances.

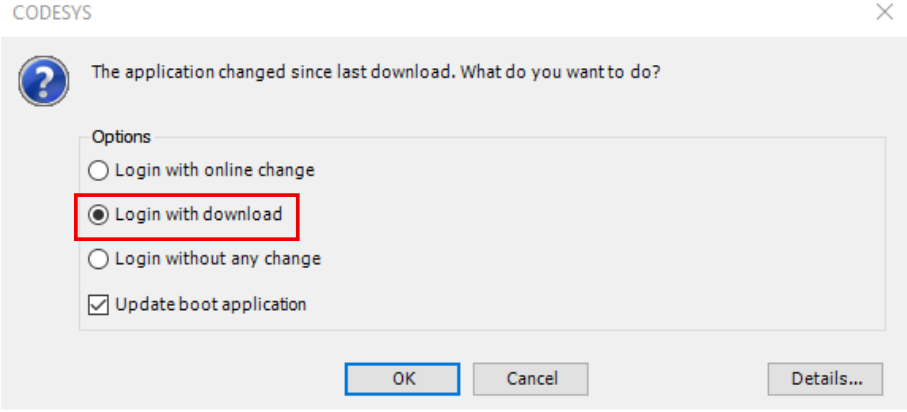
8. Drag and drop the Motion POU on to the EtherCAT_Task in the Device tree so that the Motion POU is serviced by the EtherCAT task.



9. Assign the axis name, e.g. Axis1, to the Axis input of all 3 function blocks.

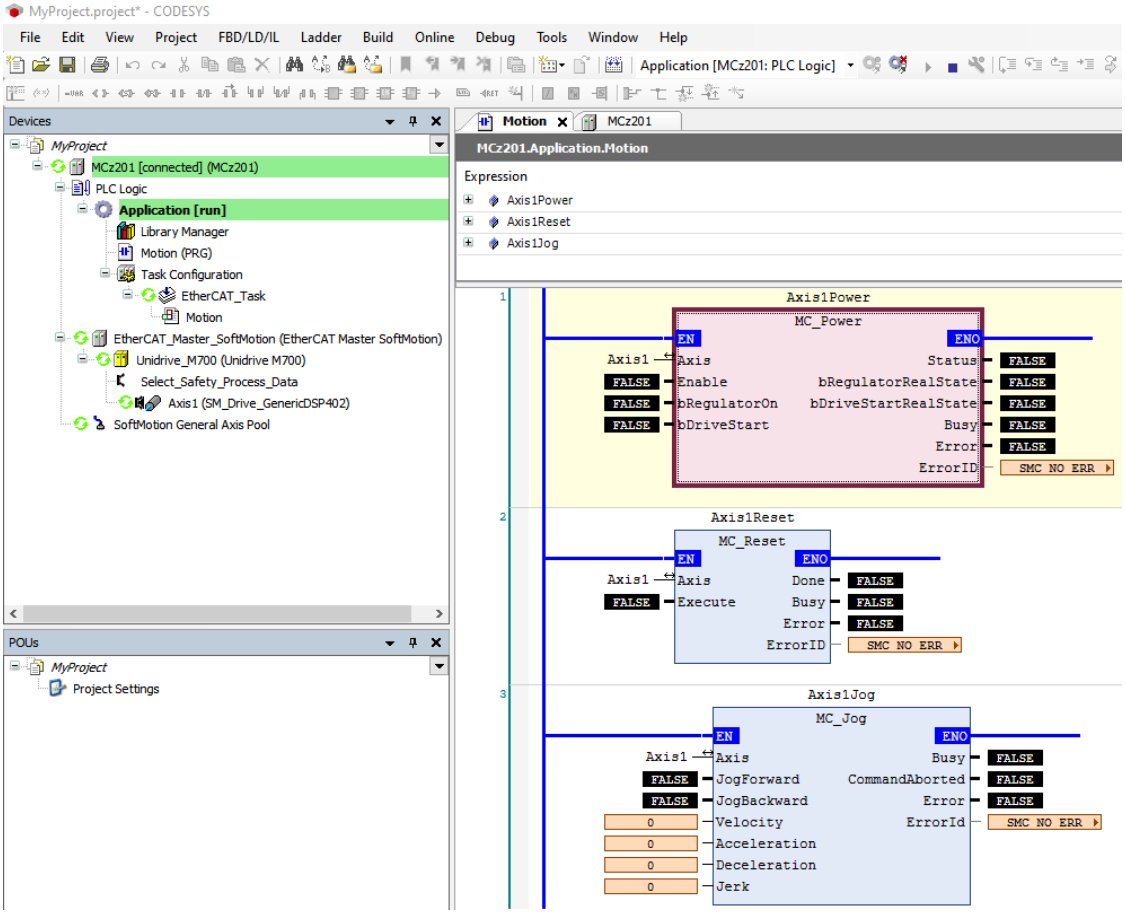


The project is ready to build and download. Press Ctrl+F8 or ,  . When the application dialog starts select “Login with download”.



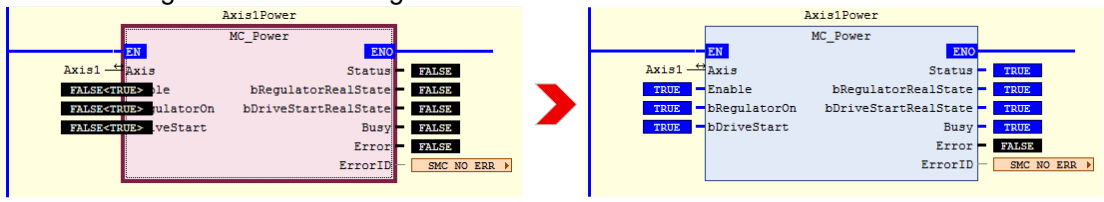
When the logging in button looks like  click the button press F5 or press the start button  .

10. The program looks like this when it is running after the download:



11. To try the features of the axis, values may be entered directly in the ladder diagram editor view by double clicking on an input, setting the new value, and then pressing Ctrl+F7 to apply the value.

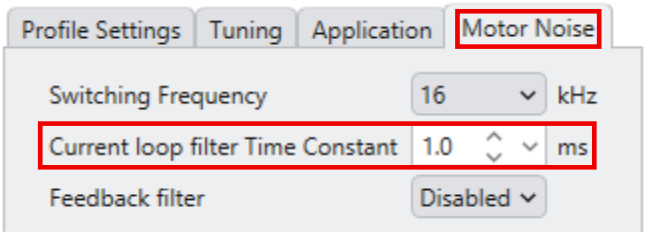
12. At this stage, **provided it is safe to do so**, jogging motion may be tested on the axis. To run the axis Set the MC_Power instance Axis1Power inputs Enable, bRegulatorOn and bDriveStart to TRUE and apply 24V to the enable / STO input(s). To set the inputs, click on “FALSE” at each input to change the setting. The axis motor should remain stationary but the drive will energise the motor and go in to “Run”.



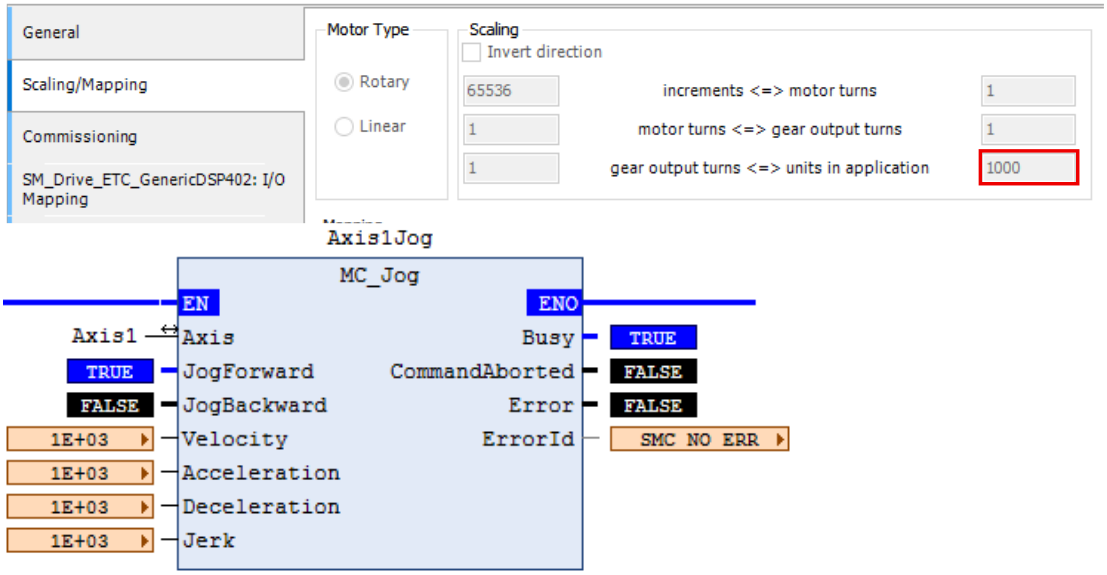
HINT: If the drive isn’t in Run, make sure that Prx.033 is set to off(0), where x is the slot menu where the EtherCAT interface is fitted. Slot1 = Menu 15, Slot2 = Menu 16, Slot2 = Menu 17, Slot4 (Factory Fit slot on Unidrive M) = Menu 24.

The factory fitted slot for the EtherCAT interface on a Digitax HD M753 is Slot3 = Menu 17

13. If the axis motor is noisy when it becomes energised e.g. due to a low resolution feedback encoder, increase the Current loop filter time constant Pr4.012 using the Motor Noise tab on the Performance Tuning step in Connect. In most cases setting this filter value to 1.0ms will help get rid of the motor noise created by a low-resolution feedback encoder.



14. Jog the axis at a slow speed <100rpm an verify the axis moves smoothly. Set the MC_Jog Velocity, Acceleration, Deceleration and Jerk inputs to the units per rev scaling value from the scaling and mapping tab e.g. 1000 – this will result in the axis running at 60rpm.



15. If activating CMD_JogForward results in reversed physical motion, tick the “Invert direction” box which can be found under the “Scaling / Mapping” tab for the SoftMotion axis to correct the direction. For more information see section **2.6.3 Scaling / Mapping setup tab**.
16. Further tuning may be required for the best performance. See section **3.12 How and when to tune the Position Loop** for details on how to do this.

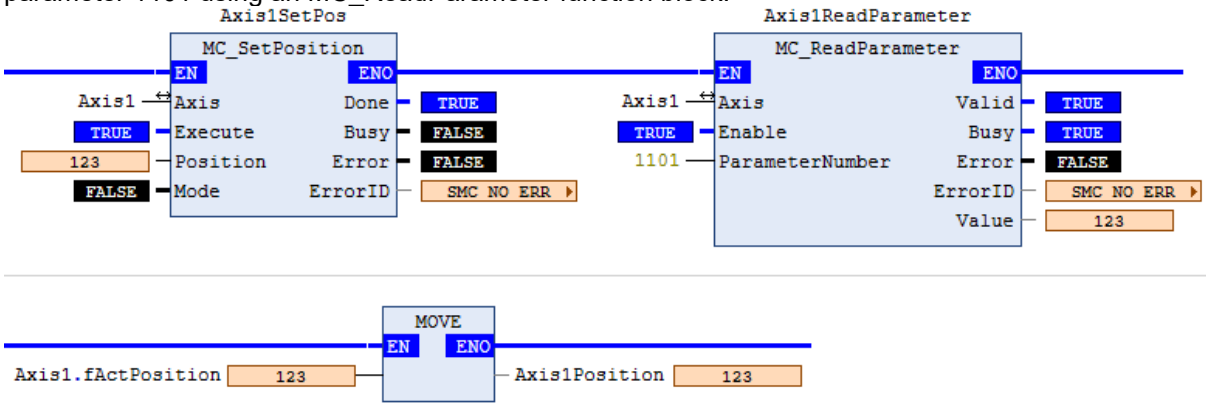
3 How to guides

3.1 How to home a Softmotion axis

Homing is simply resetting the axis position to a known value when the axis is in a known position.

3.1.1 Manual homing using MC_SetPosition

The axis can be moved manually by jogging to a known position and then the position reset by invoking MC_SetPosition where the Position input defines to new axis feedback position. The position feedback value can be read either by Looking at e.g. Axis1.fActPosition or by reading the PLCopen parameter 1101 using an MC_ReadParameter function block.

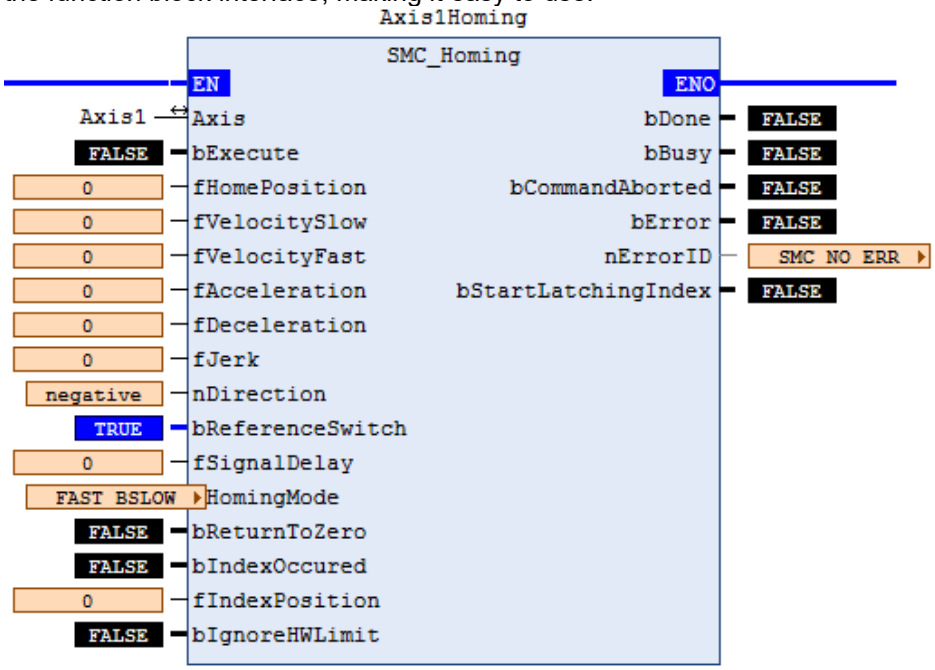


For more information see MC_SetPosition.

3.1.2 Automated homing using SMC_Homing

When homing a Softmotion axis in the CODESYS environment, it is possible to use MC_Home from the PLCopen standard however this involves setting up several CoE object writes to access the DS402 homing mechanism, where DS402 is the underlying motion system that EtherCAT uses to implement motion over comms. While this is something that could be done it isn't recommended since it is very cumbersome to setup and use.

To make things easier, CODESYS implemented the SMC_Homing function block which gives the user an automated 2 step homing routine where all of the setup values for the homing routine are in the function block interface, making it easy to use.



For more information see MC_Homing.

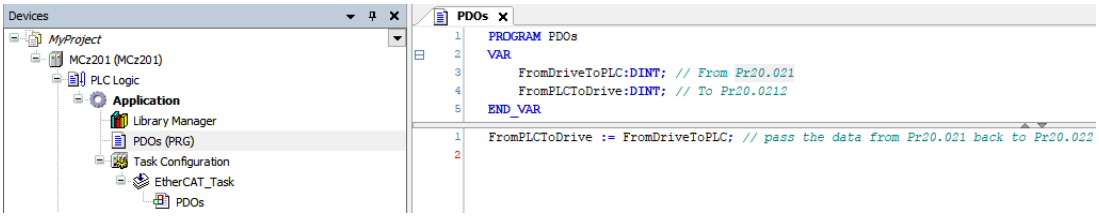
3.2 How to configure PDO mappings

3.2.1 How pass data between drive parameters and variables using PDOs

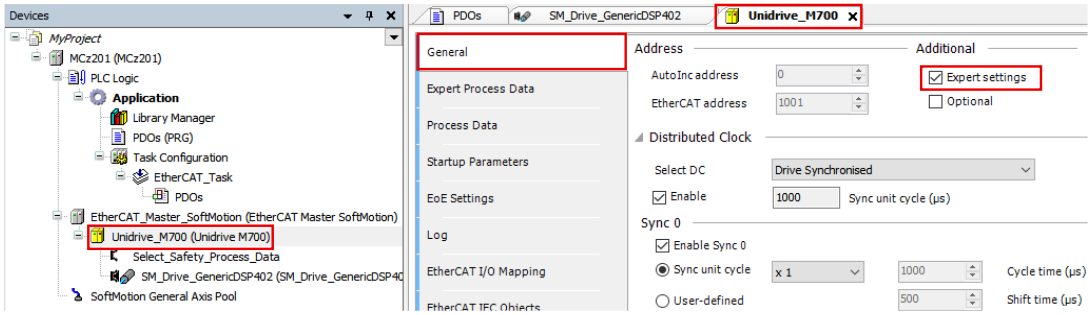
This section shows how to map data from a PLC program variable to a drive parameter via PDO mapping, and from a drive parameter to a PLC program variable via PDO mapping. The following instructions show how to map to and from Menu 20 parameters in the drive, however the same philosophy can be applied to any parameter.

1. For the purposes of this example the PDO mappings will interact with variables in a POU that has been added to the program called “POUs”.

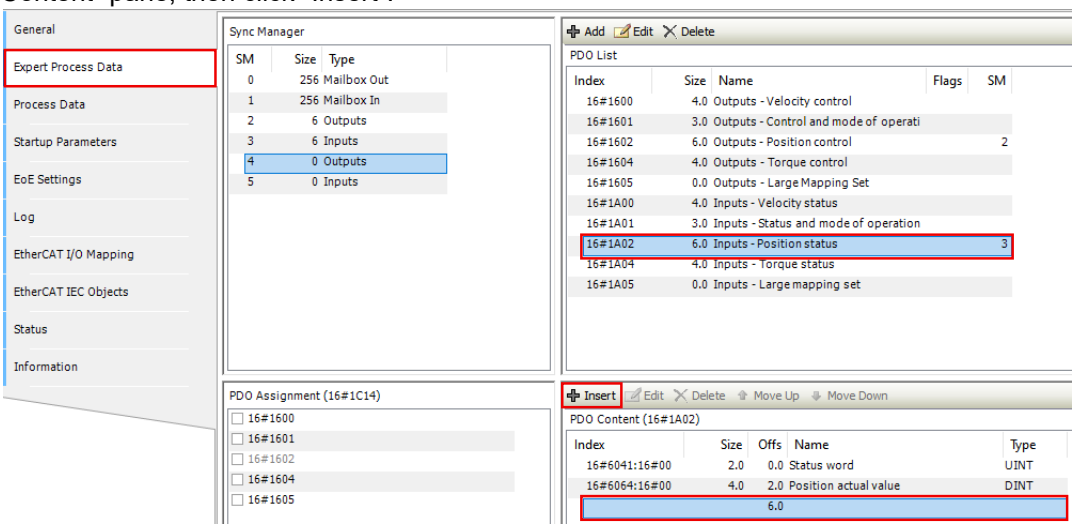
   In the POUs program, the variable “FromDriveToPLC” has been declared to hold the value of Pr20.021 in the drive. The variable “FromPLCToDrive” has been declared to hold the value that will be written to Pr20.022 in the drive. A line of program code passes the value of “FromDriveToPLC” to “FromPLCToDrive”.



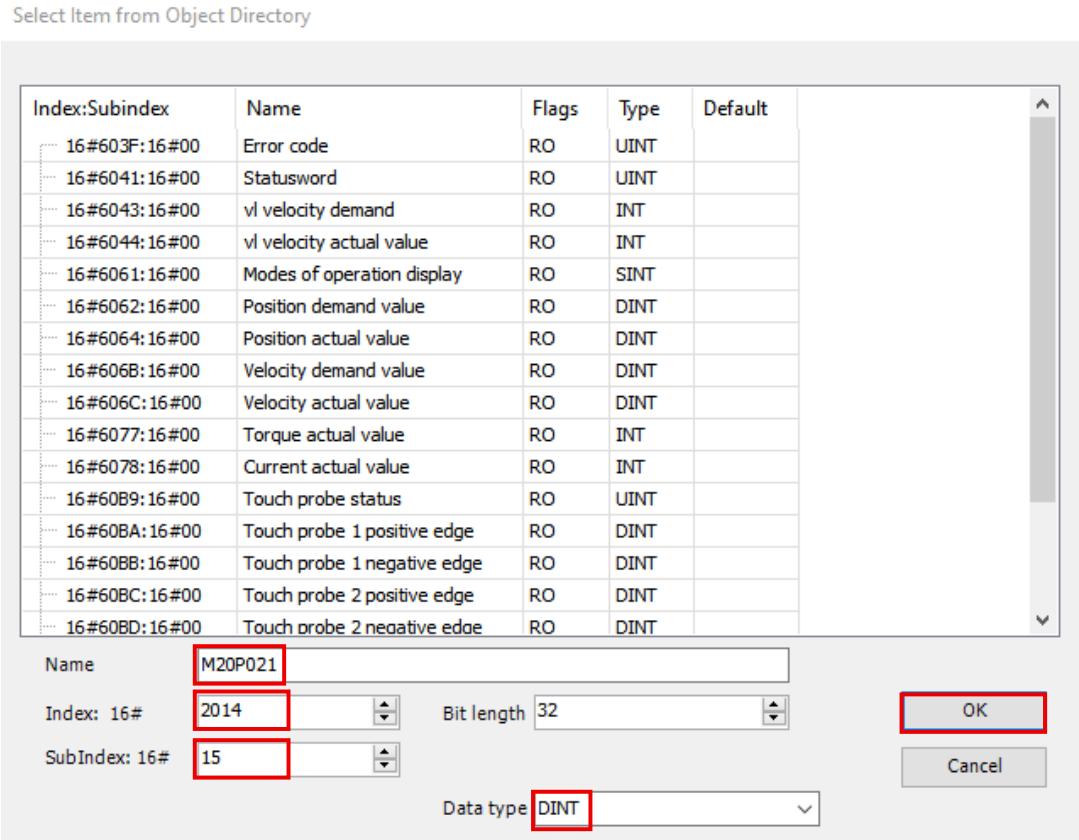
2. Enable the “Expert Process Data” tab to allow PDOs to be manually configured. Double click the EtherCAT Slave drive in the “Devices” tree e.g. “Unidrive M700 (Unidrive M700)”. When the slave device dialog starts, select the “General” tab and then check the “Expert Settings” box.



3. Next the PDO links must be made to read data from and write data to the drive parameters. Select the “Expert Process Data” tab and then check the “Expert Settings” tab. Click on the “16#1A02” row in the “PDO List” pane and then select the first empty row in the “PDO Content” pane, then click “Insert”.



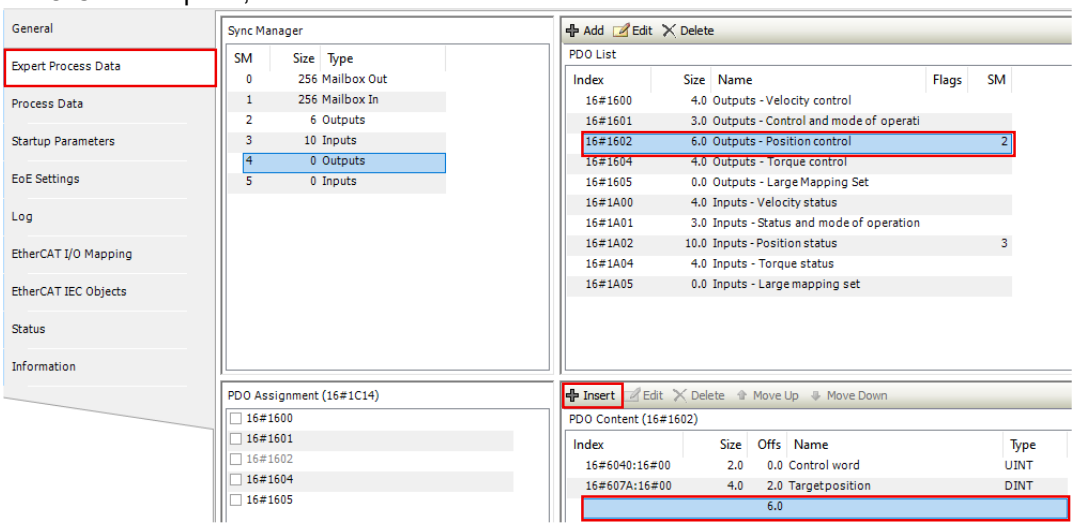
4. When the “Select Item from Object Dictionary” dialog opens set Name, Index, SubIndex and Data type as shown:



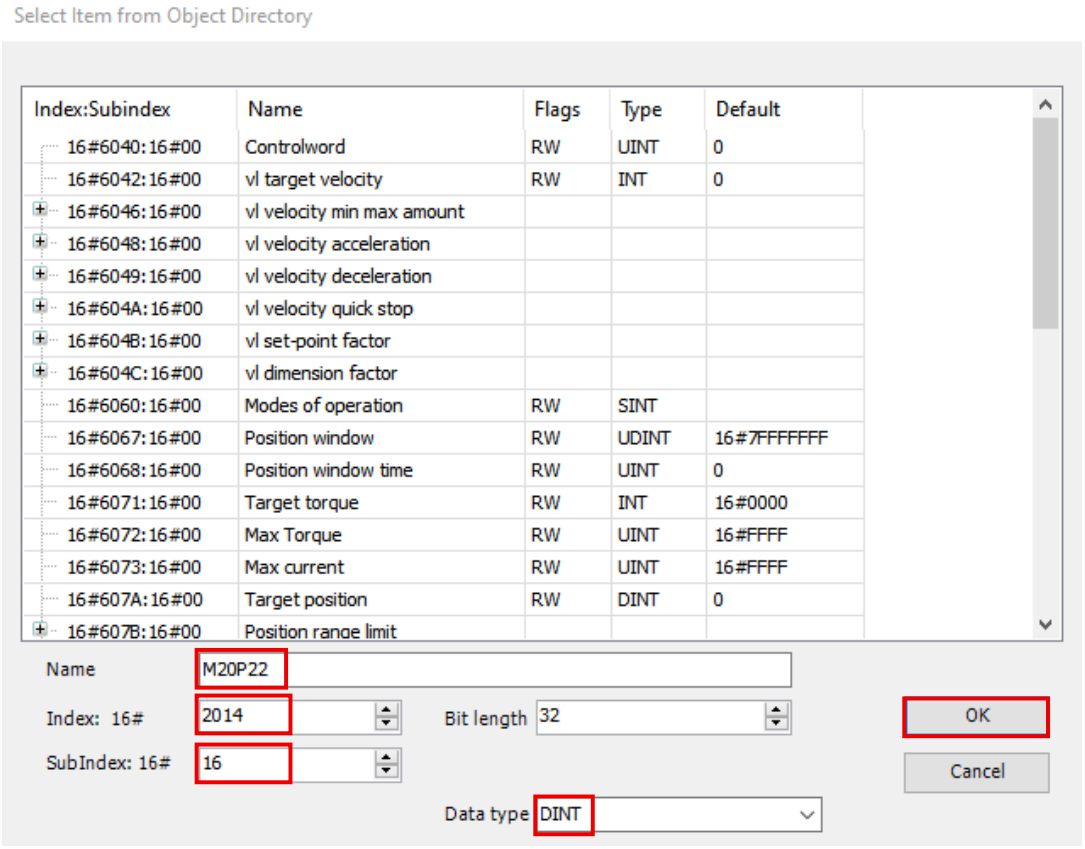
- The Index is set to 16#2000 + 16#14 (menu number converted to hexadecimal).
- The SubIndex is set to the parameter number (21) converted to hexadecimal 16#15
- Pr20.021 is a 32bit signed value which aligns with the DINT data type.
- See sections **3.2.1.1** to **3.2.1.4** for more details on parameter conversion.

Click “OK” when finished.

5. Click on the “16#1602” row in the “PDO List” pane and then select the first empty row in the “PDO Content” pane, then click “Insert”.



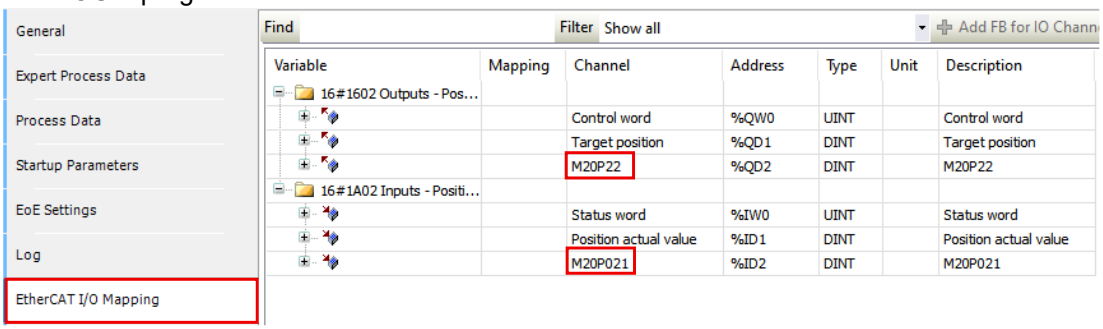
6. When the “Select Item from Object Dictionary” dialog opens set Name, Index, SubIndex and Data type as shown:



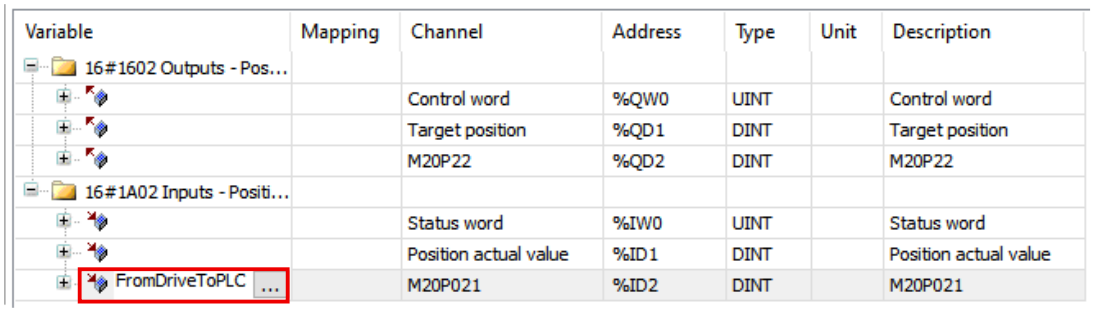
- The Index is set to 16#2000 + 16#14 (menu number converted to hexadecimal).
- The SubIndex is set to the parameter number (22) converted to hexadecimal 16#16
- Pr20.022 is a 32bit signed value which aligns with the DINT data type.
- See sections **3.2.1.1** to **3.2.1.4** for more details on parameter conversion.

Click “OK” when finished.

7. The PDO mappings are now completed. The data must now be mapped to the variables in the “POUs” program.

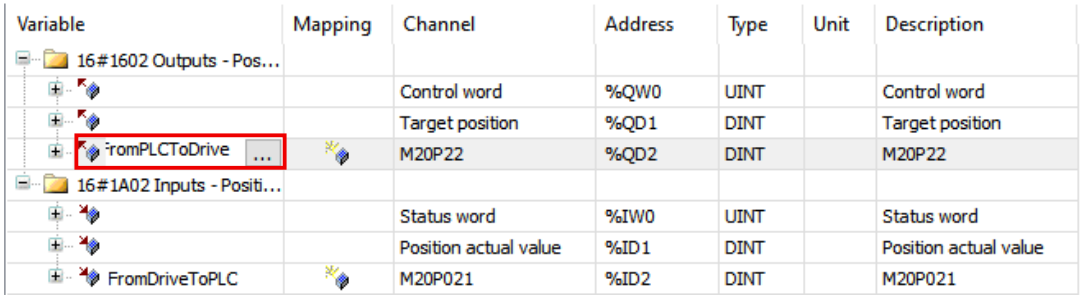


8. Double click in the “Variable” to the left of the “M20P021” cell and type in “Application.POUs.FromDriveToPLC” or select it by clicking the … button, and press enter when complete.



Variable	Mapping	Channel	Address	Type	Unit	Description
16#1602 Outputs - Pos…						
		Control word	%QW0	UINT		Control word
		Target position	%QD1	DINT		Target position
		M20P22	%QD2	DINT		M20P22
16#1A02 Inputs - Positi…						
		Status word	%IW0	UINT		Status word
		Position actual value	%ID1	DINT		Position actual value
FromDriveToPLC …		M20P021	%ID2	DINT		M20P021

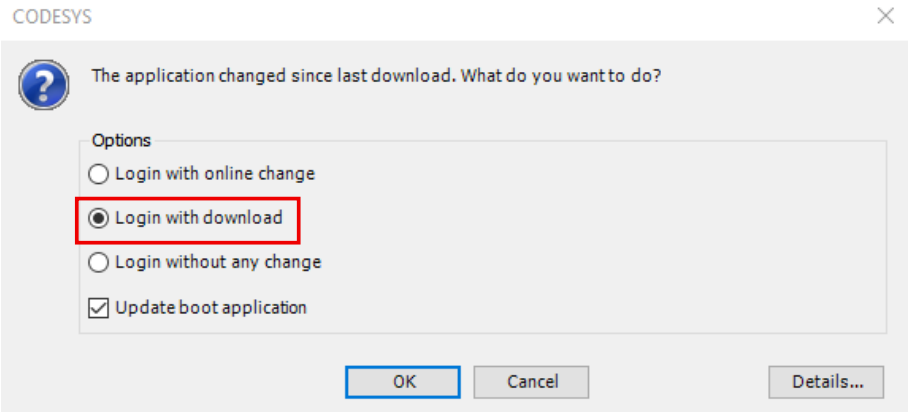
9. Repeat this process for “M20P022” and the variable “Application.POUs.FromPLCToDrive”.



Variable	Mapping	Channel	Address	Type	Unit	Description
16#1602 Outputs - Pos…						
		Control word	%QW0	UINT		Control word
		Target position	%QD1	DINT		Target position
romPLCToDrive …		M20P22	%QD2	DINT		M20P22
16#1A02 Inputs - Positi…						
		Status word	%IW0	UINT		Status word
		Position actual value	%ID1	DINT		Position actual value
FromDriveToPLC		M20P021	%ID2	DINT		M20P021

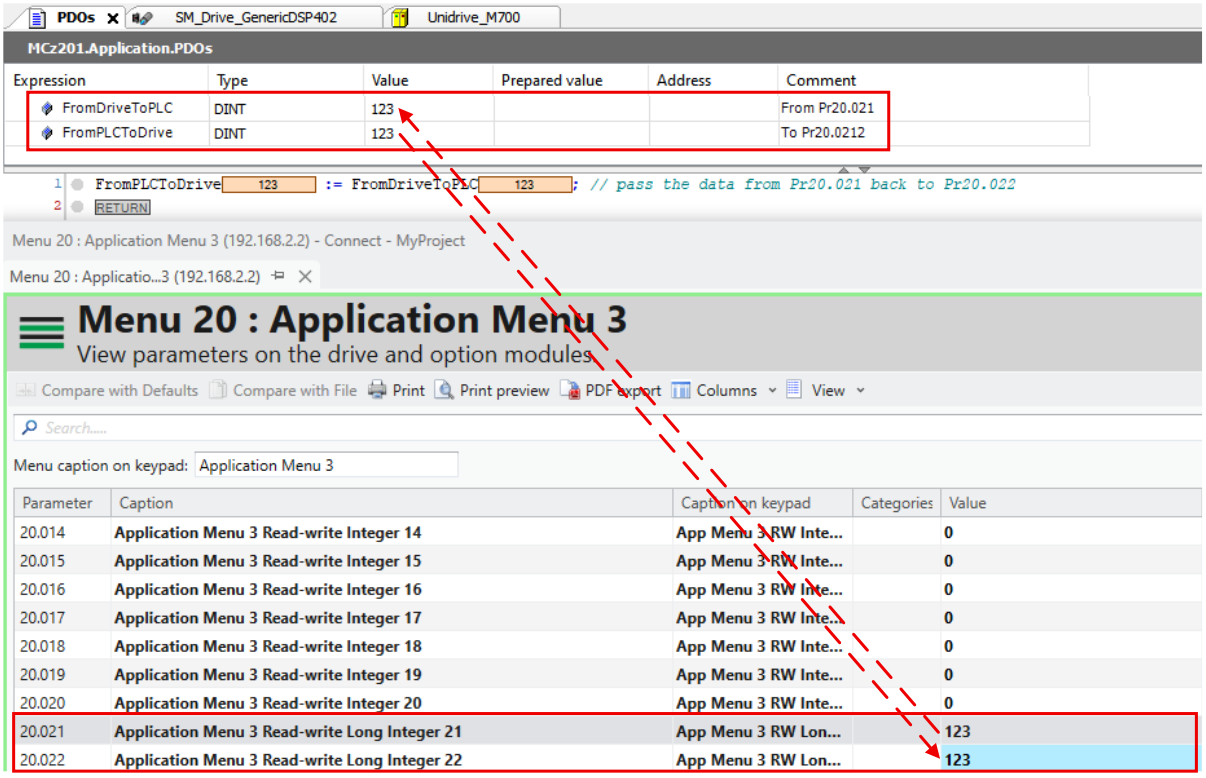
The PDO data is now mapped to the program variables.

10. The project is ready to build and download. Press Ctrl+F8 or , . When the application dialog starts select “Login with download”.



When the logging in button looks like click the button press F5 or press the start button .

11. In the image below it can be seen that “123” is set in Pr20.021 using Connect, the EtherCAT PDO links then return the value back to Pr20.022.



3.2.1.1 How to convert parameter numbers to CANopen object references

Parameters are converted to CANopen style object reference using the following formula:

Index = 0x2000 + 0x100 * Slot Number + Menu number (converted to hexadecimal)

Subindex = Parameter number (converted to hexadecimal)

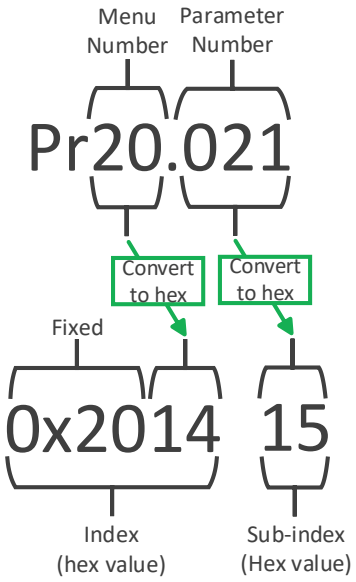
E.g. Drive parameter Pr3.017 is referenced as Index = 2003, Subindex = 11

HINT: The slot number is 0 for standard drive parameters such as Pr3.017.

3.2.1.2 Short drive parameter references

Pr20.021 → Index 0x2014 Sub-index 15

Short drive parameter references (mm.ppp)

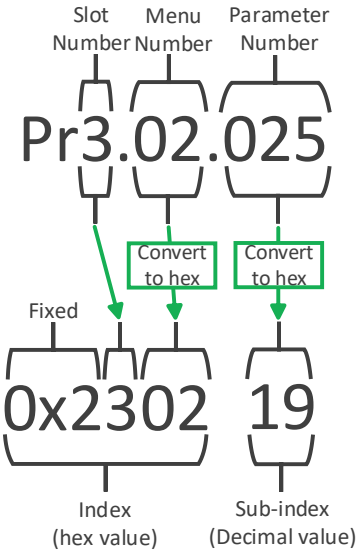


Equivalent CAN object

3.2.1.3 Long option parameter references

Pr3.02.025 → Index 0x2302 Sub-index 25

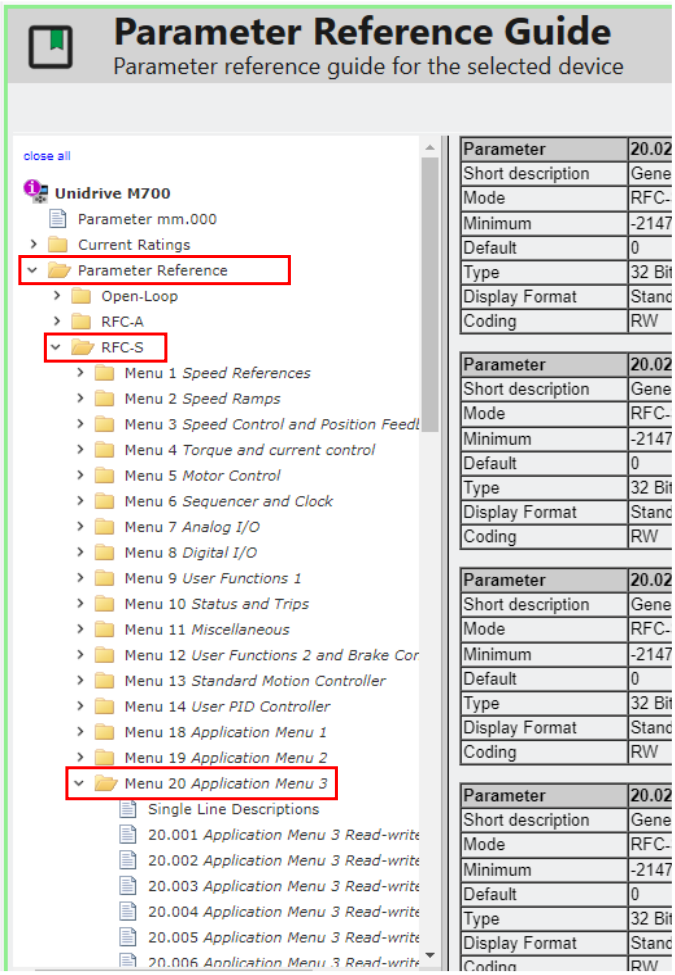
Long option slot parameter references (s.mm.ppp)



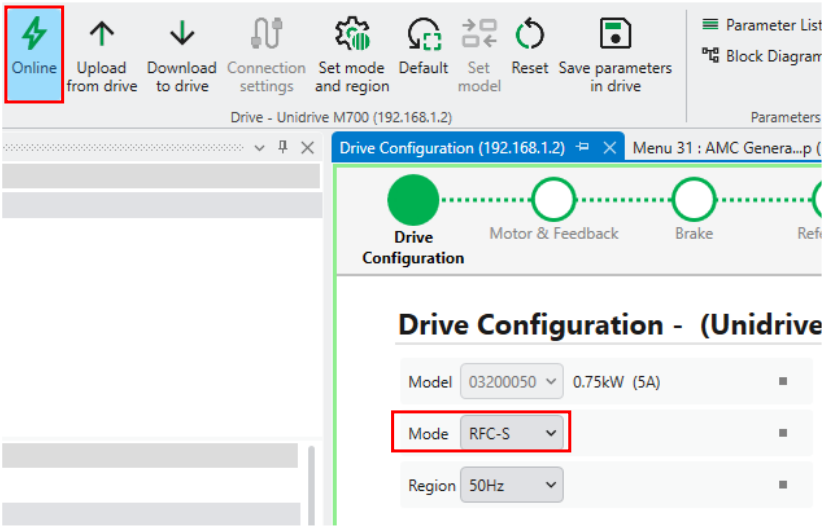
Equivalent CAN object

3.2.1.4 How to find the parameter data type

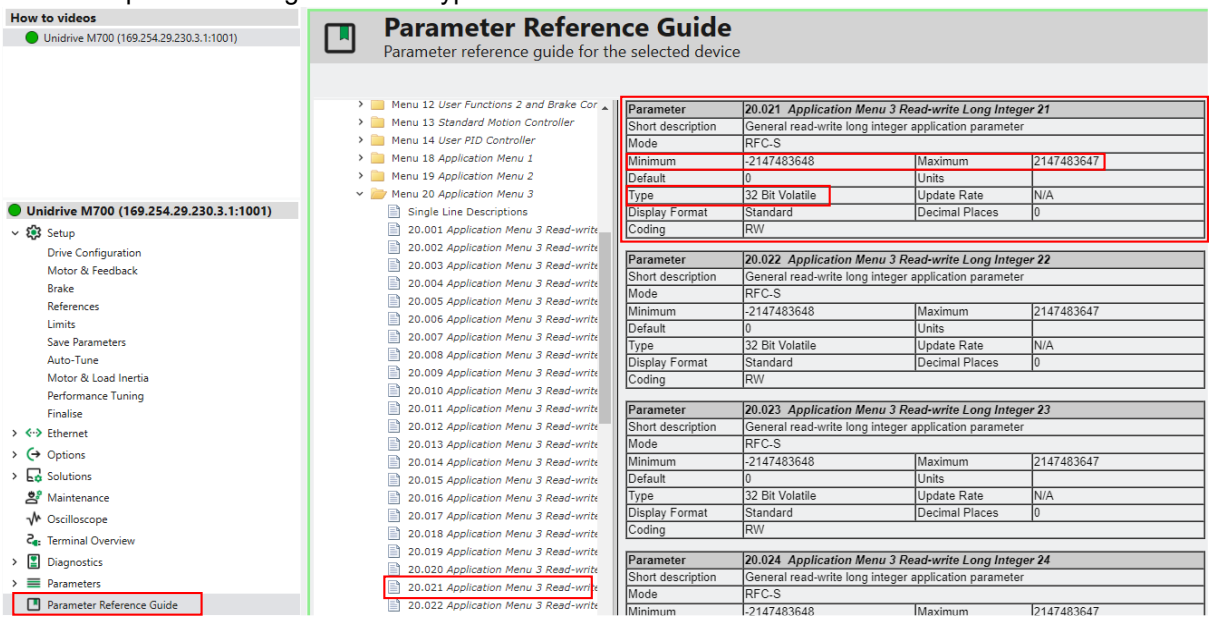
The data type can be found by looking at the parameter reference guide in Connect. Click “Parameter Reference Guide” in the Device tree and then expand “Parameter Reference”, then expand the drive mode e.g. Open-Loop, RFC-A or RFC-S, then select the menu number for the target parameter for the PDO e.g. Menu 20.



If you are unsure of the mode that the drive is in double click on “Setup” > “Drive Configuration” and the current mode will be shown when the drive is “Online”.



Select the parameter to get the data type for:



The parameter reference guide gives a number of bits and a minimum and maximum which may be used to find the IEC data type e.g. DINT. The table below show how to convert to IEC data type used by CODESYS:

Bits from parameter reference guide	Signed range	IEC data type
1	N/A	BOOL
8	No	USINT
8	Yes	SINT
16	No	UINT
16	Yes	INT
32	No	UDINT
32	Yes	DINT

E.g. Pr20.021 is a 32bit value and the range is -2147483648 to 2147483647 which shows that the value is signed, therefore the IEC data type used in CODESYS is DINT.

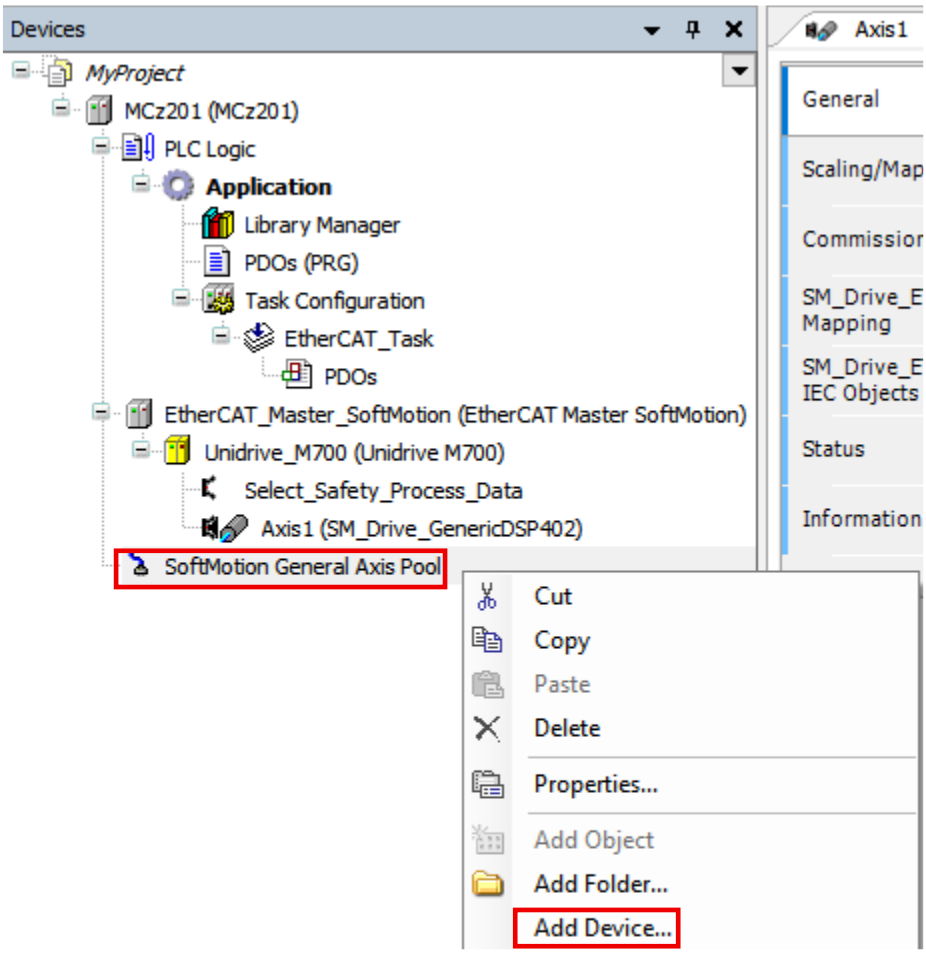
Bits from parameter reference guide Pr20.021	Signed range	IEC data type
32	Yes	DINT

3.2.2 How to map a drive encoder to a SoftMotion axis.

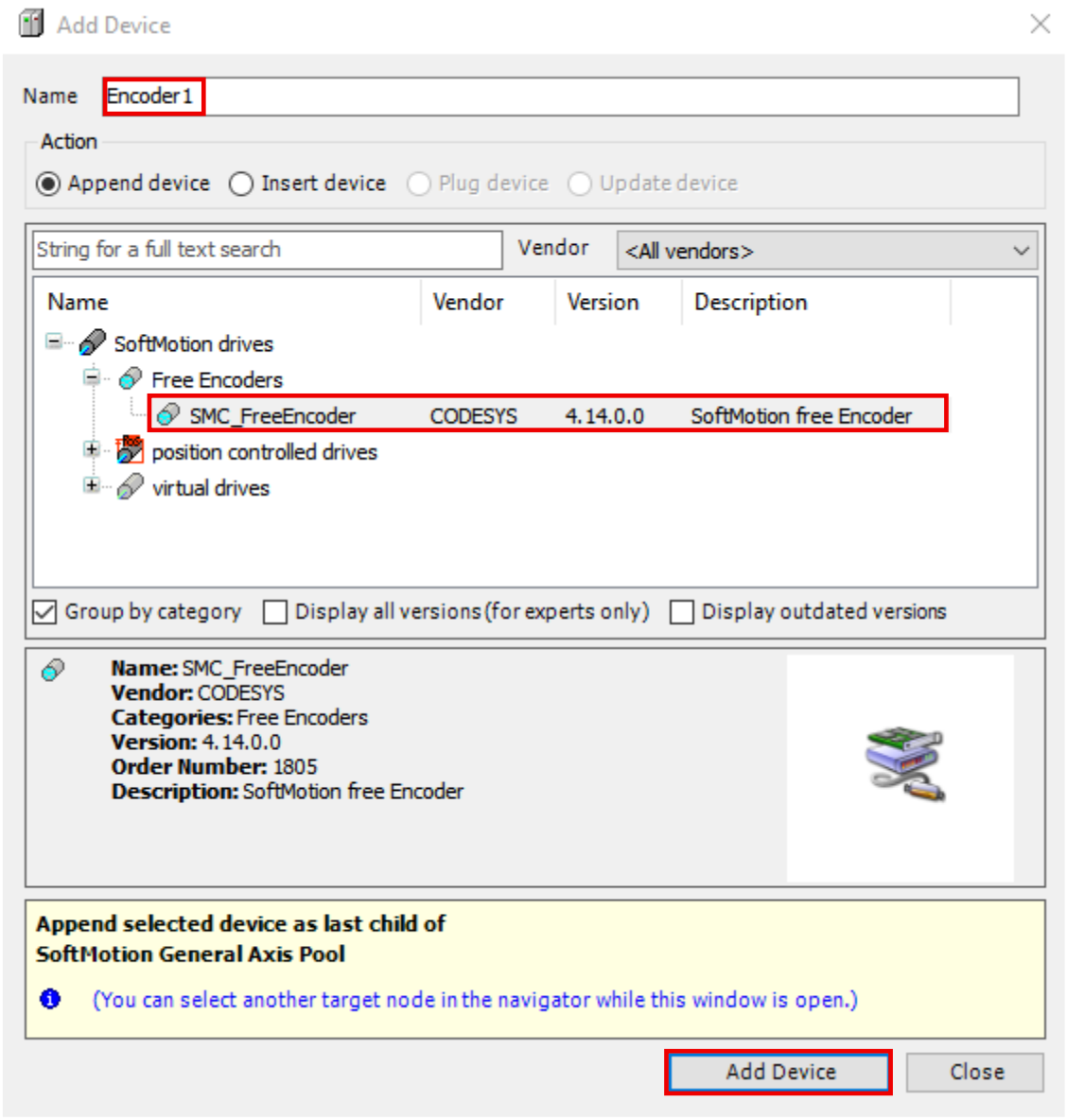
This section describes how to configure a cyclic mapping of Pr03.158 P2 Normalised Position so that it can be consumed by a SoftMotion FreeEncoder axis.

This allows the drives P2 position to be consumed by an encoder axis, where it could then be consumed by the CT SoftMotion Interface Library. An example of this is where the user would like to configure a dual loop system.

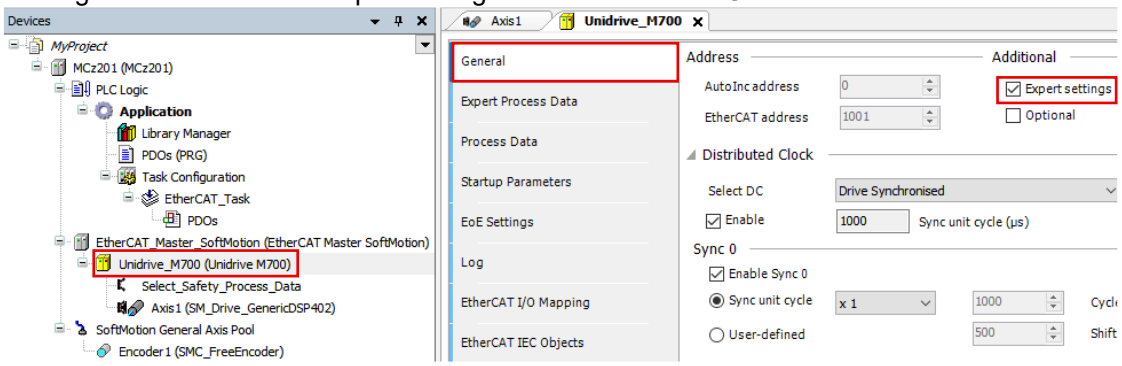
1. From the project tree, right click on the “SoftMotion General Axis Pool” device and select “Add Device…”.



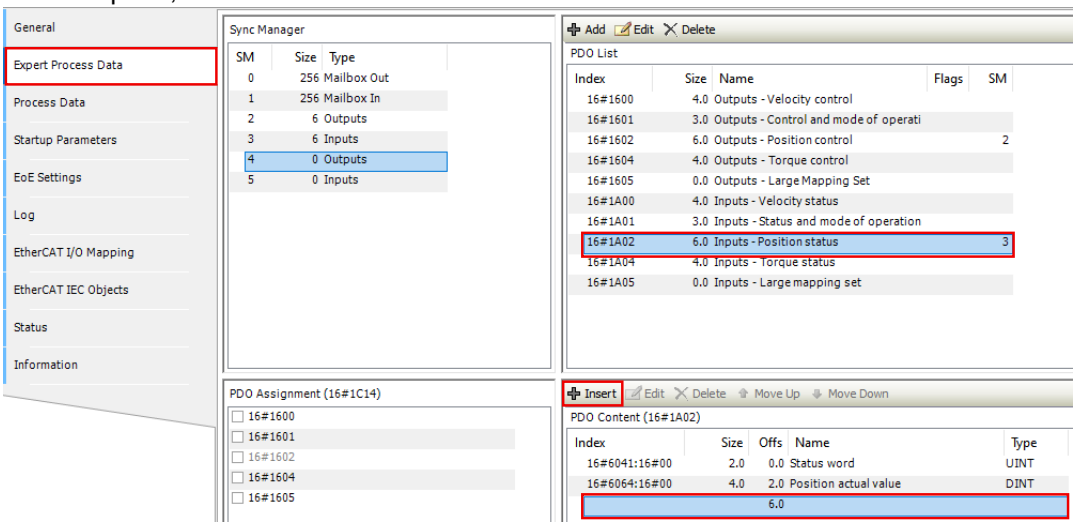
2. The “Add Device” dialog box will appear. Expand “Free Encoders” and select “SMC_FreeEncoder”, name the device e.g. “Encoder1” the followed by the “Add Device button.



3. From the project tree, double click on the drive (EtherCAT slave device) that the P2 position will originate from. Ensure “Expert settings” is sticked on the “General” tab.



4. Next the PDO links must be made to read data from and write data to the drive parameters. Select the “Expert Process Data” tab and then check the “Expert Settings” tab. Click on the “16#1A02” row in the “PDO List” pane and then select the first empty row in the “PDO Content” pane, then click “Insert”.

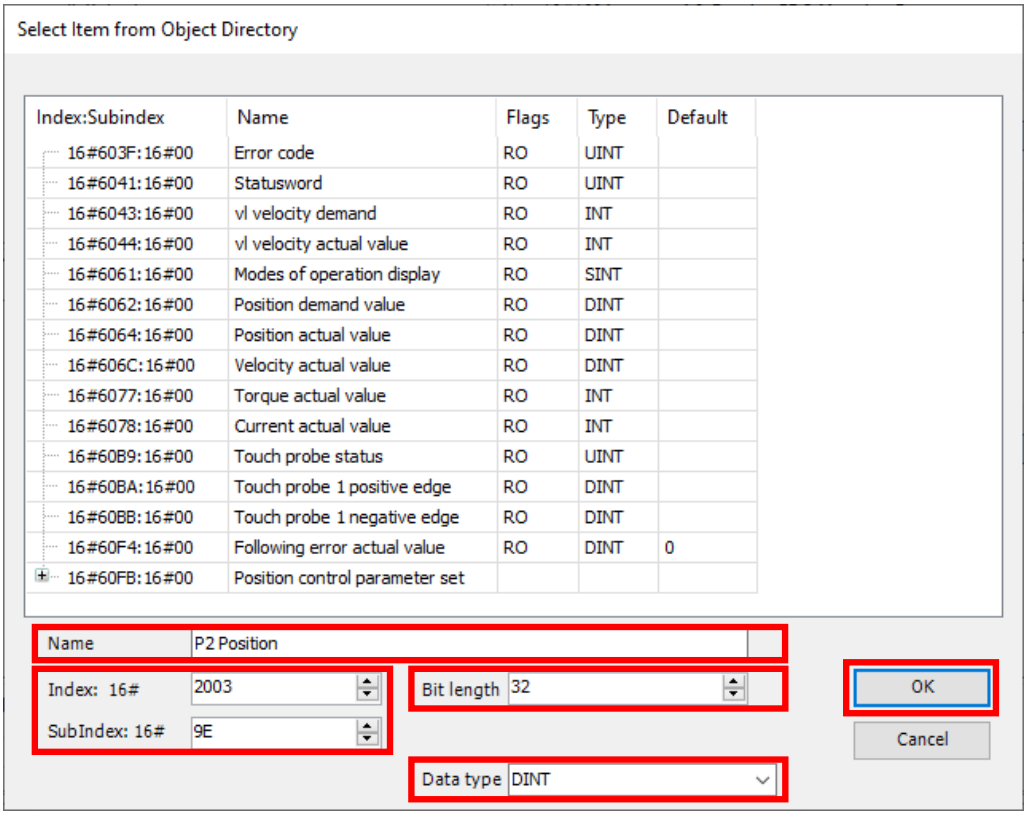


5. The “Select Item from Object Directory” dialog box will appear.

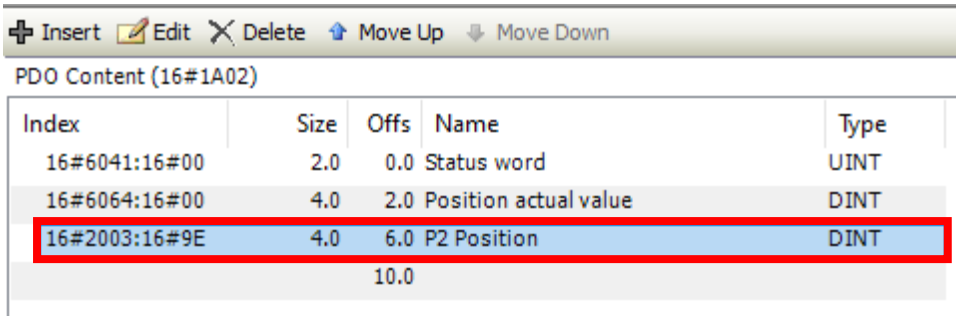
   Using the “Name” input field, enter a suitable name for the P2 Normalised Position value, for “Index” enter “2003”, for “SubIndex” enter “9E”. “Bit length” should be entered as “32” and “Data type” should be set to “DINT”.

   This creates a PDO mapping to Pr03.158 P2 Normalised Position.

   Once the above values have been entered, click “OK”.

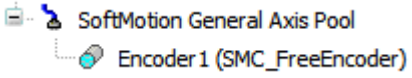


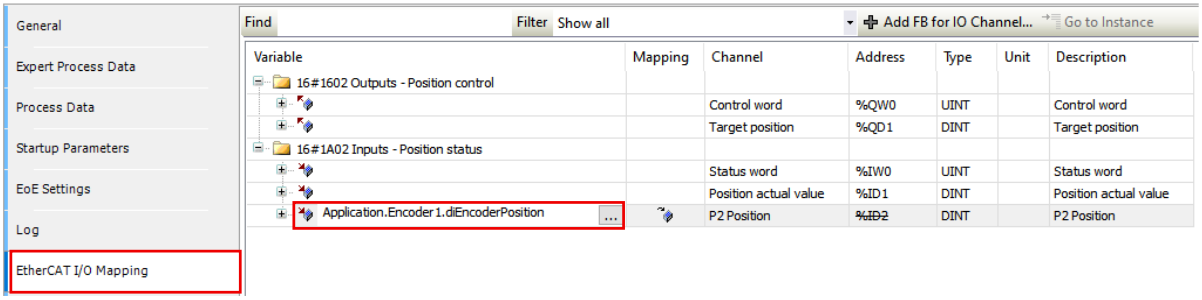
6. The new PDO mapping will appear in the PDO mapping list.



7. Open the “EtherCAT I/O Mapping” tab. If the transmit PDO’s have been organised into a folder, expand the folder and enter the following value into the “Variable” input box of the newly created PDO mapping.

   “Application.SMC_FreeEncoder.diEncoderPosition” where “Encoder1” represents the name of the FreeEncoder axis as it appears in the project tree.



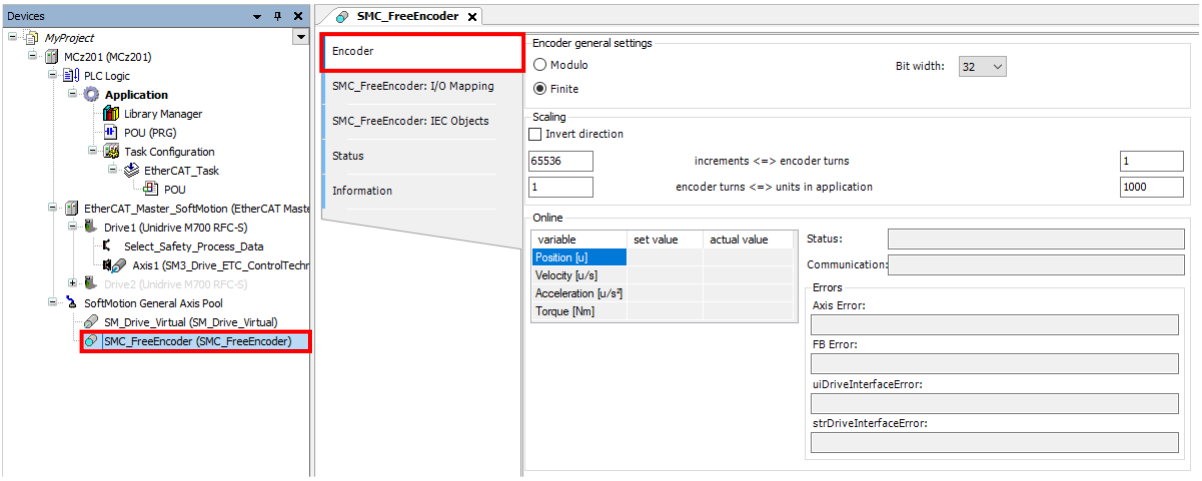


8. Ensure that the following drive parameters that relate to the P2 encoder interface have been configured:

   - Pr03.133 Rotary Turns Bits.
   - Pr03.134 Rotary Lines Per Revolution.
   - Pr03.138 P2 Device Type.

9. From the project tree, double click the “SMC_FreeEncoder” device. The “Encoder” tab allows the user to configure the axis type (modulo or finite), bit width and scaling.

   Once the application has been downloaded, this tab provides online monitoring of the encoder position, velocity and acceleration values.



The tables below provide a more detail description of the data input fields available on the “Encoder” tab.

Use the “Encoder” tab to specify how the encoder should behave.

General Encoder Settings

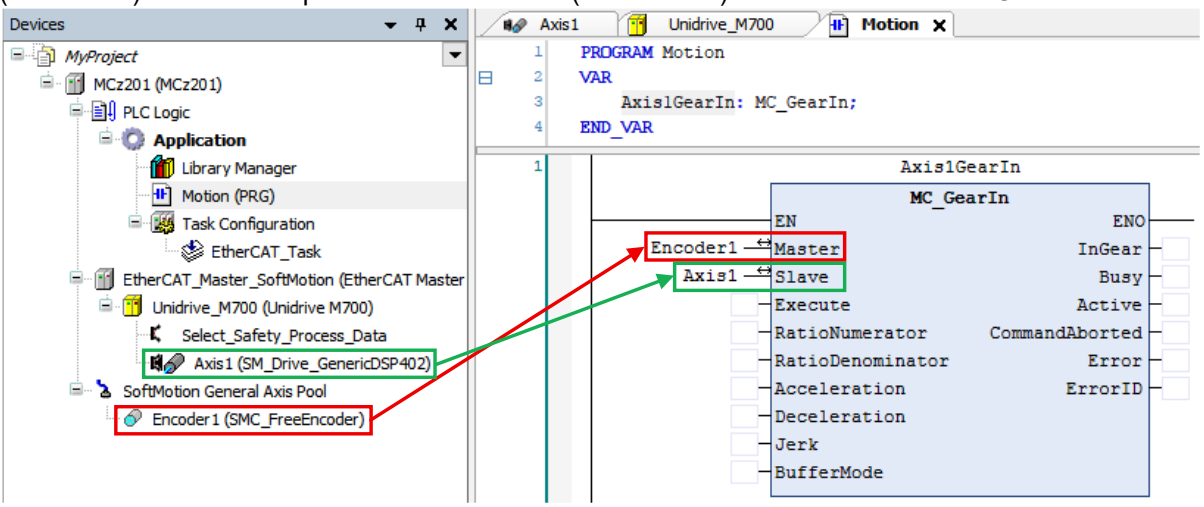
Modulo	The axis turns endlessly where the traversing range is not limited, for example, a conveyor belt. When selected, the “Modulo settings” window is displayed: The “Modulo value” is an Input field for the modulo value, the value is saved in the fPositionPeriod parameter of the AXIS_REF_SM3 function block. The encoder position displayed will wrap over when the “Modulo value” has been reached. **NOTE:** If you select the Modulo encoder type, then the product fPositionPeriod * dwRatioTechUnitsDenom has to be an integer.
Finite	The axis is limited and no modulo rollover is required.
Bit width	List box for an appropriate bit width. For DS402 using EtherCAT this must be set to 32.

Scaling

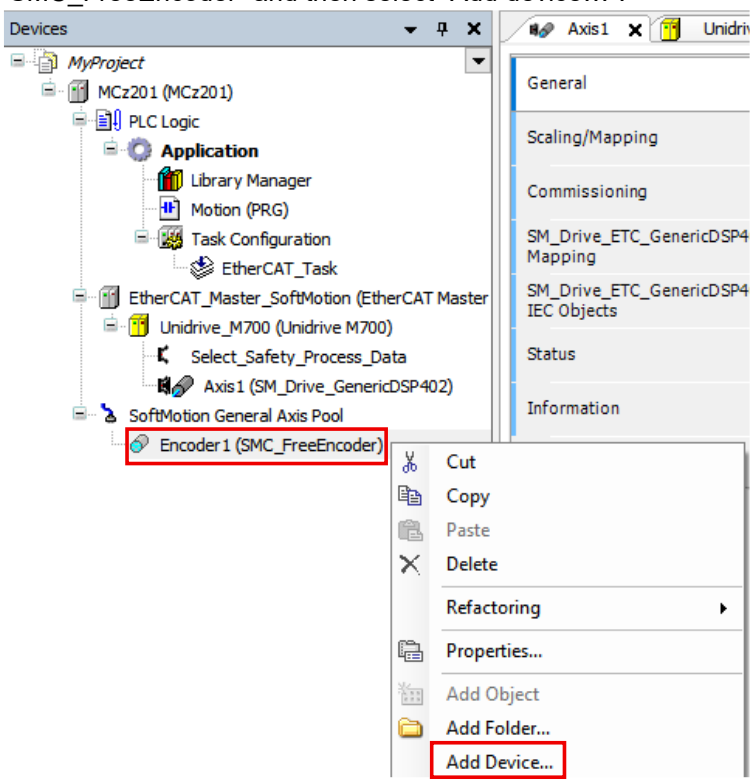
Invert direction	The encoder axis receives the specified values with the sign inverted and therefore rotates in the opposite direction.
increments < = > encoder turns	Number of “increments” which correspond to the number of complete “encoder turns”
encoder turns <=> units in application	Number of “encoder turns” which correspond to a given number of “units in application”

10. The position of the “SMC_FreeEncoder” axis can now be consumed by a CT SoftMotion Interface Library function block such as MC_GearIn.

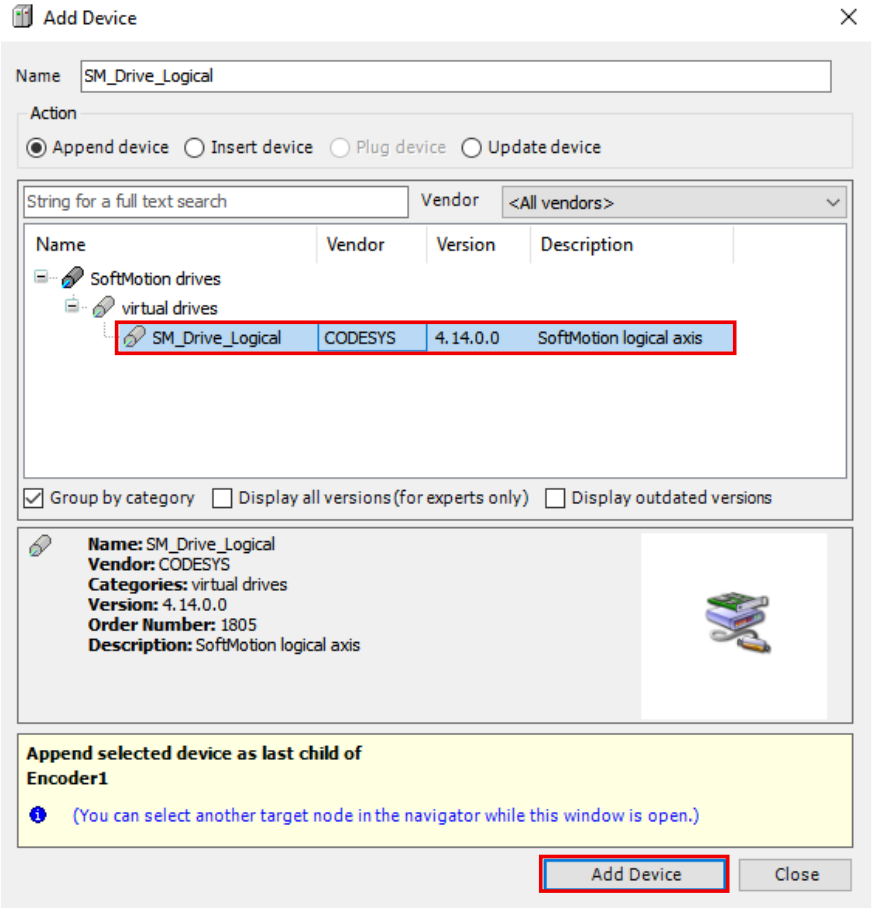
    An example of how this would look in a program can be seen below. In this instance, “Axis1” (Slave Axis) will track the position of “Encoder1” (Master Axis) when “Execute” = TRUE.



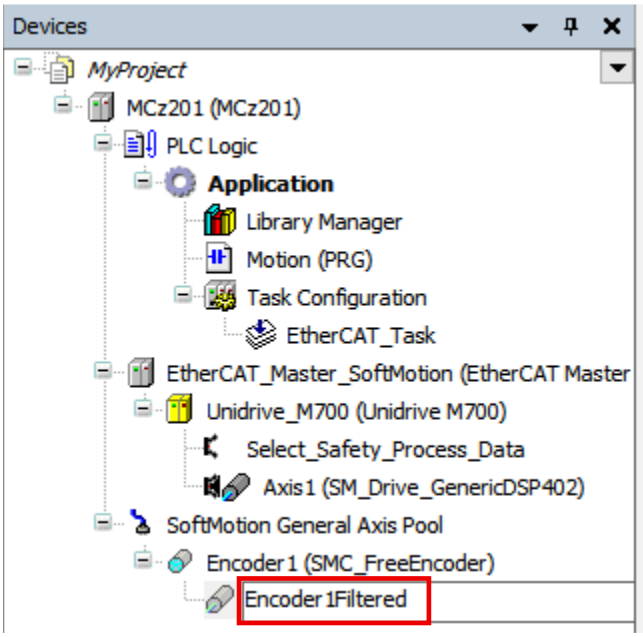
11. It may be beneficial to use a position filter to filter out noise from the raw encoder position, especially when an encoder with a low resolution is used. To use the inbuilt Softmotion filters required a logical axis to be added to the “SMC_FreeEncoder” axis. Right click on the “SMC_FreeEncoder” and then select “Add device…”.



12. When the “Add Device” dialog opens, select “Softmotion drives” > “virtual drives” > “SM_Drive_Logical” and then click “Add Device”.

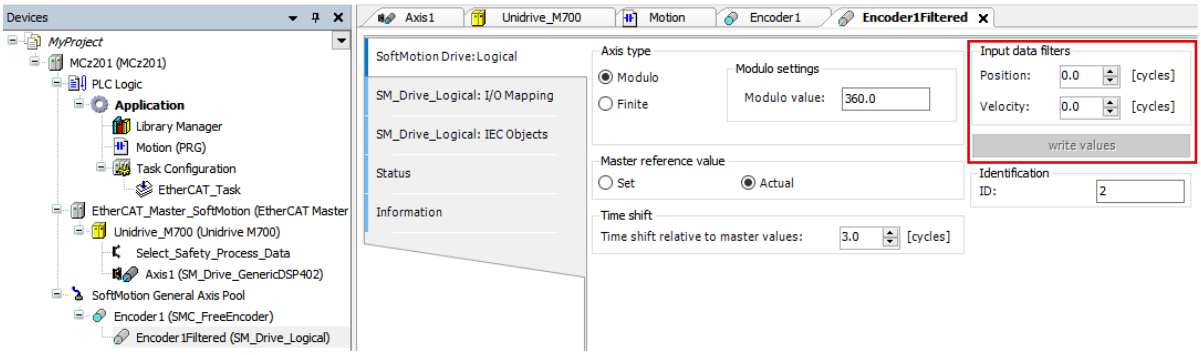


13. Rename the SM_DriveLogical e.g. Encoder1Filtered:



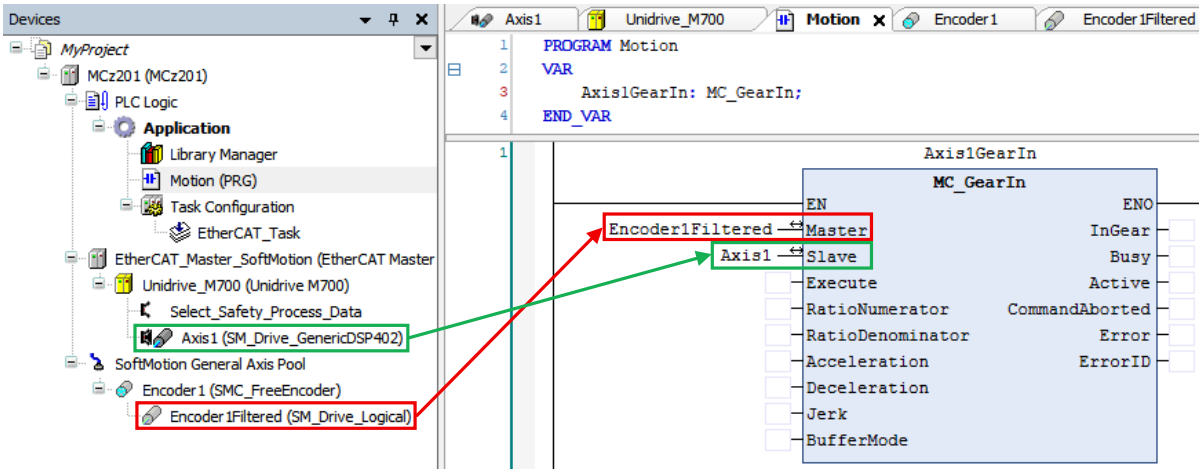
Click “Yes” and “OK” to the subsequent auto-refactoring dialogs.

14. Double click on Encoder1Filtered (SM_Drive_Logical). When the properties for Encoder1Filtered opens, the encoder value may be filtered using the “Input Data Filters” section:



The filter value may be tuned while the axis is running by clicking the “write values” button.

15. An example of how this would look in a program can be seen below. In this instance, “Axis1” (Slave Axis) will track the position of “Encoder1Filtered” (Master Axis) when “Execute” = TRUE.



3.2.3 How to define the drive action on loss of cyclic communications

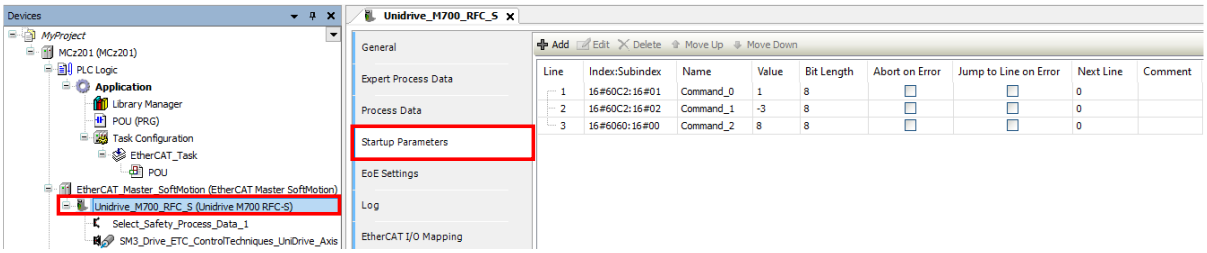
The desired drive action on loss of cyclic communications can be configured to be automatically written to the drive as the PLC starts up and EtherCAT communications are first established.

This can be achieved via the by writing to the following CANopen over EtherCAT (CoE) objects:

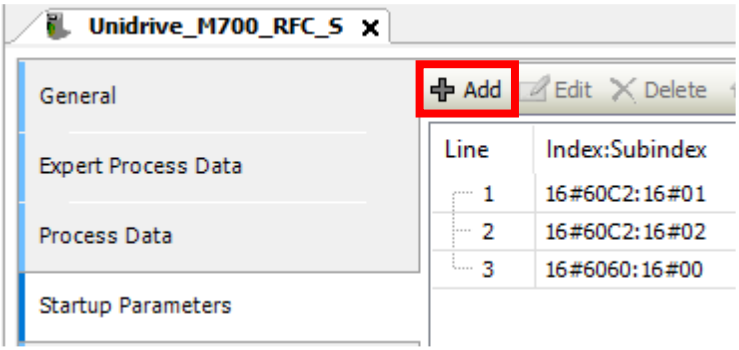
- Index: 0x3005, Sub Index: 2 – Cyclic Data Loss Action.

The following steps will describe how to cause the configuration of these CoE objects to happen when the PLC / IPC / Controller is first powered on.

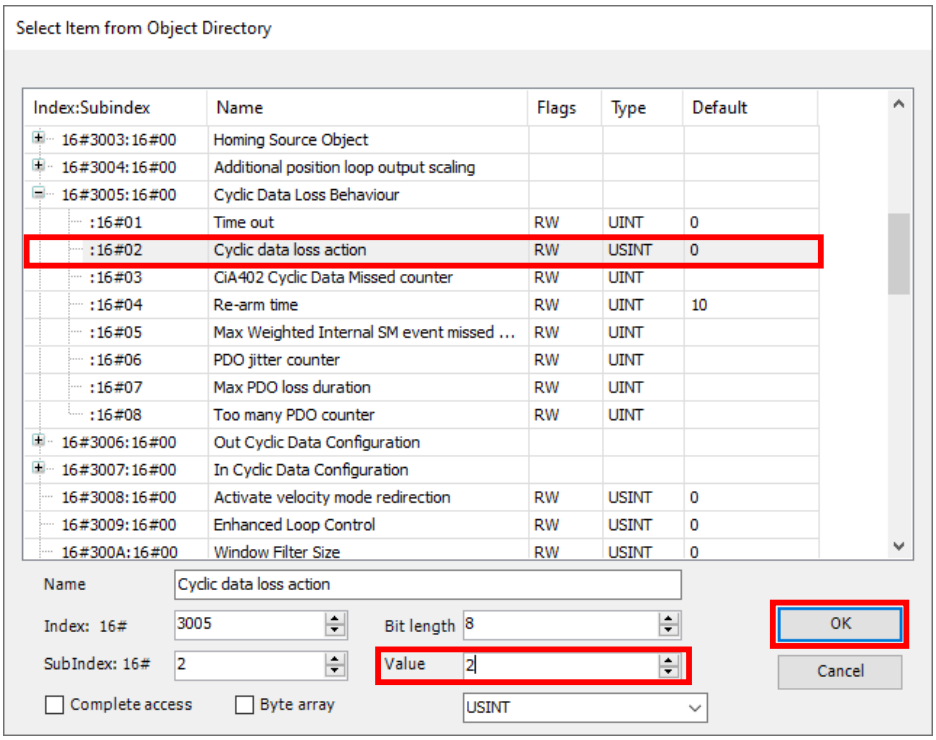
1. From the project tree, double click on “Unidrive_M700_RFC_S” and select the “Startup Parameters” tab.



2. From the “Startup Parameters” view, select the “Add” button.



3. The “Select Item from Object Directory” dialog will be displayed. From the list of available mappings, select “16#3005:16#02 Cyclic Data Loss Action”, enter “2” into the value field and click “OK”.



HINT: The value of 2 means that Raise the cyclic data loss trip and disable the drive as per the table below. Other behaviours can be selected from this table.

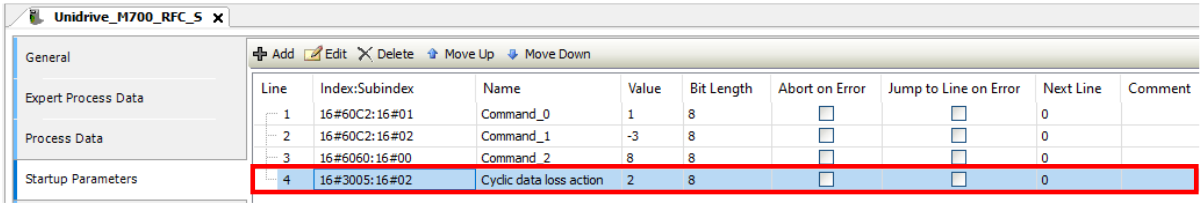
This list is an extract taken from the SI-EtherCAT User Guide which describes sub index 2 of object **0x3005**, used for setting the drive behaviour on loss of cyclic comms.

Sub-index 2: Cyclic Data Loss Action			
Access: RW	Range: 0 to 4	Size: 1 byte	Unit: N/A
Default: 0		Type: Unsigned integer / USINT	
PDO Mappable: No		Update Rate: New value used transition from SafeOp to Op	
Description:	Cyclic Data Loss Action. The value will select an action as follows: 0: Raise a warning and initiate a motor stop according to the Fault reaction option code. 1: Raise a warning and initiate a motor stop according to the Fault reaction option code. When the motor has reached zero speed raise the cyclic data loss trip. 2: Raise the cyclic data loss trip and disabled the drive invertor. 3: Only raise a warning of PDO loss. [Note: For CSP mode a PDO loss will mean SI-EtherCAT will hold the motor at the current position (if extrapolation is disabled), this could be incorrectly interpreted, by the user, as a motor stop, if it was previously moving] 4: Ignore the cyclic loss completely (i.e. disabled cyclic loss detection). [Note: For CSP mode a PDO loss will mean SI-EtherCAT will hold the motor at the current position (if extrapolation is disabled), this could be incorrectly interpreted, by the user, as a motor stop, if it was previously moving]		

If necessary, other objects such as “0x605E Fault reaction option code” can be configured in the same way so that their values are automatically set on startup of the PLC / IPC / Controller.

4. The “Cyclic data loss action” mapping will now be present in the “Startup Parameters” list which means it’s value will be written to the specified CoE object on startup of the PLC / IPC / Controller.

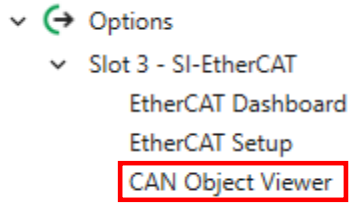
A drive configured in this way will trip on “Cyclic data loss” and enter the “Inhibit” state when EtherCAT communications are lost e.g. when the EtherCAT cable is disconnected.



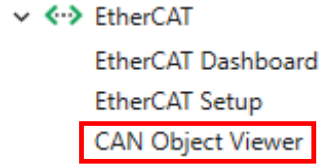
3.3 How to view CAN objects

Connect has a powerful diagnostic aid to help the user view the live value of CANopen objects called the CAN Object Viewer. This helps the user to diagnose setup and operational issues with the CiA402 system layer upon which EtherCAT and Softmotion axes run. If the EtherCAT network isn't starting, or a drive axis isn't running when it should, this tool can help the user understand what is going on.

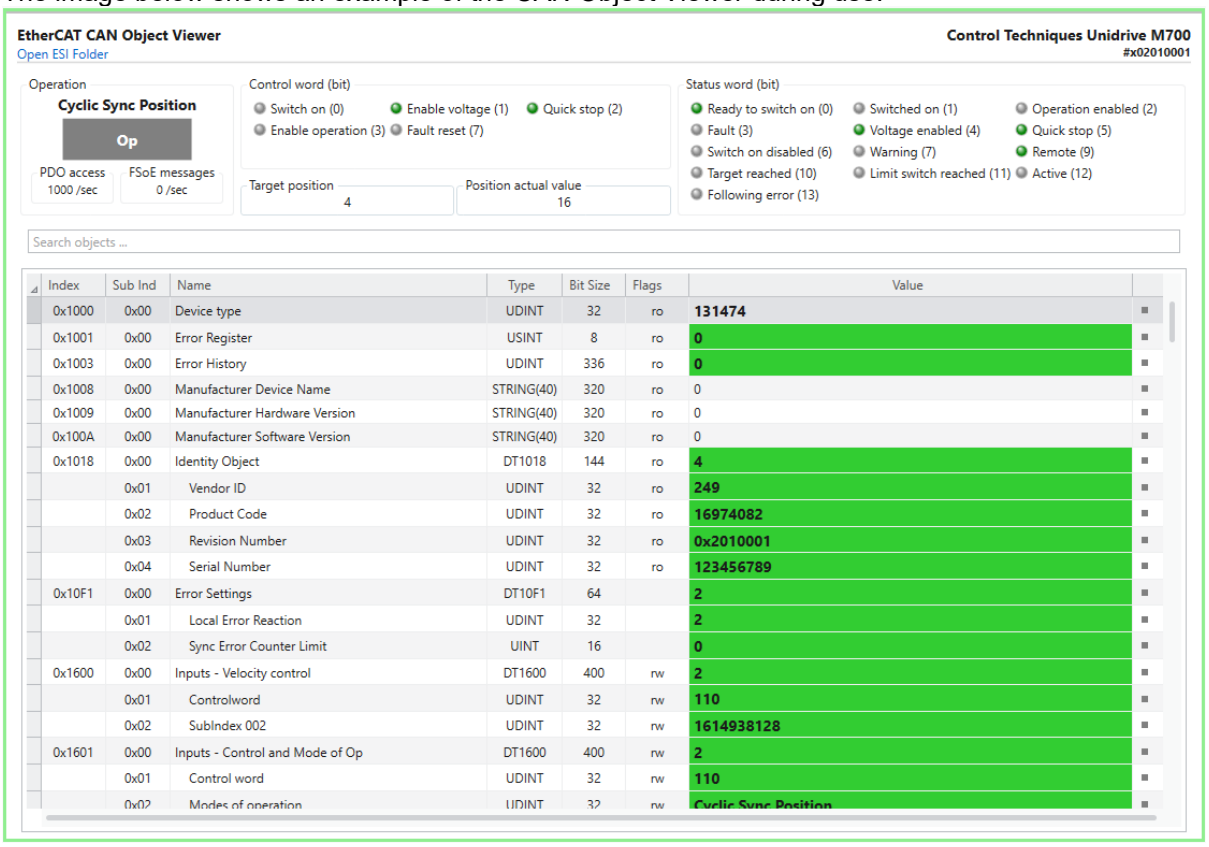
The tool is found in the Drive explorer tree under "Options" > "Slot[n]" > "CAN Object Viewer" where [n] is the drive slot in which the SI-EtherCAT interface is located.



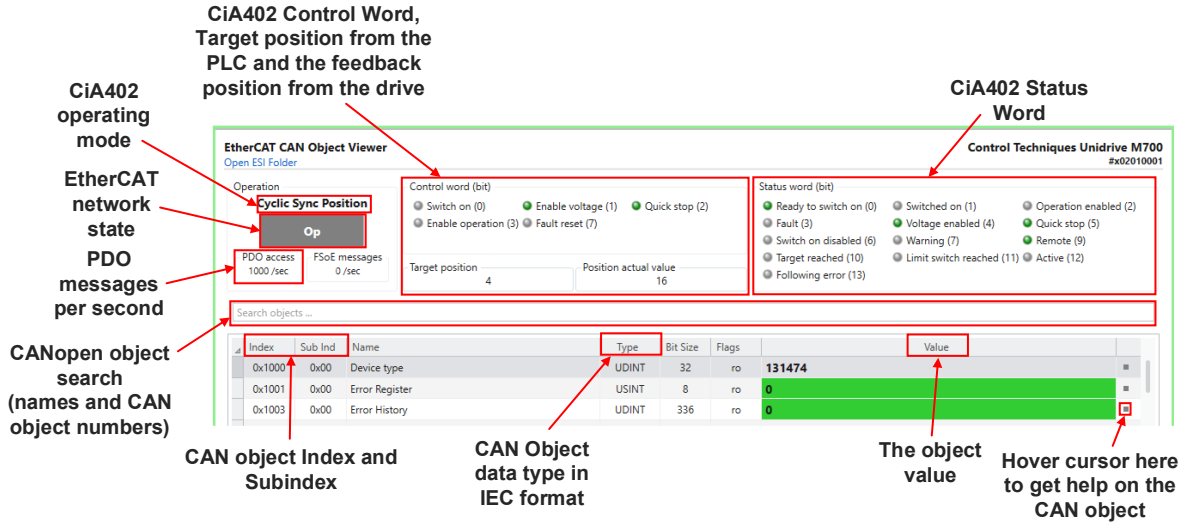
For Dedicated EtherCAT Drives Such as Digitax HD M753 it is found in the drive explorer tree under "EtherCAT" > "CAN Object Viewer".



The image below shows an example of the CAN Object Viewer during use:

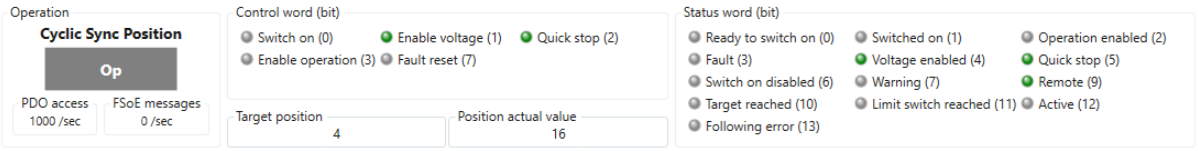


The following image highlights some of the useful features and information available within this tool.



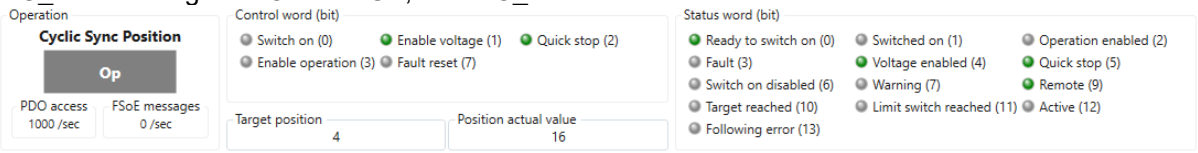
3.3.1 MC_Power Disabled and Drive Disabled

This screenshot acts a guide to understand the expected CiA402 Status and Control word for a Softmotion axis with a disabled drive (STO = 0V), and MC_Power.Enable = TRUE, and MC_Power.bRegulatorOn = TRUE, and MC_Power.bDriveStart.



3.3.2 MC_Power Disabled and Drive Enabled

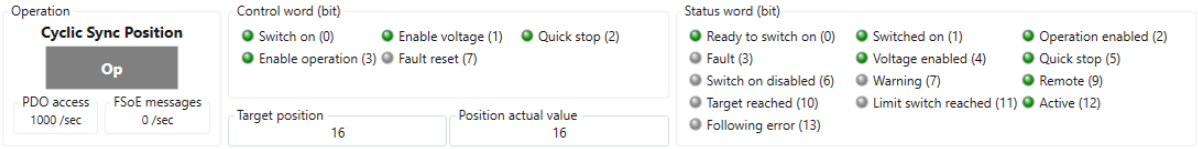
This screenshot acts a guide to understand the expected CiA402 Status and Control word for a Softmotion axis with a disabled drive (STO = 24V), and MC_Power.Enable = TRUE, and MC_Power.bRegulatorOn = TRUE, and MC_Power.bDriveStart.



HINT: “Ready to switch on (0)” is now active.

3.3.3 MC_Power Enabled and Drive Enabled

This screenshot acts a guide to understand the expected CiA402 Status and Control word for a Softmotion axis with a disabled drive (STO = 24V), and MC_Power.Enable = TRUE, and MC_Power.bRegulatorOn = TRUE, and MC_Power.bDriveStart.



HINT: “Switch on (0)”, “Enable operation (3)”, “Switched on (1)”, “Operation enabled (2)”, and “Active (12)” are now active.

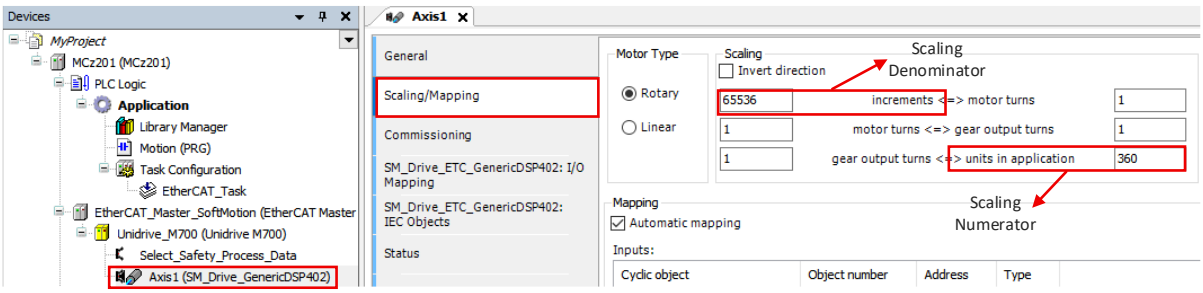
3.4 How to setup axis unit scaling and resolution

Unit scaling in most motion control systems, including a CODESYS Softmotion axis, requires the user to specify the number of feedback counts there are for a given distance, where feedback in counts is converted into Application units such as degrees or mm. Normally this is represented as a numerator and denominator where:

- The numerator is the number of distance units (“units in application”).
- The denominator is the number of position feedback counts that equals the numerator value. (“increments)

There are number of different ways to set up the position feedback scaling but the easiest is based around rotating the motor by 1 revolution and indicating how many feedback counts (as seen by the drive) there will be per revolution and how many Application units will be moved in 1 revolution.

By default, the number of feedback counts per revolution is set to 65536. If the applications units are degrees and there is no output gearbox, the unit scaling ratio would be set to a numerator of 360, and a denominator of 65536. In CODESYS it looks like this:

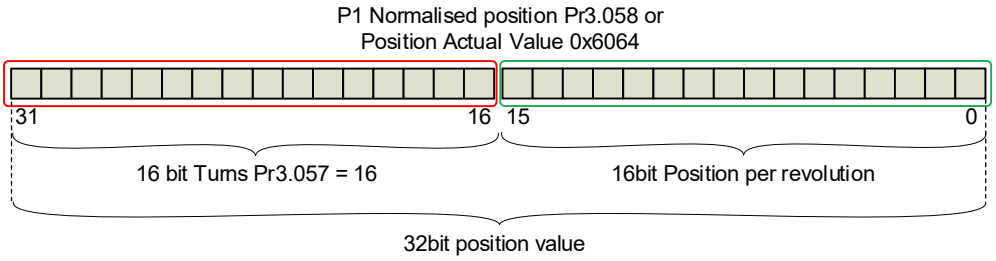


If we add a 4:1 reduction gearbox, the number of rotations at the output of the gearbox is reduced by the gearbox ratio so in this example the scaling numerator is 360 * 1 / 4 = 90.

The encoder resolution used by a Softmotion axis is set by drives feedback resolution as defined by Pr3.057 Normalisation Turns, where the number of counts per revolution can be calculated by using the following formula:

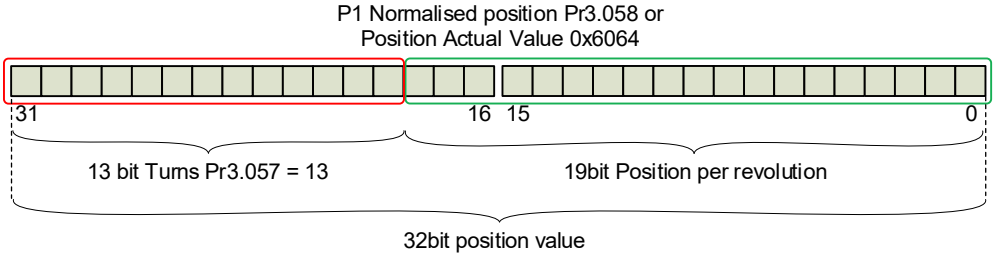
$$\text{Encoder Counts Per Revolution} = 2^{(32\text{-}\mathbf{NormalisationTurns\ Pr3.057})}$$

By default, Pr3.057 P1 Normalisation Turns = 16 so the number of feedback counts per revolution is $2^{(32-16)} = 65536$.



Provided the feedback encoder has a resolution greater than 16bit, the resolution of the axis can be increased by modifying P1 Normalisation Turns Pr3.057, where a reduction in the number of turns bits adds more feedback counts per revolution.

For example, if a 19bit per revolution encoder is used, the number of turns bits is reduced to 13 to give 19bits of turns information:



The following table can be used to help with determining the correct turns bits value to use based on the encoders number of counts per revolution.

Pr3.057 Turns bits	Counts per revolution	Bits per revolution	Turns bits as hex
16	65536	16	0x10
15	131072	17	0x0F
14	262144	18	0x0E
13	524288	19	0x0D
12	1048576	20	0x0C
11	2097152	21	0x0B
10	4194304	22	0x0A
9	8388608	23	0x09
8	16777216	24	0x08
7	33554432	25	0x07
6	67108864	26	0x06
5	134217728	27	0x05
4	268435456	28	0x04

HINT: The total resolution is always 32bit, where the sum of the turn's bits and bits per revolution always adds up to 32bits.

The two main ways to setup Pr3.057 are by configuring the value in Connect and saving it, or by setting Pr3.057 in the startup list in CODESYS directly, or by exporting the startup list from Connect once the system / application is fully commissioned.

3.4.1 How to set the resolution using the startup list in CODESYS

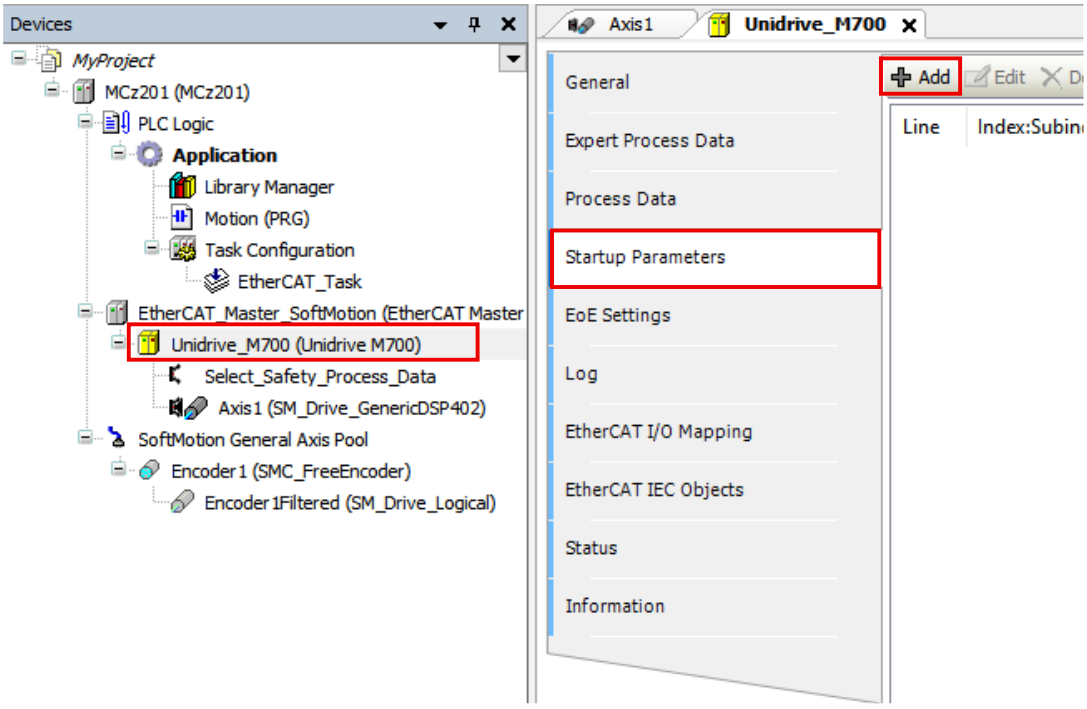
The resolution is configured by default in Pr3.057. See the previous table for the number that the turns bits should be set, (Hex value), to for a particular number of counts per revolution.

For setup in the startup list the following data is needed:

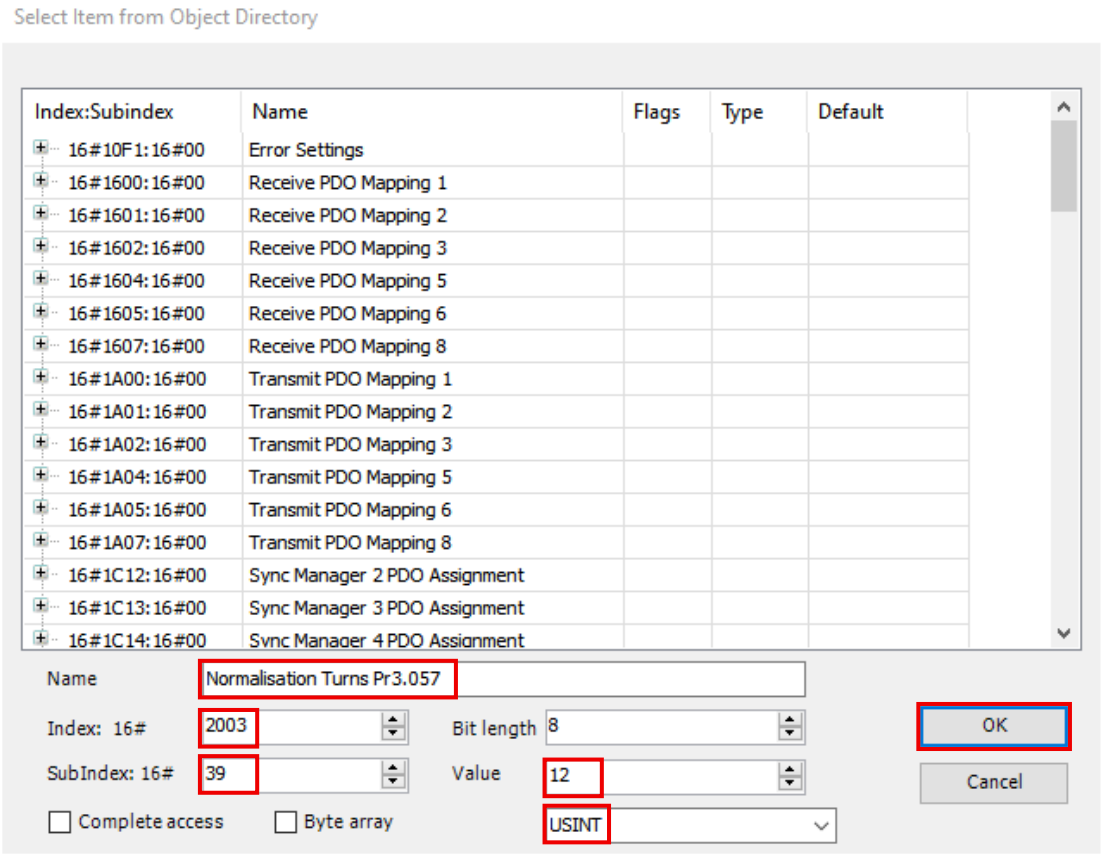
Parameter	Index (hex)	Subindex (dec)	Comment
Pr3.057	2003	57	Turns bits

Use the following steps to configure a startup list entry for a drive parameter directly:

1. Double click on the EtherCAT Slave drive in the devices tree, e.g. Unidrive_M700, and when the properties are shown select the “Startup Parameters tab”, and then click the “Add” button.



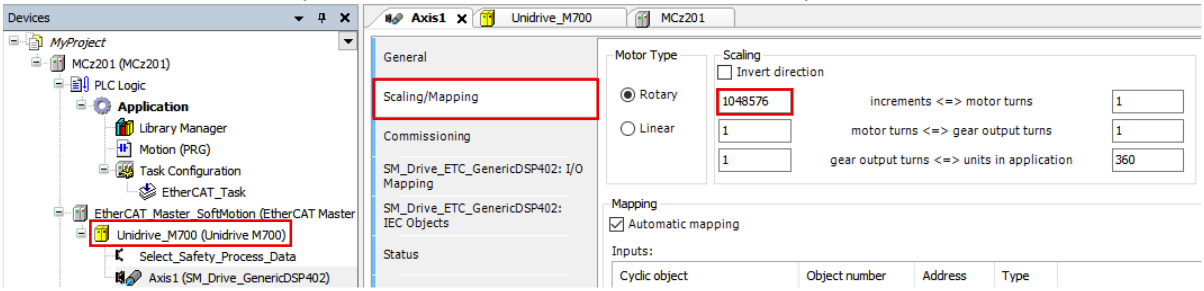
2. When the “Select Item from Object Dictionary” dialog opens, populate it as shown below:



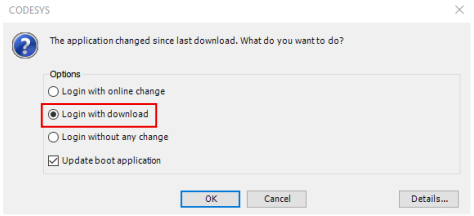
3. The new entry in the Startup list looks like this:

Line	Index:Subindex	Name	Value	Bit Length	Abort on Error	Jump to Line on Error	Next Line	Comment
1	16#2003:16#39	Normalisation Turns Pr3.057	12	8	☐	☐	0	

4. Increase the axis scaling resolution to match the updated feedback resolution. 12 turns bits results in 20bit resolution per revolution or 1048576 encoder counts per revolution.



5. The project is ready to build and download. Press Ctrl+F8 or ,  . When the application dialog starts select “Login with download”.



When the logging in button looks like  click the button press F5 or press the start button  .

3.4.2 How to scale between Softmotion axis speed and rpm

To convert an Softmotion axis speed in units/s into rpm use the following formula:

Drive speed in rpm = Speed(Units per s) * 60 / Scale Numerator

E.g. The axis is running at 1000units/s with a scaling numerator of 1000 per rev. The resulting speed in rpm observed in Pr3.002 is 1000 * 60 / 1000 = 60rpm

To convert the speed seen at the drive, in rpm, into Softmotion axis speed in units/s use the following formula:

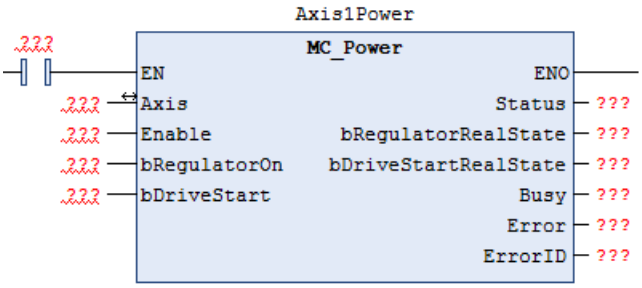
Softmotion axis speed = Speed(rpm) * Scale Numerator / 60

E.g. The axis is running at 1000rpm with a scaling numerator of 123 mm per rev. The resulting speed in rpm observed in the Softmotion Axis.fActVelocity parameter is 1000 * 123 / 60 = 2050rpm

3.5 How to remove “???” from LD diagram inputs and outputs

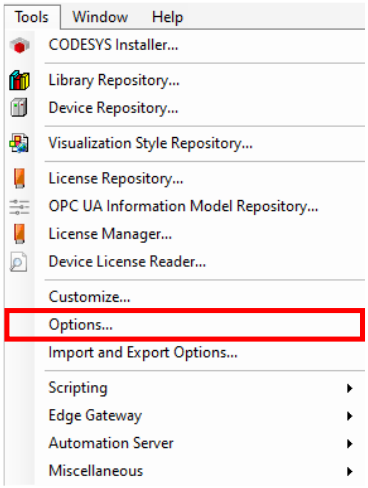
LD diagram language can be a nice way to explore the functionality of the Softmotion PLCopen function blocks. Each input and output can be seen side by side, any enumerated values selected, and all output values may be seen.

However, the default environment behaviour when using the LD language can be cumbersome when inserting function blocks to a ladder rung. All of the POU Inputs and Outputs have “???” next to them which must be deleted or overwritten before the program can be compiled which is time consuming. In the example below an instance of MC_Power has been added to the LD rung where all the inputs and outputs have “???” assigned.

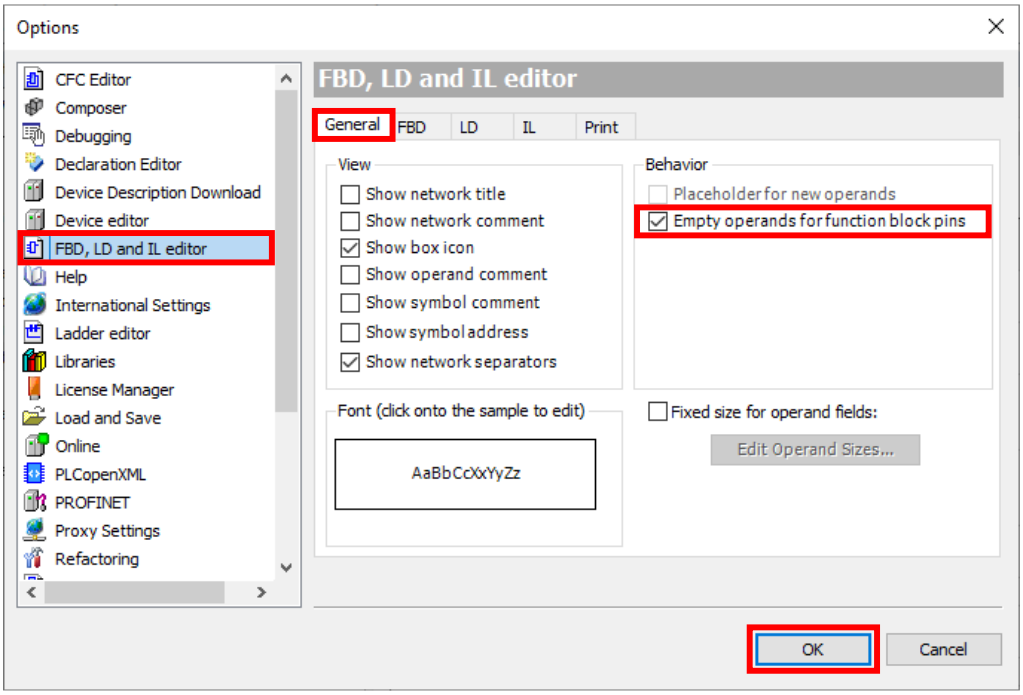


The “???” at the POU’s I/O is optional. The following steps show how to turn this off:

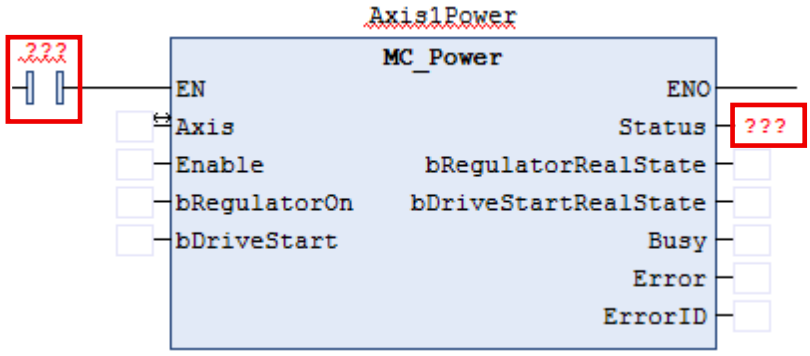
1. Go to “Tools” > ”Options”.



2. Go to “FB LD and IL editor”. The “Options dialog box will be displayed. Select “FBD, LD, and IL editor” from the category list, ensure the “General” tab is selected, tick the “Empty operands for function block pins” tick box and select “OK” to save these changes.



3. After checking the “Empty operands for function block pins” box, adding an MC_Power or any other function / function block looks like this:



Only the first output will have “???” shown so only this and the input contact need to be deleted, which is much quicker.

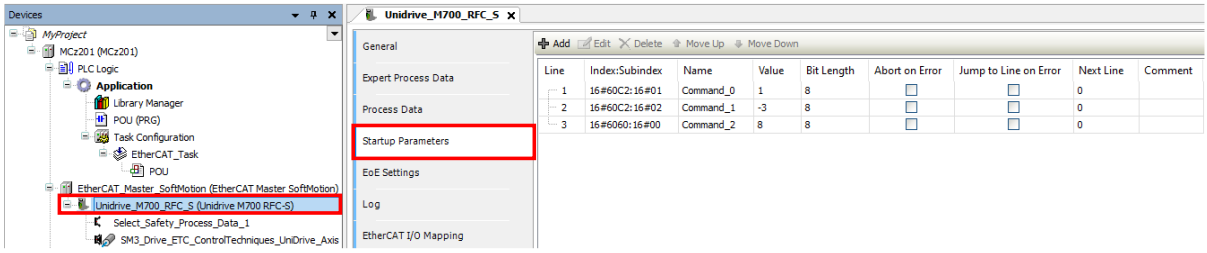
3.6 How to Import a startup parameter list

It is possible to import Startup Parameters from an .xml file such as one that has been generated at the end of the drive commissioning steps in Connect, that contain all of the changes (from default) that have been made to the drive so that every time the drive is powered up, its settings are restored.

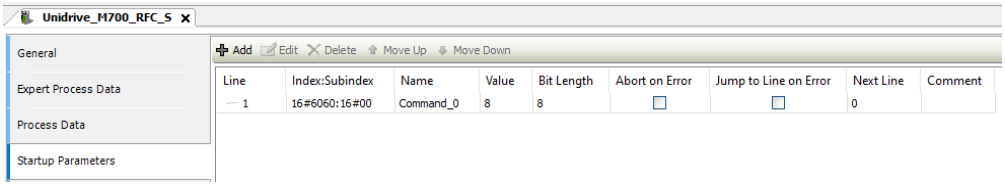
This can be helpful if a drive has to be replaced, since the previous setup is restored including things like motor map, encoder type, tuning values etc.

The following steps will describe how to import the Startup List into CODESYS.

1. From the project tree, double click on “Unidrive_M700_RFC_S” and select the “Startup Parameters” tab.



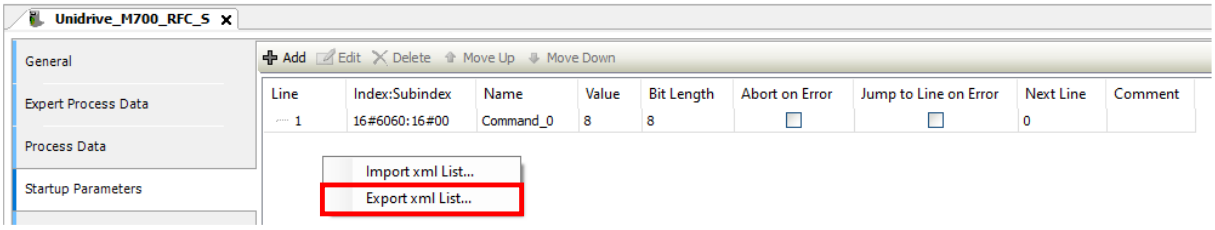
2. If you would like to preserve existing startup list entries, the existing startup list must be exported and merged with the startup list to be imported (generated by Connect).



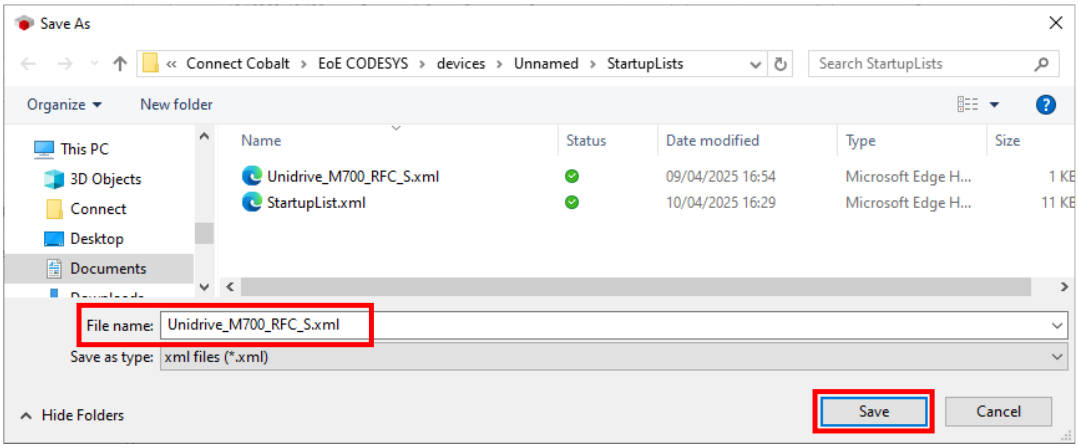
If you do not wish to preserve existing Startup List entries, skip to step 3.

To preserve existing Startup List parameters, follow these steps to export and merge startup lists:

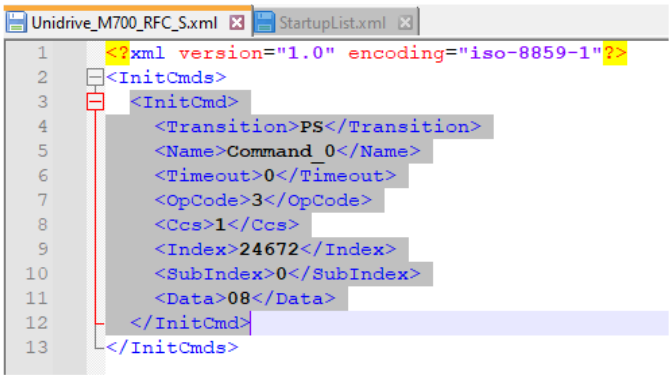
a. From the “Startup Parameters” view, right click in the white space and select “Export xml List…”.



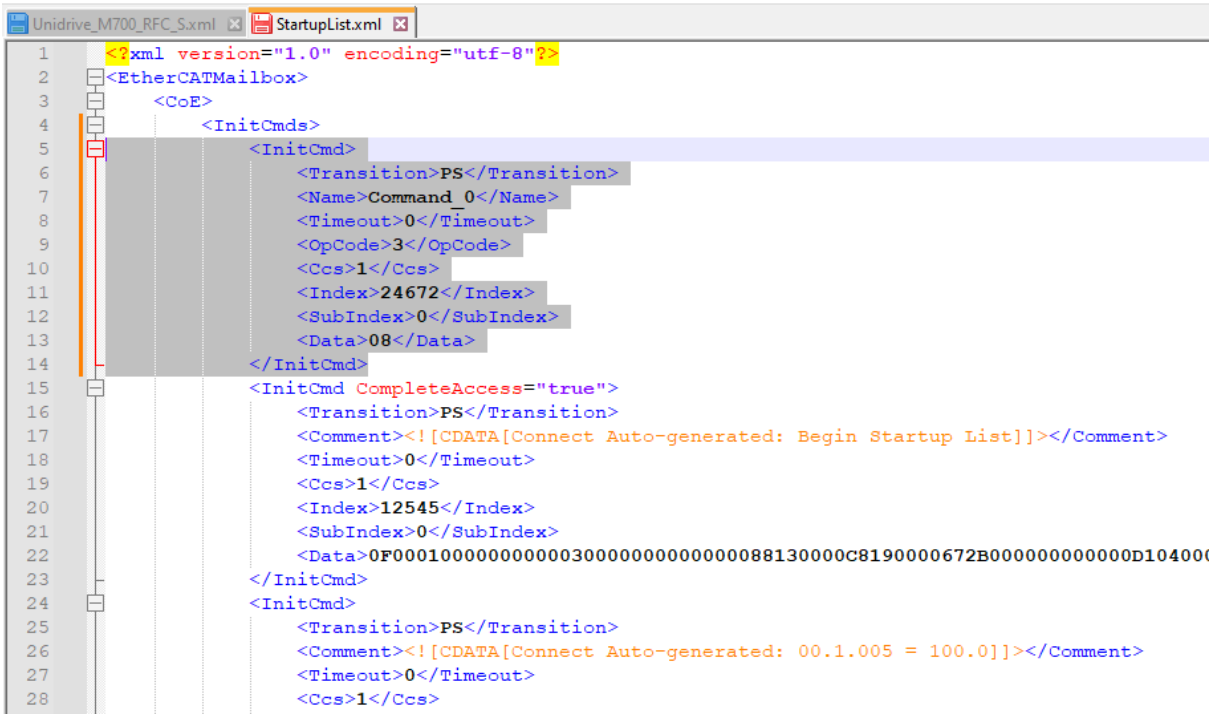
b. Provide a file name and click “Save”.



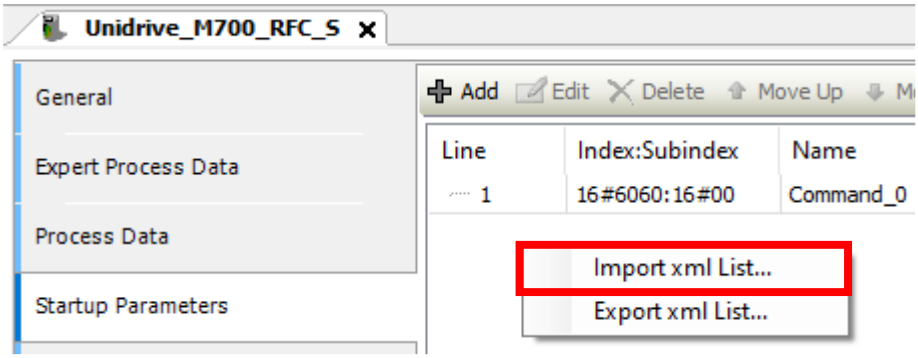
c. Using a text editor such as Notepad or Notepad++, open the .xml list that has been exported by CODESYS and copy the contents between the ”<InitCmd>” and “</InitCmd>” for each startup parameter to be preserved.



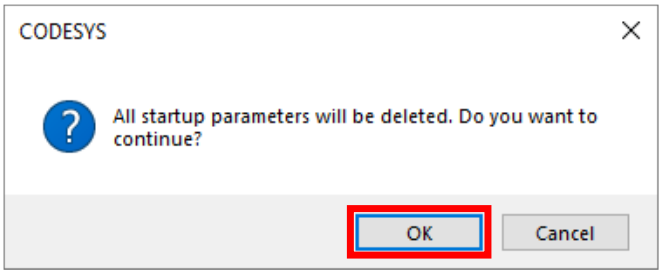
d. Using a text editor such as Notepad or Notepad++, open the startup list that has been exported by Connect and paste the copied parameter entries from the CODESYS startup list below the <InitCmds> line. Once complete, save any changes to the startup list and close the text editor.



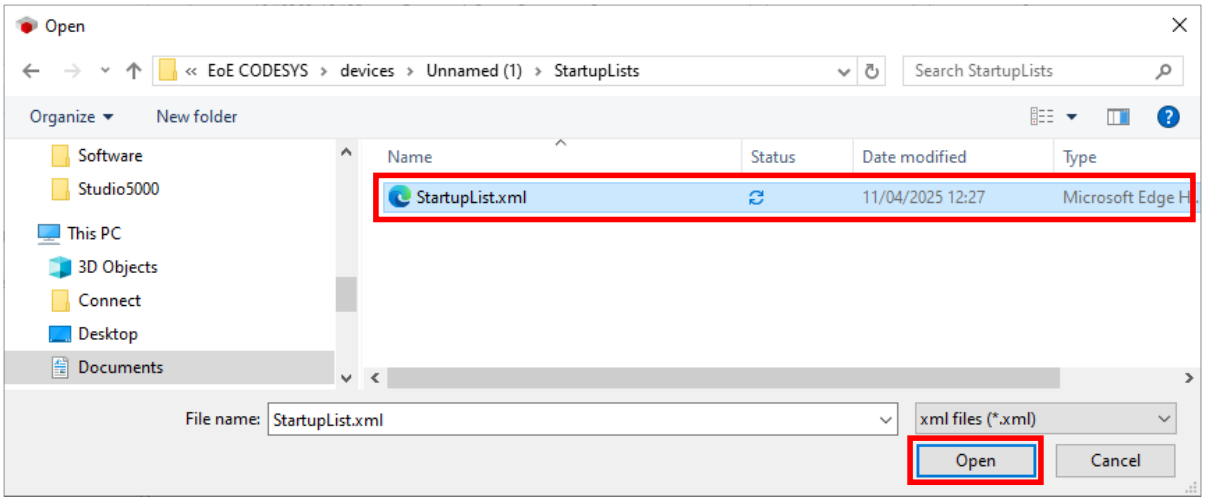
3. From the “Startup Parameters” view, right click in the white space and select “Import xml List…”.



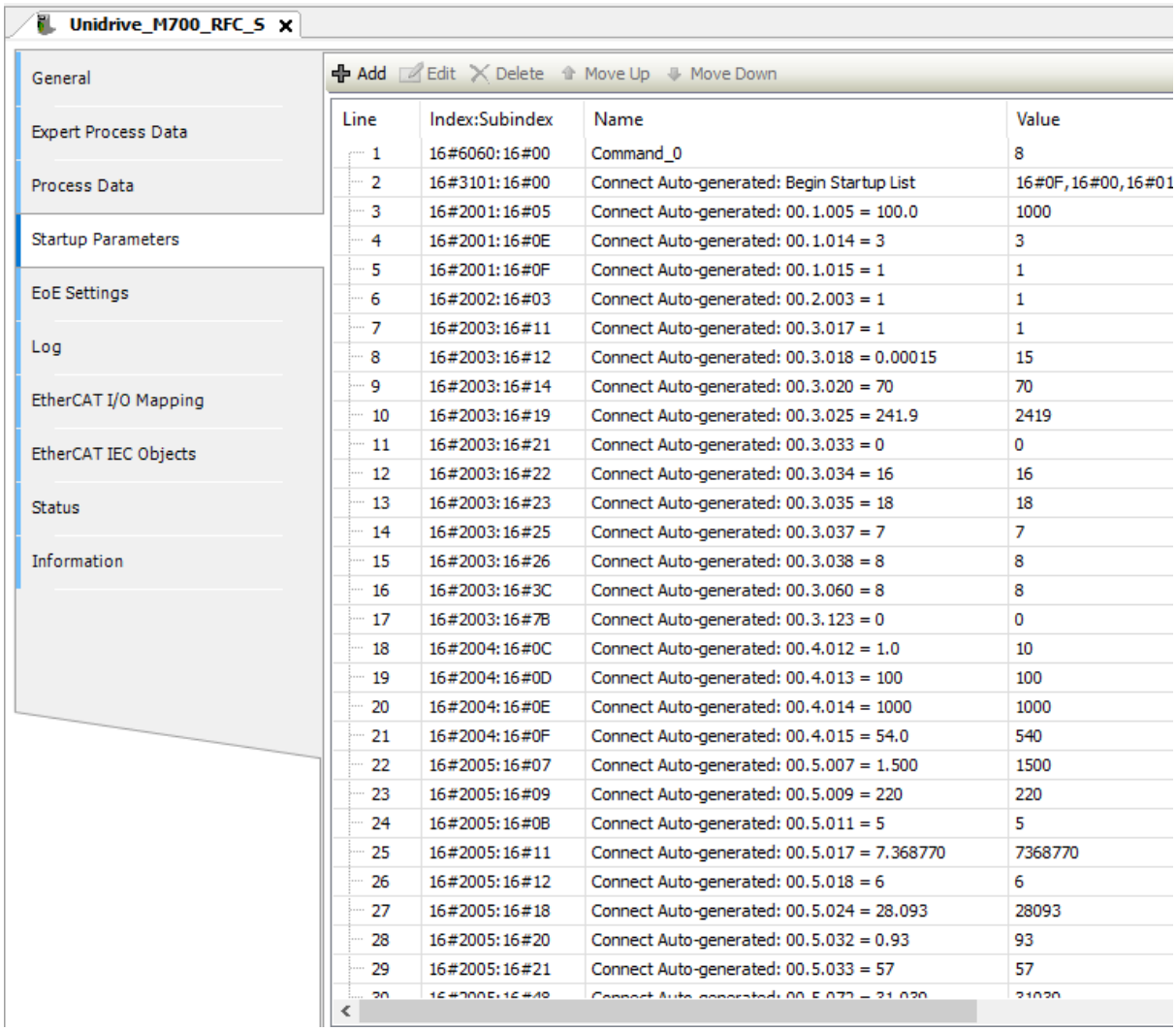
HINT: If there are already startup parameters present the following dialog box will be displayed, click “OK” to continue.



4. Browse for the Startup List and click “Open”.



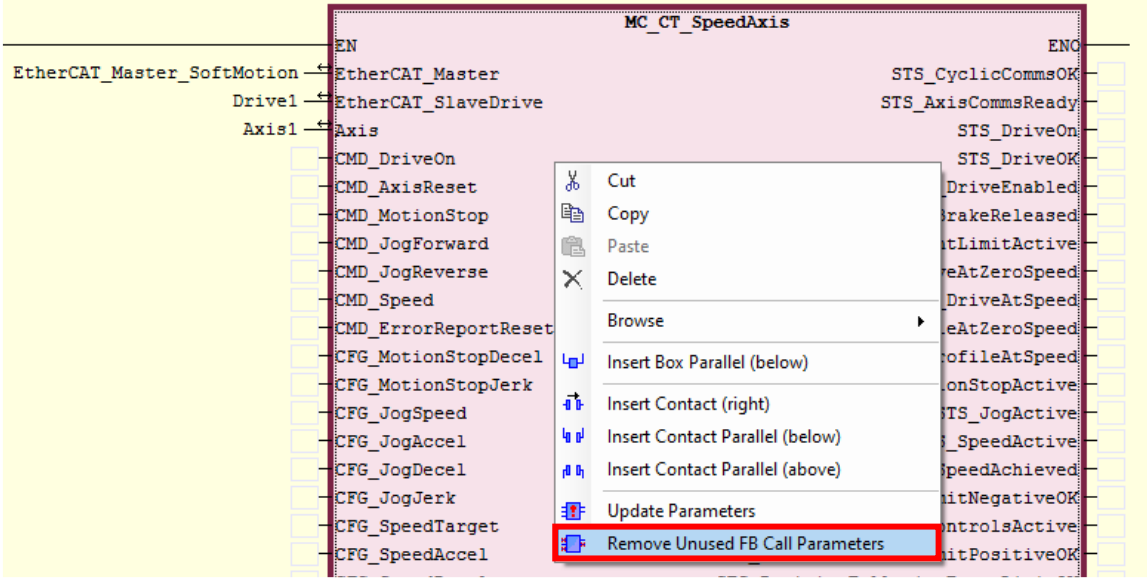
5. The startup list will now be imported.



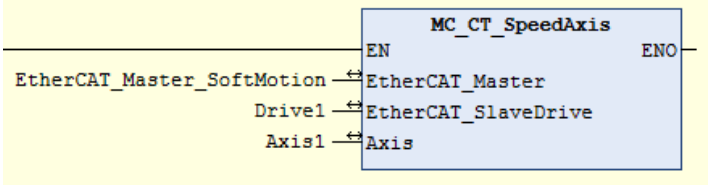
Line	Index:Subindex	Name	Value
1	16#6060:16#00	Command_0	8
2	16#3101:16#00	Connect Auto-generated: Begin Startup List	16#0F,16#00,16#01
3	16#2001:16#05	Connect Auto-generated: 00.1.005 = 100.0	1000
4	16#2001:16#0E	Connect Auto-generated: 00.1.014 = 3	3
5	16#2001:16#0F	Connect Auto-generated: 00.1.015 = 1	1
6	16#2002:16#03	Connect Auto-generated: 00.2.003 = 1	1
7	16#2003:16#11	Connect Auto-generated: 00.3.017 = 1	1
8	16#2003:16#12	Connect Auto-generated: 00.3.018 = 0.00015	15
9	16#2003:16#14	Connect Auto-generated: 00.3.020 = 70	70
10	16#2003:16#19	Connect Auto-generated: 00.3.025 = 241.9	2419
11	16#2003:16#21	Connect Auto-generated: 00.3.033 = 0	0
12	16#2003:16#22	Connect Auto-generated: 00.3.034 = 16	16
13	16#2003:16#23	Connect Auto-generated: 00.3.035 = 18	18
14	16#2003:16#25	Connect Auto-generated: 00.3.037 = 7	7
15	16#2003:16#26	Connect Auto-generated: 00.3.038 = 8	8
16	16#2003:16#3C	Connect Auto-generated: 00.3.060 = 8	8
17	16#2003:16#7B	Connect Auto-generated: 00.3.123 = 0	0
18	16#2004:16#0C	Connect Auto-generated: 00.4.012 = 1.0	10
19	16#2004:16#0D	Connect Auto-generated: 00.4.013 = 100	100
20	16#2004:16#0E	Connect Auto-generated: 00.4.014 = 1000	1000
21	16#2004:16#0F	Connect Auto-generated: 00.4.015 = 54.0	540
22	16#2005:16#07	Connect Auto-generated: 00.5.007 = 1.500	1500
23	16#2005:16#09	Connect Auto-generated: 00.5.009 = 220	220
24	16#2005:16#0B	Connect Auto-generated: 00.5.011 = 5	5
25	16#2005:16#11	Connect Auto-generated: 00.5.017 = 7.368770	7368770
26	16#2005:16#12	Connect Auto-generated: 00.5.018 = 6	6
27	16#2005:16#18	Connect Auto-generated: 00.5.024 = 28.093	28093
28	16#2005:16#20	Connect Auto-generated: 00.5.032 = 0.93	93
29	16#2005:16#21	Connect Auto-generated: 00.5.033 = 57	57

3.7 How to remove unused function block parameters

In the CODESYS ladder editor, function block parameters that are not assigned to variables or static values can be quickly removed by right clicking on the function block and selecting “Remove Unused FB Call Parameters”.

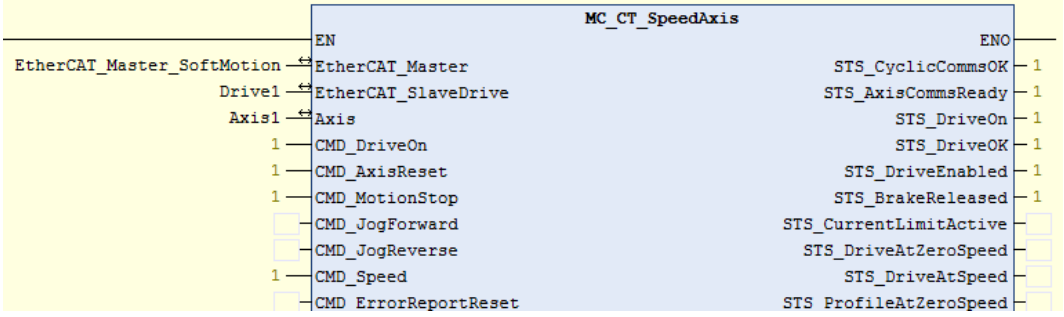


This will result in the unused parameters being hidden from the graphical representation of the function block.

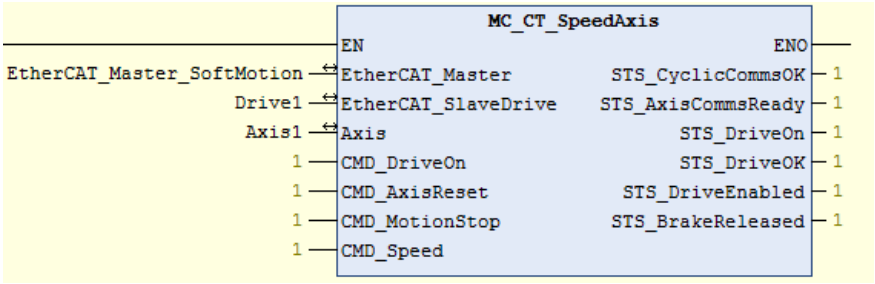


Parameters that have been removed are still accessible by user code, this is purely a visual change rather than an actual change of the function block interface and it’s functionality.

HINT: It is possible to keep unassigned parameters visible for later use, this is done by assigning a static arbitrary value such as “1” to the parameter before carrying out the “Remove Unused FB Call Parameters” action.



The parameters with arbitrary values assigned to them will be left in place after the “Remove Unused FB Call Parameters” action has been carried out.

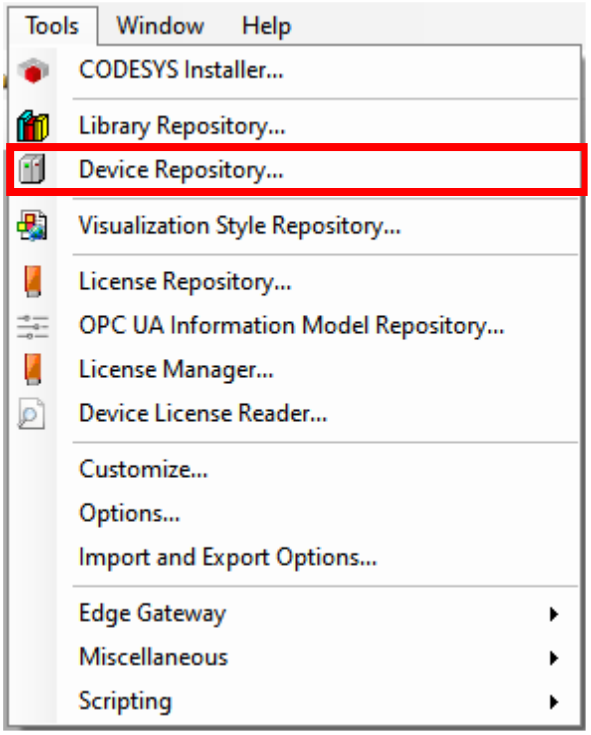


3.8 How to add a device description file

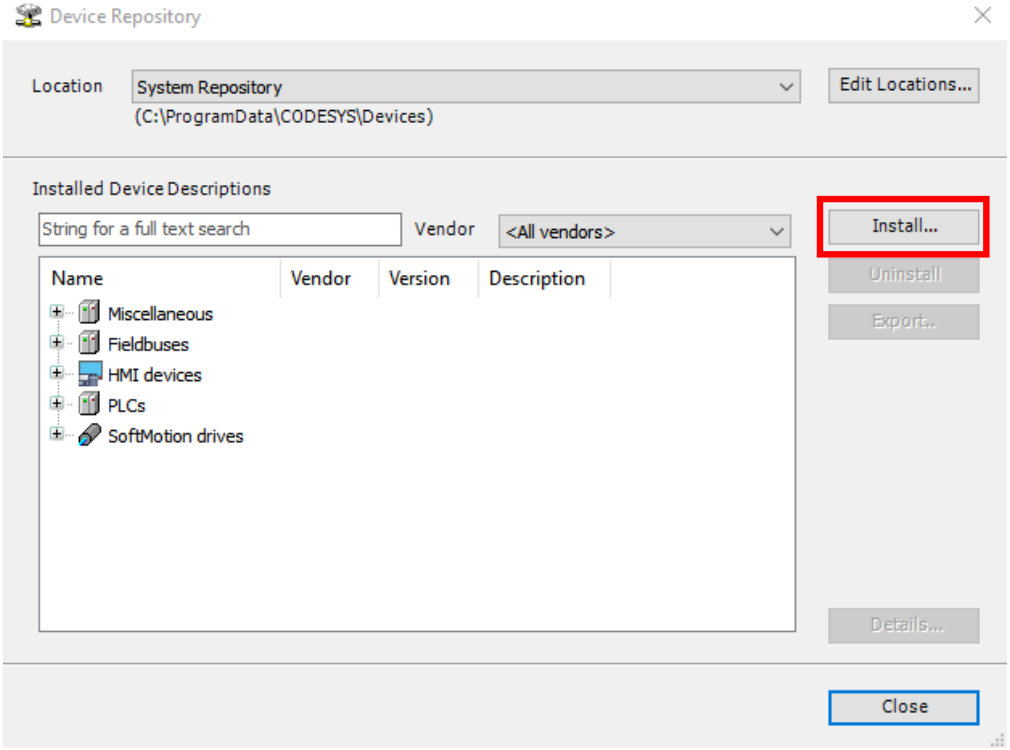
A device description file is required by CODESYS to allow a given PLC / IPC / Controller to be recognised by the development environment and added as a programmable target to the project. In preparation for this section, acquire the device description .xml file for CODESYS, for the application device.

The following instructions show how to add the device description file:

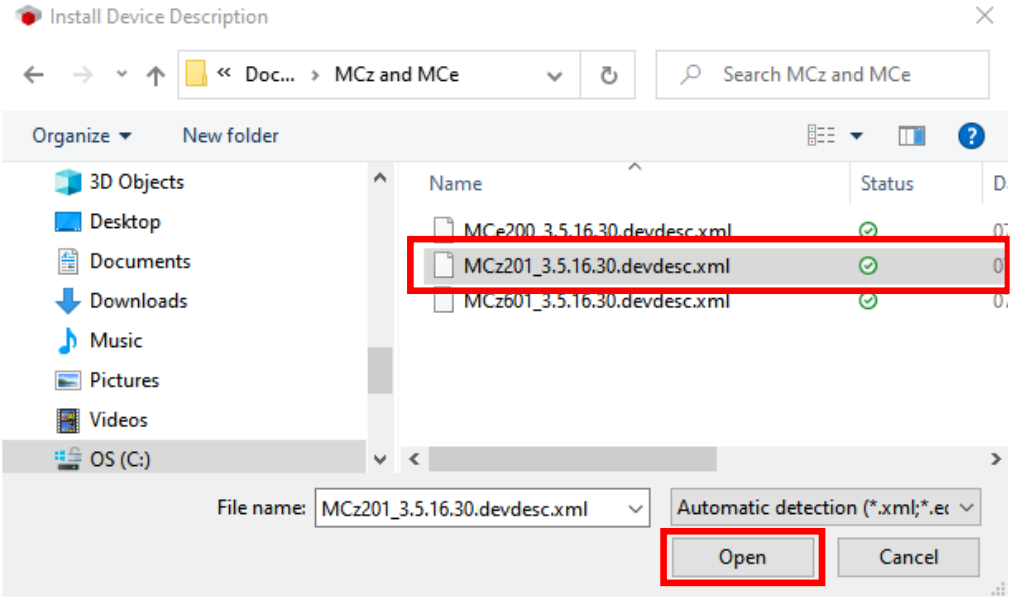
1. From the CODESYS tool bar select “Tools” > “Device Repository…”.



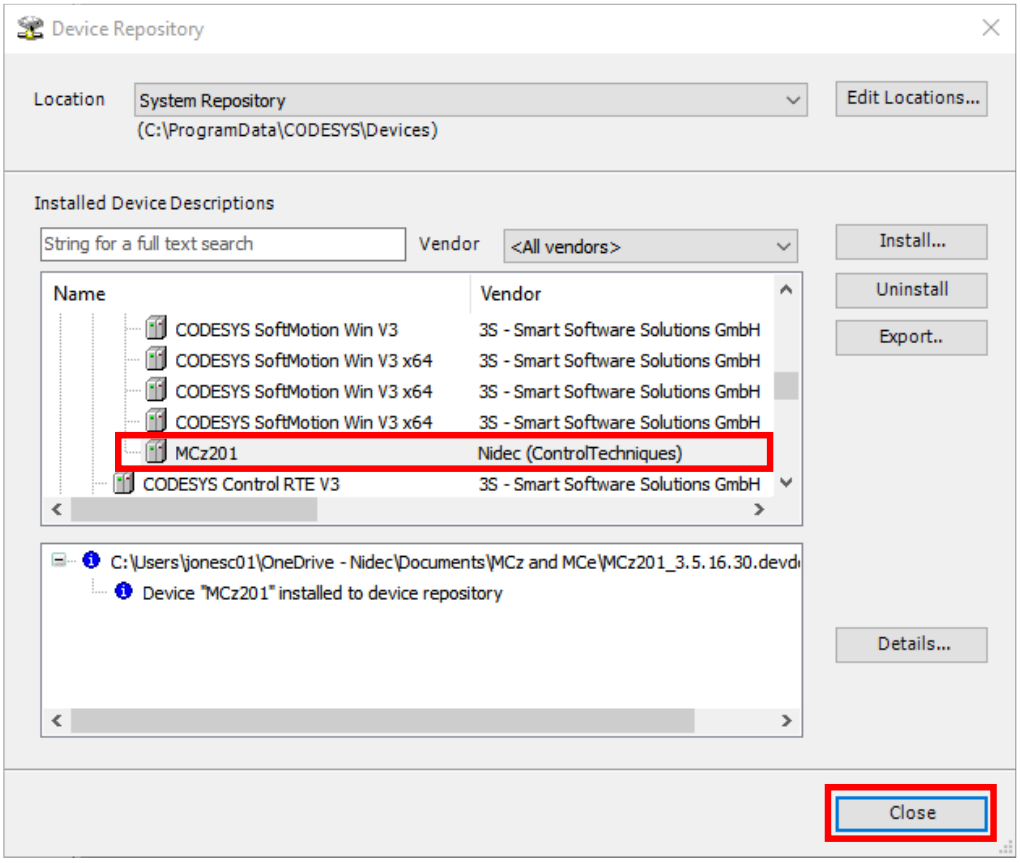
2. When the “Device Repository” dialog appears, select “Install…”



3. Browse for the location of the device description file to install, select the file, and then click “Open”.



4. The device is now added to the device repository and may be added to a CODESYS project. Click close when finished adding devices.



5. Once these files are added to the CODESYS environment this step doesn’t have to be repeated.

3.9 How to upgrade the EtherCAT interface firmware using Connect

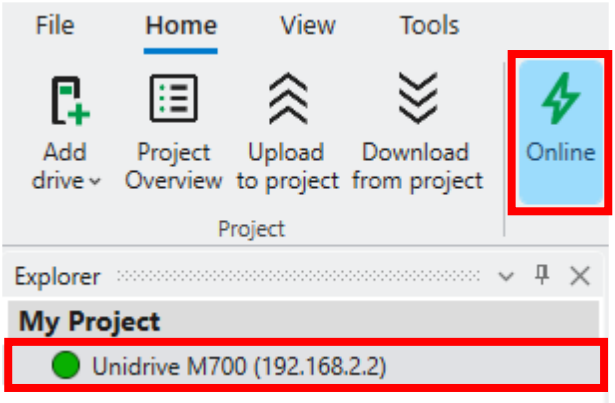
When setting up an EtherCAT network it is important to make sure the ESI file and firmware match each other to prevent issues when starting the EtherCAT network.

1. Install and open the “Connect” PC tool.

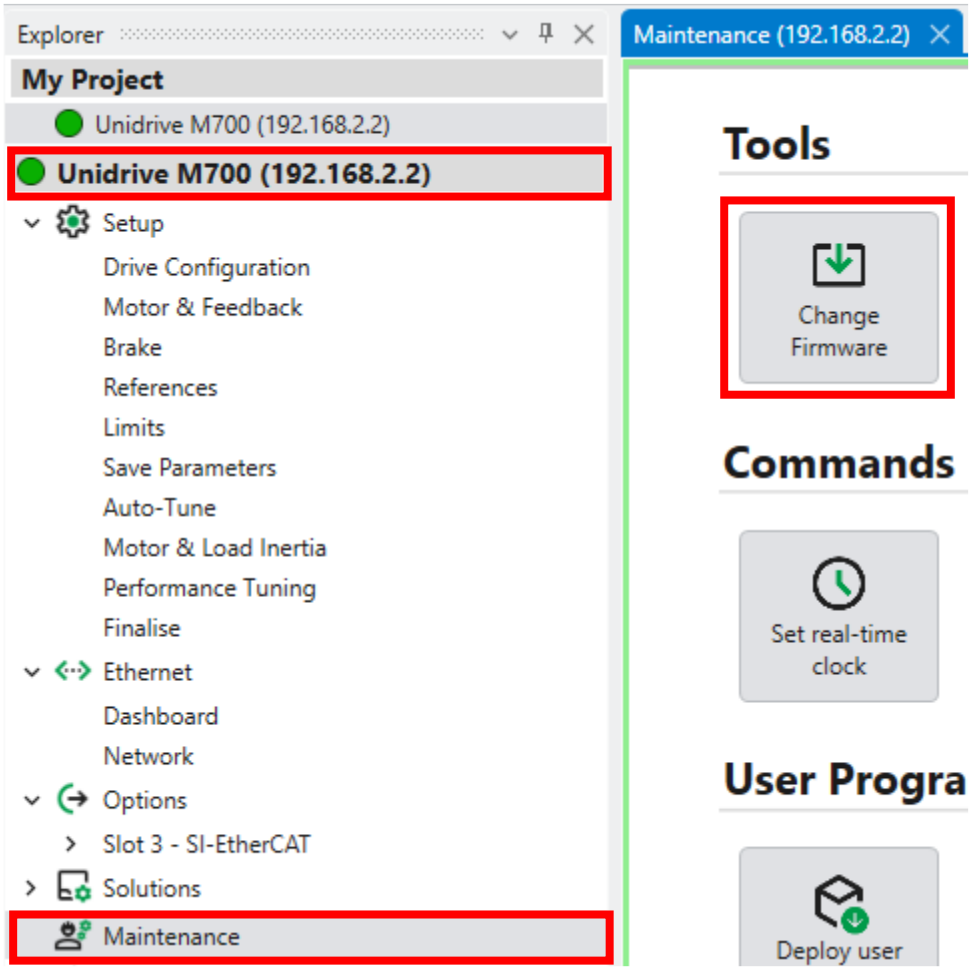
2. Establish a connection between the PC being used for Connect and the Control techniques drive, this connection can be achieved via Ethernet, EoE, or serial communications. See sections **2.3 Setup an EtherCAT EoE connection from the PLC to the drive** for details on how to establish the connection over an EtherCAT network using EoE.

   HINT: Ethernet and Serial communications may be used directly with the drive, but any EtherCAT-based communications require an EtherCAT network Master to be configured before the drive can be communicated with.

3. Highlight the drive to upgrade in the project tree and then click “Online”.

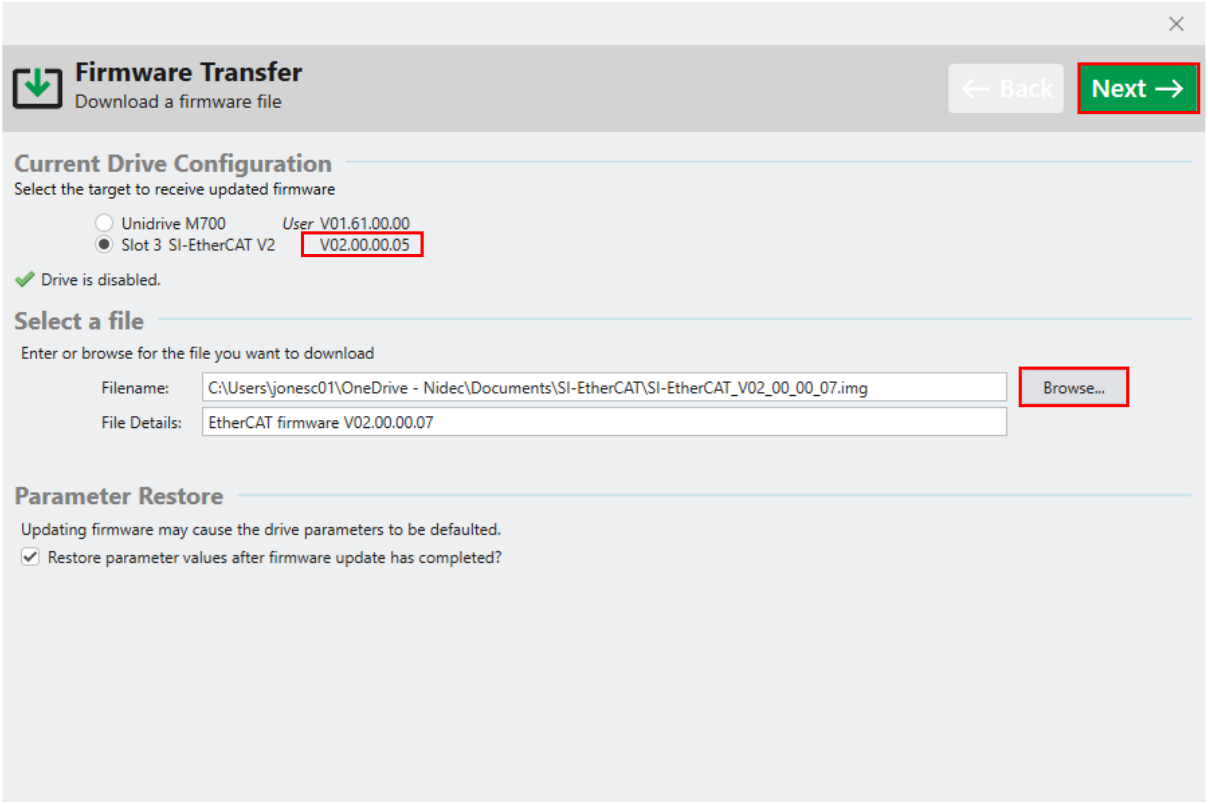


4. From the drive explorer tree double click on “Maintenance” and then “Change Firmware”:



5. The “Firmware Transfer” tool will open. This tool displays the current firmware present in the drive and any connected option modules.

   To perform a firmware upgrade, select the slot that the SI-EtherCAT module is installed in and click the “Browse…” button to locate the SI-EtherCAT firmware:



   Make sure that the drive is disabled and that any application software modules e.g. MCi210 or SI-Applications Plus have their programs stopped.

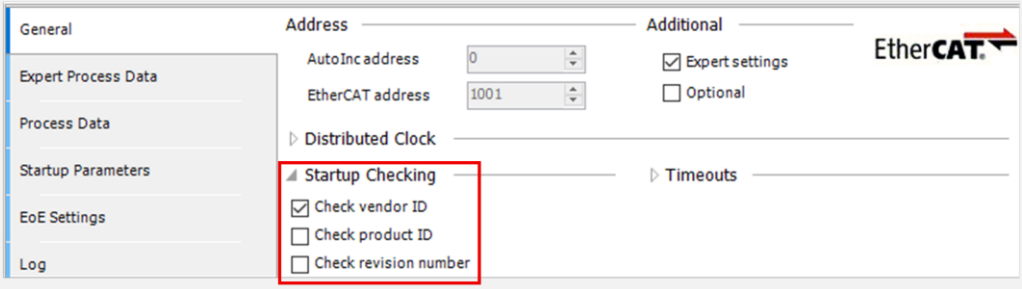
   Click 

   to continue.

6. Click “Start Download” to begin the transfer of the firmware. Please ensure that the drive remains powered up and the communications stay intact until the process has completed.

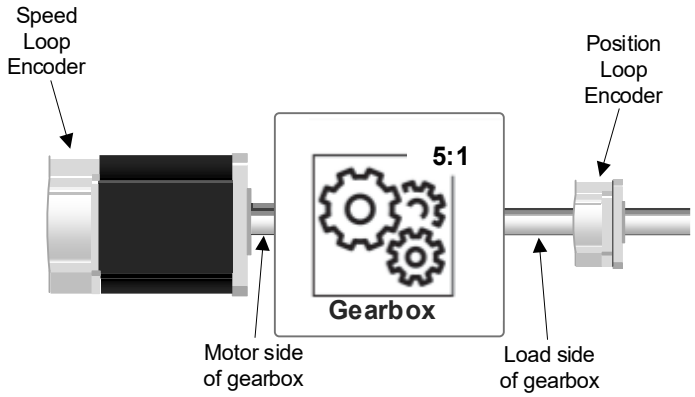
7. Power down the PLC/IPC and drives, then re-apply power to re-establish the network.

HINT: Downgrading the firmware using Connect over EoE may not fully complete since the ESI file used in the PLC / IPC will no longer match the target after the firmware has been transferred to the EtherCAT module and the EtherCAT master will stop EoE communications. To give the firmware update the best chance of completion always uncheck the “Check product ID” and “Check revision number” startup checks in the EtherCAT slave drive setup.



3.10 How to set up a dual loop system.

This section describes how to configure a dual loop system where the position loop is closed using an encoder attached directly to the load side of the gearbox as opposed to the motor side.



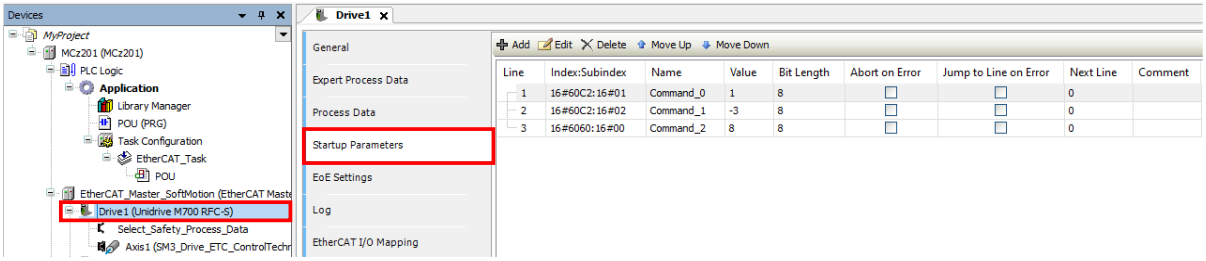
This type of system compensates for any backlash or torsional effects in the gearbox, resulting in a greater level of positioning accuracy at the load.

For this example, it is assumed that the motor encoder is connected the drives P1 interface, whilst the position feedback encoder will be connected to the drives P2 interface. Therefore, we need to configure the following CANopen over EtherCAT (CoE) objects:

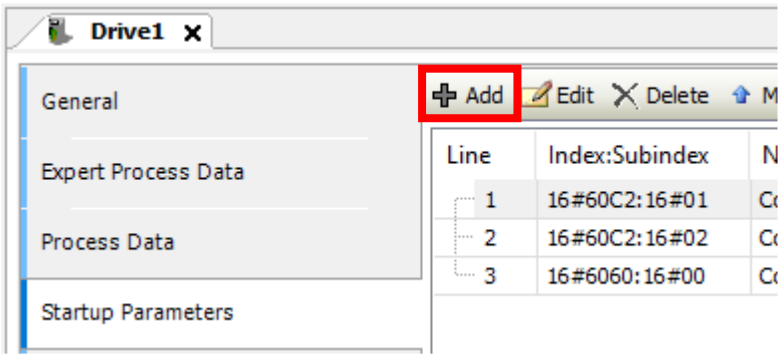
- Index: 0x3000, Sub Index: 0 – Position Feedback Encoder Configuration.
- Index: 0x3004, Sub Index: 1 – Additional Position Loop Output Scaling Numerator.
- Index: 0x3004, Sub Index: 2 – Additional Position Loop Output Scaling Denominator.

The following steps will describe how to configure these CoE objects so that they are actioned when the PLC / IPC / Controller is first powered on using the startup list.

1. Setup the axis using the information in section **2 Getting started**.
2. From the project tree, double click on the EtherCAT slave drive in this case it is called Drive1 and select the “Startup Parameters” tab.

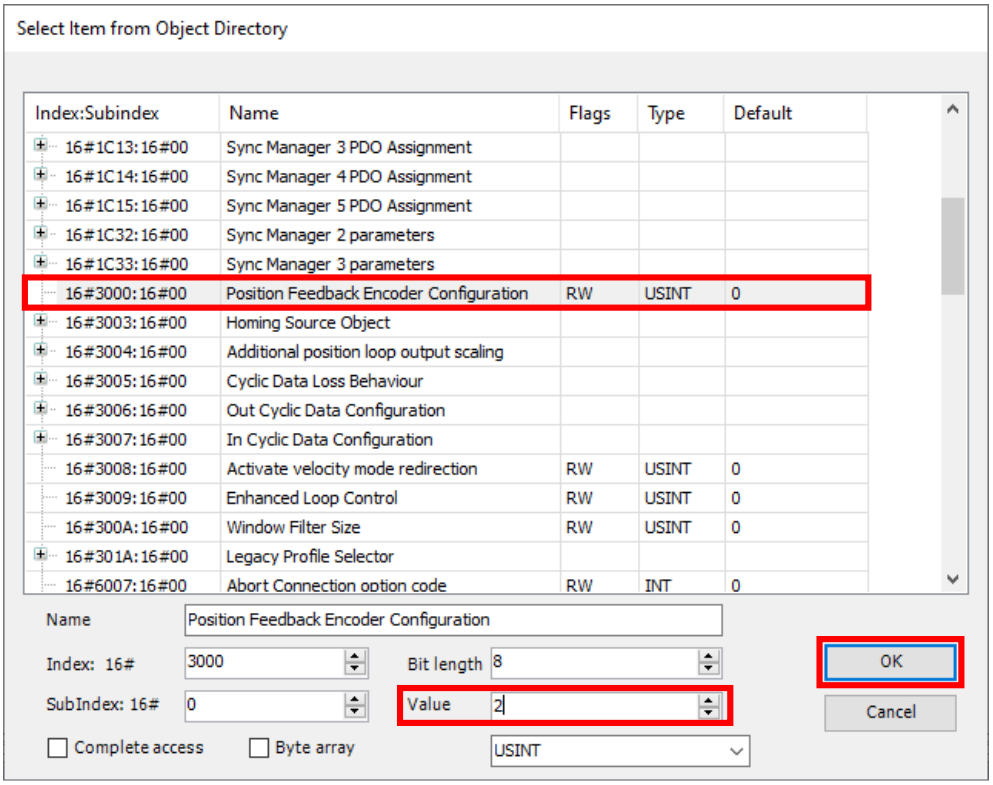


3. From the “Startup Parameters” view, select the “Add” button.



3. The “Select Item from Object Directory” dialog will be displayed.

   From the list of available mappings, select “16#3000:16#00 Position Feedback Encoder Configuration”, enter “2” into the value field and click “OK”.



HINT: The value of 2 corresponds to the drives P2 Interface which is being used as the position interface in this example. Other interfaces can be selected using the values in the list below.

This list is an extract taken from the SI-EtherCAT User Guide.

6.3.5 Feedback encoder source

Table 6-36 Feedback encoder source

0x3000	Position Feedback Encoder Configuration		
Access: RW	Range: 0 to 11	Size: 1 byte	Unit: N/A
Default: 0		Type: Unsigned Integer / USINT	
PDO Mappable: No		Update Rate: On change of CiA402 profile	
Description:	This specifies the source for position controller feedback, and the source for CiA402 position feedback objects.		

The source will have a value as follows:

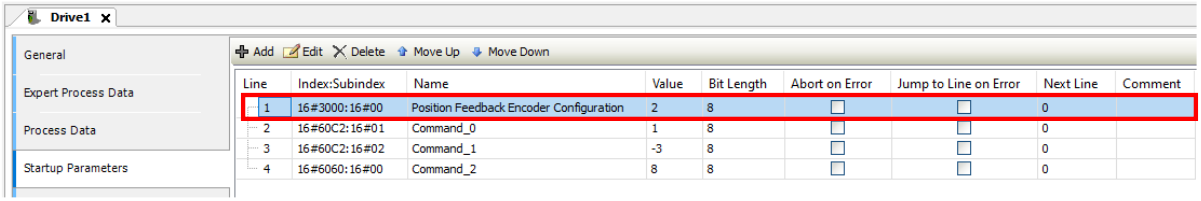
0 – The feedback source for the position controller will match the drive motor control feedback source (see below for details).

1 – Drive feedback source, P1 interface.
2 – Drive feedback source, P2 interface.
3 – Slot 1 position feedback module, P1 interface.
4 – Slot 1 position feedback module, P2 interface.
5 – Slot 2 position feedback module, P1 interface.
6 – Slot 2 position feedback module, P2 interface.
7 – Slot 3 position feedback module, P1 interface.
8 – Slot 3 position feedback module, P2 interface.
11 – Sensorless (the sensorless algorithm estimates position feedback).

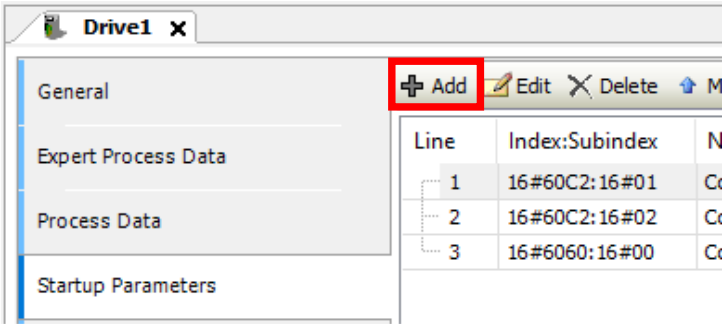
This value will be ignored on drives where no encoder input is present.

This object will be read upon a transition from the EtherCAT Pre-operational state to the Safe operational state.

4. The new entry will now appear in the “Startup Parameters” list.



5. From the “Startup Parameters” view, select the “Add” button.



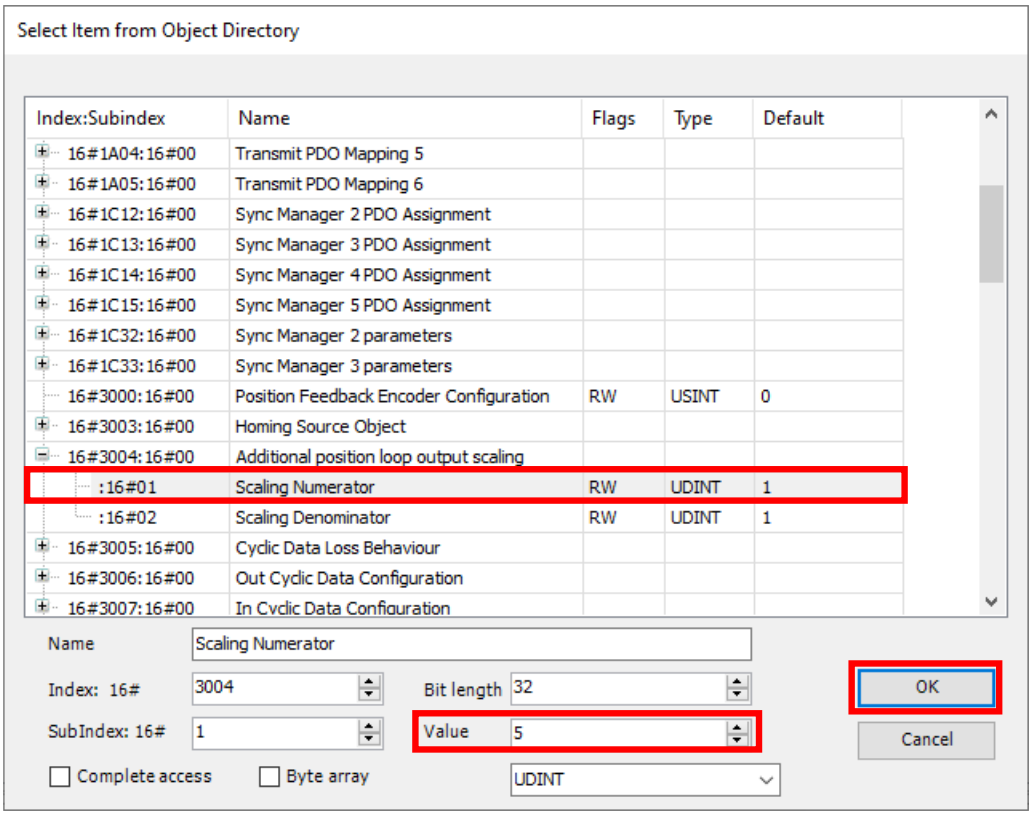
6. The “Select Item from Object Directory” dialog will be displayed.

   From the list of available mappings, select “16#3004:16#01 Scaling Numerator”.

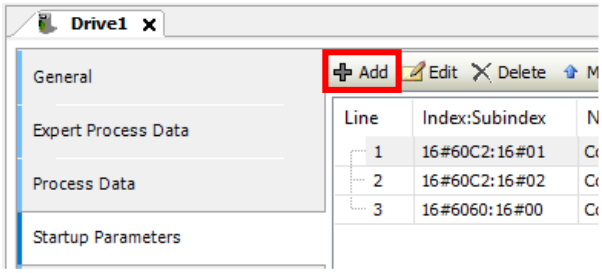
   The value of the numerator is the number motor revolutions at the gearbox input required to achieve the distance of the denominator which is the number of gearbox output revolutions.

   For example, if a motor is coupled to a 5:1 step-down gearbox and no other gearing is used in the system, the value of 5 should be entered into the numerator “Value” field.

   Enter the numerator into the value field and click “OK”.



7. From the “Startup Parameters” view, select the “Add” button.



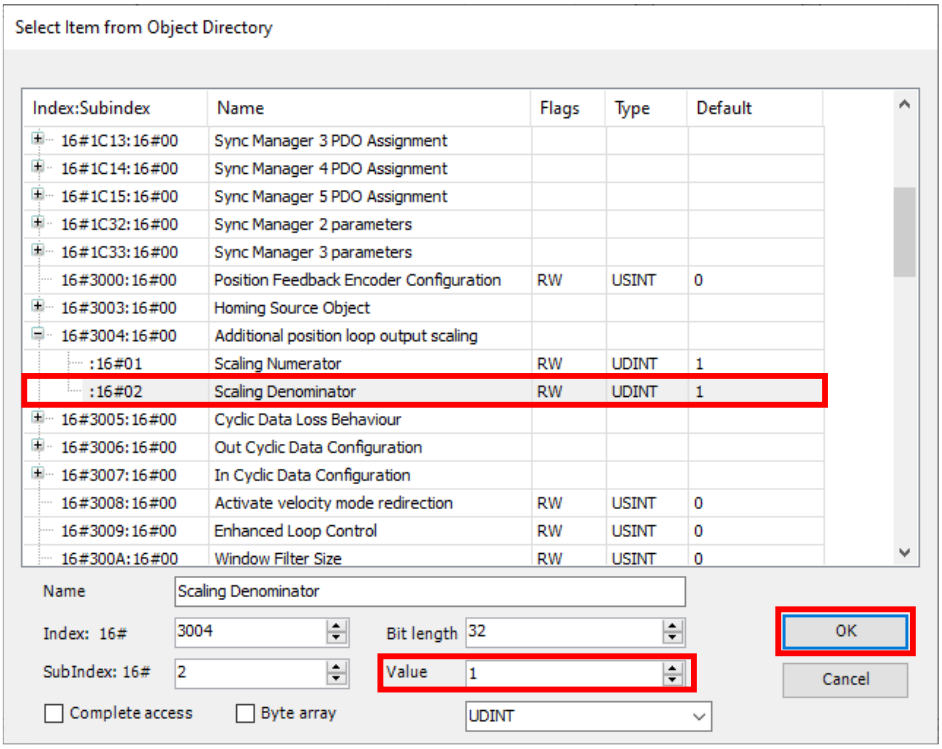
8. The “Select Item from Object Directory” dialog will be displayed.

   From the list of available mappings, select “16#3004:16#02 Scaling Denominator”.

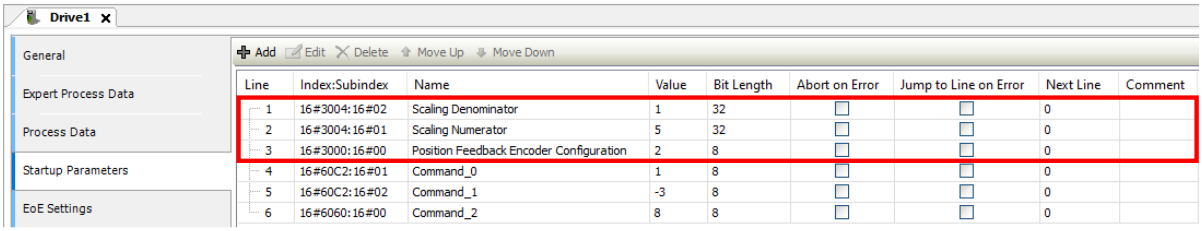
   The value of the denominator is the number of revolutions at the gearbox output required to achieve the distance of the numerator which is the number of gearbox input revolutions.

   For example, if a motor is couple to a 5:1 gearbox and no other gearing is used in the system, the value of 1 should be entered into the denominator “Value” field.

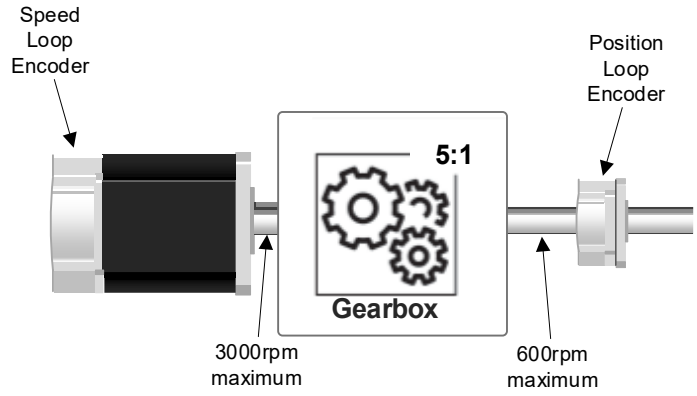
   Enter the numerator into the value field and click “OK”.



9. The Scaling Numerator, Scaling Denominator and Position Feedback Encoder Configuration mappings will now be present in the “Startup Parameters” list which means their values will be written to the specified CoE objects on startup of the PLC / IPC / Controller. A drive configured in this way will use the P2 interface for position control for any SoftMotion axis associated with this drive.



10. Calculate the maximum speed reference with the dual-loop scaling. For example, if the gearbox ratio is 5:1 stepdown and the maximum reference clamp is 3000rpm, the maximum speed at the position loop encoder is calculated from 3000rpm * 1 / 5 = 600rpm.



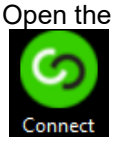
If the scaling of the position loop encoder is 1000units per revolution, the maximum velocity is 600rpm*1000units per revolution/60 seconds per minute = 10000units/s.

11. Test to see if the scaling is correct by running at constant speed and make sure that that the positional following error seen in Pr39.008 using Connect doesn't have a significant offset in it, i.e. the positional error should average out at close to 0.

3.11 How to create a Connect project using Ethernet or RS485

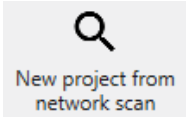
Create a new Connect project to match the requirement of the application:

1. Open the Connect PC software by double clicking the “Connect” icon.

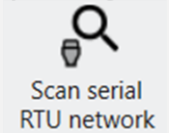


2. Ensure the version of Connect is a minimum V3.0.0. If an earlier version is installed please upgrade to V3.0.0; the software file may be obtained from your local Control Techniques Drive Centre / Distributor, or http://acim.nidec.com/drives/control-techniques, or Control Techniques Support Suite.

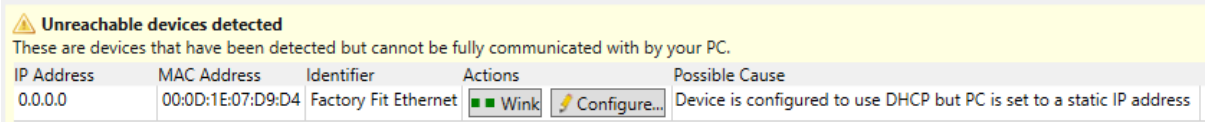
3. When Connect starts, select “New project from network scan”.



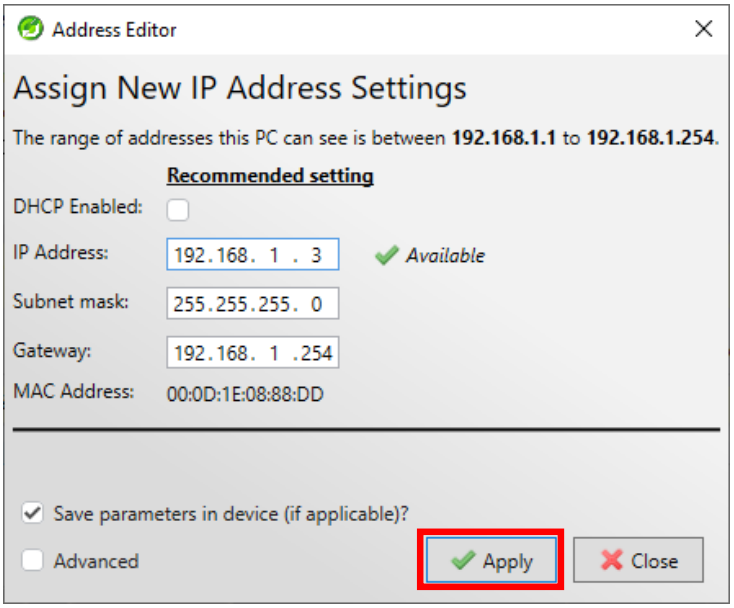
4. Select the network to scan with e.g. Ethernet.

> **HINT:** For an RS485 connection the steps are similar except that the scan is performed by clicking the click the “Scan serial RTU network” button.
>
> 
>

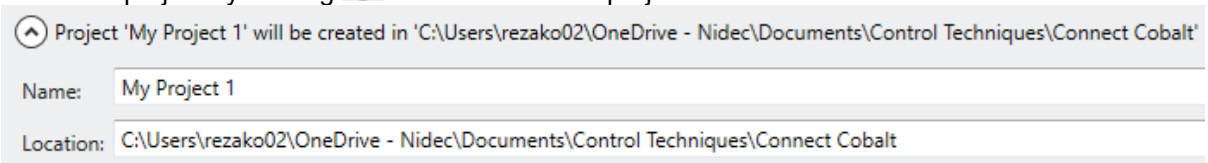
5. After scanning drive with IP addresses already configured will be automatically added to the project when it’s created. Any drive found in the scan which doesn’t have an IP address assigned, can be configured by clicking Configure...



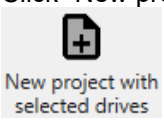
6. Specify the desired IP address configuration and click “Apply” to assign the new IP address to the drive.



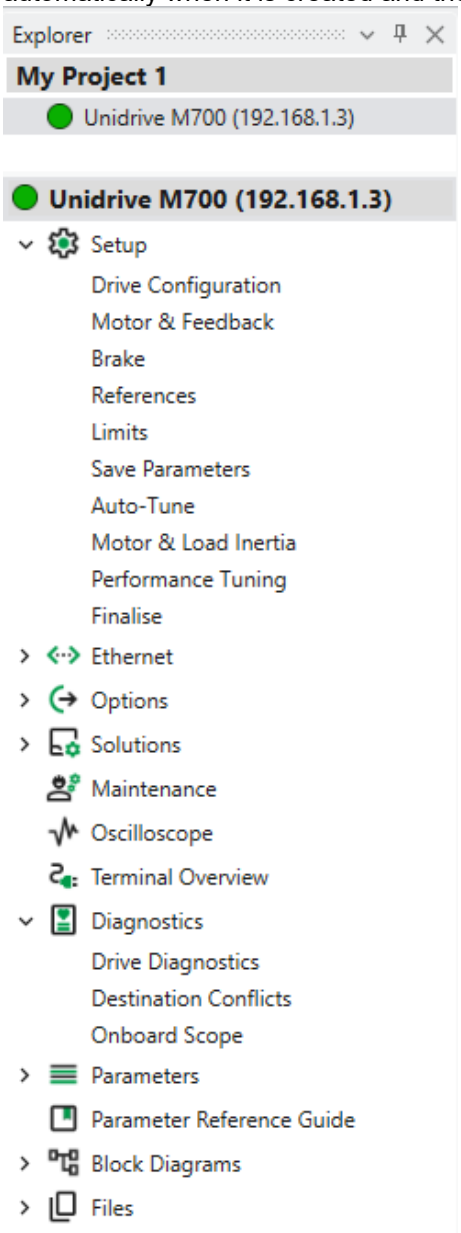
7. Name the project by clicking  and then fill in the project name.



Click “New project with selected drives” to create the project



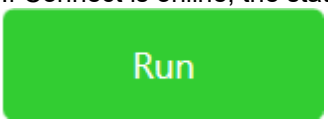
8. The project is created with the selected drives. All parameters are uploaded into the project automatically when it is created and the drive is online.



3.12 How and when to tune the Position Loop

The axis position loop in the drive is activated once the Softmotion axis is powered on, where:

- The drive is enabled by applying 24V to the STO terminal.
- The drive is healthy i.e. no trips.
- MC_Power has been used, successfully, to power on the axis.
- If a keypad is fitted, the display indicates “Run”.
- If Connect is online, the status in the bottom left shows “Run”.

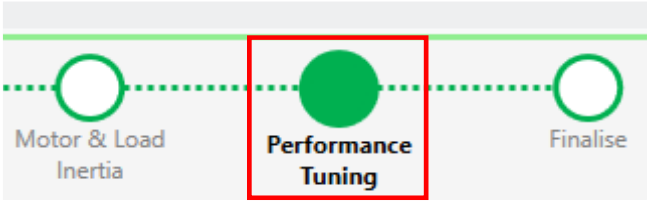


The position loop should be tuned when the application software is written, and the machine is running with the worst-case production motion profile, i.e. the most dynamic motion profile coupled with the highest load that will be experienced by the system while running.

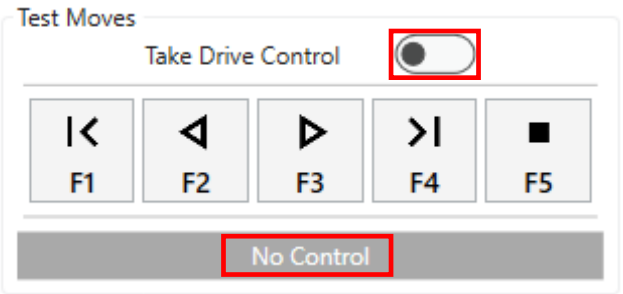
In many cases the default position loop p gain of 25.00 will give reasonable performance, but in cases where there are tight tolerances, the position loop performance can be increased to suit the requirement.

Before attempting to tune the position loop the speed loop must be tuned first as described in section **2.5 Commission the drive and motor using Connect**.

With the machine running its worst-case motion profile, open Connect and go to the Performance Tuning step

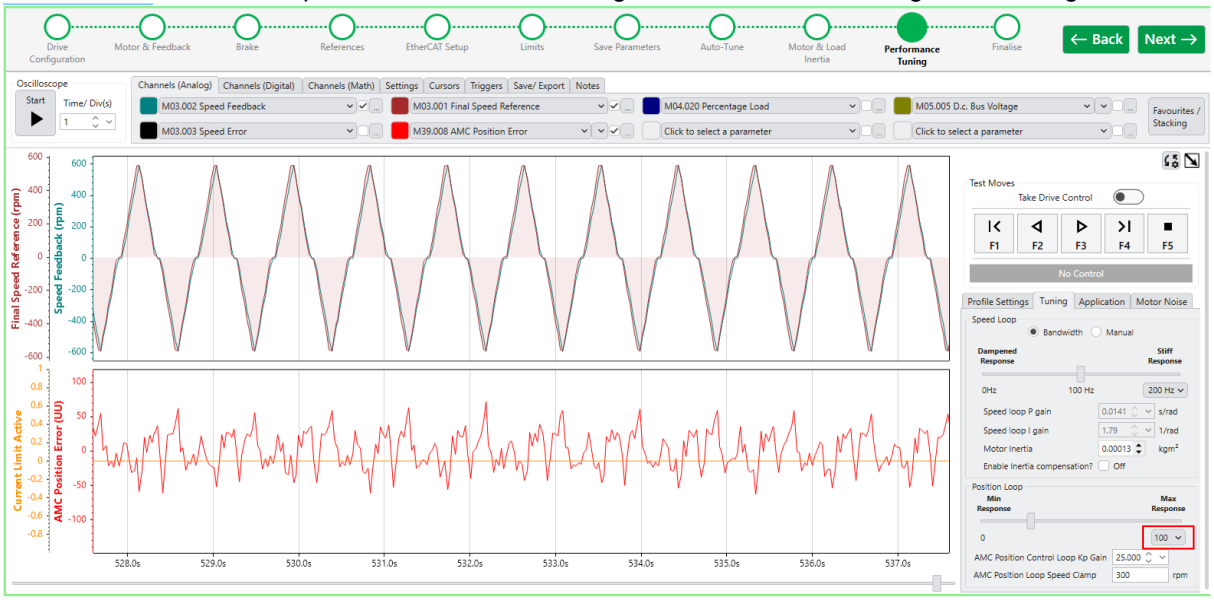


Don’t enable the test move control. The test moves status must say “No Control” to leave the application software in control of the drive and therefore the drive position loop active, ready for tuning.



Move the position loop p gain slider to optimise the tuning and observe the red AMC Position Loop Error trace. Move the slider to the right to increase the position loop gain and increase the response. The smaller the peak values the better the tuning is.

If the P Gain needs to be pushed above 100 the range can be increased using the following control:



Care must be taken when increasing the gain as it is possible to increase the gain until the system stability limit is reached and the axis motor will make additional noise or worst case oscillate. Small steps are recommended when tuning.

If the tuning has reached the best possible position error and axis noise compromise, it is possible to reduce the error further by adding Jerk to the motion profile if one has not already been used. Even very high Jerk values can make a big difference to the peak position following error values when beginning and ending acceleration or deceleration.

When the optimal value has been reached save the parameters in the drive.

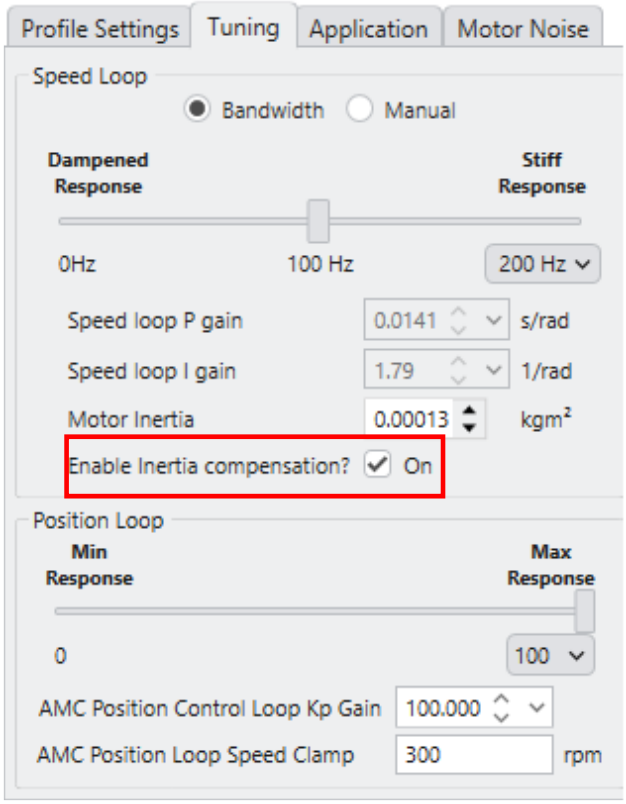
3.13 How and when to use inertia compensation

Inertia compensation, as the name suggests should only be used when there is significant load inertia and / or large acceleration rates. If the inertia is low or the acceleration rates are also low then this feature will not benefit the performance of the axis and is best not used.

Inertia compensation engages a Torque Feedforward term in the drive which, assuming the motor and load inertia has been measured correctly, applies a torque reference to the drive in proportion to the motion profile acceleration output and the inertia value. This helps reduce speed loop following error during acceleration and deceleration, at the cost of slightly increased motor noise.

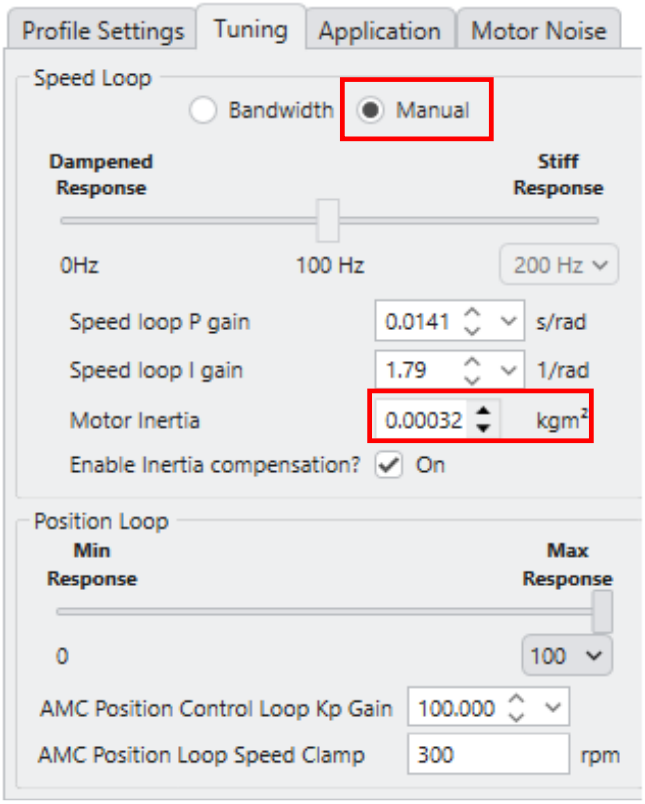
Before using this feature the speed and position loops must be tuned first as described in sections **2.5 Commission the drive and motor using Connect** and **3.12 How and when to tune the Position Loop.**

To use the inertia compensation check the Enable inertia compensation checkbox on the Performance Tuning step of the guided setup.



If motor noise induced by the inertia compensation term must be reduced, or the amount of inertia compensation must be increased to further improve the following error, select manual speed loop tuning to disconnect the inertia value from the speed loop gain setting. Then decrease the inertia value to reduce motor noise or increase the inertia to increase the output of the inertia compensation.

Before modifying the “Motor Inertia” make a note of the original value so it can be restored later if required.

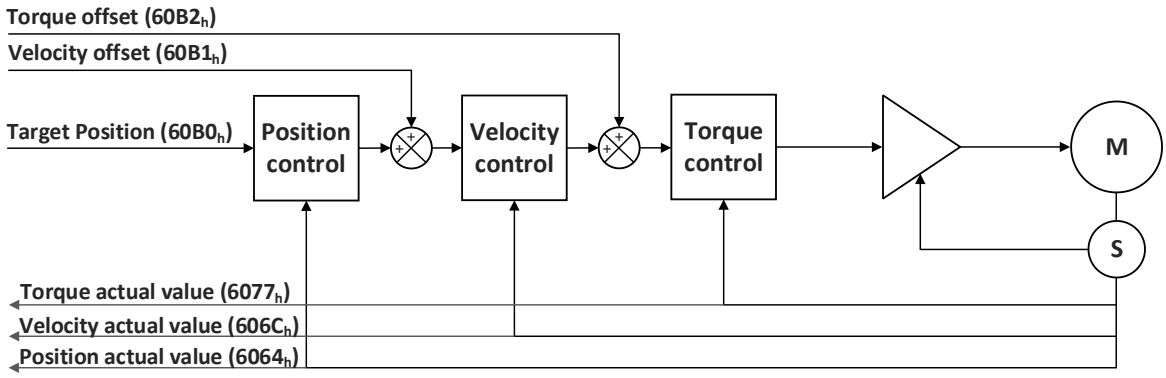


HINT: If the Inertia has been modified to tune the output of the Inertia Compensation feature, and Bandwidth mode tuning is later required, the inertia must be restored to its original value before selecting bandwidth mode, otherwise the speed loop gains may be calculated incorrectly.

3.14 How to apply velocity feedforward to a Softmotion axis

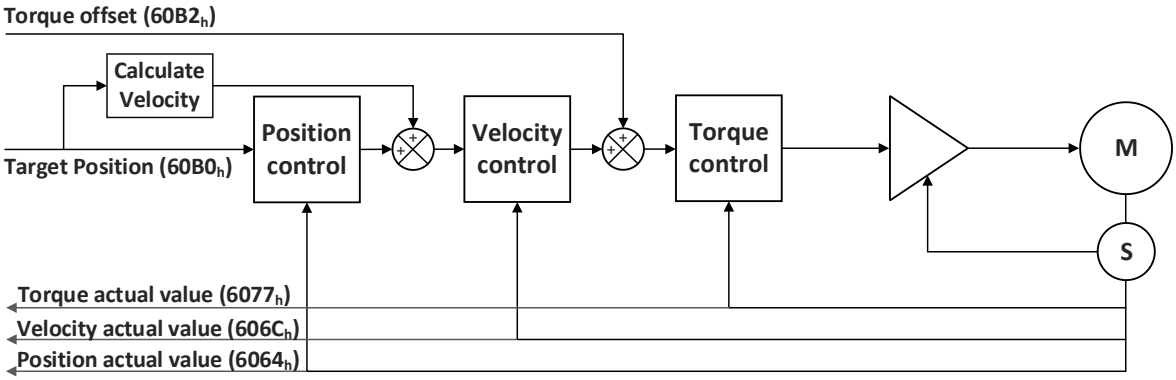
This section relates to velocity feedforward with a Softmotion axis being controlled using EtherCAT and DS402 in CSP (Cyclic Synchronous Position) mode, where the velocity calculated by the Softmotion axis motion profiler is fed forwards directly to the target axis which improves the granularity of the velocity profiling, giving the smoothest operation. The velocity feedforward must be PDO mapped to 0x60B1 and enabled by setting 0x300B = 1. The velocity feedforward is calculated from Axis.fActVelocity. This gives 100x better velocity reference resolution.

The velocity feedforward functionality is only available with >=V02.00.00.16 EtherCAT firmware and ESI files.



(Control when 0x300B = 1)

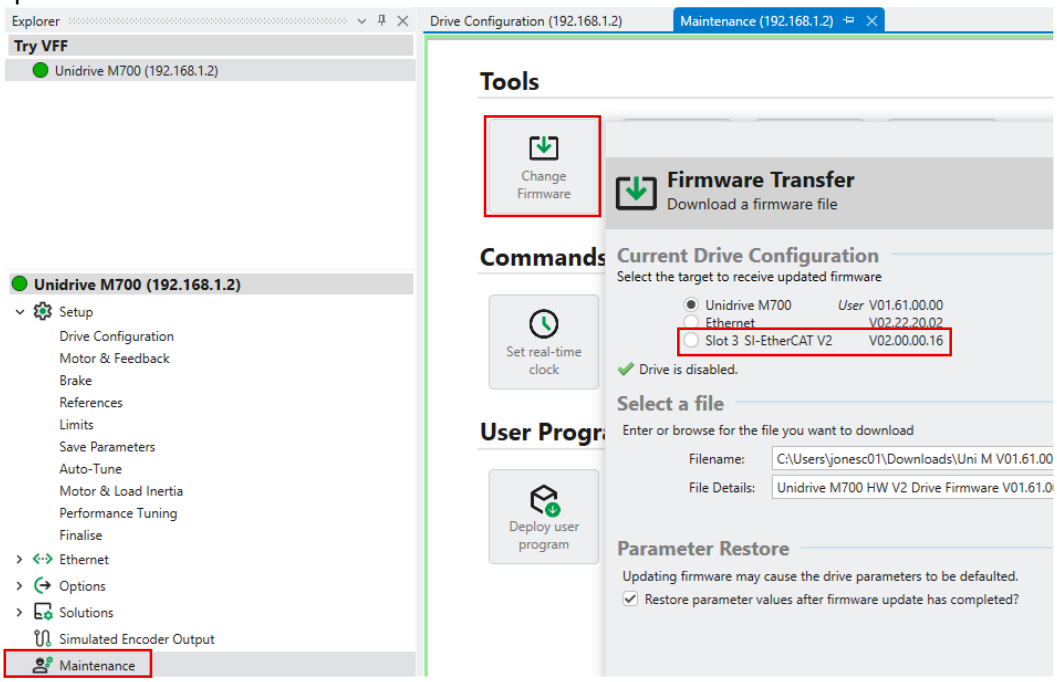
For a normal axis using the default setup i.e. no velocity feedforward implemented, the drive calculates the velocity feedforward term by differentiating the position reference passed into 0x60B0. This tends to be noisier since the velocity granularity is based on whole position units.



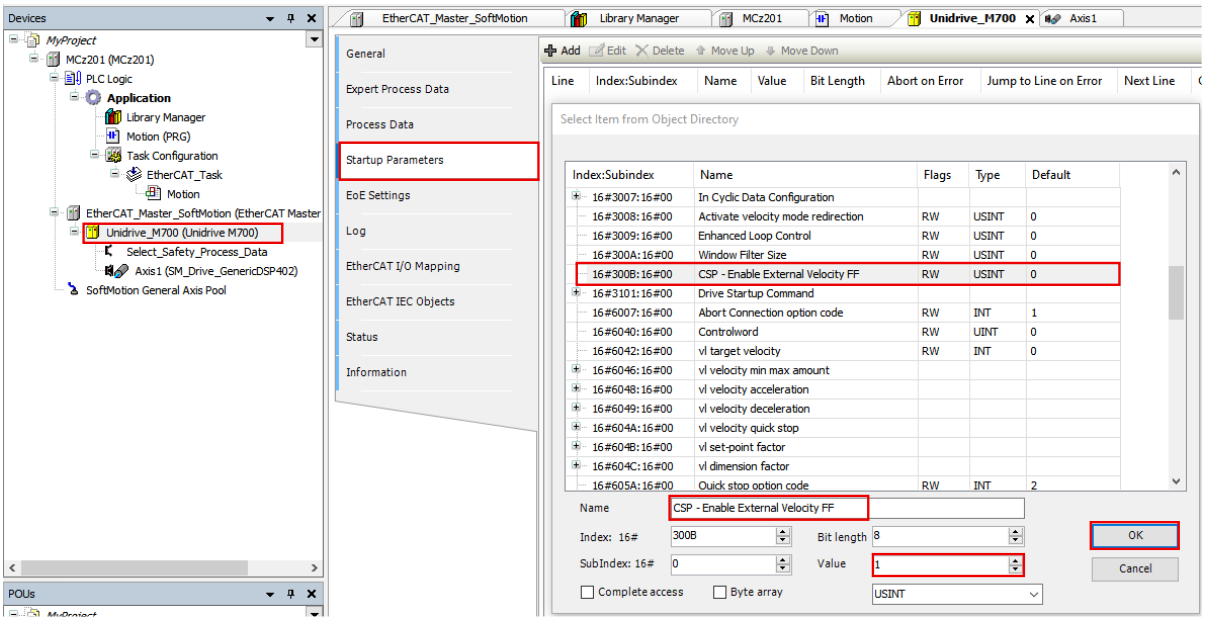
(Control when 0x300B = 0 (Default))

Use the following steps to configure the velocity feedforward:

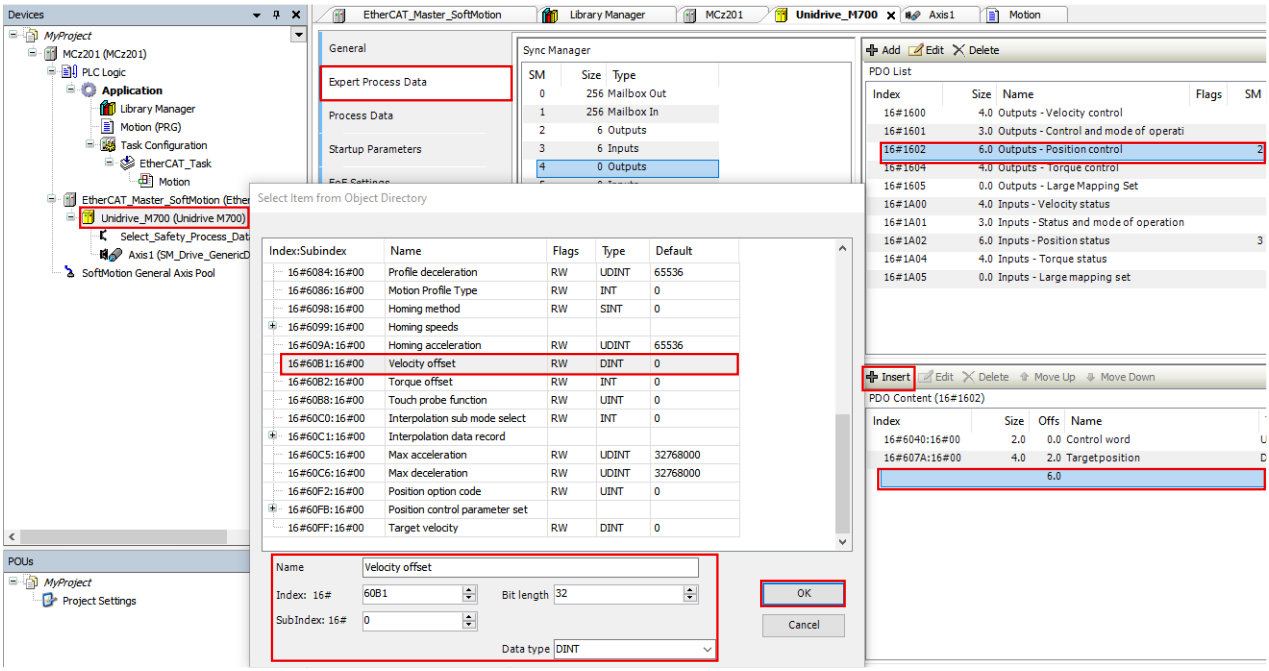
1. Make sure that the firmware in the EtherCAT interface e.g. SI-EtherCAT is at least V02.00.00.16. This is the version that supports the feedforward mechanism. In Connect select “Maintenance” > “Change Firmware”. All firmware versions for the drive and options will be listed.



2. Enable the velocity feedforward term by setting 0x300B =1 in the startup list. Double click on the EtherCAT slave in the Devices tree. Select the “Startup Parameters” tab. From the list select the entry that starts with “16#300B”. When the PDO link boxes update set “Value” to 1.



3. Map an outgoing PDO to 0x60B1. Select the “Expert Process Data” tab. In the “PDO List” select “16#1602” and then in the “PDO Content” list select an empty row and then click “Insert”. When the “Select Item from Object Dictionary” dialog opens, select “16#60B1” from the list to populate the PDO mapping. Click “OK” when finished



4. A Softmotion axis knows how to deal with 0x60B1 when it is mapped to a PDO, so no further setup is required. Compile and download the code to use the velocity feedforward feature.

4 Additional information

4.1 Terminology

FB = Function Block.

Forward = A positive speed or an increasing position.

Reverse / Backwards = A negative speed or a decreasing position.

rpm = Revolutions per minute.

CMD = Command.

CFG = Configure.

STS = Status.

DDF = Device Description File for importing new project devices in CODESYS.

ESI = The communications description file for a EtherCAT device.

PLC = Programmable Logic Controller.

EoE = Ethernet Over EtherCAT.

IPC = Industrial Personal Computer.

EGB = Electronic Gear Box.

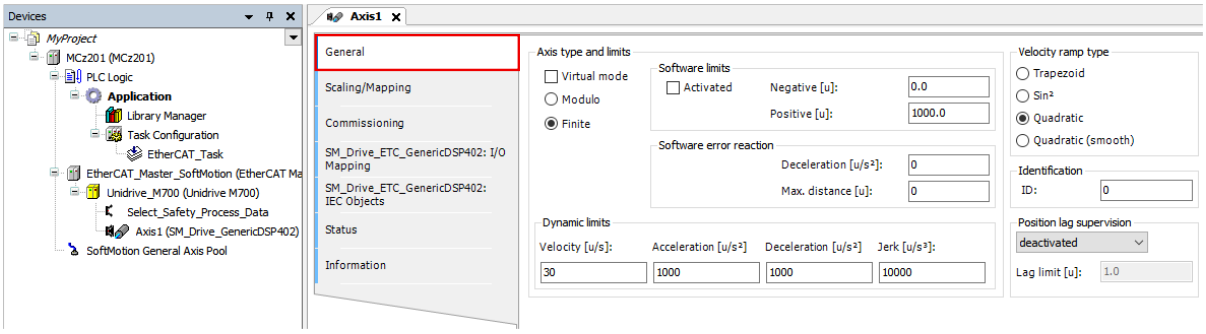
CoE = CANopen over EtherCAT.

PDO = Process Data Object.

SDO = Service Data Object.

4.2 Axis configuration information

4.2.1 General tab information

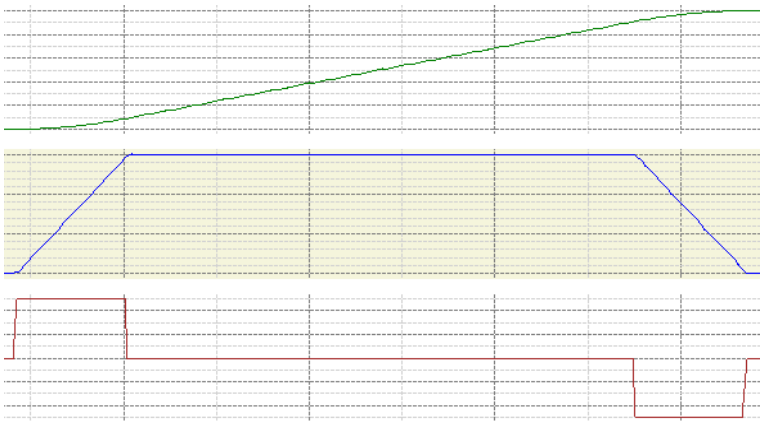
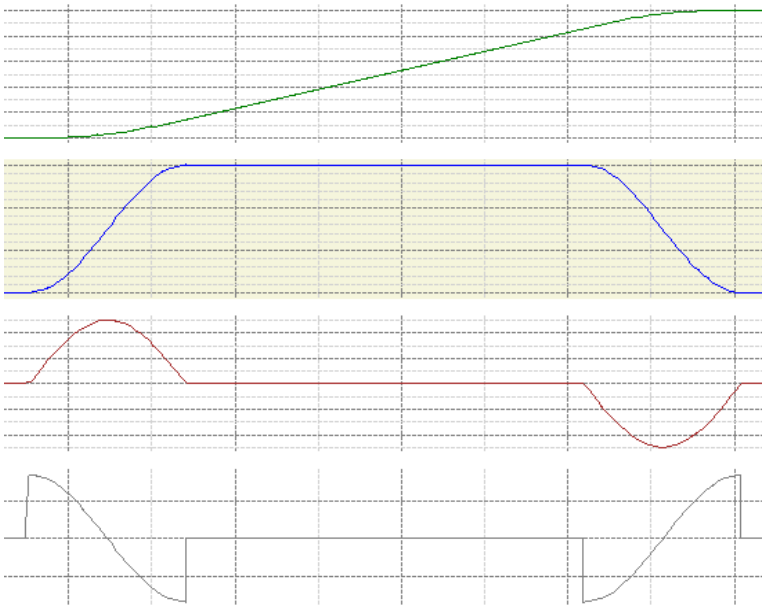
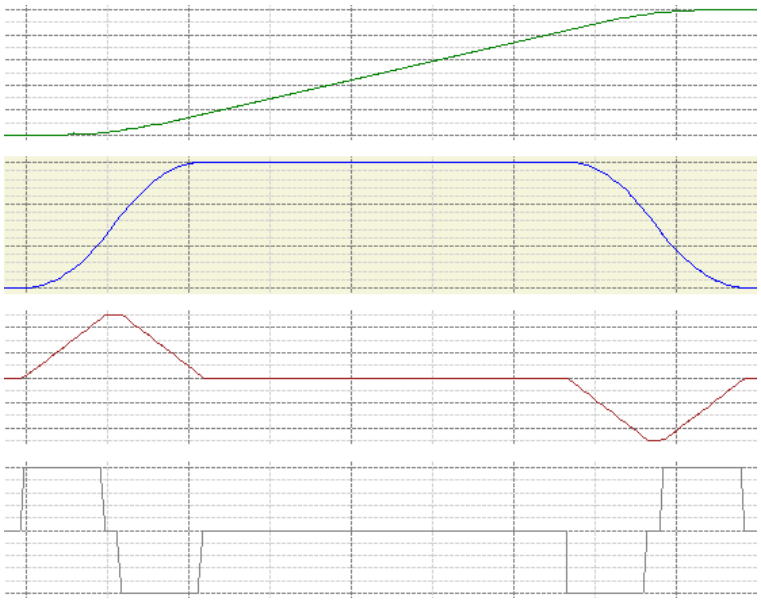


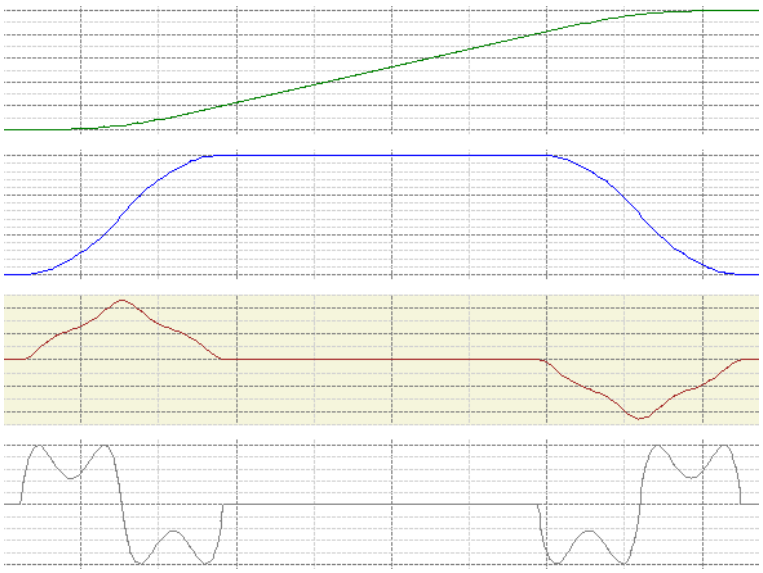
4.2.1.1 Axis Type and Limits

Virtual Mode	The drive is replaced by a simulation that is similar to a virtual drive unit. When there is a coupled drive, this does not have any effect on the fieldbus device. They function as usual without sending or receiving messages to or from a physical device. Additionally, the user can activate and deactivate virtual mode of a drive axis in IEC code by using the SMC3_ReinitDrive function block.
Modulo	The drive turns endlessly without limiting the traversing range (example: belt drive). **Modulo Settings** • **Modulo value [u]:** Value of one cycle (modulo period) The value is saved in the fPositionPeriod parameter of the AXIS_REF_SM3 function block. If the **Modulo** type is selected, then the product fPositionPeriod * dwRatioTechUnitsDenom must be an integer.
Finite	The drive has a fixed work area (example: one linear drive). **Software Limits** • **Activated:** Position values are restricted by the lower limit Negative and an upper limit Positive. • **Negative:** Input field for the negative limiting value • **Positive:** Input field for the positive limiting value

HINT: Dynamic Limits only apply to CNC axis or function blocks that are prefixed SMC_ControlAxisBy* (for detecting / preventing jumps in axis movement).

Therefore, **entering values into these limits will have no effect** on a standard SoftMotion axis using the CT SoftMotion Interface Library.

Trapezoid The velocity is partially linear and continuous, whereas the partially constant acceleration indicates jumps.	
Sin² The breaks in the velocity profile are smoothened (by using the sin² function instead of lines) to reduce the jumps in acceleration. The user cannot limit the jerk for this ramp type. The set maximum jerk has an effect only if the acceleration does not equal zero at the beginning of the movement and the interrupted deceleration and acceleration ramp cannot be continued seamlessly. Then, taking the jerk limit into account, the acceleration is decreased to zero before the current movement is started. As compared to the trapezoidal velocity profile, the deceleration takes more time in this case.	
Quadratic The acceleration is partially linear and continuous and the jerk has jumps. The velocity consists of quadratic and linear segments.	

Quadratic (smooth) The linear acceleration ramps of the quadratic ramp type are replaced by a "smooth" function with a slope value is zero at the beginning and end. As a result, the jerk is also continuous. If movement is interrupted, then jumps in the jerk can result.	

4.2.1.3 Software Error Reaction

Deceleration [u/s²]:	Deceleration value when reaching a software limit switch or when an error has occurred.
Max. distance [u]	Optional: The drive has to have reached a standstill within this distance after an error has occurred.

4.2.1.4 Velocity Ramp Type

The following images demonstrate the effect of the different ramp types. The **position** is drawn in green, the **velocity** in blue, and the **acceleration** in red.

4.2.1.5 Identification

ID	Integer identifier. Should be unique for each drive. For example, this identifier is used in the PLC log in order to identify the drive when an error occurs.

4.2.1.6 Lag Monitoring

System response to a detected lag.

A lag is detected when the difference between the set position and the compensated actual position exceeds the lag limit. The extrapolated actual position is calculated in the following formula:

extrapolated actual position := actual position + actual velocity * cycle time * Axis.fSetActTimeLagCycles

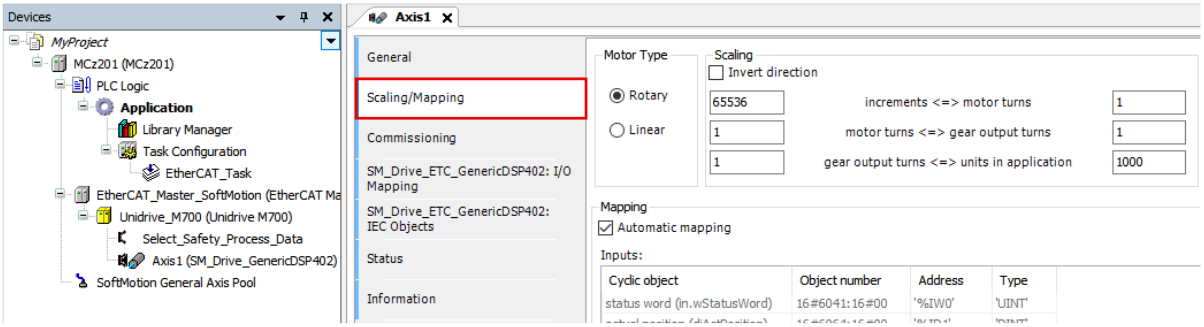
This value is approximately the position that the axis will have in fSetActTimeLagCycles cycles. The value compensates the virtual lag that forms by the time displacement due to communication. The fSetActTimeLagCycles parameter is defined in the AXIS_REF_SM3 function block.

HINT: When you monitor the lag, you should determine the value for fSetActTimeLagCycles and specify it in the parameters of the drive axis. For a description, see the "Determining the Dead Time of the System – Determining the Dead Time of the System" chapter.

HINT: Lag monitoring is not available for virtual drives.

Deactivated	Lag monitoring is deactivated.
Disable drive	The bRegulatorOn bit is forced to FALSE (compare with MC_Power input) which first forces the deceleration of the drive and then the deactivation of the drive (depending on the drive implementation).
Do quickstop	The bDriveStart bit is forced to FALSE (compare with MC_Power input) which forces the drive to perform a quickstop.
Stay enabled	The drive remains switched on, but all running movements are stopped abruptly.
Lag limit:	Lag monitoring in the controller Independent monitoring can also exist in the drive, but it is not configured in this dialog.

4.2.2 Scaling/Mapping tab information



4.2.2.1 Motor Type

Rotary	The settings in **Scaling** apply to rotary motors where the following scaling can be specified: • Motor increments per motor turn. • Motor turns per gearbox output turn. • Gearbox output turns per engineering units (used in the application).
Linear	The settings in **Scaling** apply to linear motors. (Simplified configuration without gears and motor turns).

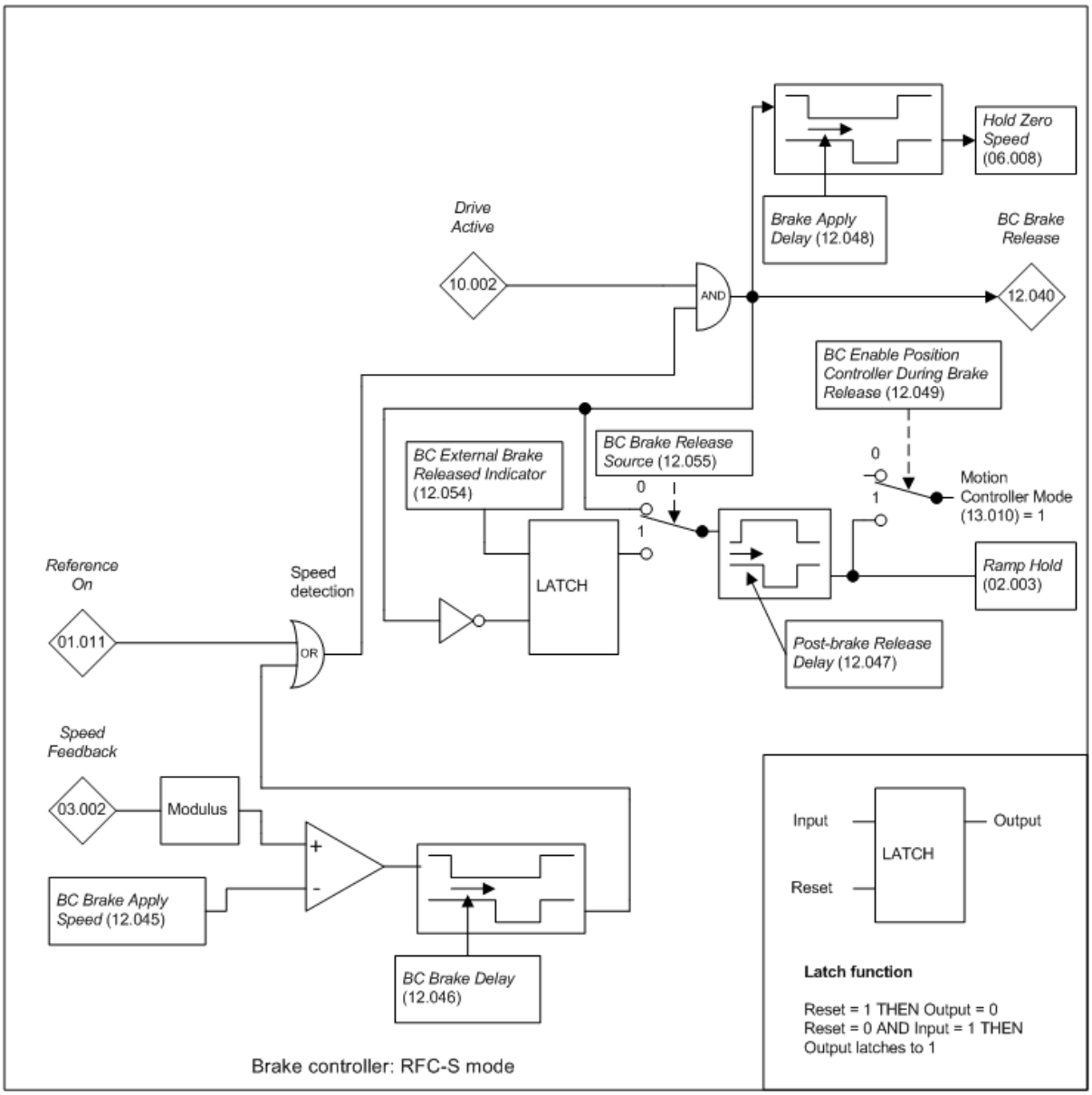
4.2.2.2 Scaling and Direction Inversion

Invert direction	The direction of turn is reversed. The drive receives the specified values with opposite signs. This is a common requirement in machines where a motor is mounted on both the left and ride side of the same machine and are mechanically coupled together.
increments <=> motor turns	Number of “increments” that correspond to a given number of “motor turns”.
motor turns <=> gear output turns	Number of “motor turns” that correspond to a given number of “gear output turns”.
gear output turns <=> units in application	Number of “gear output turns” that correspond to a given number of units in the application.

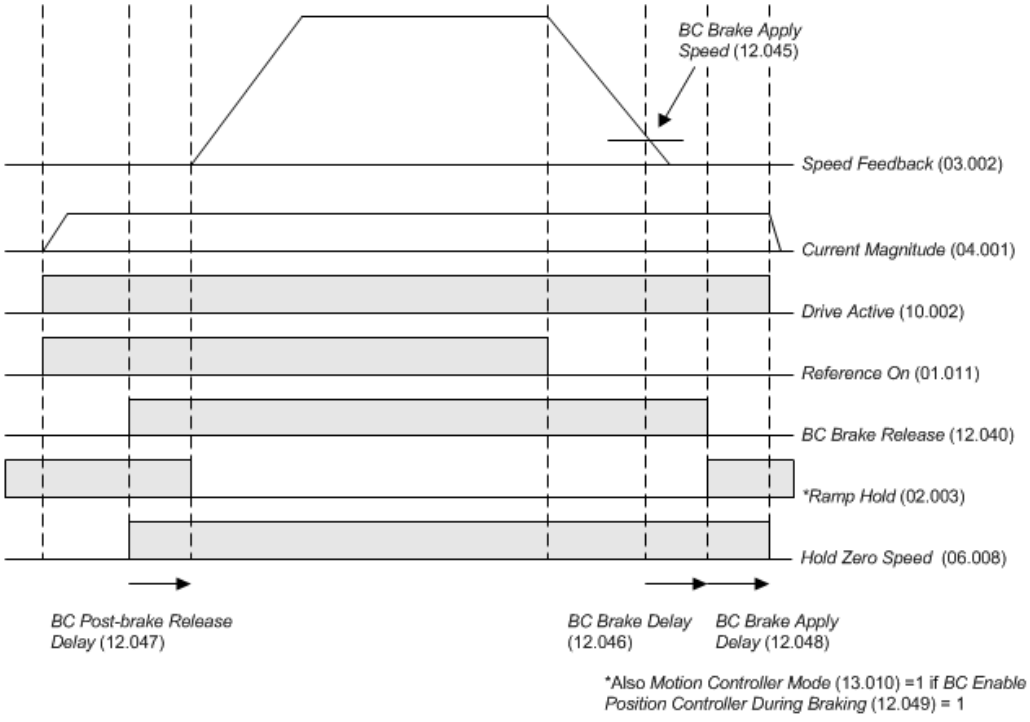
4.3 Mechanical brake controller logic

4.3.1 RFC-S closed-loop permanent-magnet motor brake controller

This section indicates the logic and timing diagrams for the brake controller in RFC-S mode for closed-loop permanent-magnet motors to help illustrate the functionality and timing.



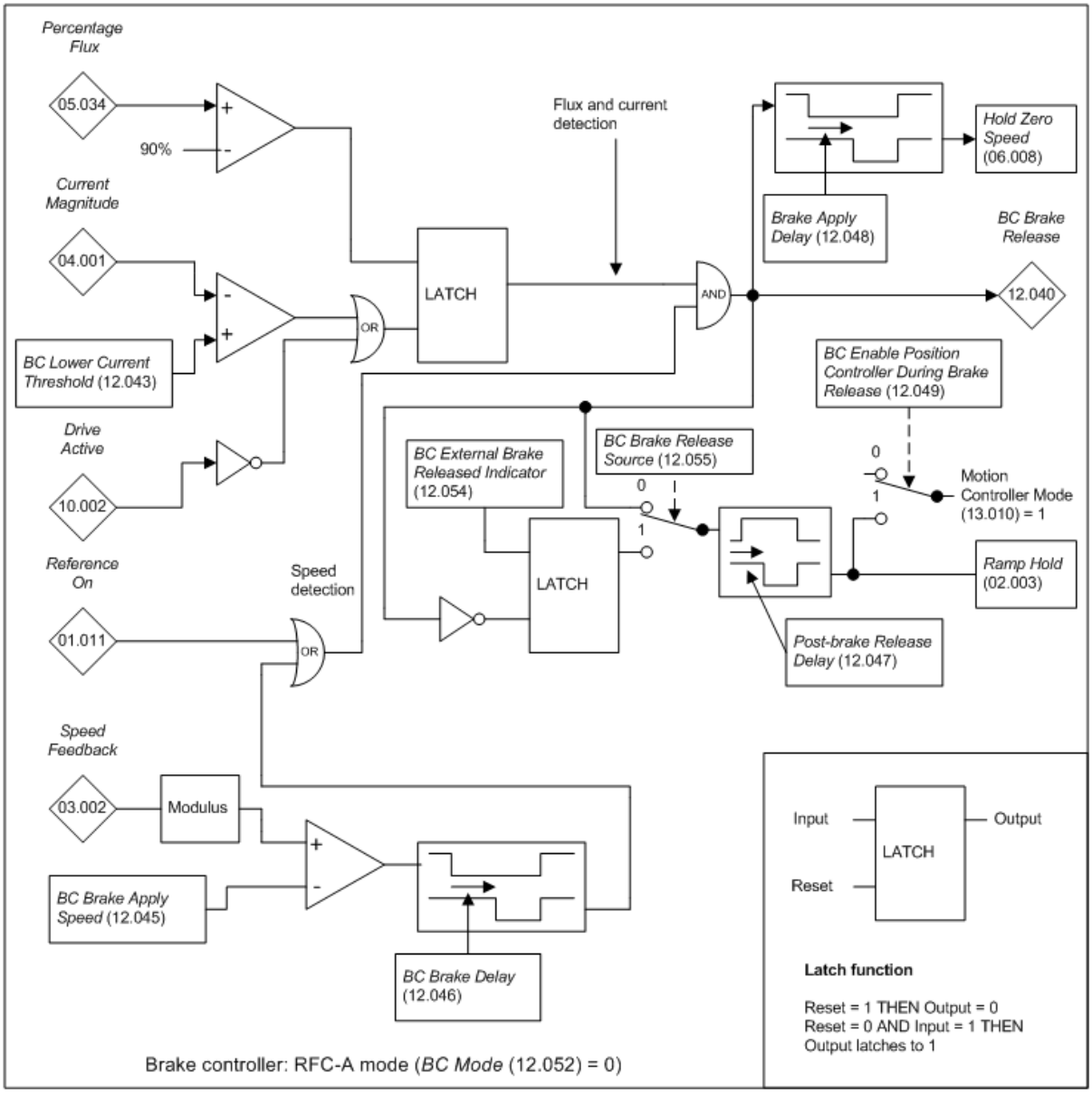
Brake controller: RFC-S mode



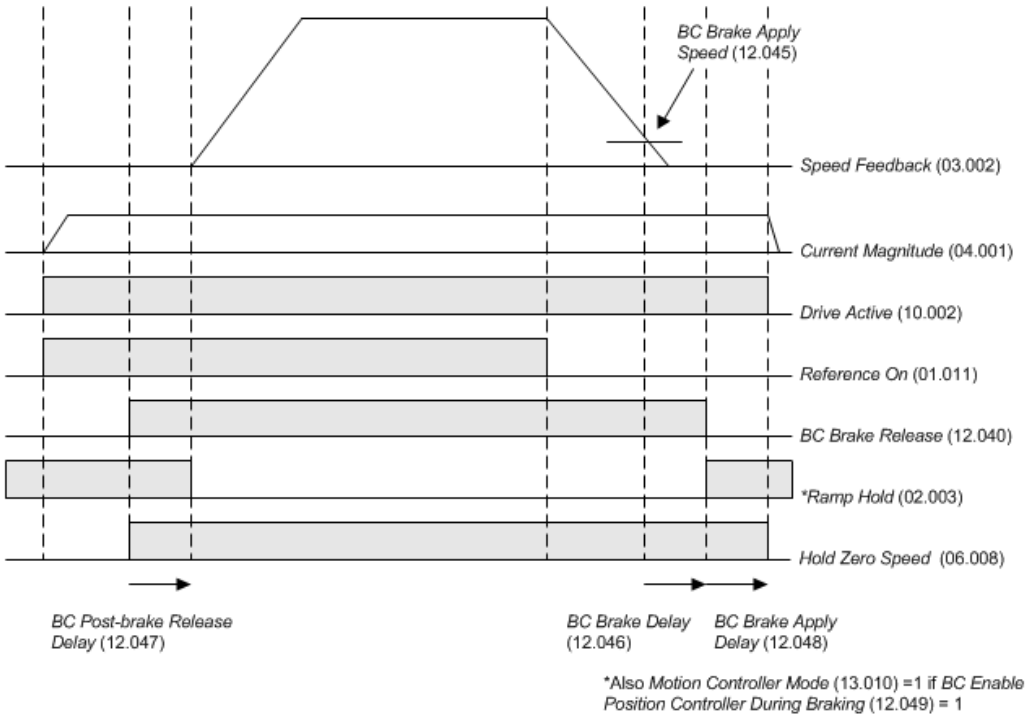
Brake controller sequence: RFC-A mode (*BC Mode* (12.052) = 0) and RFC-S mode

4.3.2 RFC-A closed-loop induction motor brake controller

This section indicates the logic and timing diagrams for the brake controller in RFC-A mode for closed-loop induction motors to help illustrate the functionality and timing.



Brake controller: RFC-A mode (*BC Mode* (12.052) = 0)



Brake controller sequence: RFC-A mode (*BC Mode* (12.052) = 0) and RFC-S mode

Nidec
All for dreams

Connect with us

www.controltechniques.com

www.kbelectronics.com

Nidec Control Techniques Limited. Registered Office: The Gro, Newtown, Powys SY16 3BE.

Registered in England and Wales. Company Reg. No. 01236886.

CONTROL TECHNIQUES

KB