User Guide

Control Techniques NC Axis Interface Library for TwinCAT PLCs

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 10
1.1 OVERVIEW 11
1.2 MINIMUM REQUIREMENTS 14
2 GETTING STARTED 15
2.1 BEST PRACTICE FOR INSTALLATION 15
2.2 INSTALL CONNECT 15
2.3 ADDING ESI FILES / DEVICE DESCRIPTION FILES TO TWINCAT 17
2.4 INITIAL TWINCAT PROJECT SETUP 18
2.5 SETUP AN ETHERCAT AOE CONNECTION FROM THE PLC TO THE DRIVE 25
2.6 CONFIGURE THE DRIVE USING CONNECT 29
2.7 COMMISSION THE DRIVE AND MOTOR USING CONNECT 39
2.8 INSTALLING THE CT NC AXIS INTERFACE LIBRARY TO THE TWINCAT LIBRARY REPOSITORY 57
2.9 ADDING A PLC PROJECT 59
2.10 ADDING THE CT NC AXIS INTERFACE LIBRARY TO THE TWINCAT REFERENCES 61
2.11 ADDING A CALL OF THE NC AXIS INTERFACE LIBRARY 65
2.12 REMOVING UNUSED FUNCTION BLOCK PARAMETERS 70
2.13 NC AXIS CONFIGURATION 72
  2.13.1 Creating PDO mappings for Touch Probe functionality 72
  2.13.2 Linking an axis to the dirve 77
  2.13.3 Axis configuration 79
  2.13.4 Moving the axis using TwinCAT test moves 84
2.14 SETTING THE PROJECT TASK RATES 87
2.15 VERIFY THE AXIS 90
2.16 MOVING THE AXIS WITH THE CT NC AXIS INTERFACE LIBRARY 92
3 HOW TO GUIDES 93
3.1 HOW TO SETUP AXIS UNIT SCALING AND RESOLUTION 93
  3.1.1 How to set the scaling using the startup list in TwinCAT 95
  3.1.2 How to scale between NC axis speed and rpm 98
3.2 HOW TO USE A STARTUP LIST IN TWINCAT 99
  3.2.1 How to import a startup list .xml file from Connect into TwinCAT 99
  3.2.2 How to add a single CAN object to the TwinCAT Startup List & Cyclic comms loss action 103
3.3 HOW TO CONFIGURE PDO MAPPINGS 106
  3.3.1 How to convert parameter numbers to CANopen object references 106
    3.3.1.1 Short drive parameter references 106
    3.3.1.2 Long option parameter references 107
    3.3.1.3 How to find the parameter data type 108
  3.3.2 How pass data between drive parameters and variables using PDOs 110

3.3.3 How to route the drive P2 encoder interface to an encoder axis 115
3.4 HOW TO USE A VIRTUAL MASTER AXIS TO CONTROL TWO SLAVE DRIVES USING THE EGB INTERFACE 121
3.5 HOW TO UPDATE THE CT NC AXIS INTERFACE LIBRARY 132
3.6 HOW TO SET UP AN ETHERCAT EOE CONNECTION FROM THE PLC TO THE DRIVE 134
3.7 HOW TO CREATE A CONNECT PROJECT FOR THE APPLICATION (EOE) 149
3.8 HOW TO CREATE A CONNECT PROJECT FOR THE APPLICATION (ETHERNET & RS485) 153
3.9 HOW TO UPGRADE THE ETHERCAT INTERFACE FIRMWARE 156
3.9.1 How to upgrade the EtherCAT interface firmware using Connect 156
3.9.2 How to use FoE to update the EtherCAT interface firmware 159
3.10 HOW AND WHEN TO TUNE THE POSITION LOOP 168
3.11 HOW AND WHEN TO USE INERTIA COMPENSATION 171
3.12 HOW TO SET UP A DUAL LOOP SYSTEM 173
3.13 HOW TO VIEW CAN OBJECTS 179
3.13.1 MC_Power Disabled and Drive Disabled 180
3.13.2 MC_Power Disabled and Drive Enabled 180
3.13.3 MC_Power Enabled and Drive Enabled 180
3.14 HOW TO INSTALL TWINCAT RUNTIME AND CONFIGURE ETHERNET PORTS FOR ETHERCAT USE ON A BECKHOFF IPC 181
3.15 HOW TO REMOVE "???" FROM LD DIAGRAM POU INPUTS AND OUTPUTS 186
3.16 HOW TO APPLY VELOCITY FEEDFORWARD TO AN NC AXIS 188
4 CT NC AXIS INTERFACE LIBRARY FUNCTIONALITY 193
4.1 MOTION INTERFACE 193
4.1.1 FB Diagrams 193
4.1.2 Common FB Functionality 197
4.1.2.1 General 197
4.1.2.2 Limits 209
4.1.2.3 Common Motion 218
4.1.2.4 Motion Stop 225
4.1.2.5 Jog 228
4.1.2.6 Home 231
4.1.2.7 Position 252
4.1.2.8 EGB 259
4.1.3 Timing Diagrams 266
4.1.3.1 CMD_Home 266
4.1.3.2 CMD_DriveOn 266
4.1.3.3 CMD_JogForward and CMD_JogReverse 267
4.1.3.4 CMD_Position 267
4.1.3.5 CMD_EGB 268
4.1.3.6 Behaviour when more than one reference is selected 268

4.2 MASTER REFERENCE FUNCTIONALITY 269
  4.2.1 MC_CT_AxisFilter 269
    4.2.1.1 FB Diagram 270
    4.2.1.2 Inputs / Outputs 270
    4.2.1.3 Inputs 272
    4.2.1.4 Outputs 275
  4.2.2 MC_CT_AxisSelector 277
    4.2.2.1 FB Diagram 278
    4.2.2.2 Inputs / Outputs 278
    4.2.2.3 Inputs 281
    4.2.2.4 Outputs 285
4.3 DRIVE FUNCTIONALITY 287
  4.3.1 CT_DriveStatusWord 287
    4.3.1.1 FB Diagram 287
    4.3.1.2 Inputs / Outputs 288
    4.3.1.3 Inputs 288
    4.3.1.4 Outputs 289
  4.3.2 CT_AdditionalStatusWord 295
    4.3.2.1 FB Diagram 295
    4.3.2.2 Inputs / Outputs 296
    4.3.2.3 Inputs 296
    4.3.2.4 Outputs 297
  4.3.3 CT_DigitalInputWord 302
    4.3.3.1 FB Diagram 302
    4.3.3.2 Inputs / Outputs 303
    4.3.3.3 Inputs 303
    4.3.3.4 Outputs 304
  4.3.4 CT_DigitalOutputWord 310
    4.3.4.1 FB Diagram 310
    4.3.4.2 Inputs / Outputs 310
    4.3.4.3 Inputs 311
    4.3.4.4 Outputs 311
4.4 FILTER FUNCTIONALITY 316
  4.4.1 MC_CT_WeightedAverageFilter_DINT 316
    4.4.1.1 FB Diagram 316
    4.4.1.2 Inputs 316
    4.4.1.3 Outputs 317
  4.4.2 MC_CT_WeightedAverageFilter_LREAL 318
    4.4.2.1 FB Diagram 318
    4.4.2.2 Inputs 318

4.4.2.3 Outputs 319
4.4.3 MC_CT_WeightedAveragePositionFilter_DINT 320
4.4.3.1 FB Diagram 320
4.4.3.2 Inputs 320
4.4.3.3 Outputs 322
4.4.4 MC_CT_WeightedAveragePositionFilter_LREAL 323
4.4.4.1 FB Diagram 323
4.4.4.2 Inputs 323
4.4.4.3 Outputs 325
4.4.5 MC_CT_WindowingAverageFilterDINT 326
4.4.5.1 FB Diagram 326
4.4.5.2 Inputs 326
4.4.5.3 Outputs 327
4.4.6 MC_CT_WindowingAverageFilter_LREAL 328
4.4.6.1 FB Diagram 328
4.4.6.2 Inputs 328
4.4.6.3 Outputs 329
4.4.7 MC_CT_WindowingAveragePositionFilter_DINT 330
4.4.7.1 FB Diagram 330
4.4.7.2 Inputs 330
4.4.7.3 Outputs 332
4.4.8 MC_CT_WindowingAveragePositionFilter_LREAL 333
4.4.8.1 FB Diagram 333
4.4.8.2 Inputs 333
4.4.8.3 Outputs 335
4.5 SDO READ / WRITE 336
4.5.1 CT_SDO_ReadCoE 336
4.5.1.1 FB Diagram 336
4.5.1.2 Inputs / Outputs 336
4.5.1.3 Inputs 337
4.5.1.4 Outputs 338
4.5.2 CT_SDO_ReadCoE2 340
4.5.2.1 FB Diagram 340
4.5.2.2 Inputs / Outputs 340
4.5.2.3 Inputs 341
4.5.2.4 Outputs 342
4.5.3 CT_SDO_ReadParameter 345
4.5.3.1 FB Diagram 345
4.5.3.2 Inputs / Outputs 345
4.5.3.3 Inputs 346

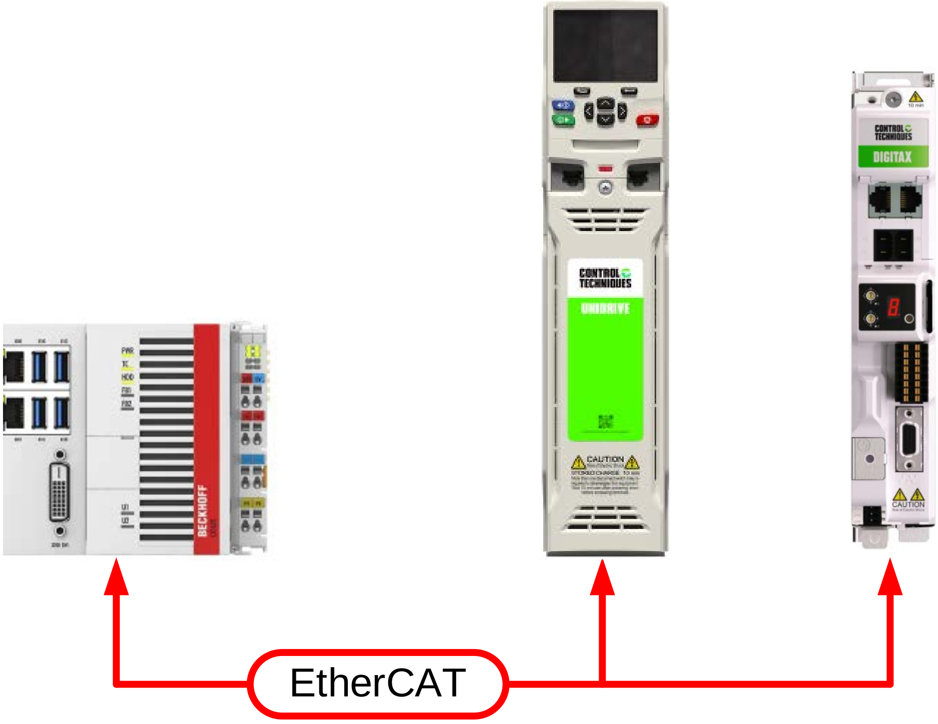
4.5.3.4 Outputs 348
4.5.4 CT_SDO_ReadParameter2 350
4.5.4.1 FB Diagram 350
4.5.4.2 Inputs / Outputs 350
4.5.4.3 Inputs 351
4.5.4.4 Outputs 353
4.5.5 CT_SDO_WriteCoE 356
4.5.5.1 FB Diagram 356
4.5.5.2 Inputs / Outputs 356
4.5.5.3 Inputs 357
4.5.5.4 Outputs 358
4.5.6 CT_SDO_WriteCoE2 360
4.5.6.1 FB Diagram 360
4.5.6.2 Inputs / Outputs 360
4.5.6.3 Inputs 361
4.5.6.4 Outputs 363
4.5.7 CT_SDO_WriteParameter 365
4.5.7.1 FB Diagram 365
4.5.7.2 Inputs / Outputs 365
4.5.7.3 Inputs 366
4.5.7.4 Outputs 368
4.5.8 CT_SDO_WriteParameter2 370
4.5.8.1 FB Diagram 370
4.5.8.2 Inputs / Outputs 370
4.5.8.3 Inputs 371
4.5.8.4 Outputs 374
4.6 BASIC NC AXIS FUNCTION BLOCKS 376
4.6.1 MC_CT_GearIn 376
4.6.1.1 FB Diagram 376
4.6.1.2 Inputs / Outputs 377
4.6.1.3 Inputs 378
4.6.1.4 Outputs 381
4.6.2 MC_CT_Halt 383
4.6.2.1 FB Diagram 383
4.6.2.2 Inputs / Outputs 383
4.6.2.3 Inputs 384
4.6.2.4 Outputs 385
4.6.3 MC_CT_Jog 387
4.6.3.1 FB Diagram 387
4.6.3.2 Inputs / Outputs 387

4.6.3.3 Inputs 388
4.6.3.4 Outputs 390
4.6.4 MC_CT_MoveAbsolute 392
4.6.4.1 FB Diagram 392
4.6.4.2 Inputs / Outputs 392
4.6.4.3 Inputs 393
4.6.4.4 Outputs 395
4.6.5 MC_CT_MoveRelative 397
4.6.5.1 FB Diagram 397
4.6.5.2 Inputs / Outputs 397
4.6.5.3 Inputs 398
4.6.5.4 Outputs 400
4.6.6 MC_CT_MoveSuperimposed 402
4.6.6.1 FB Diagram 402
4.6.6.2 Inputs / Outputs 403
4.6.6.3 Inputs 404
4.6.6.4 Outputs 407
4.6.7 MC_CT_MoveSuperimposedAbsolute 410
4.6.7.1 FB Diagram 410
4.6.7.2 Inputs / Outputs 411
4.6.7.3 Inputs 412
4.6.7.4 Outputs 415
4.6.8 MC_CT_MoveVelocity 418
4.6.8.1 FB Diagram 418
4.6.8.2 Inputs / Outputs 418
4.6.8.3 Inputs 419
4.6.8.4 Outputs 422
4.6.9 MC_CT_SoftwareLimits 424
4.6.9.1 FB Diagram 424
4.6.9.2 Inputs / Outputs 424
4.6.9.3 Inputs 425
4.6.9.4 Outputs 427
4.6.10 MC_CT_MoveModulo 429
4.6.10.1 FB Diagram 429
4.6.10.2 Inputs / Outputs 430
4.6.10.3 Inputs 430
4.6.10.4 Outputs 434
4.6.11 MC_CT_Homing 436
4.6.11.1 FB Diagram 437
4.6.11.2 Inputs / Outputs 437

4.6.11.3 Inputs....438
4.6.11.4 Outputs....457
5 ENUMERATIONS....459
5.1 MC_CT_ACTIVEALARMS....459
5.2 MC_CT_AXISMODE....460
5.3 MC_CT_AXISSTATES....460
5.4 MC_DIRECTION....462
5.5 MC_CT_DRIVEDIGITALINPUTS....462
5.6 MC_CT_EGB_MODES....463
5.7 MC_CT_ERRORCATEGORIES....464
5.8 MC_CT_ERRORLIST....465
5.9 MC_CT_ERRORMODES....466
5.10 MC_CT_FB_TYPES....466
5.11 MC_CT_FB_FILTERTYPES....467
5.12 MC_CT_FREEZECAPTURESOURCE....468
5.13 MC_CT_HOMINGMODES....469
5.14 MC_CT_NC_ERROR....474
5.15 MC_CT_POSITIONMOTIONTYPES....474
5.16 MC_CT_SDO_ERRORLIST....476
5.17 MC_CT_STOPONFOLLOWINGERRORMODES....481
5.18 MC_CT_SWITCHLOGIC....481
5.19 MC_CT_SYNCMODE....482
5.20 MC_CT_TRIPSTRINGS....483
5.21 MC_CT_WRITEMODES....494
6 ADDITIONAL INFORMATION....495
6.1 TERMINOLOGY....495
6.2 MECHANICAL BRAKE CONTROLLER LOGIC....496
6.2.1 RFC-S closed-loop permanent-magnet motor brake controller....496
6.2.2 RFC-A closed-loop induction motor brake controller....498

1 Introduction

The Control Techniques “CT NC Axis Interface Library” is a standardised set of motion control interfaces from a PLC / IPC / Controller to a Control Techniques drive. This library is designed for Unidrive M and Digitax HD products in running with closed loop feedback.



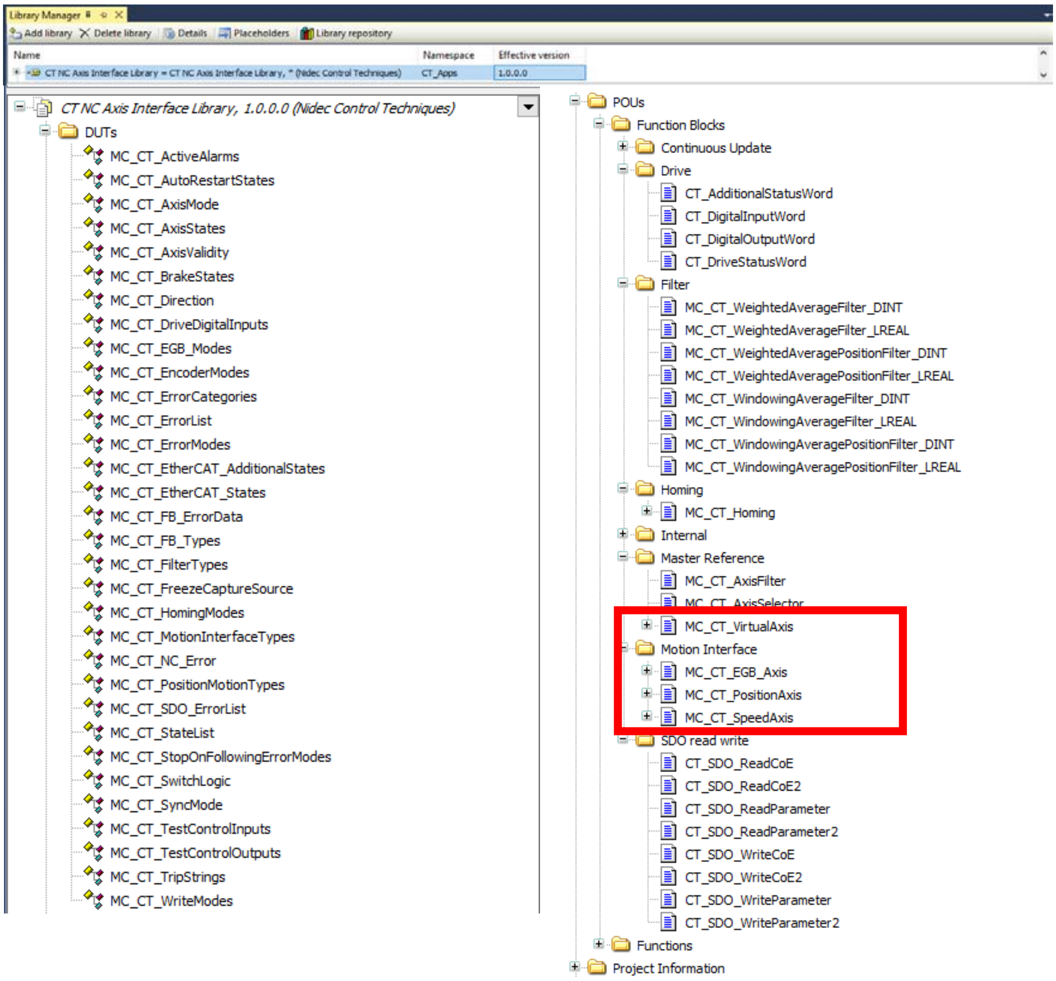
The CT NC Axis Interface Library is designed to work with TwinCAT 3.1 – it is recommended that the latest version of TwinCAT XAE Shell is installed from the BECKHOFF website:

[TwinCAT 3.1 Build 4026 | Beckhoff United Kingdom]

1.1 Overview

Once the CT NC Axis Interface Library has been referenced in a TwinCAT project, the 4 main “MC_CT” function blocks that are capable of invoking motion become available to use:

- MC_CT_EGB_Axis
- MC_CT_PositionAxis
- MC_CT_SpeedAxis
- MC_CT_VirtualAxis



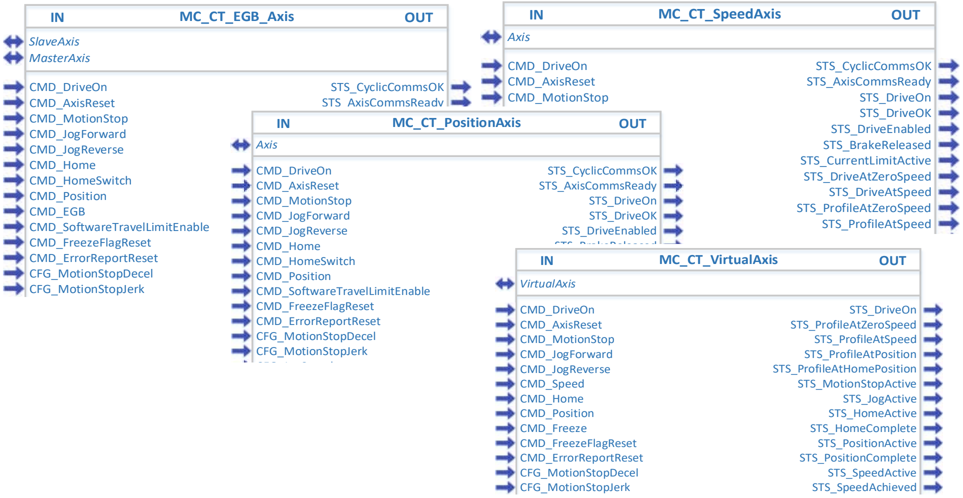
Library Manager

Add library | Delete library | Details | Placeholders | Library repository

Name	Namespace	Effective version
CT NC Axis Interface Library = CT NC Axis Interface Library, * (Nidec Control Techniques)	CT_Apps	1.0.0.0

- *CT NC Axis Interface Library, 1.0.0.0 (Nidec Control Techniques)*
  - DUTs
    - MC_CT_ActiveAlarms
    - MC_CT_AutoRestartStates
    - MC_CT_AxisMode
    - MC_CT_AxisStates
    - MC_CT_AxisValidity
    - MC_CT_BrakeStates
    - MC_CT_Direction
    - MC_CT_DriveDigitalInputs
    - MC_CT_EGB_Modes
    - MC_CT_EncoderModes
    - MC_CT_ErrorCategories
    - MC_CT_ErrorList
    - MC_CT_ErrorModes
    - MC_CT_EtherCAT_AdditionalStates
    - MC_CT_EtherCAT_States
    - MC_CT_FB_ErrorData
    - MC_CT_FB_Types
    - MC_CT_FilterTypes
    - MC_CT_FreezeCaptureSource
    - MC_CT_HomingModes
    - MC_CT_MotionInterfaceTypes
    - MC_CT_NC_Error
    - MC_CT_PositionMotionTypes
    - MC_CT_SDO_ErrorList
    - MC_CT_StateList
    - MC_CT_StopOnFollowingErrorModes
    - MC_CT_SwitchLogic
    - MC_CT_SyncMode
    - MC_CT_TestControlInputs
    - MC_CT_TestControlOutputs
    - MC_CT_TripStrings
    - MC_CT_WriteModes

- POUs
  - Function Blocks
    - Continuous Update
    - Drive
      - CT_AdditionalStatusWord
      - CT_DigitalInputWord
      - CT_DigitalOutputWord
      - CT_DriveStatusWord
    - Filter
      - MC_CT_WeightedAverageFilter_DINT
      - MC_CT_WeightedAverageFilter_LREAL
      - MC_CT_WeightedAveragePositionFilter_DINT
      - MC_CT_WeightedAveragePositionFilter_LREAL
      - MC_CT_WindowingAverageFilter_DINT
      - MC_CT_WindowingAverageFilter_LREAL
      - MC_CT_WindowingAveragePositionFilter_DINT
      - MC_CT_WindowingAveragePositionFilter_LREAL
    - Homing
      - MC_CT_Homing
    - Internal
    - Master Reference
      - MC_CT_AxisFilter
      - MC_CT_AxisSelector
      - MC_CT_VirtualAxis
    - Motion Interface
      - MC_CT_EGB_Axis
      - MC_CT_PositionAxis
      - MC_CT_SpeedAxis
    - SDO read write
      - CT_SDO_ReadCoE
      - CT_SDO_ReadCoE2
      - CT_SDO_ReadParameter
      - CT_SDO_ReadParameter2
      - CT_SDO_WriteCoE
      - CT_SDO_WriteCoE2
      - CT_SDO_WriteParameter
      - CT_SDO_WriteParameter2
  - Functions
- Project Information

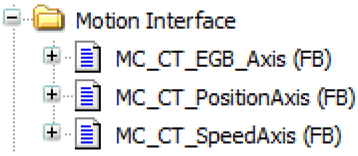


This guide walks the user through the necessary steps to get an axis working with each of the 4 main "MC_CT" function blocks including drive commissioning, PLC setup and a detailed description of each component that makes up the CT NC Axis Interface Library.

Section **2 Getting Started** is designed to be a step by step guide consisting of multiple sections. Once all sections have been followed, the user is left with working axis which is controlled by the CT NC Axis Interface Library.

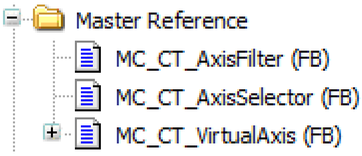
The CT NC Axis Interface Library includes the following items:

- The main Motion Interface function blocks used for controlling a physical drive axis.



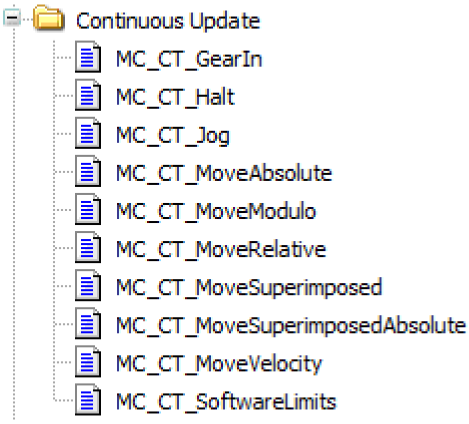
Motion Interface
- MC_CT_EGB_Axis (FB)
- MC_CT_PositionAxis (FB)
- MC_CT_SpeedAxis (FB)

- Master Reference functionality – these function blocks are intended for use in an Electronic Line Shaft system, or when a distributed master encoder reference is consumed by a physical drive as a reference e.g. an encoder axis with a master / slave function block such as MC_GearIn or MC_CT_EGB_Axis.



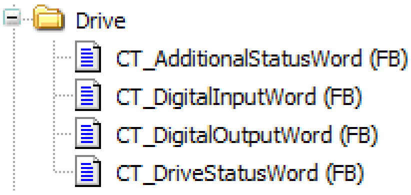
Master Reference
- MC_CT_AxisFilter (FB)
- MC_CT_AxisSelector (FB)
- MC_CT_VirtualAxis (FB)

- Continuous Update Motion Control function blocks – these function blocks replicate those included with the NC Axis library but with continuous update functionality, where any change to any of the rates or references take instant effect.



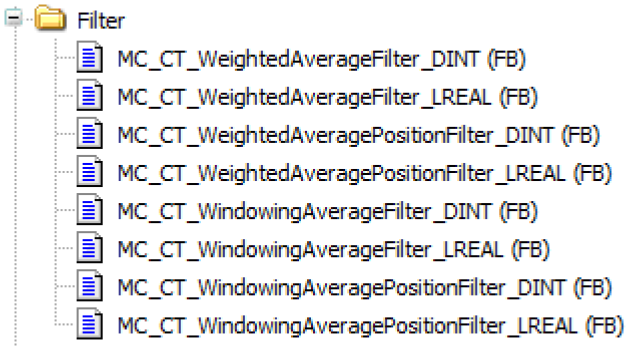
Continuous Update
- MC_CT_GearIn
- MC_CT_Halt
- MC_CT_Jog
- MC_CT_MoveAbsolute
- MC_CT_MoveModulo
- MC_CT_MoveRelative
- MC_CT_MoveSuperimposed
- MC_CT_MoveSuperimposedAbsolute
- MC_CT_MoveVelocity
- MC_CT_SoftwareLimits

- Drive status function blocks – these function blocks are used for monitoring drive IO and status words.



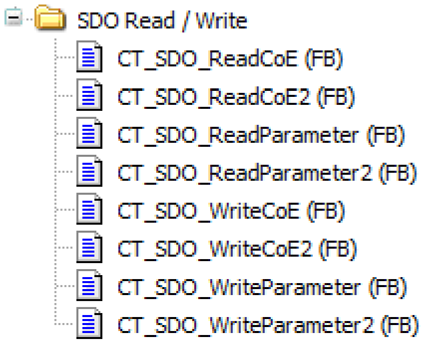
Drive
- CT_AdditionalStatusWord (FB)
- CT_DigitalInputWord (FB)
- CT_DigitalOutputWord (FB)
- CT_DriveStatusWord (FB)

- Filter functionality – these function blocks are used for filtering LREAL and DINT values such as those generated by an encoder axis.



Filter
  - MC_CT_WeightedAverageFilter_DINT (FB)
  - MC_CT_WeightedAverageFilter_LREAL (FB)
  - MC_CT_WeightedAveragePositionFilter_DINT (FB)
  - MC_CT_WeightedAveragePositionFilter_LREAL (FB)
  - MC_CT_WindowingAverageFilter_DINT (FB)
  - MC_CT_WindowingAverageFilter_LREAL (FB)
  - MC_CT_WindowingAveragePositionFilter_DINT (FB)
  - MC_CT_WindowingAveragePositionFilter_LREAL (FB)

- SDO Read / SDO Write – these function blocks provide an interface for reading and writing both CiA402 CANopen objects and drive parameters.



SDO Read / Write
  - CT_SDO_ReadCoE (FB)
  - CT_SDO_ReadCoE2 (FB)
  - CT_SDO_ReadParameter (FB)
  - CT_SDO_ReadParameter2 (FB)
  - CT_SDO_WriteCoE (FB)
  - CT_SDO_WriteCoE2 (FB)
  - CT_SDO_WriteParameter (FB)
  - CT_SDO_WriteParameter2 (FB)

1.2 Minimum requirements

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

- An installation of TwinCAT 3 on the programming PC.
- An installation of TwinCAT XAR Runtime on the TwinCAT PLC / IPC / Controller.
- The TwinCAT PLC / IPC / Controller must support EtherCAT communications.
- A suitable EtherCAT connection between the drive and TwinCAT PLC / IPC / Controller.
- An Ethernet connection between the TwinCAT 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**.
- An ESI file / device description .xml file is required for the Control Techniques drive.

For guidance on how to install the TwinCAT runtime on an TwinCAT PLC / IPC / Controller see section **3.14 How to install TwinCAT Runtime and configure Ethernet ports for EtherCAT use on a BECKHOFF IPC**.

2 Getting Started

This section describes how to commission the drive and implementing the CT NC Axis Interface Library in TwinCAT. 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 Best practice for installation

When installing any software including installers from Control Techniques, it is best practice to avoid installing the software directly from a USB memory device and instead copy the files to the target PC and install from that copy.

This is because most virus scanning software installed on a target PC have strict rules inhibiting the functionality of executable files from a USB memory device to prevent virus infection, that will also prevent a genuine installer file from operating correctly.

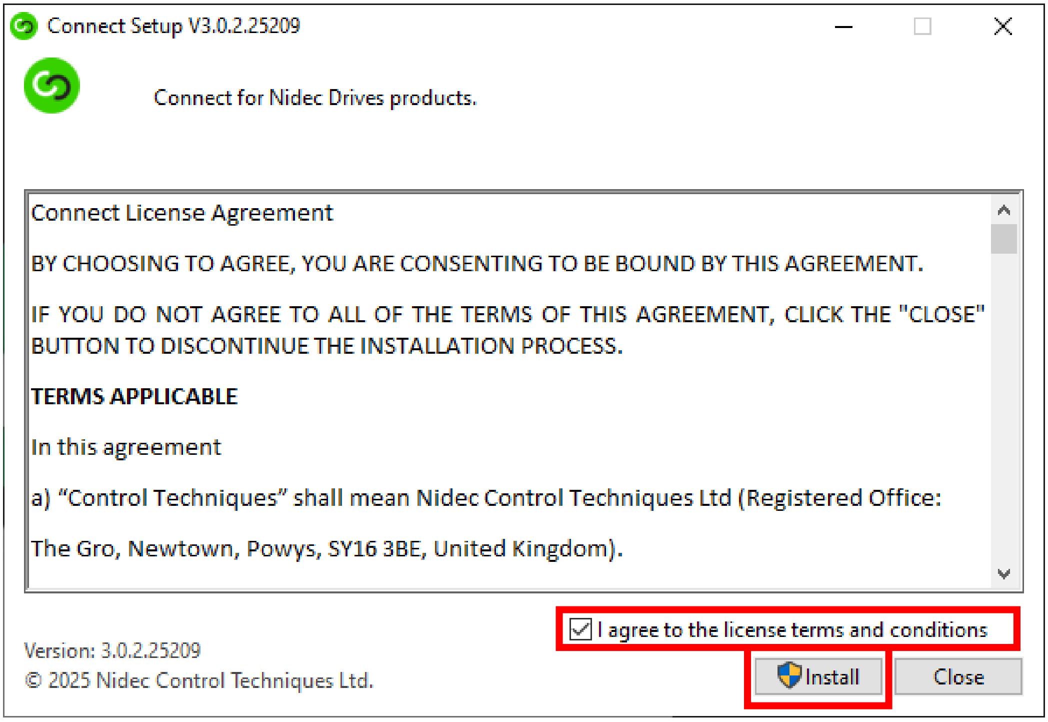
Uninstall any previous version of Connect before installing Connect V03.00.00.

2.2 Install Connect

The user must install a minimum of version Connect V03.00.00. The software file may be obtained from your local Control Techniques Drive Centre / Distributor or the Control Techniques website.

1. Double click on the Connect Installer file.

2. The “Connect Setup” window will be displayed. Once the license agreement has been read and accepted, tick the “I agree to the license terms and conditions” check box and click “install” to begin the installation.



Connect for Nidec Drives products.

Connect License Agreement

BY CHOOSING TO AGREE, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT.

IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE "CLOSE" BUTTON TO DISCONTINUE THE INSTALLATION PROCESS.

TERMS APPLICABLE

In this agreement

a) “Control Techniques” shall mean Nidec Control Techniques Ltd (Registered Office:

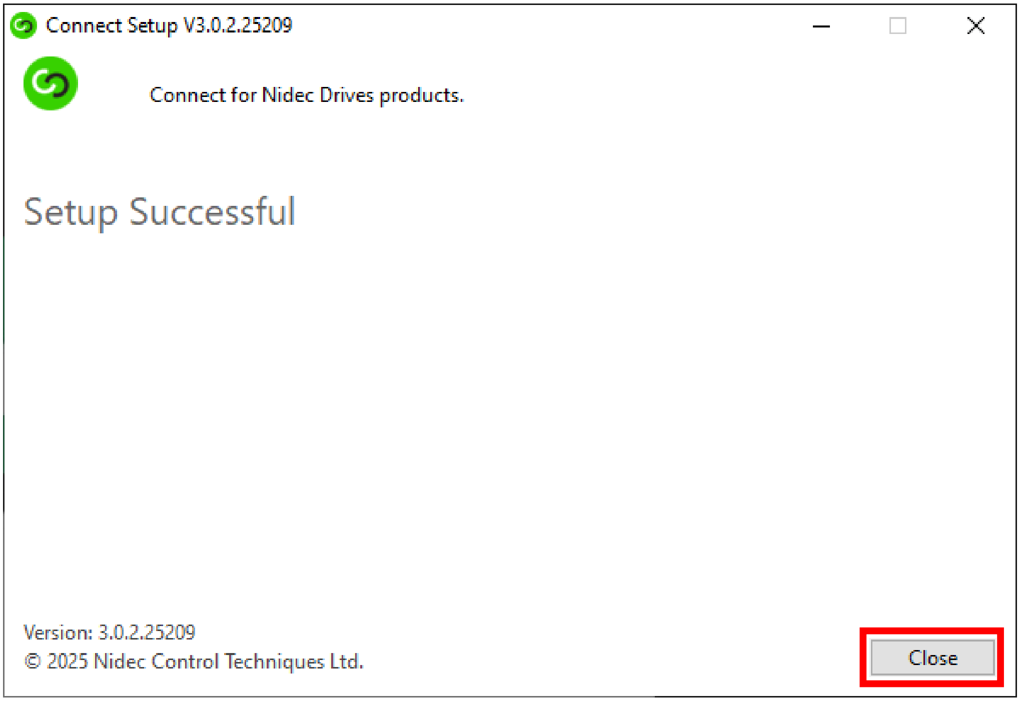
The Gro, Newtown, Powys, SY16 3BE, United Kingdom).

☑ I agree to the license terms and conditions

Version: 3.0.2.25209

© 2025 Nidec Control Techniques Ltd.

3. Once the installation is complete, the "Setup Successful" message will be displayed. Click "Close" to exit the installer. Connect will now be installed and ready to use.



Connect for Nidec Drives products.

Setup Successful

Version: 3.0.2.25209

© 2025 Nidec Control Techniques Ltd.

4. The rest of section 2 describes how to commission the drive via an AoE (ADS over EtherCAT) connection. This method is recommended for applications where EtherCAT is the only communications protocol supported by the drive, such as with Digitax HD M753.

   Alternatively guidance for applications using EoE (Ethernet over EtherCAT) can be found in section **3.7 How to create a Connect project for the application (EoE)**.

   For applications which support standard Ethernet or RS485 comms go to section **3.8 How to Create a Connect project for the application (Ethernet & RS485).**

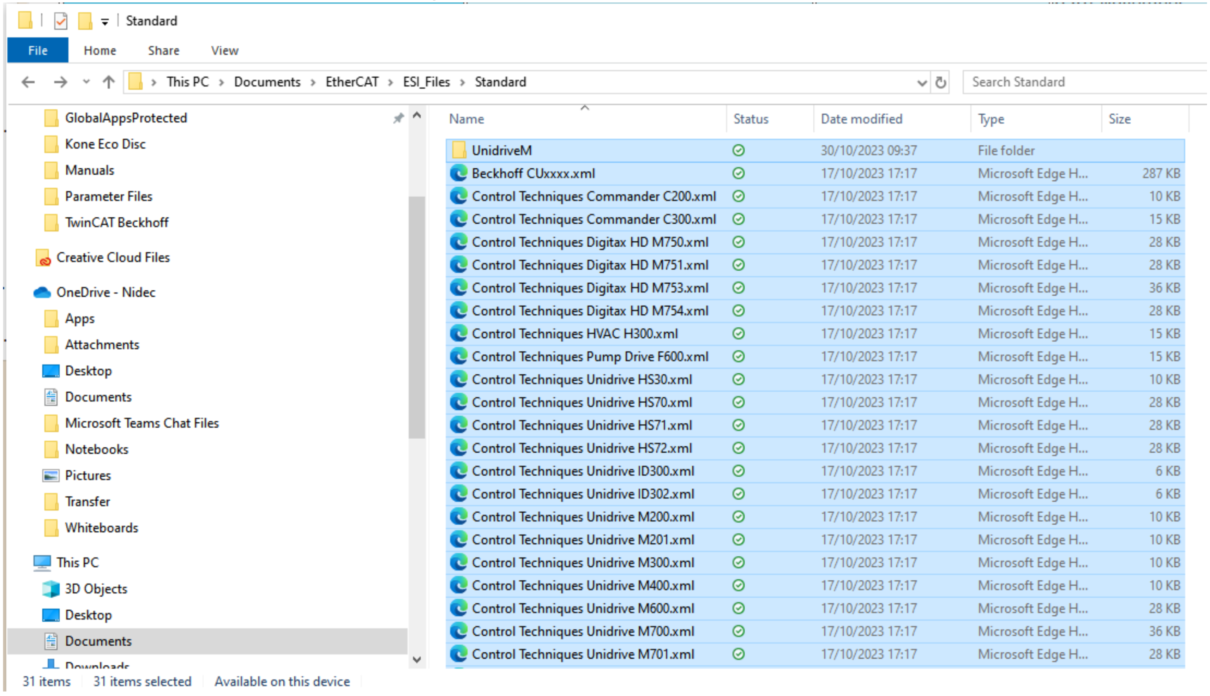
2.3 Adding ESI files / device description files to TwinCAT

Device description files (ESI files) are required by TwinCAT to allow a given PLC / IPC / Controller to properly recognise a Control Techniques drive. In preparation for this section, acquire the appropriate set of ESI / device description .xml files from Support Suite or a Control Techniques Drive Centre or Distributor.

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. Make sure that TwinCAT is not running. Close the program if it is.

3. Extract the contents of the ESI file zip.



GlobalAppsProtected
Kone Eco Disc
Manuals
Parameter Files
TwinCAT Beckhoff
Creative Cloud Files
OneDrive - Nidec
Apps
Attachments
Desktop
Documents
Microsoft Teams Chat Files
Notebooks
Pictures
Transfer
Whiteboards
This PC
3D Objects
Desktop
Documents
Downloads

Name	Status	Date modified	Type	Size
UnidriveM	✓	30/10/2023 09:37	File folder	
Beckhoff CUxxxx.xml	✓	17/10/2023 17:17	Microsoft Edge H...	287 KB
Control Techniques Commander C200.xml	✓	17/10/2023 17:17	Microsoft Edge H...	10 KB
Control Techniques Commander C300.xml	✓	17/10/2023 17:17	Microsoft Edge H...	15 KB
Control Techniques Digitax HD M750.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB
Control Techniques Digitax HD M751.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB
Control Techniques Digitax HD M753.xml	✓	17/10/2023 17:17	Microsoft Edge H...	36 KB
Control Techniques Digitax HD M754.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB
Control Techniques HVAC H300.xml	✓	17/10/2023 17:17	Microsoft Edge H...	15 KB
Control Techniques Pump Drive F600.xml	✓	17/10/2023 17:17	Microsoft Edge H...	15 KB
Control Techniques Unidrive HS30.xml	✓	17/10/2023 17:17	Microsoft Edge H...	10 KB
Control Techniques Unidrive HS70.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB
Control Techniques Unidrive HS71.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB
Control Techniques Unidrive HS72.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB
Control Techniques Unidrive ID300.xml	✓	17/10/2023 17:17	Microsoft Edge H...	6 KB
Control Techniques Unidrive ID302.xml	✓	17/10/2023 17:17	Microsoft Edge H...	6 KB
Control Techniques Unidrive M200.xml	✓	17/10/2023 17:17	Microsoft Edge H...	10 KB
Control Techniques Unidrive M201.xml	✓	17/10/2023 17:17	Microsoft Edge H...	10 KB
Control Techniques Unidrive M300.xml	✓	17/10/2023 17:17	Microsoft Edge H...	10 KB
Control Techniques Unidrive M400.xml	✓	17/10/2023 17:17	Microsoft Edge H...	10 KB
Control Techniques Unidrive M600.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB
Control Techniques Unidrive M700.xml	✓	17/10/2023 17:17	Microsoft Edge H...	36 KB
Control Techniques Unidrive M701.xml	✓	17/10/2023 17:17	Microsoft Edge H...	28 KB

31 items 31 items selected Available on this device

Copy the contents of the zip to “C:\TwinCAT\3.1\Config\Io\EtherCAT” or for newer installations of TwinCAT “C:\Program Files (x86)\Beckhoff\TwinCAT\3.1\Config\Io\EtherCAT”

4. When TwinCAT is next run, it will recognise Control Techniques drives after a “Scan”.

HINT: If the ESI files don’t match the firmware in the EtherCAT option, after scanning the network and adding the EtherCAT master in the “I/O” section, the drives will appear as “Box 1”, “Box 2” etc.

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

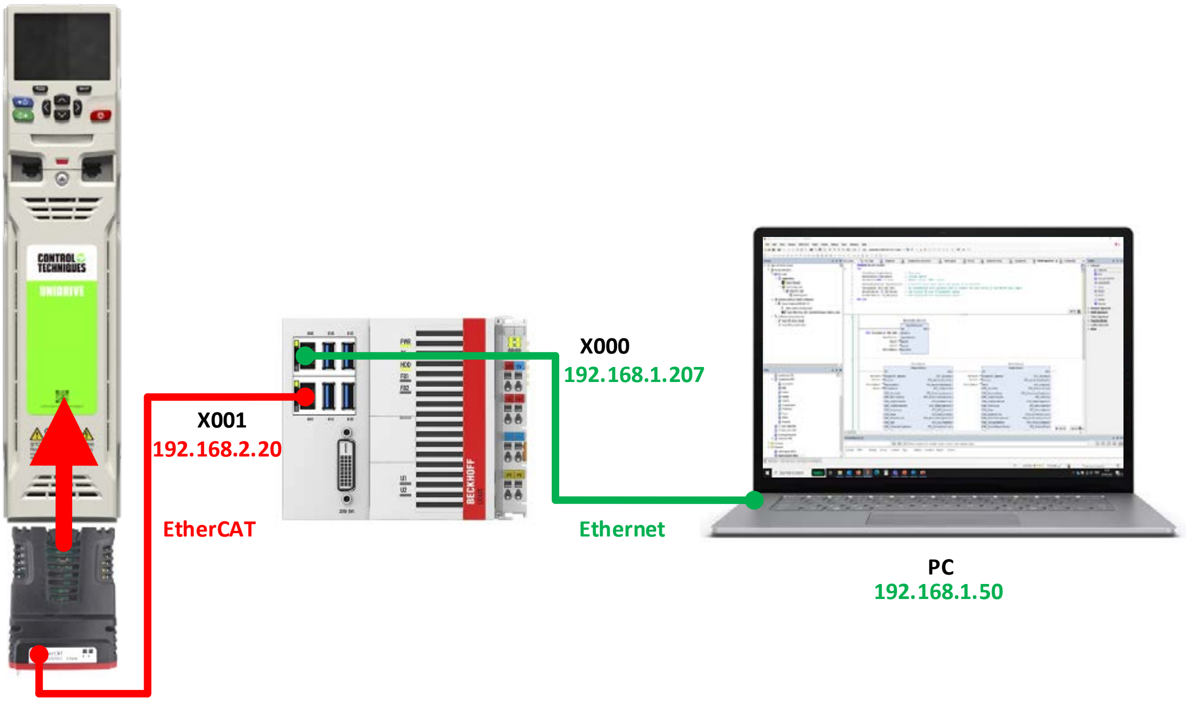
2.4 Initial TwinCAT project setup

To allow Connect to communicate to the slave drives in the EtherCAT network, an AoE (ADS over EtherCAT) connection must be configured.

Before continuing, an Ethernet cable should be connected between the Ethernet port of the programming PC and the Ethernet port of the PLC / IPC / Controller (X000 in the diagram below).

A second Ethernet cable should be connected between the EtherCAT port of the PLC / IPC / Controller (X001 in the diagram below) and the EtherCAT post of the Control Techniques Drive / SI-EtherCAT module in use.

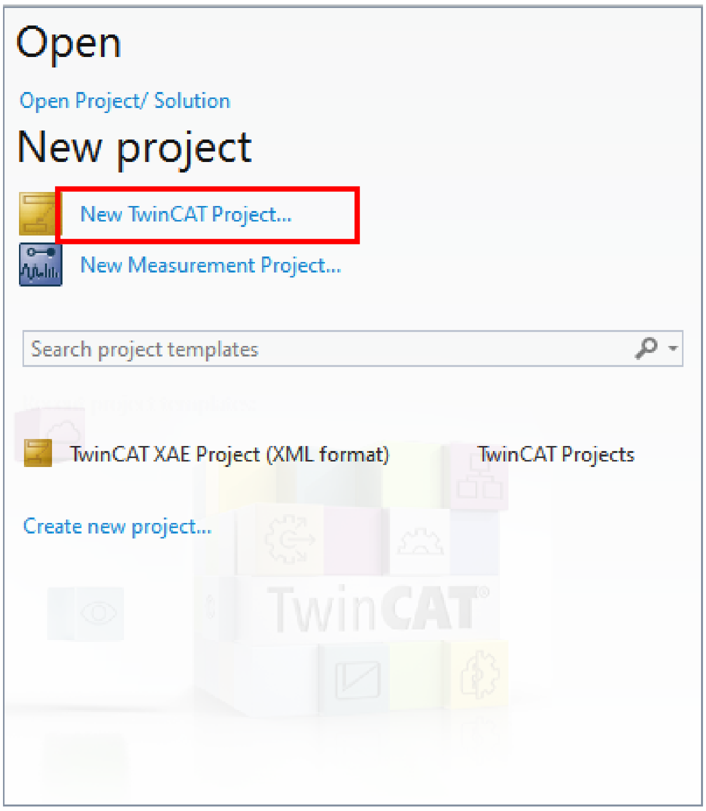
It is assumed that the Ethernet addresses of the programming PC and PLC / IPC / Controller are in the same range e.g.192.168.1.x.



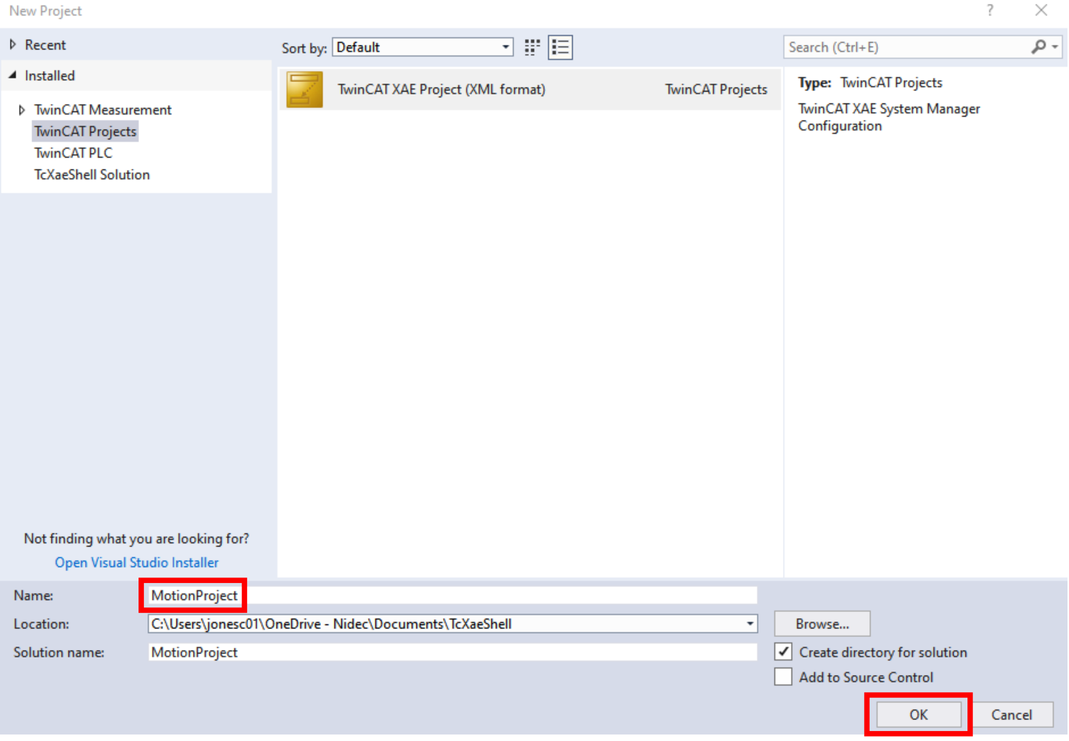
This section shows the first steps required to get connected to a Beckhoff PLC/IPC.

1. Start TwinCAT. For the purposes of these examples the TwinCAT XAE Shell will be used to run the TwinCAT IDE. This is available even if the PC doesn't have Visual Studio installed.

2. Select “New TwinCAT Project”:



3. Give the project a sensible name e.g. “MotionProject” and then click “OK”.



Type: TwinCAT Projects

TwinCAT XAE System Manager Configuration

Name: MotionProject

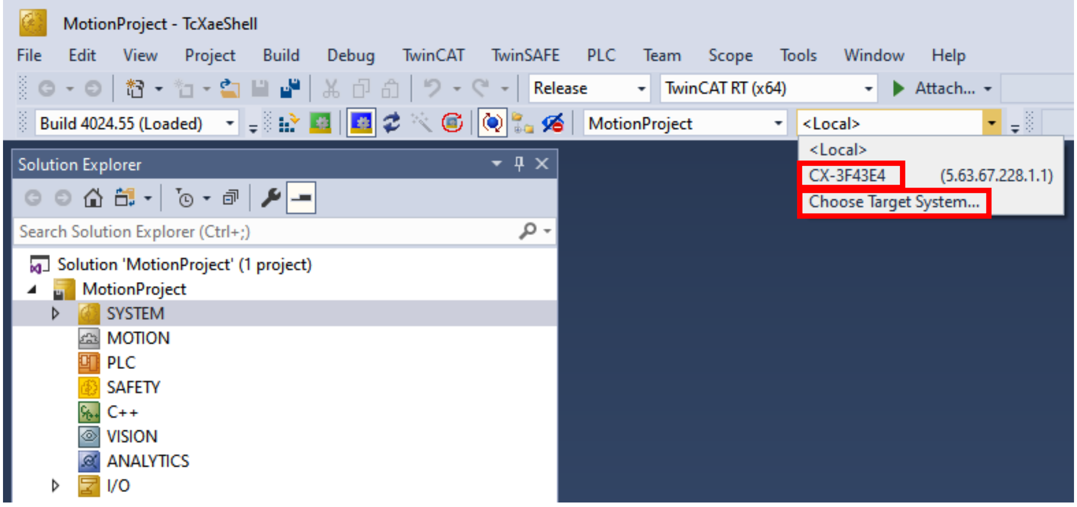
Location: C:\Users\jonesc01\OneDrive - Nidec\Documents\TcXaeShell

Solution name: MotionProject

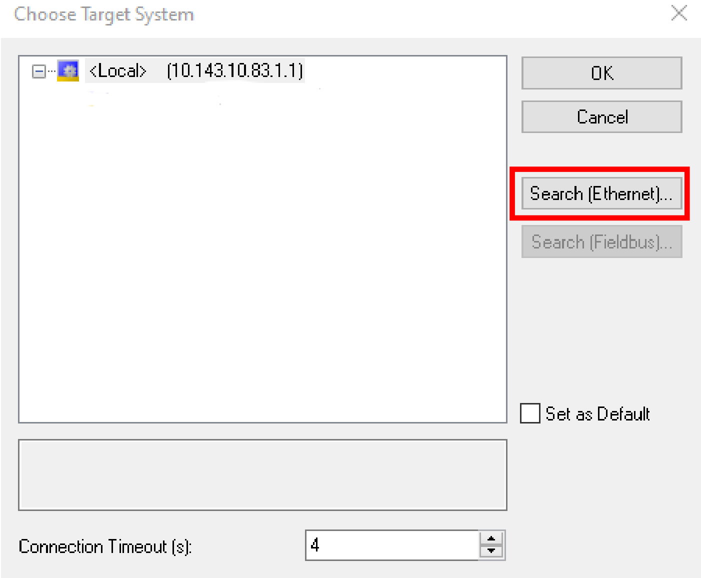
☑ Create directory for solution

☐ Add to Source Control

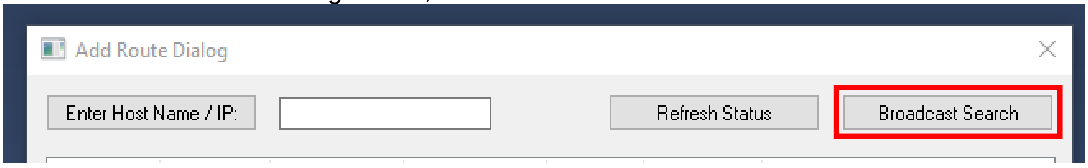
4. Select the target PLC. If your PLC has previously been connected to a TwinCAT project, then the target name and the ADS Address of the PLC is shown.



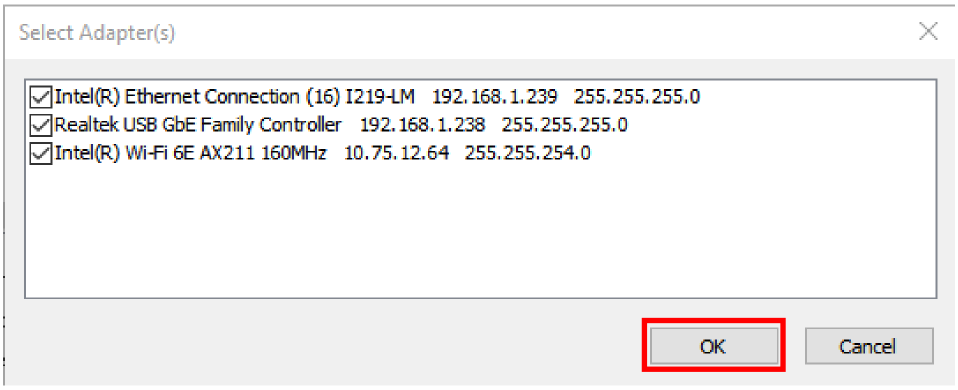
If the PLC hasn't previously been connected to a TwinCAT project, then select "Choose Target System…" and then "Search (Ethernet)" from the "Choose Target System" dialog.



5. When the “Add Route Dialog” starts, click “Broadcast search”.

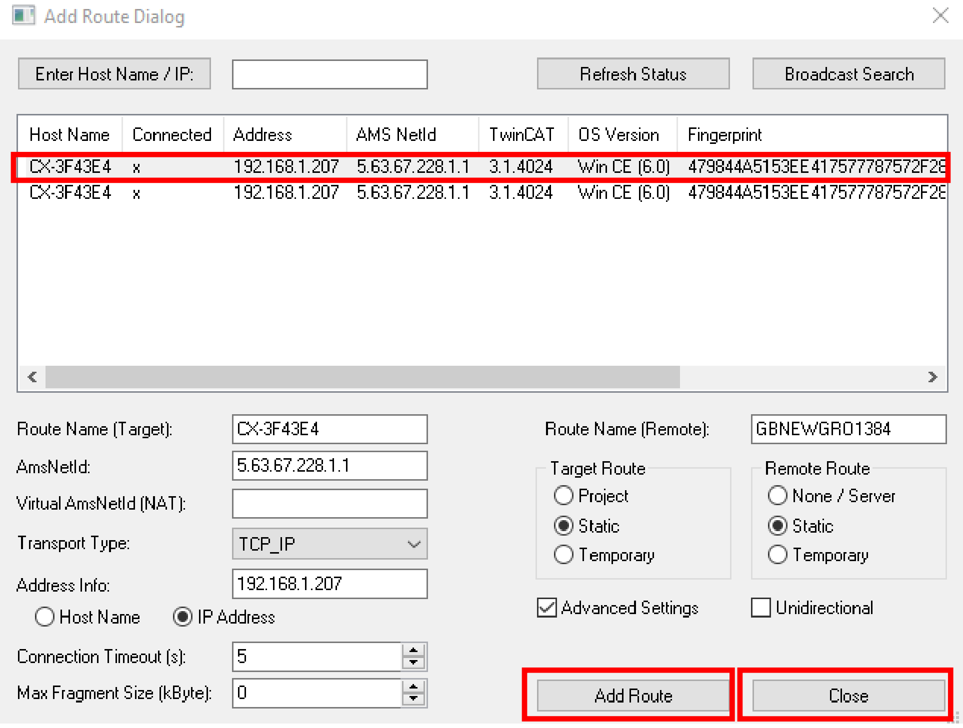


and then select the adapter to search for compatible devices:



☑ Intel(R) Ethernet Connection (16) I219-LM 192.168.1.239 255.255.255.0
☑ Realtek USB GbE Family Controller 192.168.1.238 255.255.255.0
☑ Intel(R) Wi-Fi 6E AX211 160MHz 10.75.12.64 255.255.254.0

6. Select the PLC to connect to and then click “Add Route” and then “Close”.



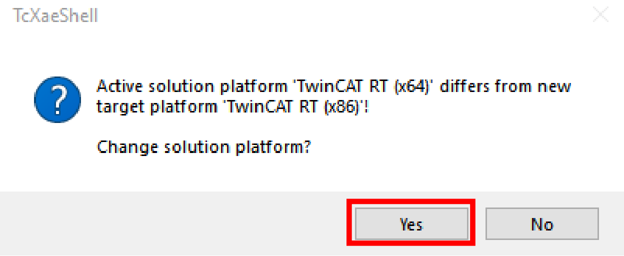
Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Fingerprint
CX-3F43E4	x	192.168.1.207	5.63.67.228.1.1	3.1.4024	Win CE (6.0)	479844A5153EE417577787572F2
CX-3F43E4	x	192.168.1.207	5.63.67.228.1.1	3.1.4024	Win CE (6.0)	479844A5153EE417577787572F2

Route Name (Target): CX-3F43E4
AmsNetId: 5.63.67.228.1.1
Virtual AmsNetId (NAT):
Transport Type: TCP_IP
Address Info: 192.168.1.207
○ Host Name ● IP Address
Connection Timeout (s): 5
Max Fragment Size (kByte): 0

Route Name (Remote): GBNEWGR01384
Target Route: ○ Project ● Static ○ Temporary
Remote Route: ○ None / Server ● Static ○ Temporary
☑ Advanced Settings ☐ Unidirectional

HINT: If there is an Administrator password challenge, the default password is “1”

Click “Yes” if the message below appears:



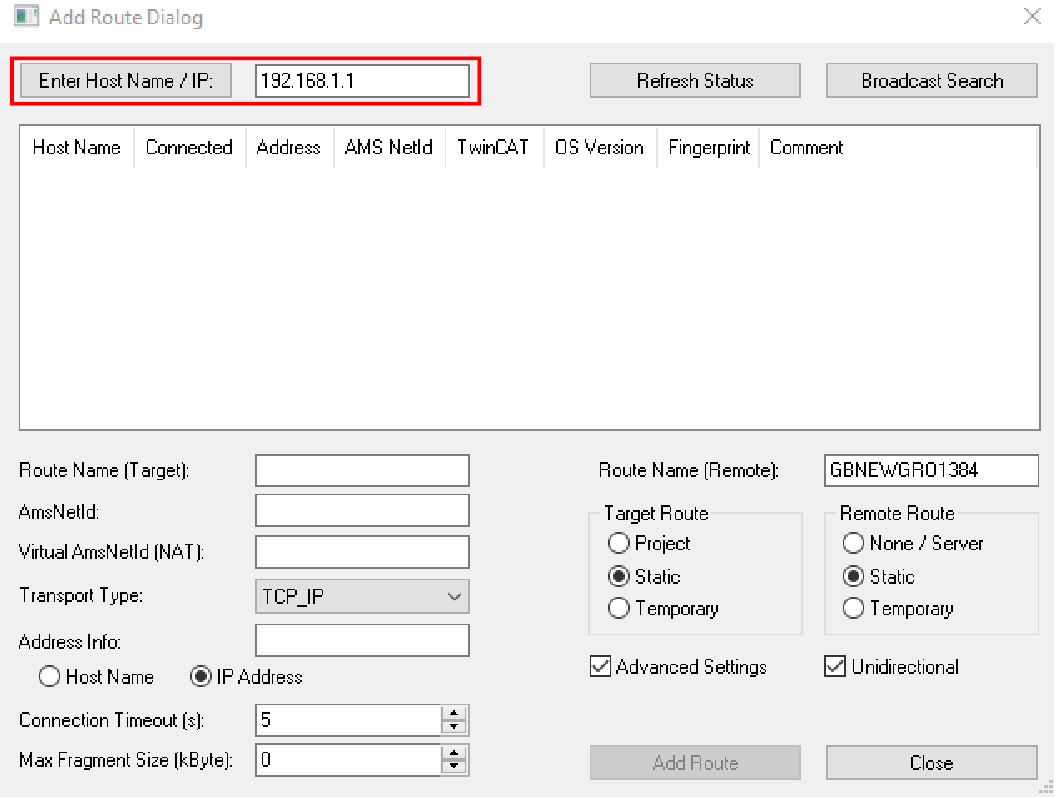
TcXaeShell

Active solution platform 'TwinCAT RT (x64)' differs from new target platform 'TwinCAT RT (x86)'!

Change solution platform?

HINT: For brand new IPCs it is likely that the runtime hasn’t been installed, and that the EtherCAT port hasn’t been assigned. For more information, see section **3.14 How to install TwinCAT Runtime and configure Ethernet ports for EtherCAT use on a BECKHOFF IPC.**

If this doesn’t allow connection, Ping the IP address for the device or connect via Remote Desktop to ensure the connection to the PC is good. Then enter the IP address into the “Enter Host Name / IP:” input box and press enter.



Add Route Dialog

Enter Host Name / IP: 192.168.1.1

Route Name (Target):
AmsNetId:
Virtual AmsNetId (NAT):
Transport Type: TCP_IP
Address Info:
○ Host Name ● IP Address
Connection Timeout (s): 5
Max Fragment Size (kByte): 0

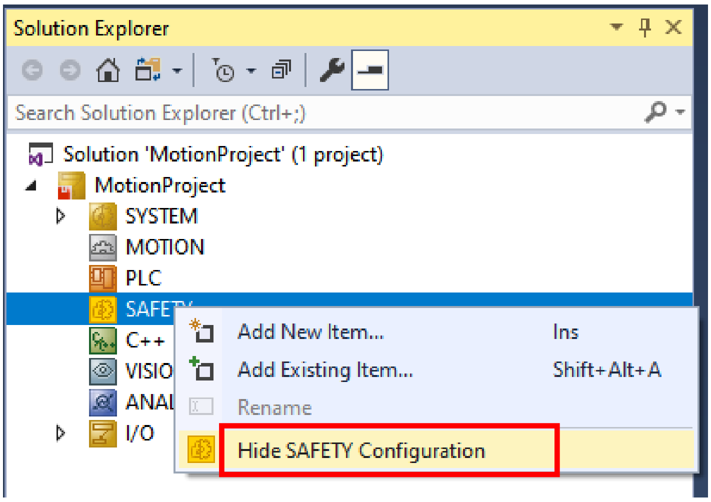
Route Name (Remote): GBNEWGR01384

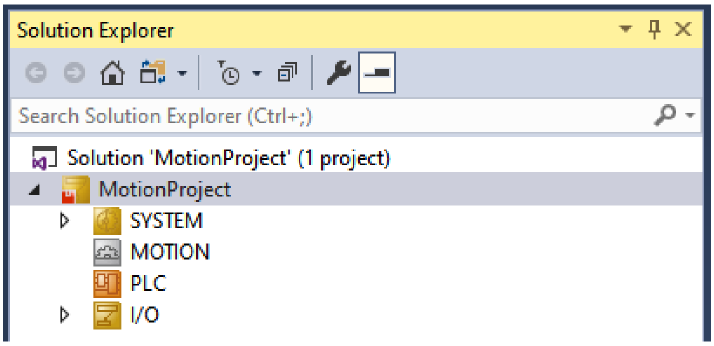
Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None / Server ● Static ○ Temporary

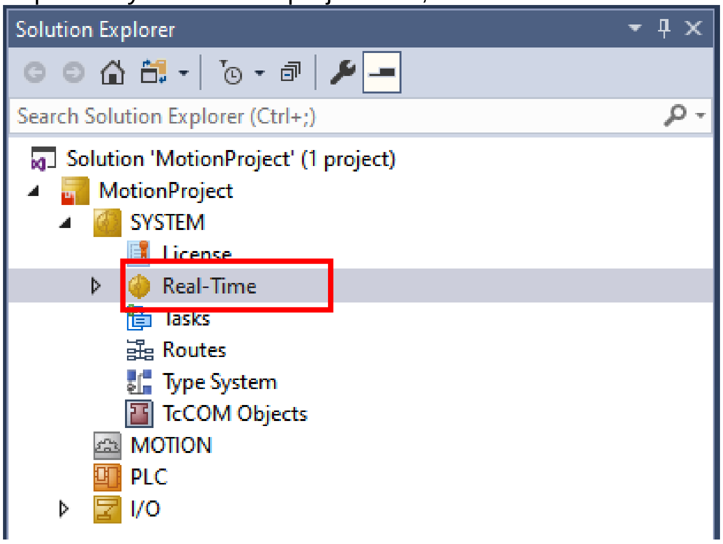
☑ Advanced Settings ☑ Unidirectional

7. An optional step is to clean up any unrequired items from the project tree e.g. this example doesn't require Safety, C++, or Vision etc. To do this right click on the item and select "Hide xxxx".

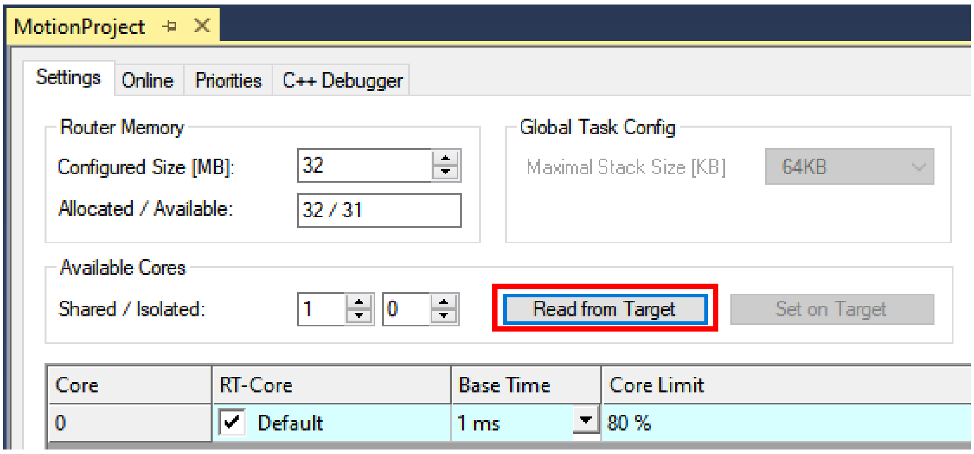




8. Expand "System" in the project tree, the double click on "Real-Time":



When the project settings tab appears, Select "Read from Target" to update the information on the number of processing cores.



Configured Size [MB]: 32

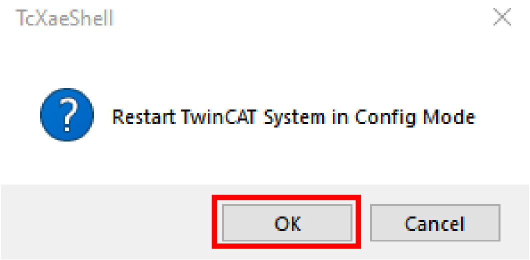
Allocated / Available: 32 / 31

Maximal Stack Size [KB] 64KB

Shared / Isolated: 1 0

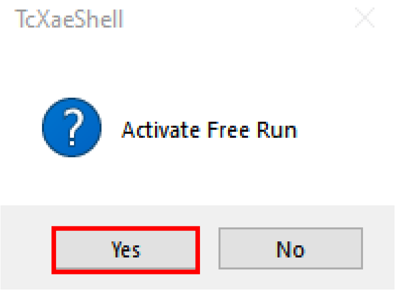
Core	RT-Core	Base Time	Core Limit
0	☑ Default	1 ms	80 %

9. Make sure the TwinCAT is in Config Mode by clicking on in the tool bar and then click “OK” when prompted with the “Restart TwinCAT system in Config Mode” message:



Restart TwinCAT System in Config Mode

Failure to do this may result in the “Scan” feature being greyed out. Click “Yes” to the “Activate Free Run” pop-up.

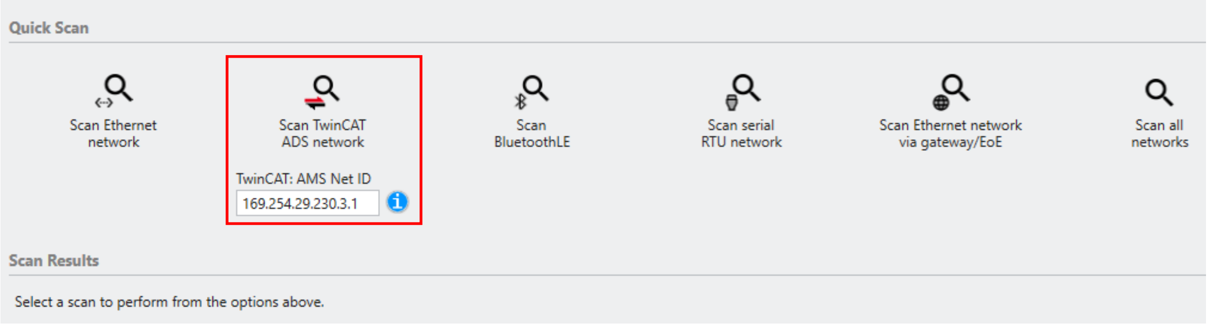


Activate Free Run

2.5 Setup an EtherCAT AoE connection from the PLC to the drive

An EtherCAT AoE (ADS over EtherCAT) connection allows communications between the drive(s) and the Connect PC tool in systems where the only communications connection to the drives is EtherCAT.

In the latest version of Connect it is possible to access the drive using the EtherCAT masters AMS network ID:

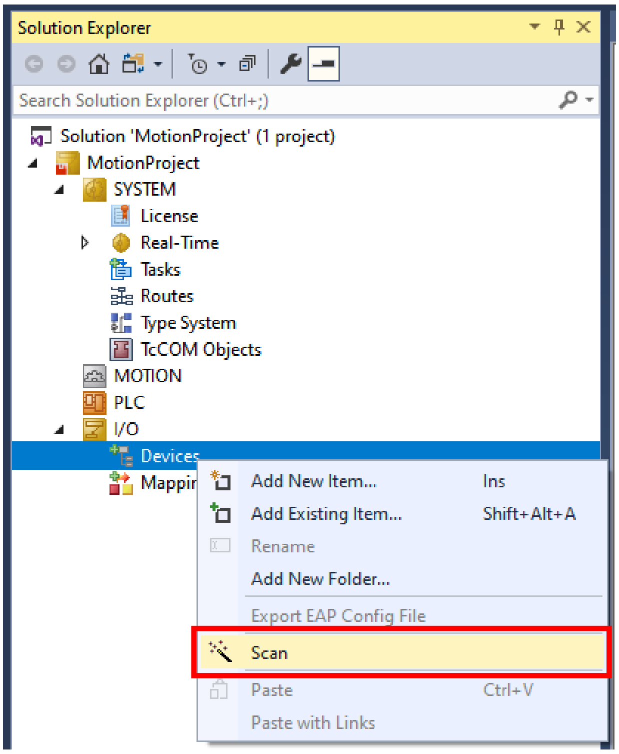


TwinCAT: AMS Net ID: 169.254.29.230.3.1

Select a scan to perform from the options above.

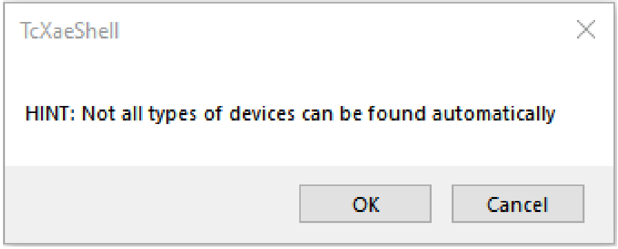
Expand the “I/O” category in the project tree, and then right click on “Devices” and select

1. Expand the “I/O” category in the project tree, and then right click on “Devices” and select “Scan”:



HINT: If “Scan” is greyed out, put TwinCAT in “Config” mode by pressing the following button .

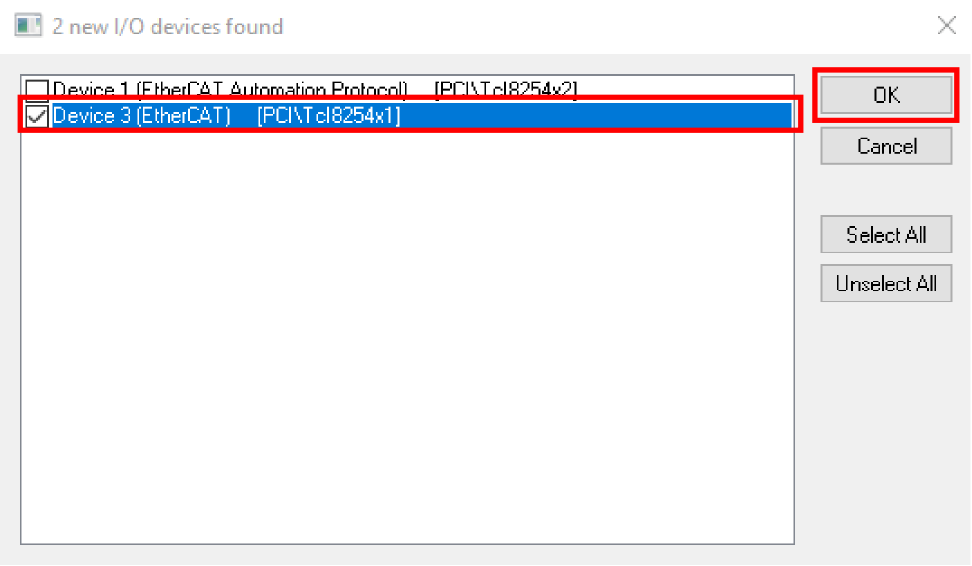
2. Click OK on the message below:



TcXaeShell

HINT: Not all types of devices can be found automatically

3. Select the “(EtherCAT)” device and then click “OK”.

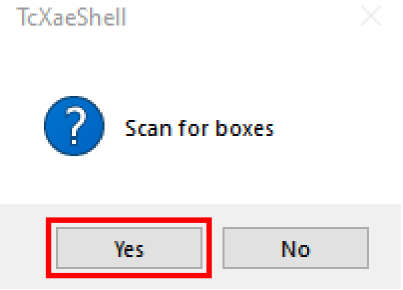


2 new I/O devices found

☐ Device 1 (EtherCAT Automation Protocol)   [PCI\TcI8254x2]
☑ Device 3 (EtherCAT)   [PCI\TcI8254x1]

4. Make sure the Control Techniques drive has a suitable Ethernet cable connecting the drives EtherCAT port to the PLC’s EtherCAT port. To use the PLCopen functionality the drive must be either a Unidrive M or a Digitax HD.

5. Next, TwinCAT will prompt the user to scan the network for boxes. In this context, boxes means EtherCAT devices connected to the PLC’s EtherCAT port. Click “Yes” to accept this.

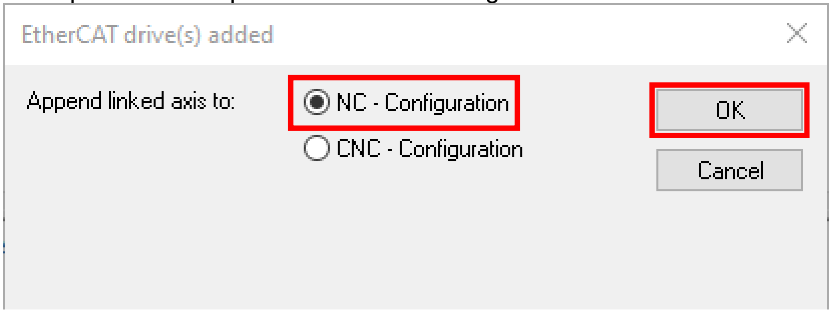


TcXaeShell

Scan for boxes

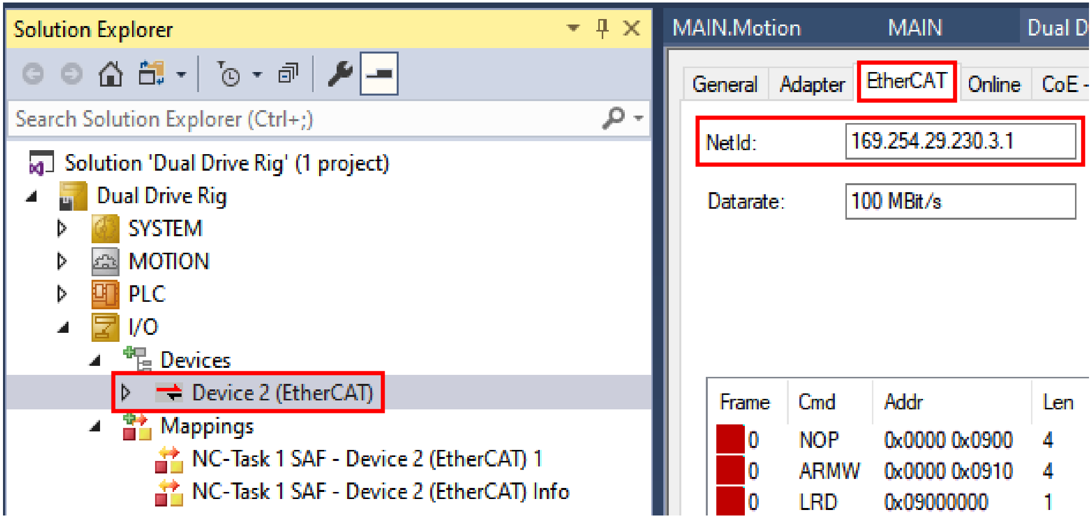
HINT: If there are any issues connecting to the target drives, the ESI files in TwinCAT may not match the firmware in the EtherCAT interface on the drive. The ESI file **must** match the firmware of the EtherCAT module. Section **3.9 How to upgrade the EtherCAT interface firmware** describes how to check the current SI-EtherCAT firmware version and how to perform an upgrade if necessary.

6. When the axis is found the “EtherCAT drive(s) added” dialog is shown. This guide is for PLCopen which requires the “NC – Configuration”. Select this and then click “OK”.



Append linked axis to:
● NC - Configuration
○ CNC - Configuration

7. When the scan process is completed the PLC/IPC ADS Net ID, (used by Connect), may be located. Double click on the EtherCAT network Master in the Solution Explorer, e.g. “Device 2”, and then select the “EtherCAT” tab. The AMS network ID is given next to “NetId:”.



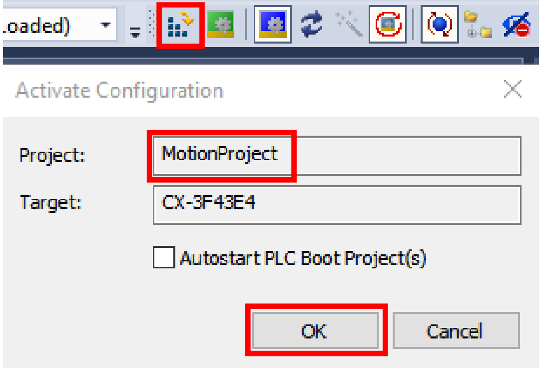
NetId: 169.254.29.230.3.1

Datarate: 100 MBit/s

Frame	Cmd	Addr	Len
0	NOP	0x0000 0x0900	4
0	ARMW	0x0000 0x0910	4
0	LRD	0x09000000	1

HINT: The EtherCAT port device number varies between different PLCs, depending on the network ports available. In general, the port required will be described as Device (n) (EtherCAT).

8. Click the “Activate Configuration” button and then click “OK” if the “Activate Configuration” dialog appears:

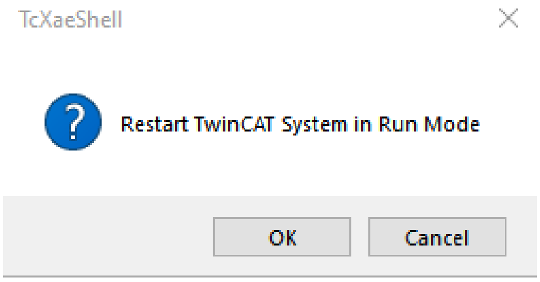


Project: MotionProject

Target: CX-3F43E4

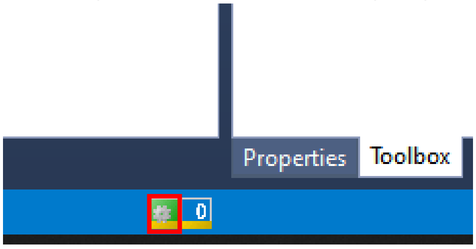
☐ Autostart PLC Boot Project(s)

Click “OK” on the “Restart TwinCAT in Run Mode” dialog:

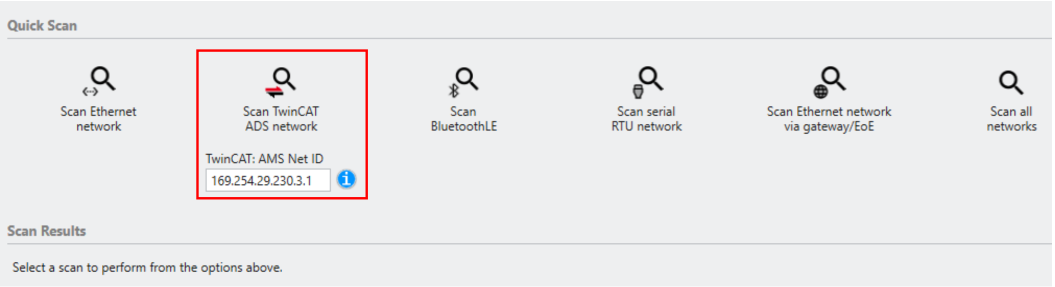


Restart TwinCAT System in Run Mode

9. If the system is operating properly the status icon in the bottom right corner of TwinCAT will show a green box with a rotating cog:



10. Communications will now be running and Connect will be able to search for the drive(s) on the EtherCAT network using the “NetId:” from step 7.



TwinCAT: AMS Net ID 169.254.29.230.3.1

Select a scan to perform from the options above.

2.6 Configure the drive using Connect

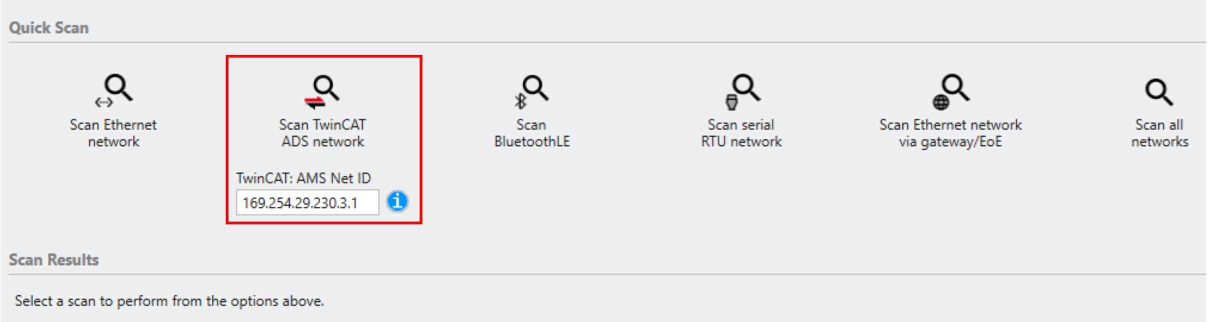
Create a new Connect project to configure and commission the drive for the motion application:

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, or http://acim.nidec.com/drives/control-techniques, or Control Techniques Support Suite.

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

4. Select the communications type, such as ADS, to scan that network and locate the drives to be configured:

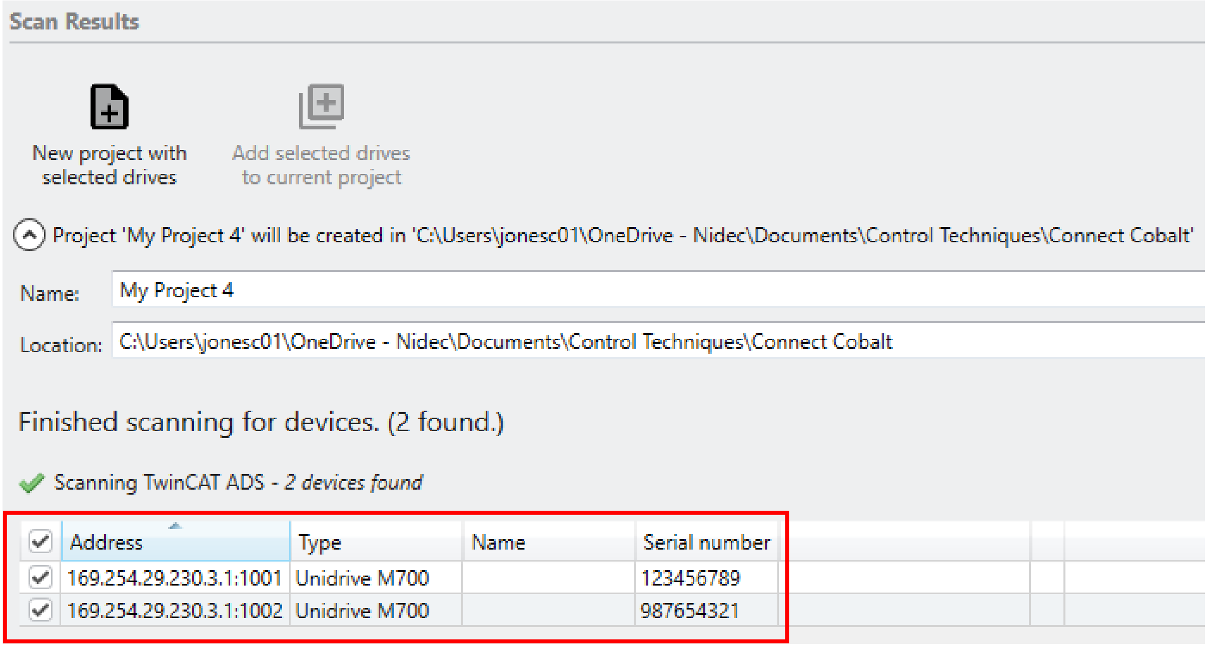


TwinCAT: AMS Net ID 169.254.29.230.3.1

Select a scan to perform from the options above.

HINT: Section **2.5 Setup an EtherCAT AoE connection from the PLC to the drive** shows where to get the EtherCAT AMS Net ID / ADS Network ID from TwinCAT.

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



Project 'My Project 4' will be created in 'C:\Users\jonesc01\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt'

Name: My Project 4

Location: C:\Users\jonesc01\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt

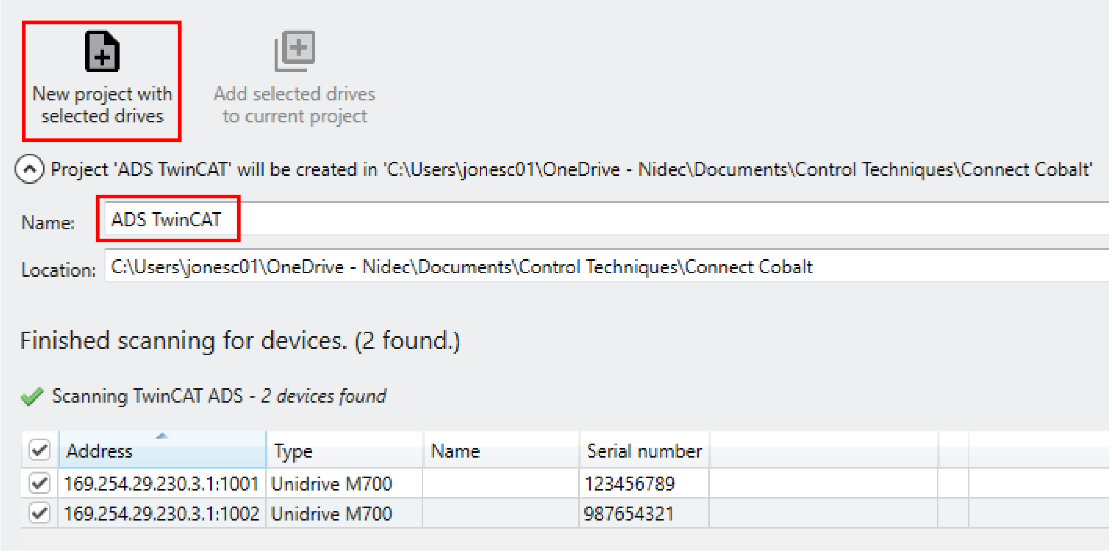
Finished scanning for devices. (2 found.)

✓ Scanning TwinCAT ADS - *2 devices found*

☑	Address	Type	Name	Serial number
☑	169.254.29.230.3.1:1001	Unidrive M700		123456789
☑	169.254.29.230.3.1:1002	Unidrive M700		987654321

HINT: For this to be successful on an EtherCAT-based network, the Beckhoff TwinCAT PLC/IPC must be running, as indicated by a green “TC” LED on the PLC/IPC and the activated symbol highlighted in TwinCAT.

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



Project 'ADS TwinCAT' will be created in 'C:\Users\jonesc01\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt'

Name: ADS TwinCAT

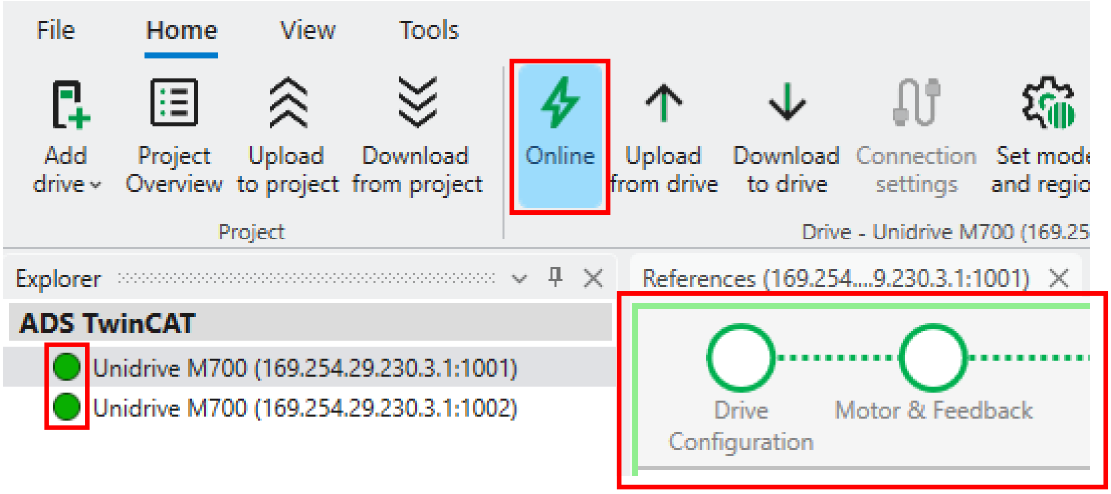
Location: C:\Users\jonesc01\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt

Finished scanning for devices. (2 found.)

✓ Scanning TwinCAT ADS - *2 devices found*

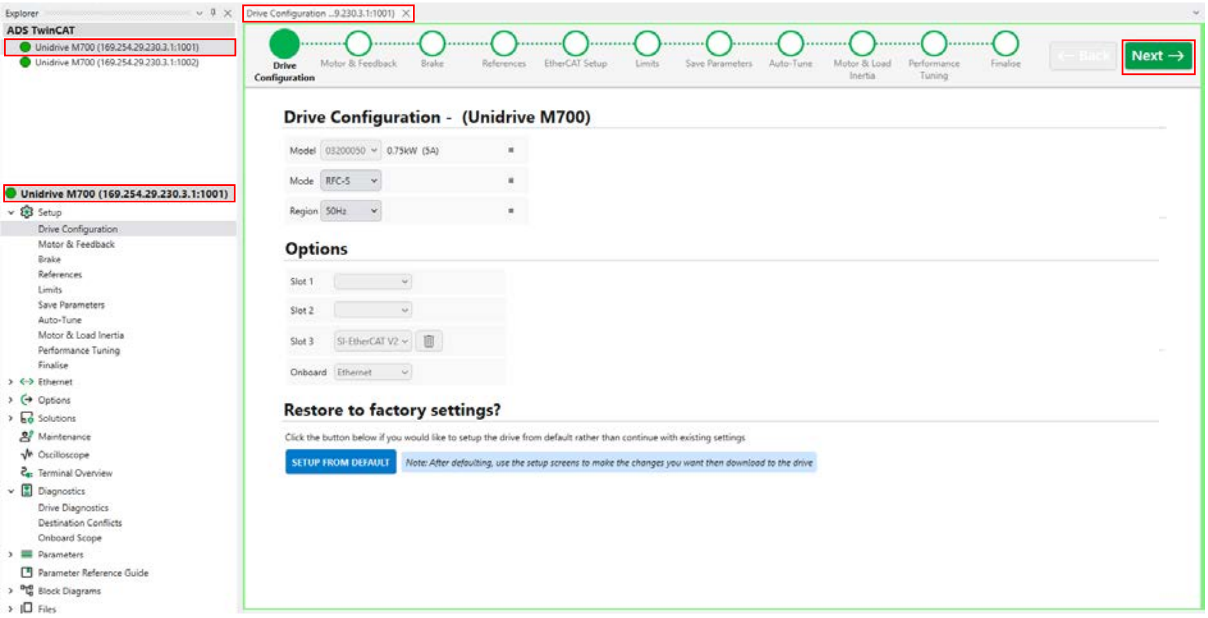
☑	Address	Type	Name	Serial number
☑	169.254.29.230.3.1:1001	Unidrive M700		123456789
☑	169.254.29.230.3.1:1002	Unidrive M700		987654321

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.



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 setup wizard opened.



Drive Configuration - (Unidrive M700)

Model 03200050 0.75kW (5A)

Mode RFC-S

Region 50Hz

Options

Slot 1

Slot 2

Slot 3 SI-EtherCAT V2

Onboard Ethernet

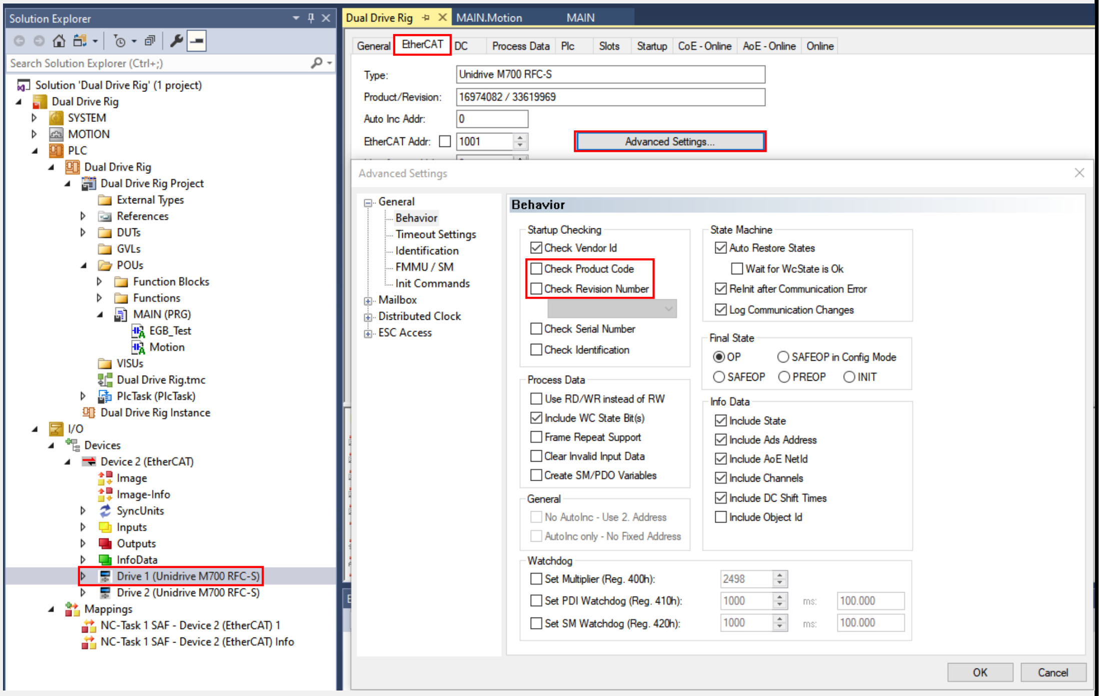
Restore to factory settings?

Click the button below if you would like to setup the drive from default rather than continue with existing settings

Note: After defaulting, use the setup screens to make the changes you want then download to the drive

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: Has comms been lost after changing the drive mode? This might be because the EtherCAT interface has V1 firmware installed which as a part of the start-up checks includes the drive mode. V2 firmware removes this issue. If the drive mode is changed the comms connection stops completely, in TwinCAT, change to “Config mode”, delete the EtherCAT drive nodes, and then right click on the network master and select “Scan” to add the nodes in their new mode, and then activate the configuration. Alternatively, uncheck the “Check Product Code” and “Check Revision Number”, and then activate the configuration. See also section **3.9 How to upgrade the EtherCAT interface firmware**.



Type: Unidrive M700 RFC-S

Product/Revision: 16974082 / 33619969

Auto Inc Addr: 0

EtherCAT Addr: ☐ 1001

Behavior

Startup Checking
- ☑ Check Vendor Id
- ☐ Check Product Code
- ☐ Check Revision Number
- ☐ Check Serial Number
- ☐ Check Identification

State Machine
- ☑ Auto Restore States
  - ☐ Wait for WcState is Ok
- ☑ ReInit after Communication Error
- ☑ Log Communication Changes

Final State
- ● OP ○ SAFEOP in Config Mode
- ○ SAFEOP ○ PREOP ○ INIT

Process Data
- ☐ Use RD/WR instead of RW
- ☑ Include WC State Bit(s)
- ☐ Frame Repeat Support
- ☐ Clear Invalid Input Data
- ☐ Create SM/PDO Variables

Info Data
- ☑ Include State
- ☑ Include Ads Address
- ☑ Include AoE NetId
- ☑ Include Channels
- ☑ Include DC Shift Times
- ☐ Include Object Id

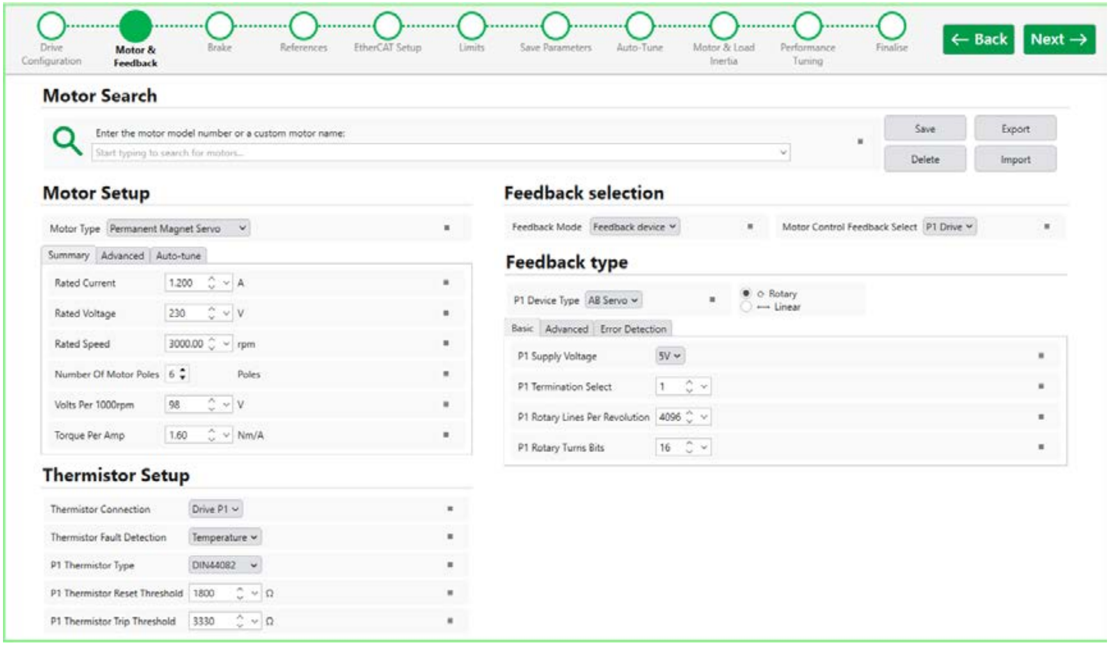
General
- ☐ No AutoInc - Use 2. Address
- ☐ AutoInc only - No Fixed Address

Watchdog
- ☐ Set Multiplier (Reg. 400h): 2498
- ☐ Set PDI Watchdog (Reg. 410h): 1000 ms: 100.000
- ☐ Set SM Watchdog (Reg. 420h): 1000 ms: 100.000

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.



Motor Search

Enter the motor model number or a custom motor name:

Start typing to search for motors...

Motor Setup

Motor Type: Permanent Magnet Servo

Rated Current	1.200	A
Rated Voltage	230	V
Rated Speed	3000.00	rpm
Number Of Motor Poles	6	Poles
Volts Per 1000rpm	98	V
Torque Per Amp	1.60	Nm/A

Thermistor Setup

Thermistor Connection	Drive P1	
Thermistor Fault Detection	Temperature	
P1 Thermistor Type	DIN44082	
P1 Thermistor Reset Threshold	1800	Ω
P1 Thermistor Trip Threshold	3330	Ω

Feedback selection

Feedback Mode: Feedback device

Motor Control Feedback Select: P1 Drive

Feedback type

P1 Device Type: AB Servo

● Rotary

○ Linear

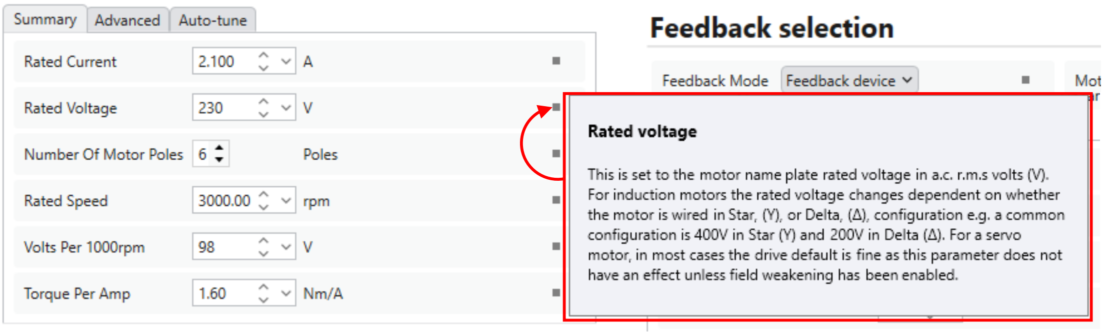
P1 Supply Voltage	5V
P1 Termination Select	1
P1 Rotary Lines Per Revolution	4096
P1 Rotary Turns Bits	16

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

It is important to setup the **Torque Per Amp** field in the motor data. If the value isn’t available for an induction motor, it can be approximated by dividing the Motor rated torque by the Motor rated current. This value is used by the motor and load inertia calculation, which helps make the motor tuning simple with a single slider.

> **HINT:** The Motor rated torque for an induction motor is calculated from:
>
> Rated Torque (Nm) = (Rated Power (W) x 60) / (2 x π x Motor Rated Speed (rpm))

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



Rated Current	2.100	A
Rated Voltage	230	V
Number Of Motor Poles	6	Poles
Rated Speed	3000.00	rpm
Volts Per 1000rpm	98	V
Torque Per Amp	1.60	Nm/A

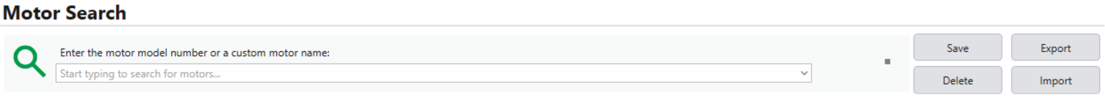
Feedback selection

Feedback Mode: Feedback device

Rated voltage

This is set to the motor name plate rated voltage in a.c. r.m.s volts (V). For induction motors the rated voltage changes dependent on whether the motor is wired in Star, (Y), or Delta, (Δ), configuration e.g. a common configuration is 400V in Star (Y) and 200V in Delta (Δ). For a servo motor, in most cases the drive default is fine as this parameter does not have an effect unless field weakening has been enabled.

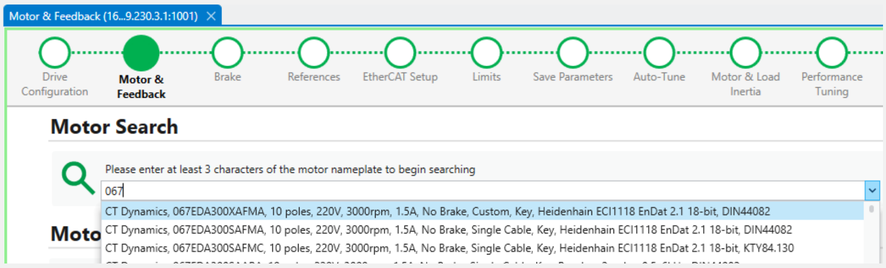
It is possible to save a motor and encoder configuration for later use; these are searchable too.



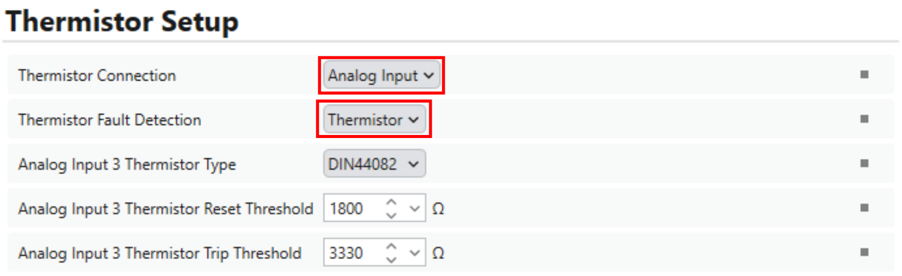
Motor Search

Enter the motor model number or a custom motor name:

Start typing to search for motors...

> **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:
>
> 
>
>
>
> **Motor Search**
>
> Please enter at least 3 characters of the motor nameplate to begin searching
>
> 067
>
> CT Dynamics, 067EDA300XAFMA, 10 poles, 220V, 3000rpm, 1.5A, No Brake, Custom, Key, Heidenhain ECI1118 EnDat 2.1 18-bit, DIN44082
>
> CT Dynamics, 067EDA300SAFMA, 10 poles, 220V, 3000rpm, 1.5A, No Brake, Single Cable, Key, Heidenhain ECI1118 EnDat 2.1 18-bit, DIN44082
>
> CT Dynamics, 067EDA300SAFMC, 10 poles, 220V, 3000rpm, 1.5A, No Brake, Single Cable, Key, Heidenhain ECI1118 EnDat 2.1 18-bit, KTY84.130

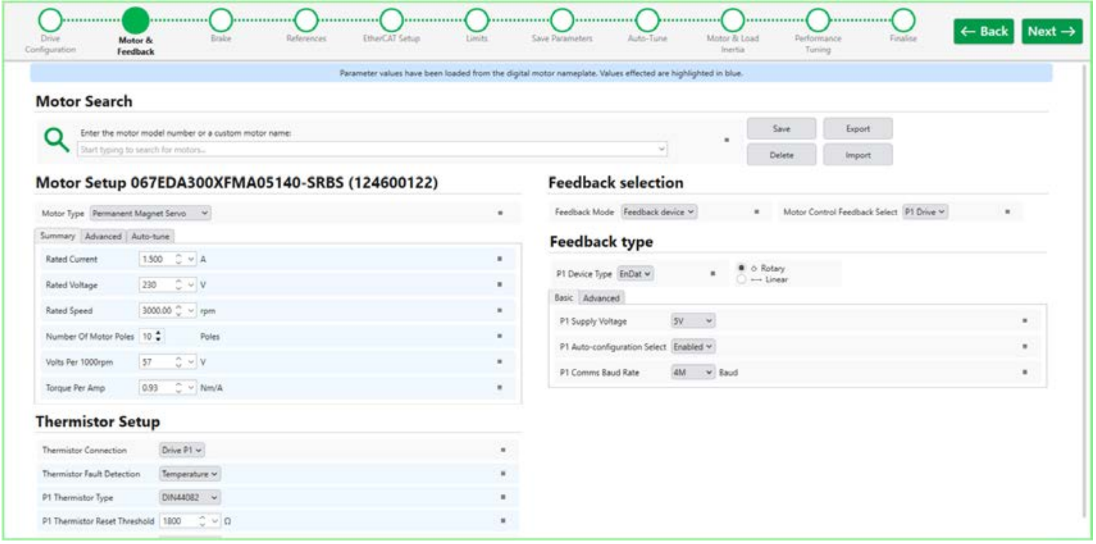
For induction motors, where the thermistor is not typically connected to the encoder interface D Type, and instead, is connected to a terminal on the drive (e.g. Analogue input 3). Set Analogue Input 3 Mode to Thermistor.



Thermistor Setup

Thermistor Connection	Analog Input	
Thermistor Fault Detection	Thermistor	
Analog Input 3 Thermistor Type	DIN44082	
Analog Input 3 Thermistor Reset Threshold	1800	Ω
Analog Input 3 Thermistor Trip Threshold	3330	Ω

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:



Parameter values have been loaded from the digital motor nameplate. Values effected are highlighted in blue.

Motor Search

Enter the motor model number or a custom motor name:

Start typing to search for motors...

Motor Setup 067EDA300XFMA05140-SRBS (124600122)

Motor Type: Permanent Magnet Servo

Rated Current	1.500	A
Rated Voltage	230	V
Rated Speed	3000.00	rpm
Number Of Motor Poles	10	Poles
Volts Per 1000rpm	57	V
Torque Per Amp	0.93	Nm/A

Thermistor Setup

Thermistor Connection	Drive P1	
Thermistor Fault Detection	Temperature	
P1 Thermistor Type	DIN44082	
P1 Thermistor Reset Threshold	1800	Ω

Feedback selection

Feedback Mode: Feedback device

Motor Control Feedback Select: P1 Drive

Feedback type

P1 Device Type: EnDat

● Rotary
○ Linear

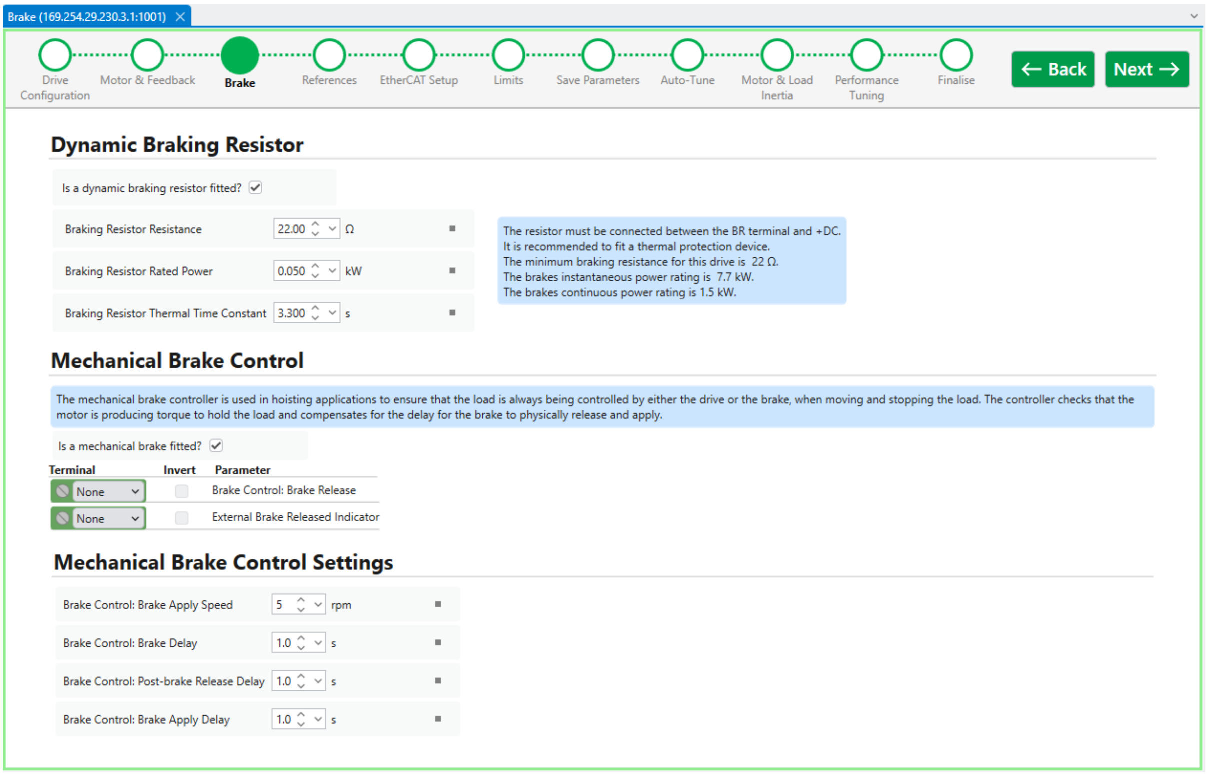
P1 Supply Voltage	5V	
P1 Auto-configuration Select	Enabled	
P1 Comms Baud Rate	4M	Baud

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.

    This page is also 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 controlled by either the motor 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.



Dynamic Braking Resistor

Is a dynamic braking resistor fitted? ☑

Braking Resistor Resistance 22.00 Ω

Braking Resistor Rated Power 0.050 kW

Braking Resistor Thermal Time Constant 3.300 s

The resistor must be connected between the BR terminal and +DC.
It is recommended to fit a thermal protection device.
The minimum braking resistance for this drive is 22 Ω.
The brakes instantaneous power rating is 7.7 kW.
The brakes continuous power rating is 1.5 kW.

Mechanical Brake Control

The mechanical brake controller is used in hoisting applications to ensure that the load is always being controlled by either the drive or the brake, when moving and stopping the load. The controller checks that the motor is producing torque to hold the load and compensates for the delay for the brake to physically release and apply.

Is a mechanical brake fitted? ☑

Terminal	Invert	Parameter
None	☐	Brake Control: Brake Release
None	☐	External Brake Released Indicator

Mechanical Brake Control Settings

Brake Control: Brake Apply Speed 5 rpm

Brake Control: Brake Delay 1.0 s

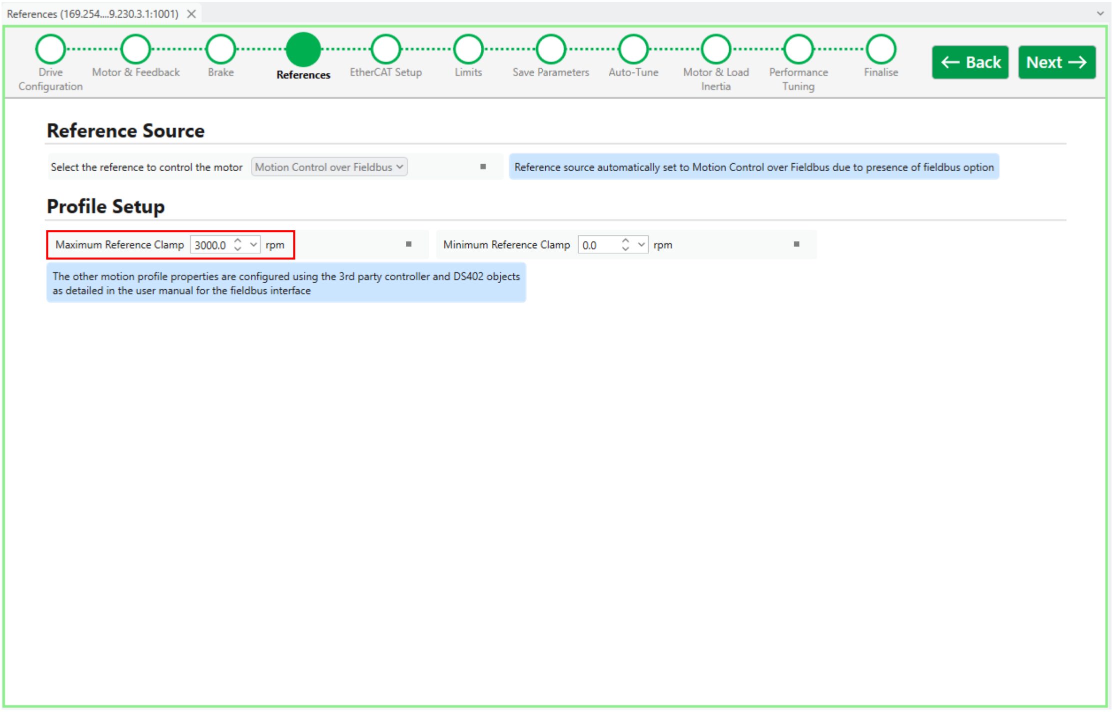
Brake Control: Post-brake Release Delay 1.0 s

Brake Control: Brake Apply Delay 1.0 s

For more information on the mechanical brake controller logic and timings see section **6.2 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 increased and may be adjusted from this page.



Reference Source

Select the reference to control the motor: Motion Control over Fieldbus

Reference source automatically set to Motion Control over Fieldbus due to presence of fieldbus option

Profile Setup

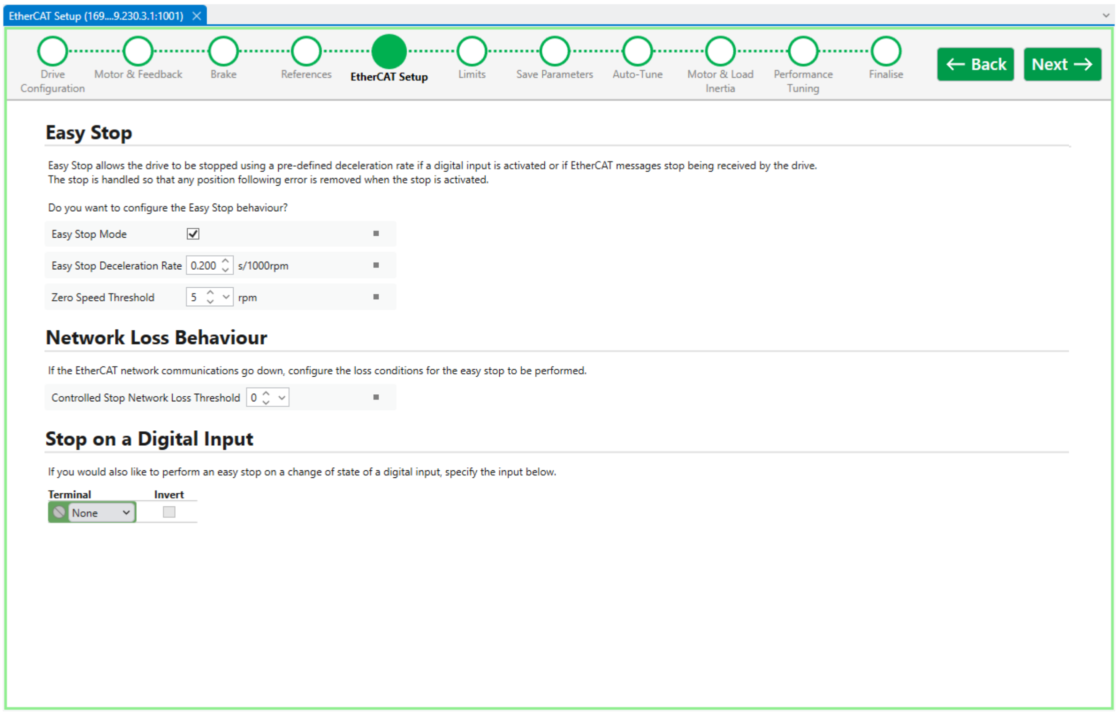
Maximum Reference Clamp: 3000.0 rpm

Minimum Reference Clamp: 0.0 rpm

The other motion profile properties are configured using the 3rd party controller and DS402 objects as detailed in the user manual for the fieldbus interface

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.



Easy Stop

Easy Stop allows the drive to be stopped using a pre-defined deceleration rate if a digital input is activated or if EtherCAT messages stop being received by the drive.
The stop is handled so that any position following error is removed when the stop is activated.

Do you want to configure the Easy Stop behaviour?

Easy Stop Mode ☑

Easy Stop Deceleration Rate 0.200 s/1000rpm

Zero Speed Threshold 5 rpm

Network Loss Behaviour

If the EtherCAT network communications go down, configure the loss conditions for the easy stop to be performed.

Controlled Stop Network Loss Threshold 0

Stop on a Digital Input

If you would also like to perform an easy stop on a change of state of a digital input, specify the input below.

Terminal	Invert
None	☐

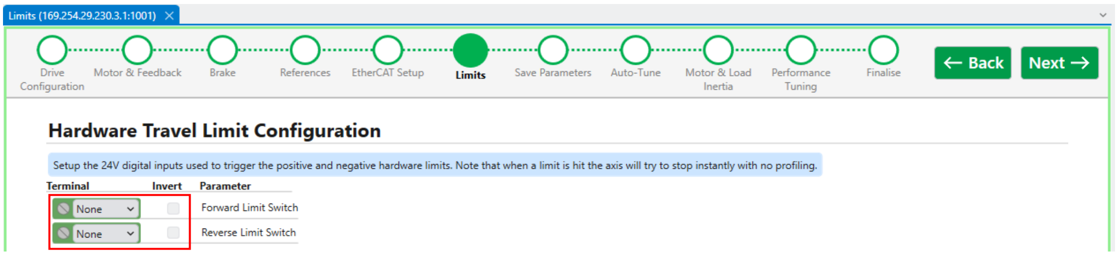
When the "Easy Stop Mode" box is checked, the user can assign a digital input that will cause the axis to stop. The axis is permitted to run when 24V is applied to it, but if 24V is removed the axis will stop, using the "Easy Stop Deceleration Rate".

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.



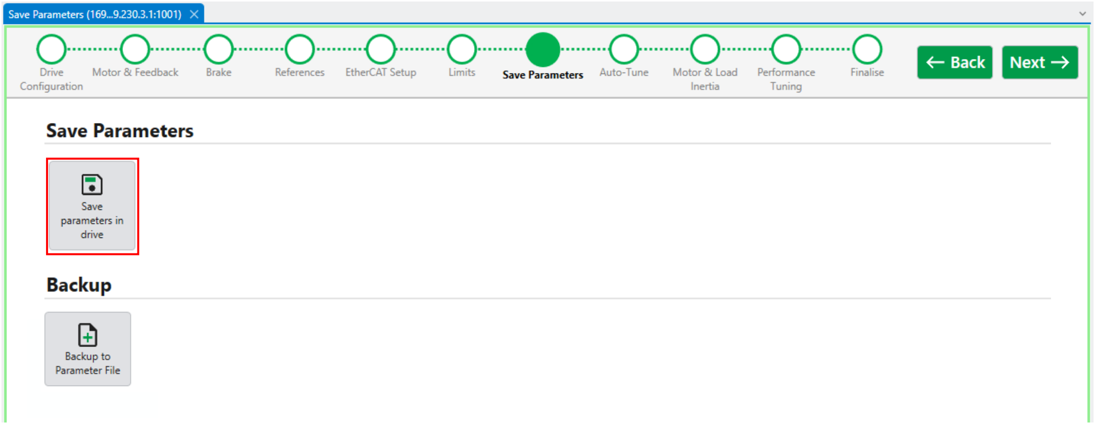
Hardware Travel Limit Configuration

Setup the 24V digital inputs used to trigger the positive and negative hardware limits. Note that when a limit is hit the axis will try to stop instantly with no profiling.

Terminal	Invert	Parameter
None	☐	Forward Limit Switch
None	☐	Reverse Limit Switch

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.

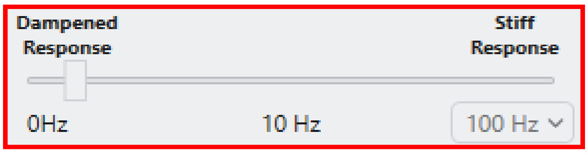


This completes the configuration section of the guided setup. The next section of the manual describes the commissioning of the drive and motor using the remaining steps in the guided setup.

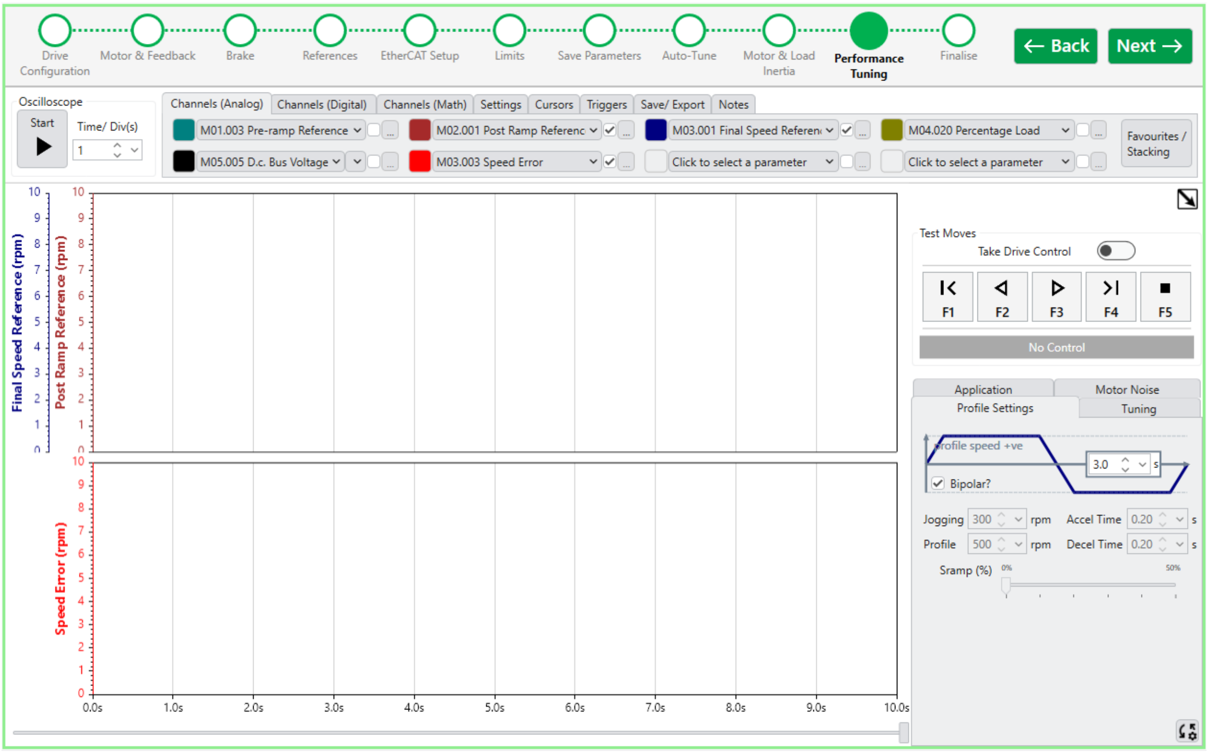
2.7 Commission the drive and motor using Connect

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.



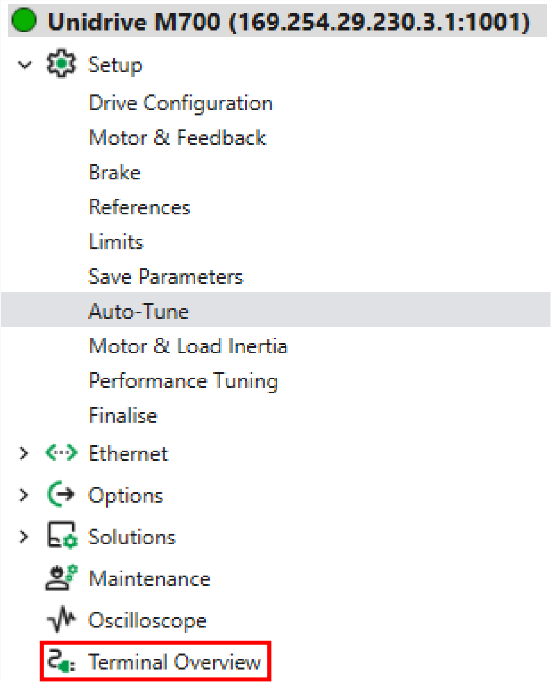
- Performance tuning and test moves. This page is a “one stop shop” with test move controls, motion profile setup, oscilloscope, and tuning controls on a single page, ideal for site commissioning using a laptop.

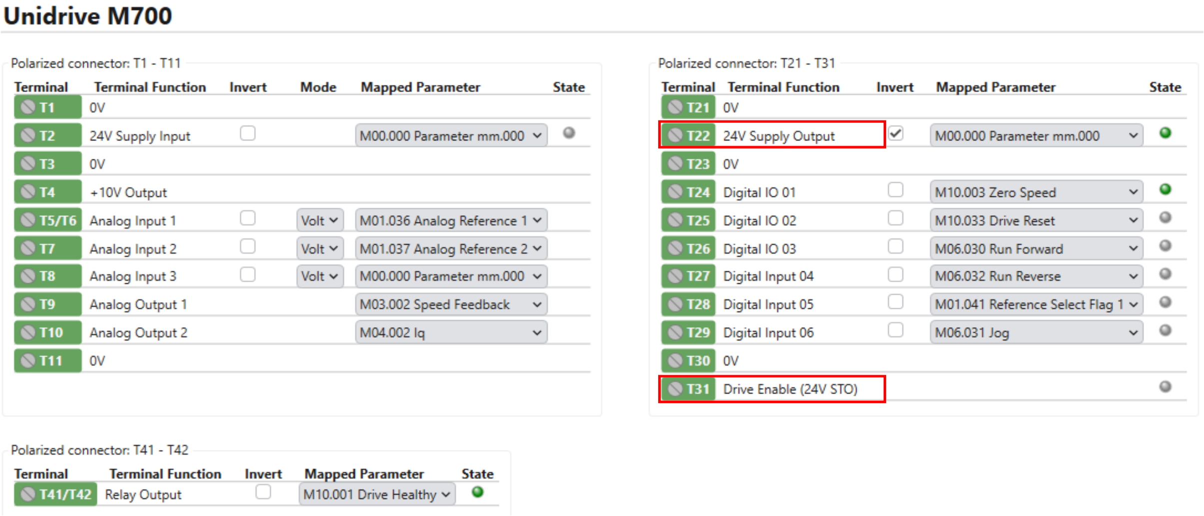


- Finalising the application by saving the tuned parameters and creating a startup list allowing the PLC / IPC to configure the drive, each time the system powers up; this is helpful if the drive must be replaced.

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.





Unidrive M700

Polarized connector: T1 - T11

Terminal	Terminal Function	Invert	Mode	Mapped Parameter	State
T1	0V				
T2	24V Supply Input	☐		M00.000 Parameter mm.000	●
T3	0V				
T4	+10V Output				
T5/T6	Analog Input 1	☐	Volt	M01.036 Analog Reference 1	
T7	Analog Input 2	☐	Volt	M01.037 Analog Reference 2	
T8	Analog Input 3	☐	Volt	M00.000 Parameter mm.000	
T9	Analog Output 1			M03.002 Speed Feedback	
T10	Analog Output 2			M04.002 Iq	
T11	0V				

Polarized connector: T21 - T31

Terminal	Terminal Function	Invert	Mapped Parameter	State
T21	0V			
T22	24V Supply Output	☑	M00.000 Parameter mm.000	●
T23	0V			
T24	Digital IO 01	☐	M10.003 Zero Speed	●
T25	Digital IO 02	☐	M10.033 Drive Reset	●
T26	Digital IO 03	☐	M06.030 Run Forward	●
T27	Digital Input 04	☐	M06.032 Run Reverse	●
T28	Digital Input 05	☐	M01.041 Reference Select Flag 1	●
T29	Digital Input 06	☐	M06.031 Jog	●
T30	0V			
T31	Drive Enable (24V STO)			●

Polarized connector: T41 - T42

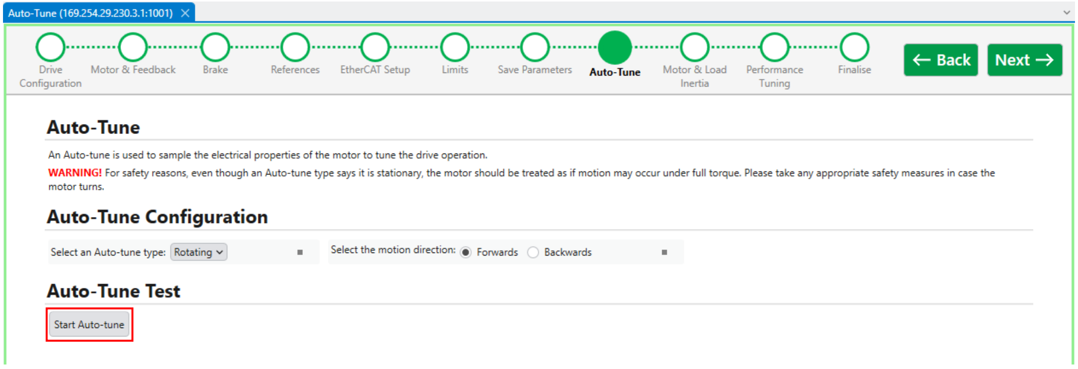
Terminal	Terminal Function	Invert	Mapped Parameter	State
T41/T42	Relay Output	☐	M10.001 Drive Healthy	●

> **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. If the motor is a Control Techniques Dynamic type with an electronic name plate, then this autotune is not required.

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



Auto-Tune

An Auto-tune is used to sample the electrical properties of the motor to tune the drive operation.

WARNING! For safety reasons, even though an Auto-tune type says it is stationary, the motor should be treated as if motion may occur under full torque. Please take any appropriate safety measures in case the motor turns.

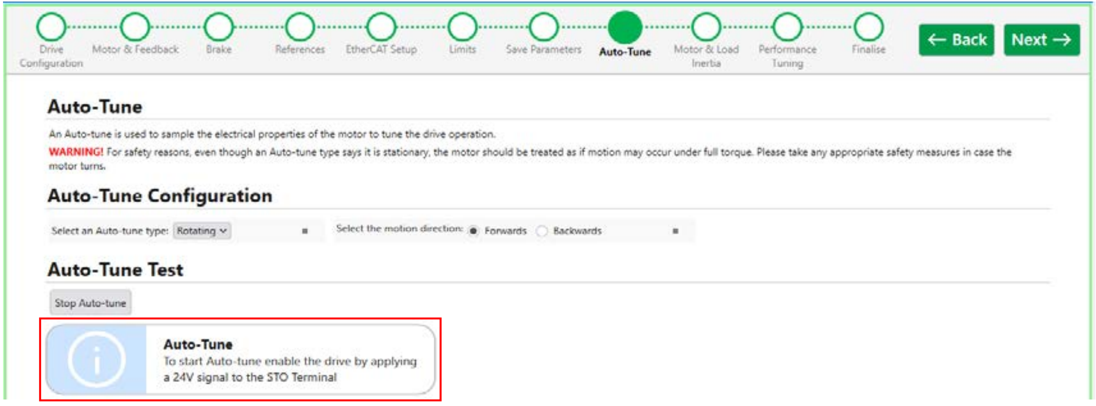
Auto-Tune Configuration

Select an Auto-tune type: Rotating

Select the motion direction: ● Forwards ○ Backwards

Auto-Tune Test

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



Auto-Tune

An Auto-tune is used to sample the electrical properties of the motor to tune the drive operation.

WARNING! For safety reasons, even though an Auto-tune type says it is stationary, the motor should be treated as if motion may occur under full torque. Please take any appropriate safety measures in case the motor turns.

Auto-Tune Configuration

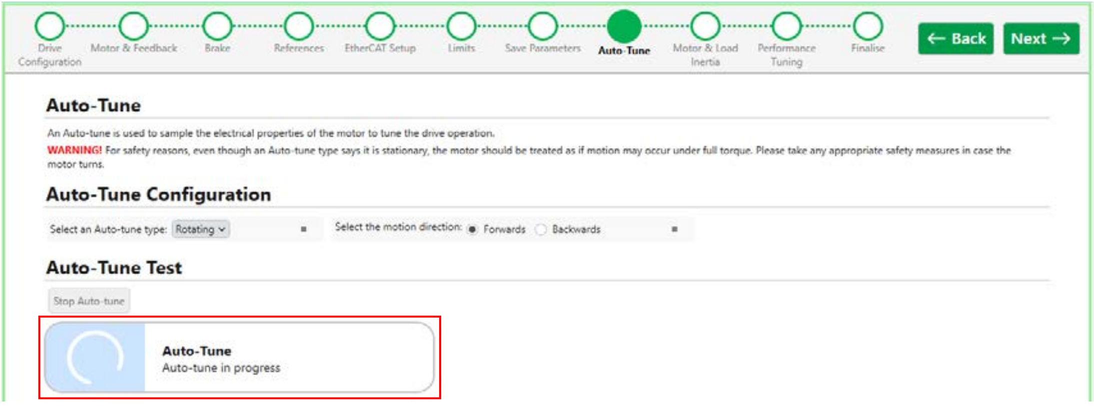
Select an Auto-tune type: Rotating

Select the motion direction: ● Forwards ○ Backwards

Auto-Tune Test

Auto-Tune
To start Auto-tune enable the drive by applying a 24V signal to the STO Terminal

   The test takes approximately 30s to complete



Auto-Tune

An Auto-tune is used to sample the electrical properties of the motor to tune the drive operation.

WARNING! For safety reasons, even though an Auto-tune type says it is stationary, the motor should be treated as if motion may occur under full torque. Please take any appropriate safety measures in case the motor turns.

Auto-Tune Configuration

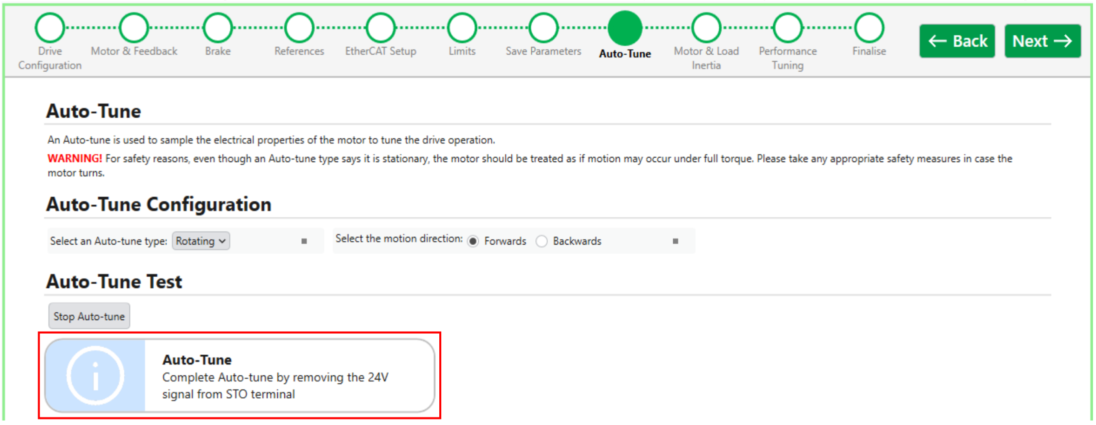
Select an Auto-tune type: Rotating

Select the motion direction: ● Forwards ○ Backwards

Auto-Tune Test

Auto-Tune
Auto-tune in progress

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



Auto-Tune

An Auto-tune is used to sample the electrical properties of the motor to tune the drive operation.

WARNING! For safety reasons, even though an Auto-tune type says it is stationary, the motor should be treated as if motion may occur under full torque. Please take any appropriate safety measures in case the motor turns.

Auto-Tune Configuration

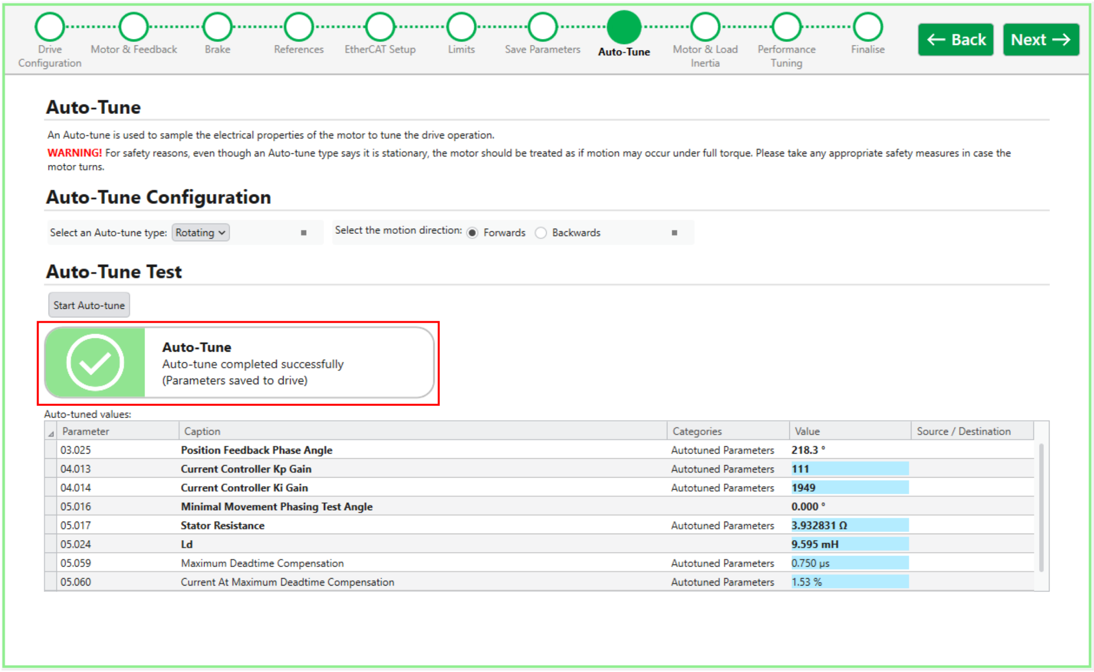
Select an Auto-tune type: Rotating

Select the motion direction: ● Forwards ○ Backwards

Auto-Tune Test

Auto-Tune
Complete Auto-tune by removing the 24V signal from STO terminal

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



Auto-Tune

An Auto-tune is used to sample the electrical properties of the motor to tune the drive operation.

WARNING! For safety reasons, even though an Auto-tune type says it is stationary, the motor should be treated as if motion may occur under full torque. Please take any appropriate safety measures in case the motor turns.

Auto-Tune Configuration

Select an Auto-tune type: Rotating

Select the motion direction: ● Forwards ○ Backwards

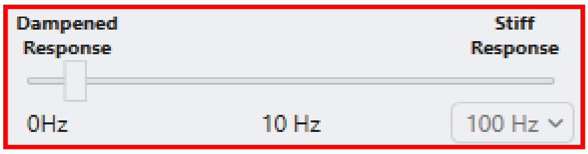
Auto-Tune Test

Auto-Tune
Auto-tune completed successfully
(Parameters saved to drive)

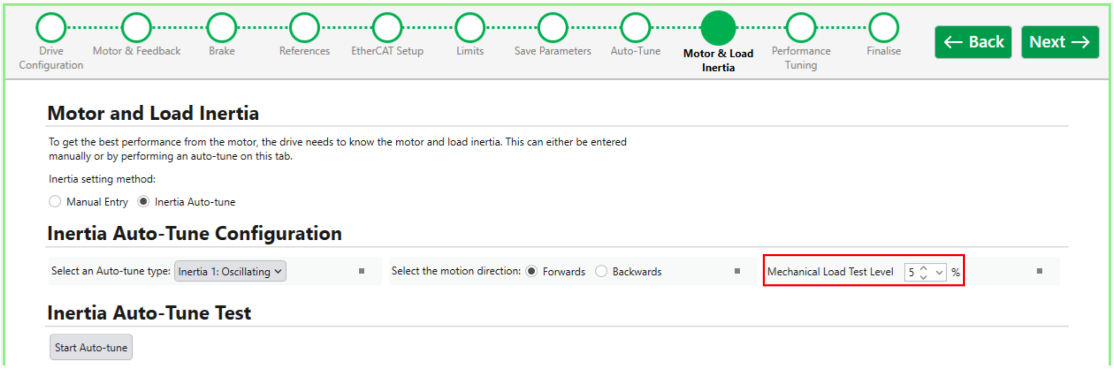
Auto-tuned values:

Parameter	Caption	Categories	Value	Source / Destination
03.025	**Position Feedback Phase Angle**	Autotuned Parameters	**218.3 °**	
04.013	**Current Controller Kp Gain**	Autotuned Parameters	**111**	
04.014	**Current Controller Ki Gain**	Autotuned Parameters	**1949**	
05.016	**Minimal Movement Phasing Test Angle**		**0.000 °**	
05.017	**Stator Resistance**	Autotuned Parameters	**3.932831 Ω**	
05.024	**Ld**		**9.595 mH**	
05.059	Maximum Deadtime Compensation	Autotuned Parameters	0.750 µs	
05.060	Current At Maximum Deadtime Compensation	Autotuned Parameters	1.53 %	

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.



Motor and Load Inertia

To get the best performance from the motor, the drive needs to know the motor and load inertia. This can either be entered manually or by performing an auto-tune on this tab.

Inertia setting method:

○ Manual Entry ● Inertia Auto-tune

Inertia Auto-Tune Configuration

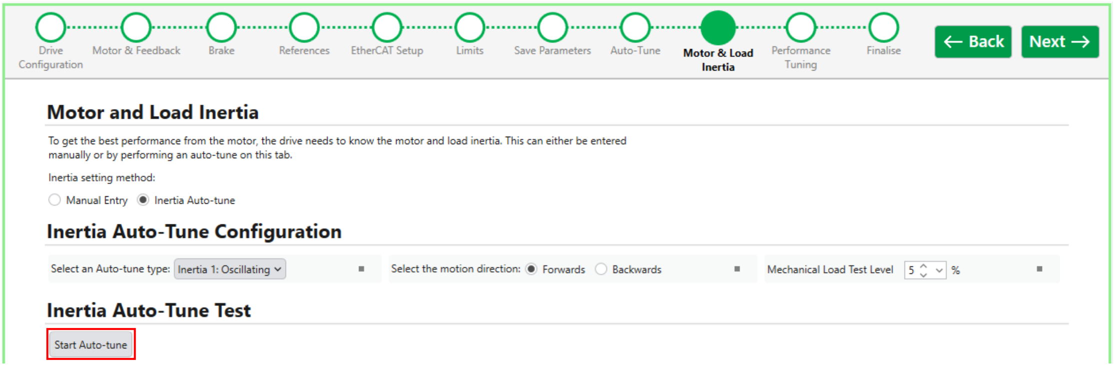
Select an Auto-tune type: Inertia 1: Oscillating

Select the motion direction: ● Forwards ○ Backwards

Mechanical Load Test Level 5 %

Inertia Auto-Tune Test

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



Motor and Load Inertia

To get the best performance from the motor, the drive needs to know the motor and load inertia. This can either be entered manually or by performing an auto-tune on this tab.

Inertia setting method:

○ Manual Entry ● Inertia Auto-tune

Inertia Auto-Tune Configuration

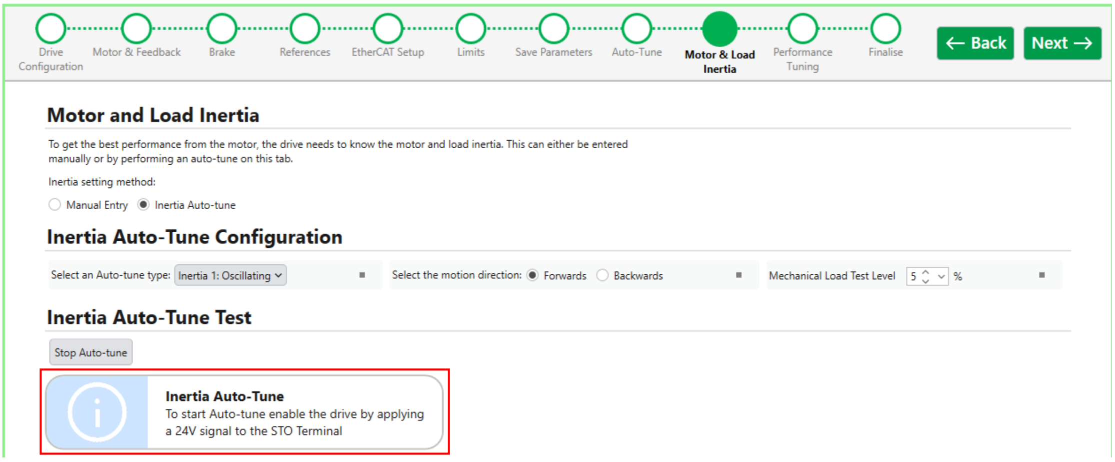
Select an Auto-tune type: Inertia 1: Oscillating

Select the motion direction: ● Forwards ○ Backwards

Mechanical Load Test Level 5 %

Inertia Auto-Tune Test

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



Motor and Load Inertia

To get the best performance from the motor, the drive needs to know the motor and load inertia. This can either be entered manually or by performing an auto-tune on this tab.

Inertia setting method:

○ Manual Entry ● Inertia Auto-tune

Inertia Auto-Tune Configuration

Select an Auto-tune type: Inertia 1: Oscillating

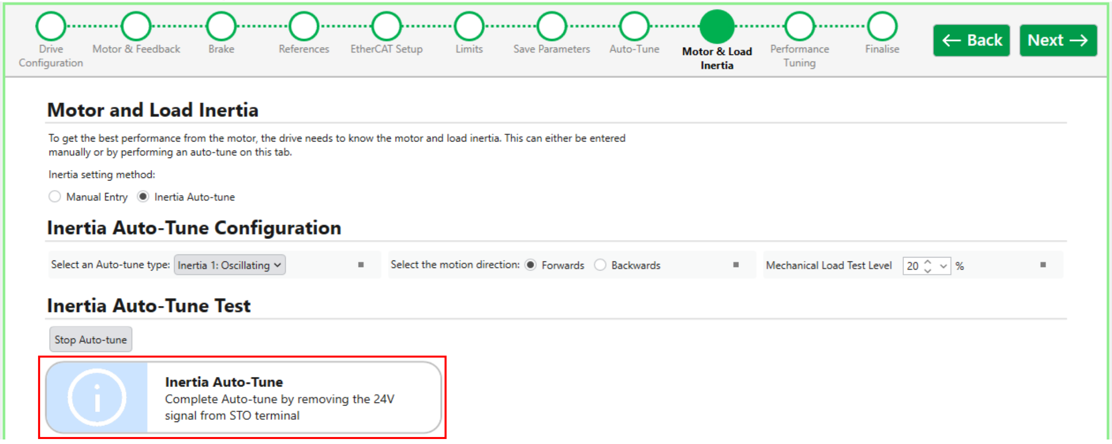
Select the motion direction: ● Forwards ○ Backwards

Mechanical Load Test Level 5 %

Inertia Auto-Tune Test

Inertia Auto-Tune
To start Auto-tune enable the drive by applying a 24V signal to the STO Terminal

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



Motor and Load Inertia

To get the best performance from the motor, the drive needs to know the motor and load inertia. This can either be entered manually or by performing an auto-tune on this tab.

Inertia setting method:

○ Manual Entry ● Inertia Auto-tune

Inertia Auto-Tune Configuration

Select an Auto-tune type: Inertia 1: Oscillating

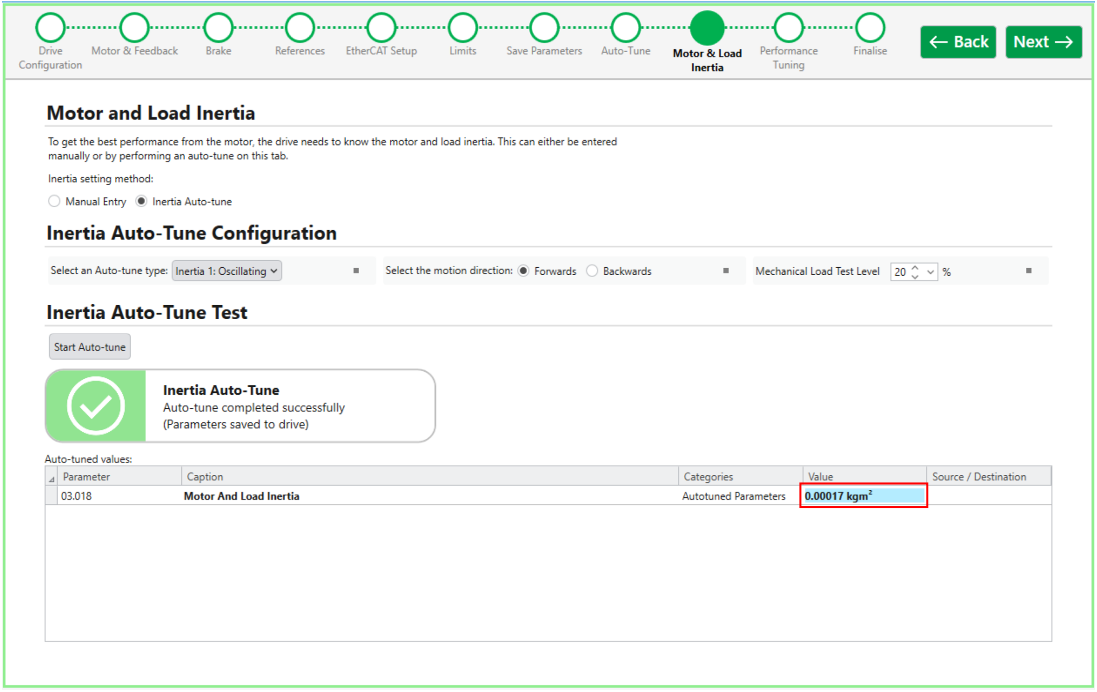
Select the motion direction: ● Forwards ○ Backwards

Mechanical Load Test Level 20 %

Inertia Auto-Tune Test

Inertia Auto-Tune
Complete Auto-tune by removing the 24V signal from STO terminal

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



Motor and Load Inertia

To get the best performance from the motor, the drive needs to know the motor and load inertia. This can either be entered manually or by performing an auto-tune on this tab.

Inertia setting method:

○ Manual Entry ● Inertia Auto-tune

Inertia Auto-Tune Configuration

Select an Auto-tune type: Inertia 1: Oscillating

Select the motion direction: ● Forwards ○ Backwards

Mechanical Load Test Level 20 %

Inertia Auto-Tune Test

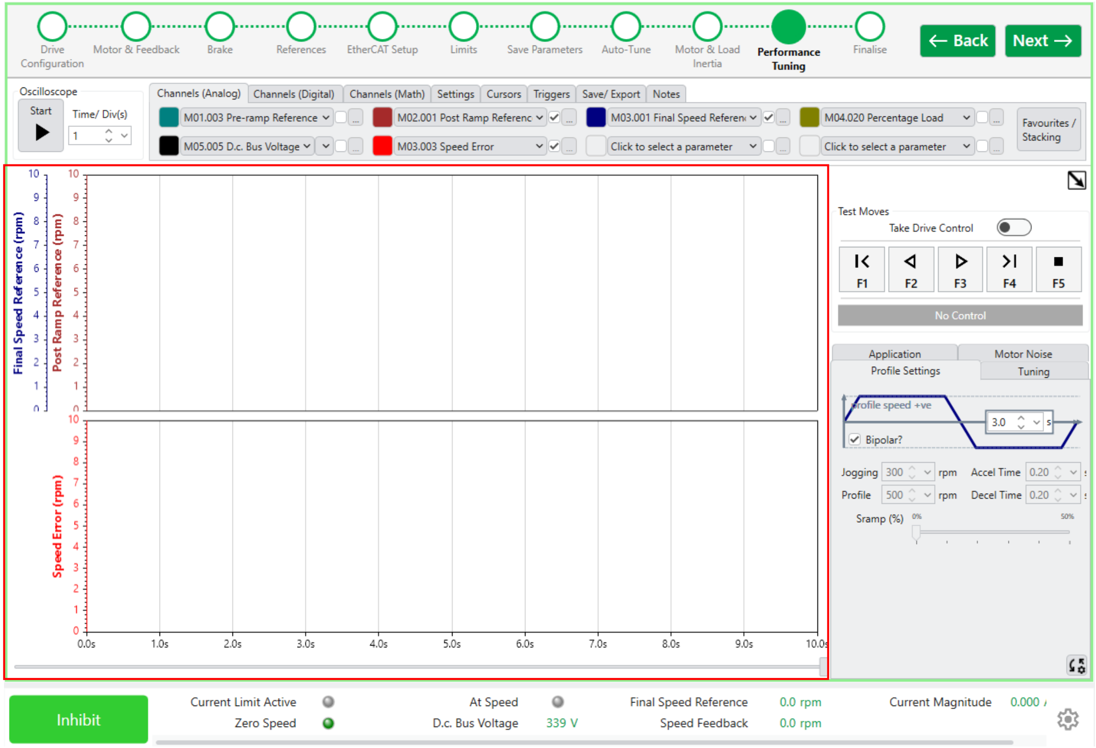
Inertia Auto-Tune
Auto-tune completed successfully
(Parameters saved to drive)

Auto-tuned values:

Parameter	Caption	Categories	Value	Source / Destination
03.018	**Motor And Load Inertia**	Autotuned Parameters	**0.00017 kgm^2**	

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:



Time/ Div(s) 1

Jogging 300 rpm Accel Time 0.20 s

Profile 500 rpm Decel Time 0.20 s

Sramp (%) 0% 50%

☑ Bipolar? 3.0 s

Current Limit Active ●

Zero Speed ●

At Speed ●

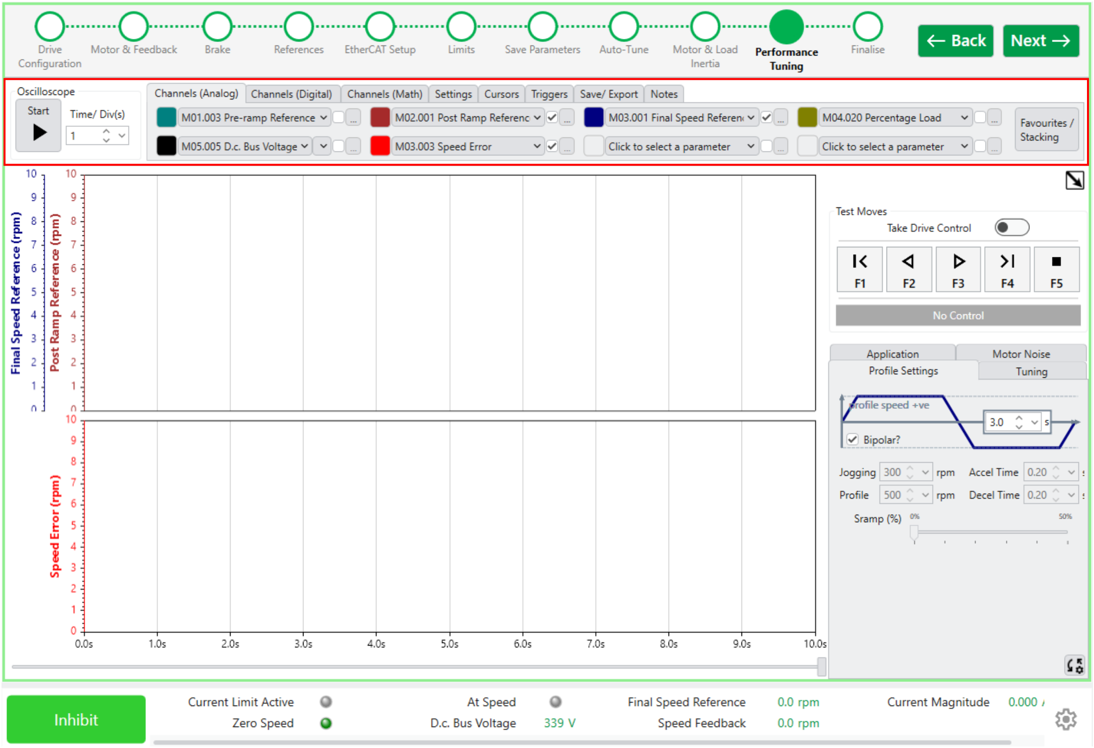
D.c. Bus Voltage 339 V

Final Speed Reference 0.0 rpm

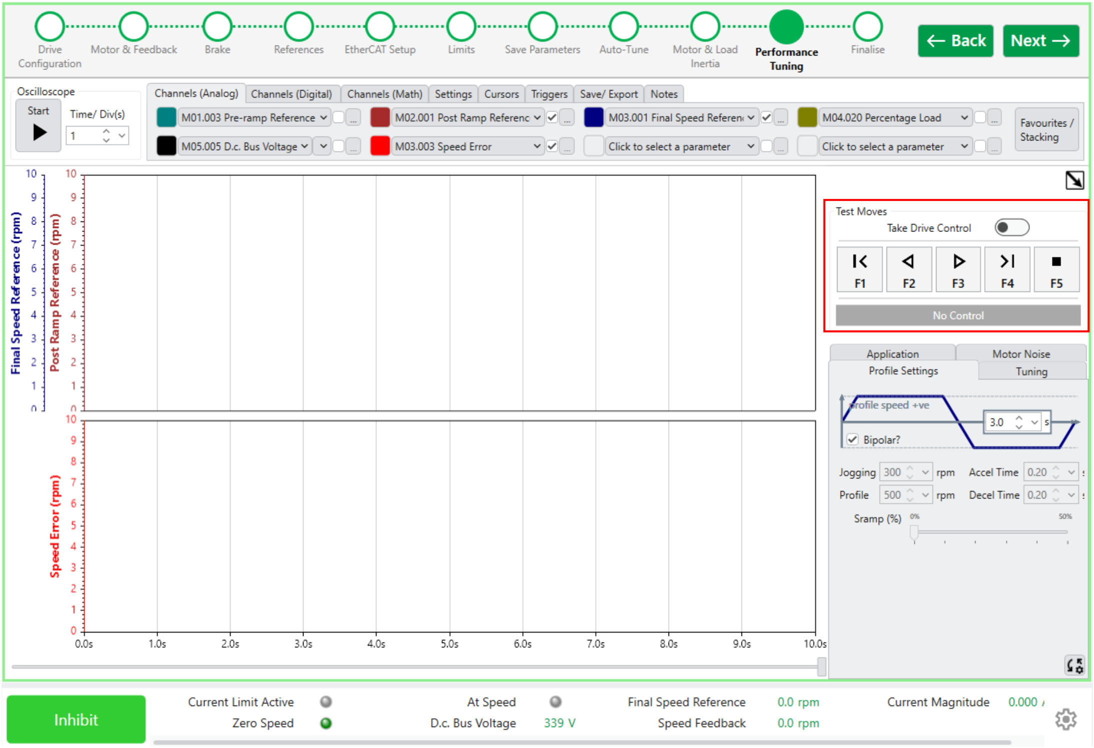
Speed Feedback 0.0 rpm

Current Magnitude 0.000

   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.



Time/ Div(s) 1

M01.003 Pre-ramp Reference

M02.001 Post Ramp Reference ☑

M03.001 Final Speed Reference ☑

M04.020 Percentage Load

M05.005 D.c. Bus Voltage

M03.003 Speed Error ☑

Click to select a parameter

Click to select a parameter

3.0 s

☑ Bipolar?

Jogging 300 rpm Accel Time 0.20

Profile 500 rpm Decel Time 0.20

Sramp (%) 0% 50%

Current Limit Active ●

Zero Speed ●

At Speed ●

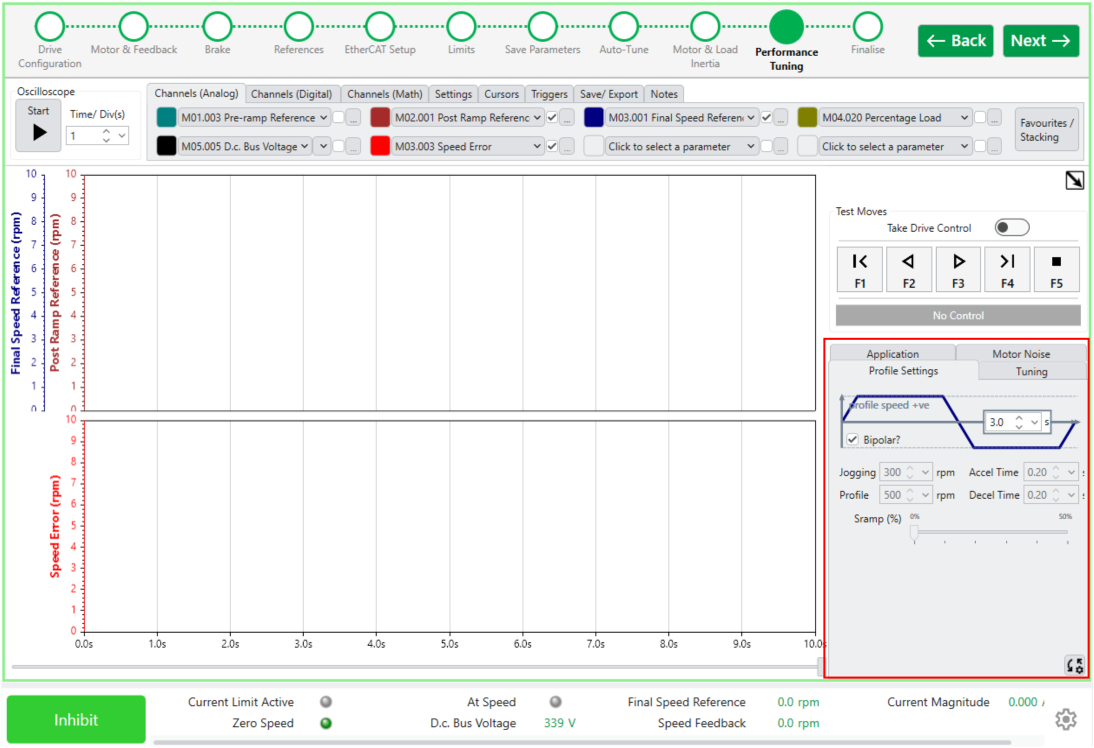
D.c. Bus Voltage 339 V

Final Speed Reference 0.0 rpm

Speed Feedback 0.0 rpm

Current Magnitude 0.000

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



Time/ Div(s) 1

M01.003 Pre-ramp Reference

M02.001 Post Ramp Reference ☑

M03.001 Final Speed Reference ☑

M04.020 Percentage Load

M05.005 D.c. Bus Voltage

M03.003 Speed Error ☑

Click to select a parameter

Click to select a parameter

3.0 s

☑ Bipolar?

Jogging 300 rpm Accel Time 0.20

Profile 500 rpm Decel Time 0.20

Sramp (%) 0% 50%

Current Limit Active ●

Zero Speed ●

At Speed ●

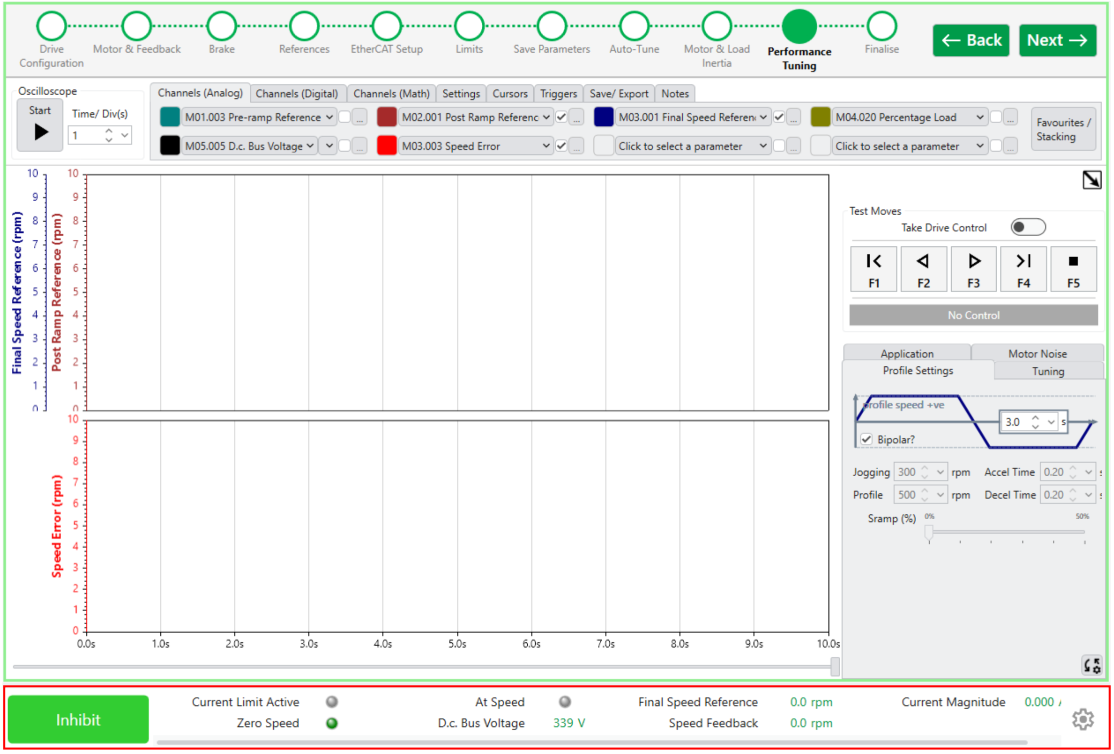
D.c. Bus Voltage 339 V

Final Speed Reference 0.0 rpm

Speed Feedback 0.0 rpm

Current Magnitude 0.000

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



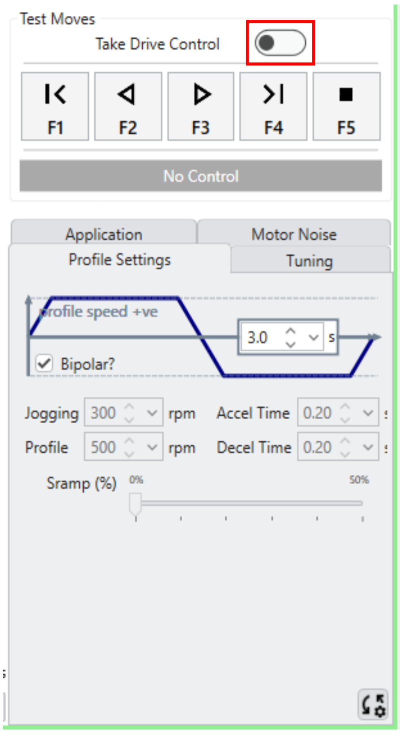
Time/ Div(s) 1

M01.003 Pre-ramp Reference ☐
M02.001 Post Ramp Reference ☑
M03.001 Final Speed Reference ☑
M04.020 Percentage Load ☐
M05.005 D.c. Bus Voltage ☐
M03.003 Speed Error ☑
Click to select a parameter ☐
Click to select a parameter ☐

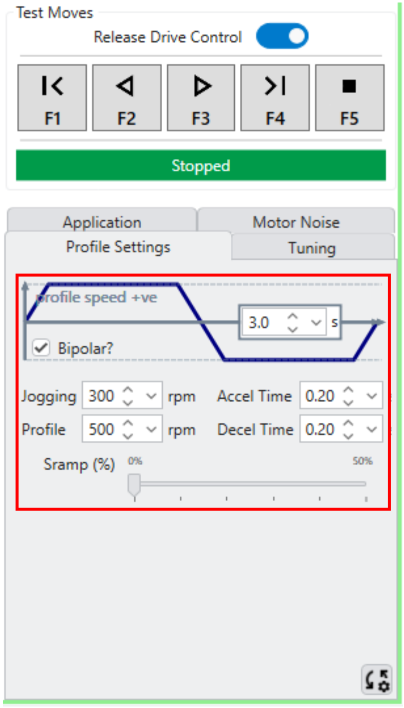
3.0 s
☑ Bipolar?
Jogging 300 rpm
Profile 500 rpm
Accel Time 0.20
Decel Time 0.20
Sramp (%) 0% 50%

Current Limit Active ●
Zero Speed ●
At Speed ●
D.c. Bus Voltage 339 V
Final Speed Reference 0.0 rpm
Speed Feedback 0.0 rpm
Current Magnitude 0.000

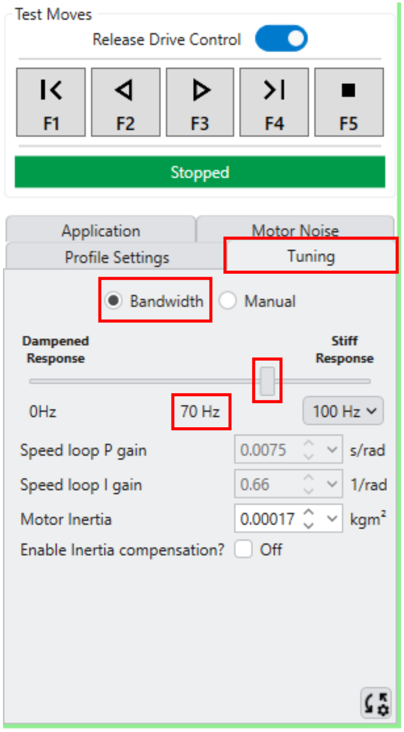
6. Begin commissioning by jogging the axis (slowly) to determine if the motor direction is correct for the application. Ensure it is safe to move the motor. Click on “Take Drive Control” and then apply 24V to the STO input.



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.



● Bandwidth ○ Manual

70 Hz

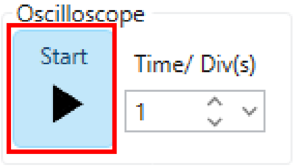
Speed loop P gain 0.0075 s/rad

Speed loop I gain 0.66 1/rad

Motor Inertia 0.00017 kgm^2

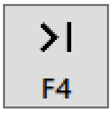
Enable Inertia compensation? ☐ Off

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

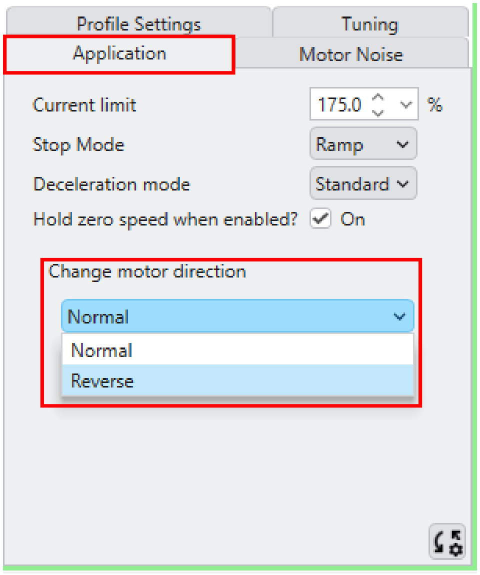


Time/ Div(s) 1

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



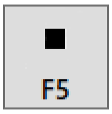
Verify that the movement is stable and that the forward direction of rotation is correct for the application. If the movement is unstable, modify the tuning slider until the desired performance is achieved. If the motor is turning the wrong way when jogging forward, remove 24V from the STO Input, select 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.



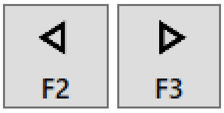
Current limit 175.0 %
Stop Mode Ramp
Deceleration mode Standard
Hold zero speed when enabled? ☑ On
Change motor direction Normal

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.

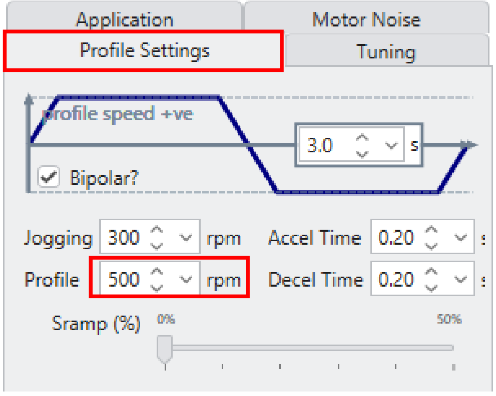
8. Verify the step response of the motor and load by enabling automatic running. Before tuning using these commands make sure that if the axis has physical limits that it is placed in the centre of these limits and that Bipolar mode is selected so that the axis moves back and forth. To stop the axis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



3.0 s

☑ Bipolar?

Jogging 300 rpm

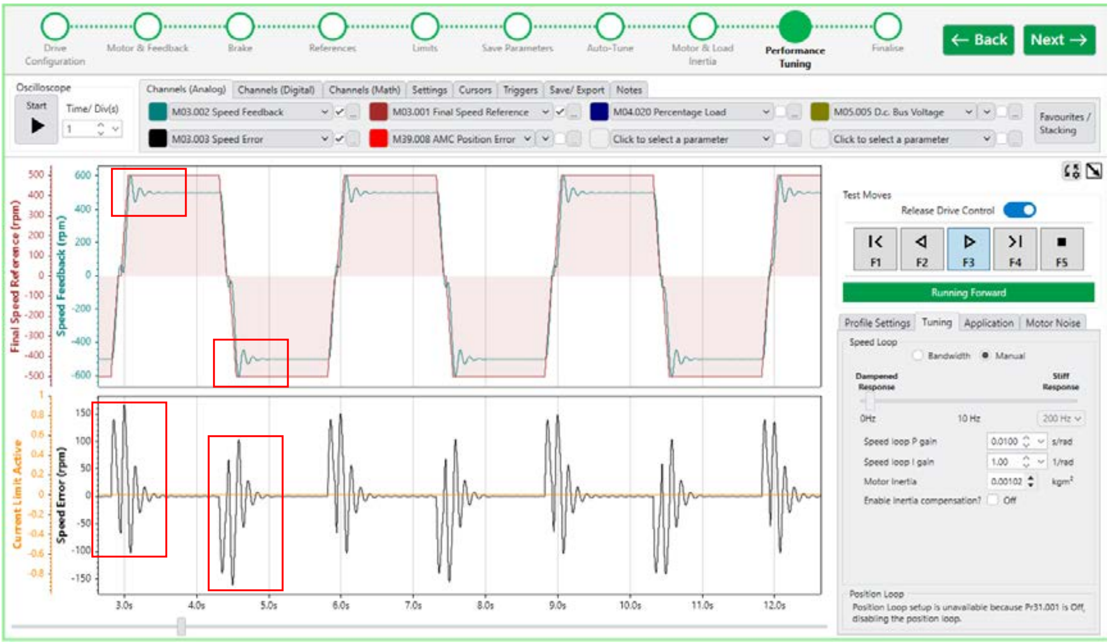
Accel Time 0.20 s

Profile 500 rpm

Decel Time 0.20 s

Sramp (%) 0%

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:



M03.002 Speed Feedback
M03.001 Final Speed Reference
M04.020 Percentage Load
M05.005 D.c. Bus Voltage
M03.003 Speed Error
M39.008 AMC Position Error
Click to select a parameter
Click to select a parameter

○ Bandwidth ● Manual

Speed loop P gain 0.0100 s/rad

Speed loop I gain 1.00 1/rad

Motor Inertia 0.00102 kgm²

Enable Inertia compensation? ☐ Off

Position Loop setup is unavailable because Pr31.001 is Off, disabling the position loop.

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:

M03.002 Speed Feedback
M03.001 Final Speed Reference
M04.020 Percentage Load
M05.005 D.c. Bus Voltage
M03.003 Speed Error
M39.008 AMC Position Error
Click to select a parameter
Click to select a parameter

● Bandwidth ○ Manual

Speed loop P gain 0.2198 s/rad

Speed loop I gain 27.82 1/rad

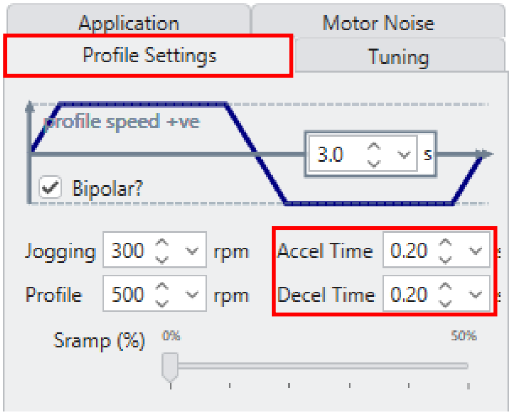
Motor Inertia 0.00102 kgm²

Enable Inertia compensation? ☐ Off

Position Loop setup is unavailable because Pr31.001 is Off, disabling the position loop.

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.



3.0 s

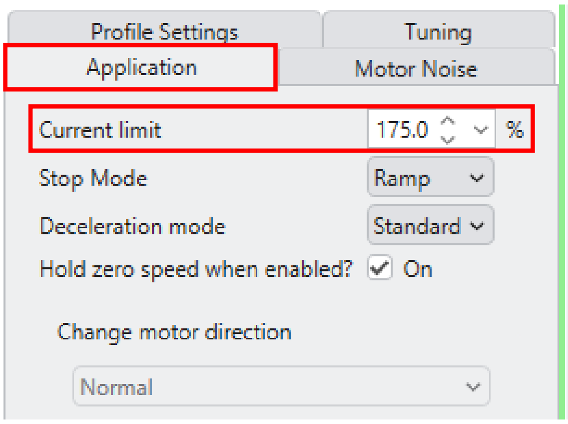
☑ Bipolar?

Jogging 300 rpm

Profile 500 rpm

Accel Time 0.20

Decel Time 0.20



Current limit 175.0 %

Stop Mode Ramp

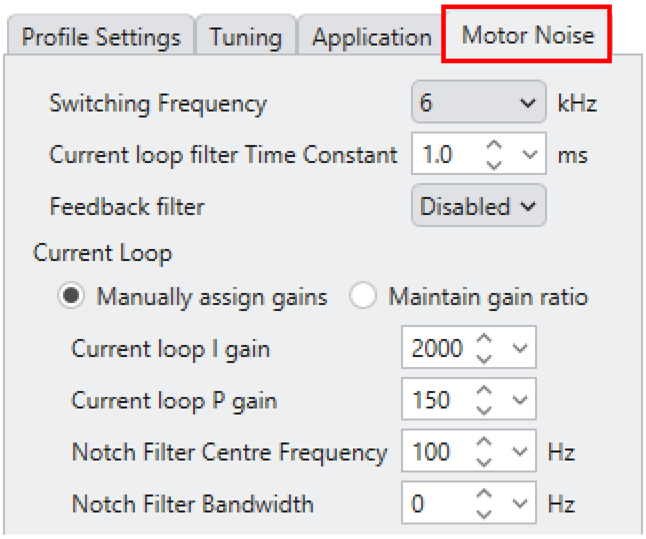
Deceleration mode Standard

Hold zero speed when enabled? ☑ On

Change motor direction

Normal

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



Switching Frequency 6 kHz

Current loop filter Time Constant 1.0 ms

Feedback filter Disabled

Current Loop

● Manually assign gains ○ Maintain gain ratio

Current loop I gain 2000

Current loop P gain 150

Notch Filter Centre Frequency 100 Hz

Notch Filter Bandwidth 0 Hz

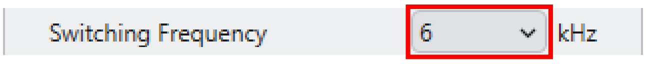
For most applications it is recommended to use a Current Loop Filter Time Constant of 1.0ms, unless the application is ultra-dynamic and needs a very fast response where a lower value is needed. A 1.0ms filter gives a 160Hz response bandwidth, where most systems will be well tuned at between 50Hz and 100Hz.

Current loop filter Time Constant 1.0 ms

This helps in the following situations:

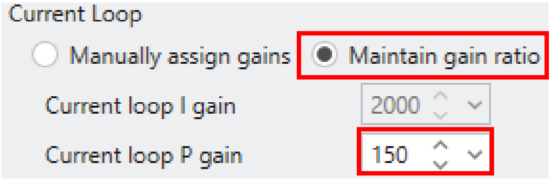
- The speed reference or speed feedback contains a noise component e.g. if the feedback device is of low resolution.
- If the speed loop gains have been raised to the point where the motor noise has become unacceptable, and the required speed loop performance hasn't been reached.
- The application often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

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



Switching Frequency 6 kHz

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.



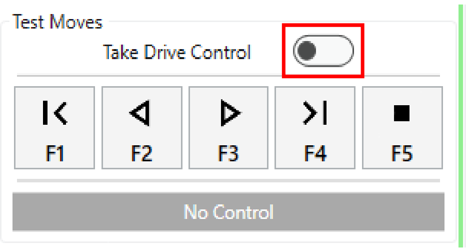
Current Loop

○ Manually assign gains ● Maintain gain ratio

Current loop I gain 2000

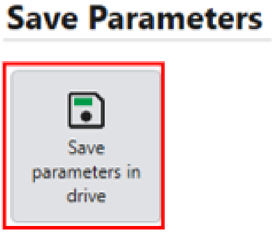
Current loop P gain 150

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 TwinCAT 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:



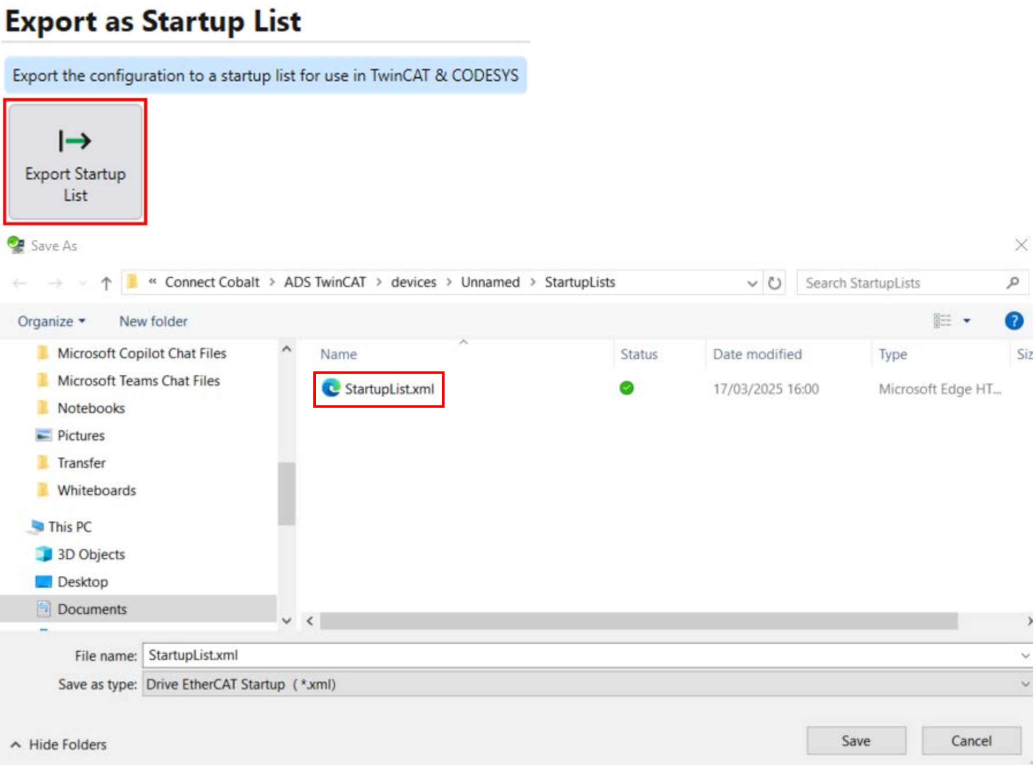
Click Next to move to the “Finalise” page.

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 TwinCAT PLC/IPC. The startup list is in the form of an startup list.xml file that may be imported into TwinCAT.

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



Export as Startup List

Export the configuration to a startup list for use in TwinCAT & CODESYS

File name: StartupList.xml

Save as type: Drive EtherCAT Startup (*.xml)

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

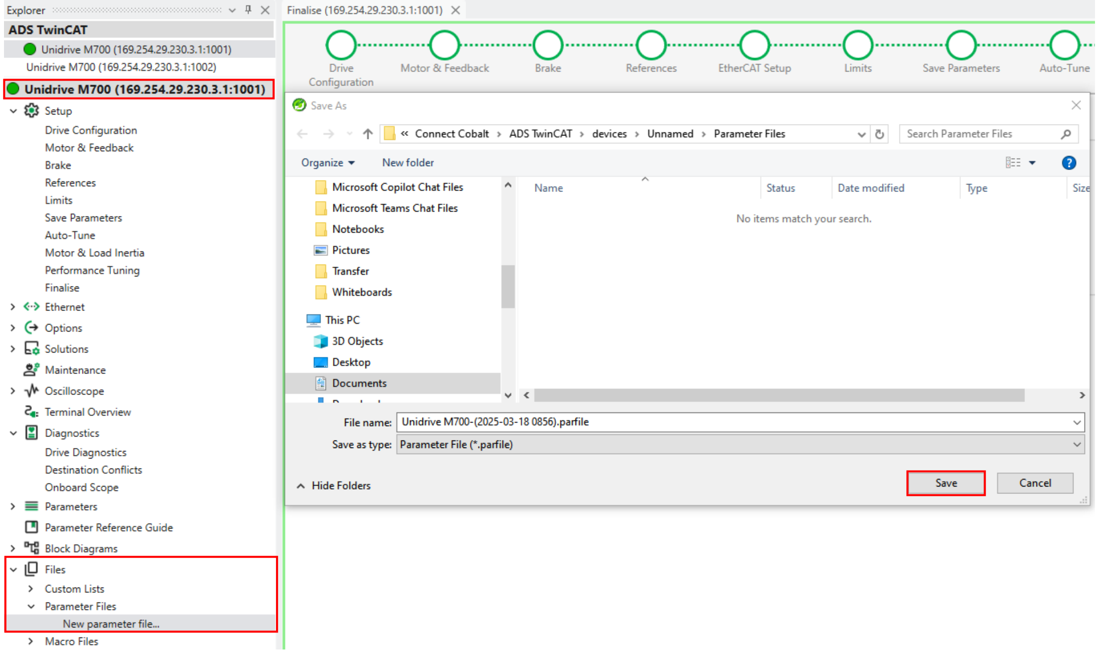
See section **3.2 How to use a startup list in TwinCAT** for more information on how to import a startup list file into TwinCAT.

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 couple of standard ways in which this feature is used:

    - 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.



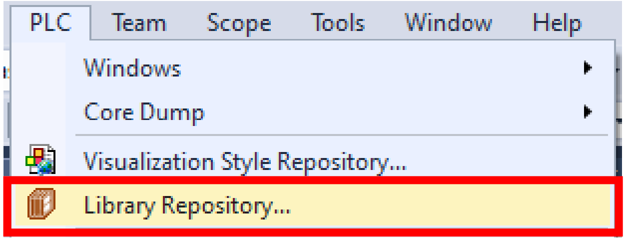
File name: Unidrive M700-(2025-03-18 0856).parfile

Save as type: Parameter File (*.parfile)

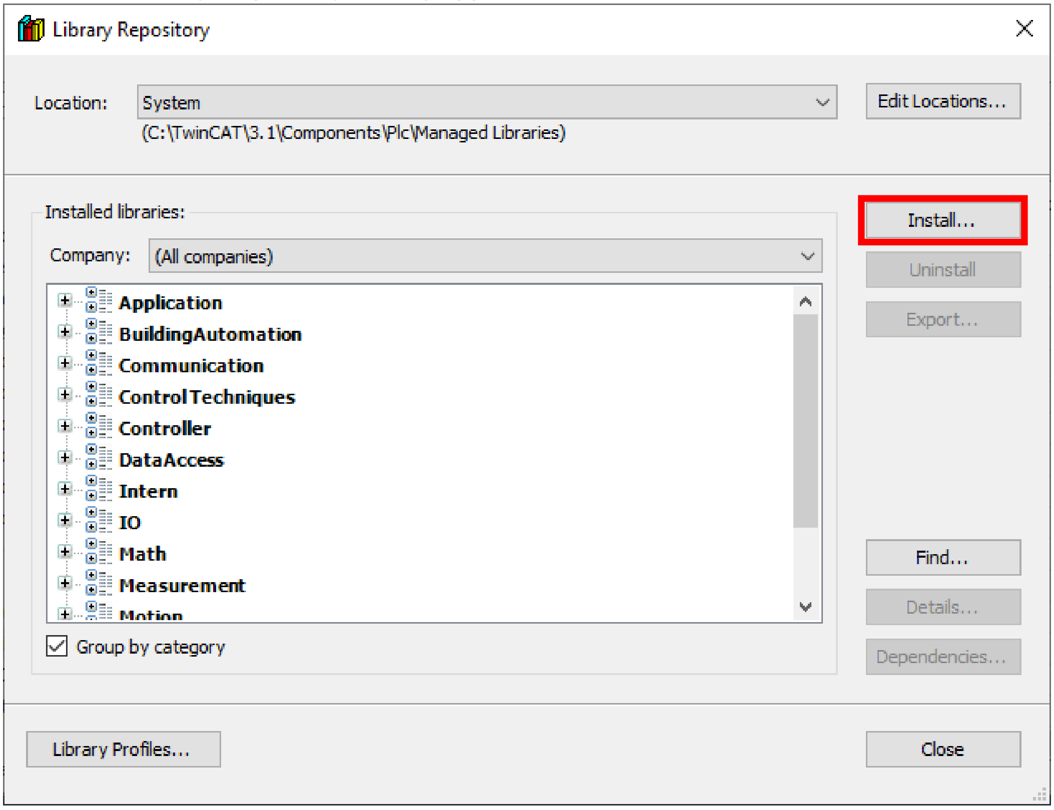
2.8 Installing the CT NC Axis Interface library to the TwinCAT Library Repository

This section describes how to add the CT NC Axis Interface Library to TwinCAT so that it may be used in a project:

1. From the TwinCAT tool bar select “PLC” > “Library Repository…”.



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

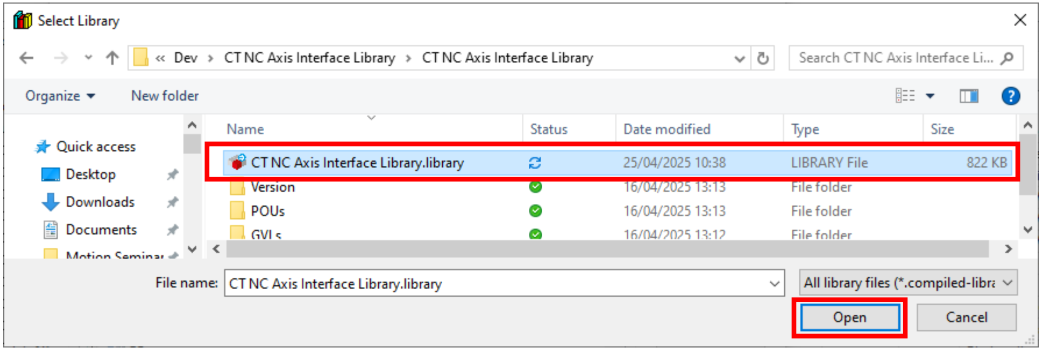


Location: System
(C:\TwinCAT\3.1\Components\Plc\Managed Libraries)

Company: (All companies)

☑ Group by category

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



« Dev › CT NC Axis Interface Library › CT NC Axis Interface Library

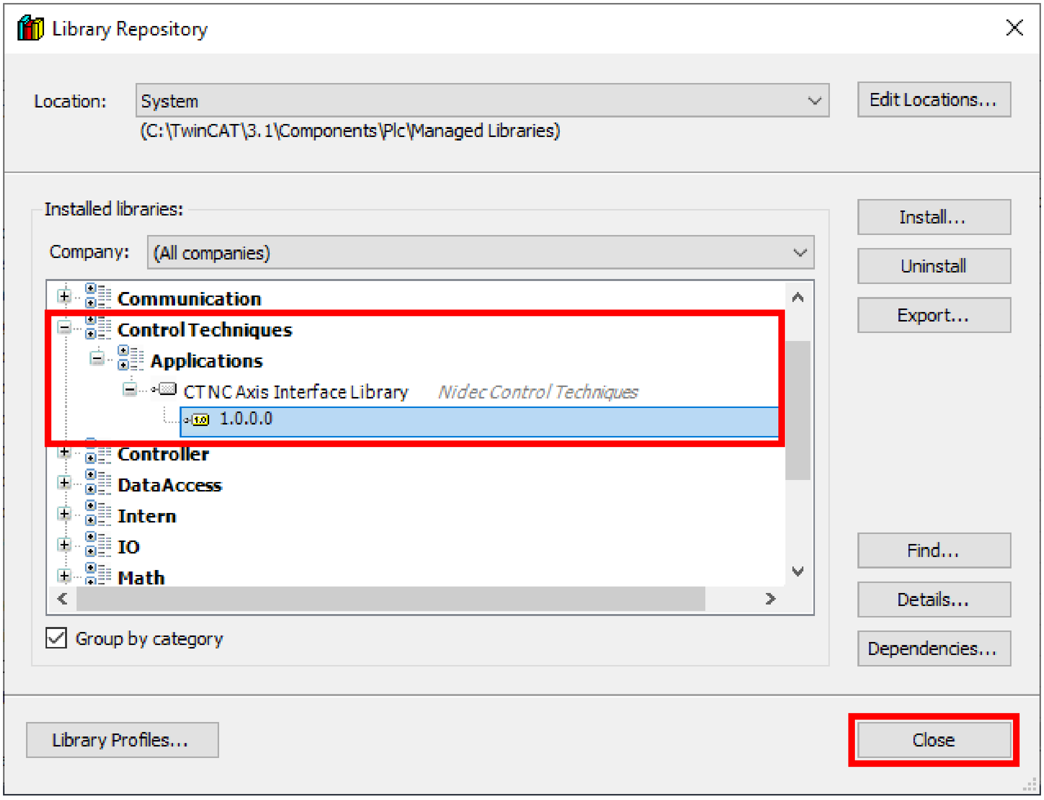
Search CT NC Axis Interface Li...

Name	Status	Date modified	Type	Size
CT NC Axis Interface Library.library		25/04/2025 10:38	LIBRARY File	822 KB
Version		16/04/2025 13:13	File folder	
POUs		16/04/2025 13:13	File folder	
GVLs		16/04/2025 13:12	File folder	

File name: CT NC Axis Interface Library.library

All library files (*.compiled-libra

4. The library is now added to the “Library Repository” and may be added to a TwinCAT project. Click “Close” to finish adding libraries.



2.9 Adding a PLC project

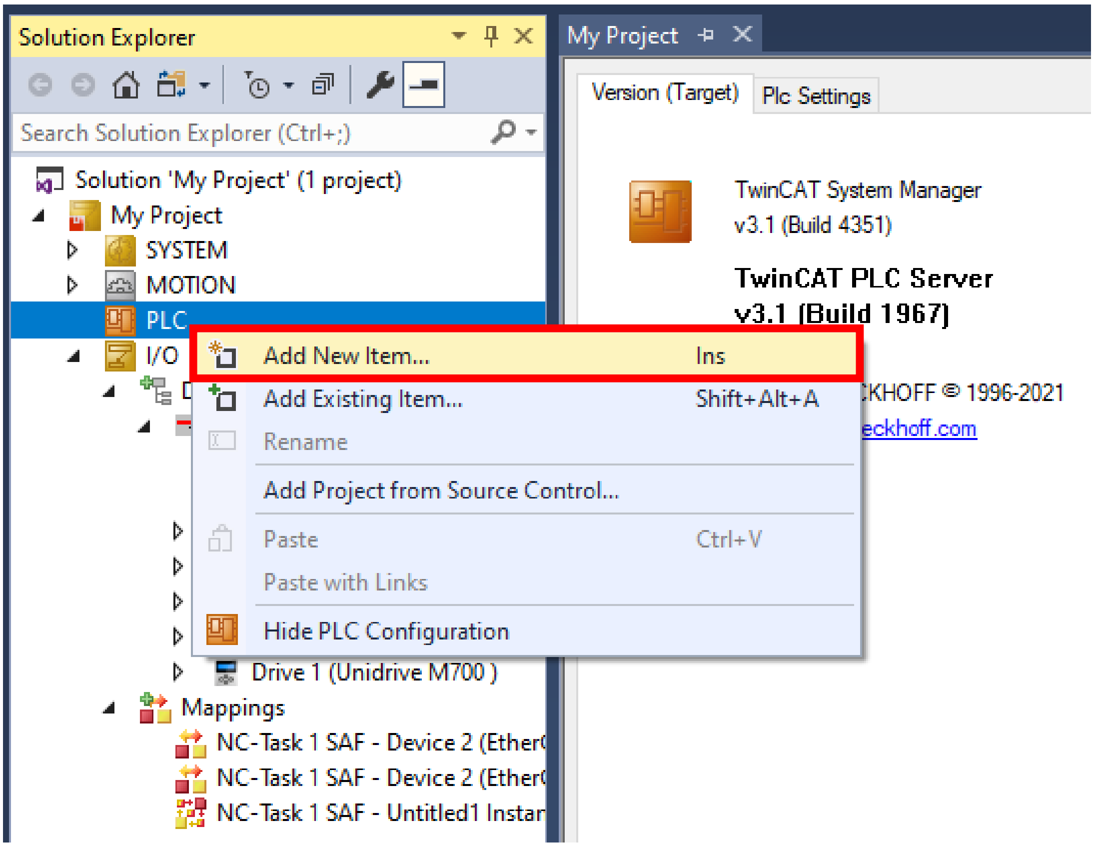
The CT NC Axis Interface Library must be added to the project before it can be called in a POU.

For the library to work correctly, there are several other libraries that must be added to the project.

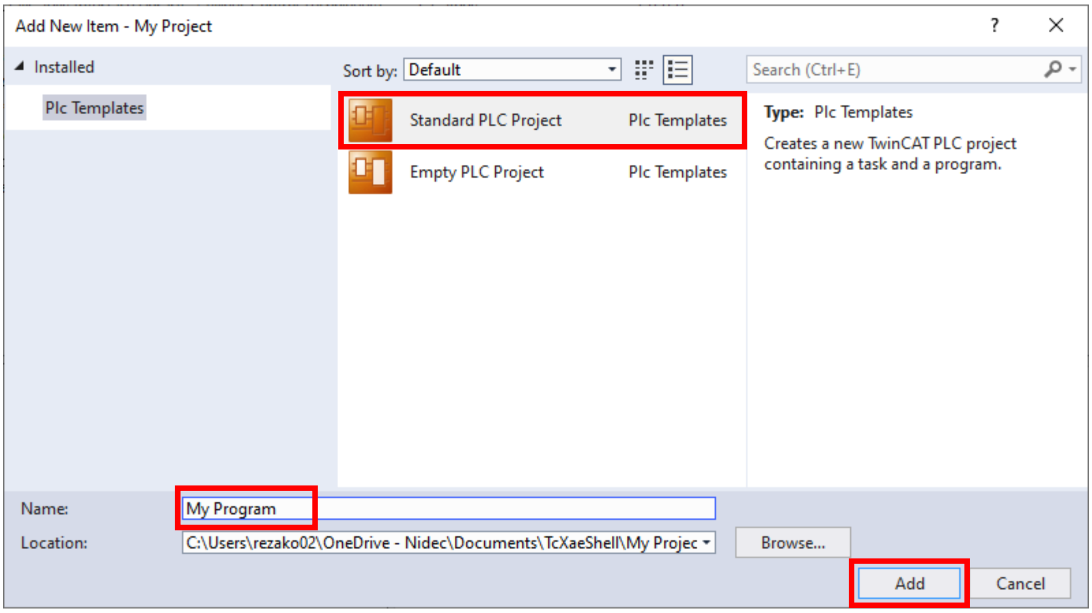
This section describes how to add the necessary libraries to the project for the NC Axis Interface Library to function correctly.

HINT: It is assumed that the CT NC Axis Interface Library has already been added to the Library Repository, section **2.8 Installing the CT NC Axis Interface library to the TwinCAT Library Repository** describes how to do this.

1. From the project tree, right click on the “PLC” item and click “Add New Item…”.



2. The “Add New Item dialog box will appear. Select “Standard PLC Project” and give the project a suitable name before clicking the “Add” button.



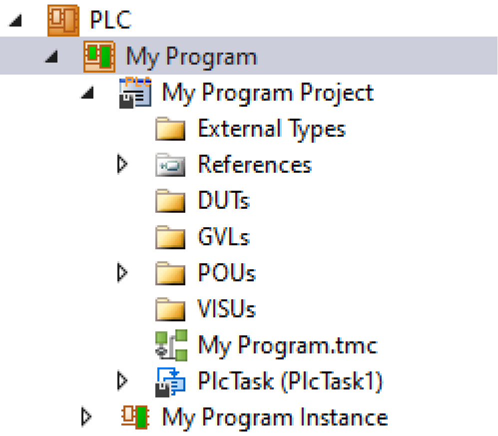
Type: Plc Templates

Creates a new TwinCAT PLC project containing a task and a program.

Name: My Program

Location: C:\Users\rezako02\OneDrive - Nidec\Documents\TcXaeShell\My Projec

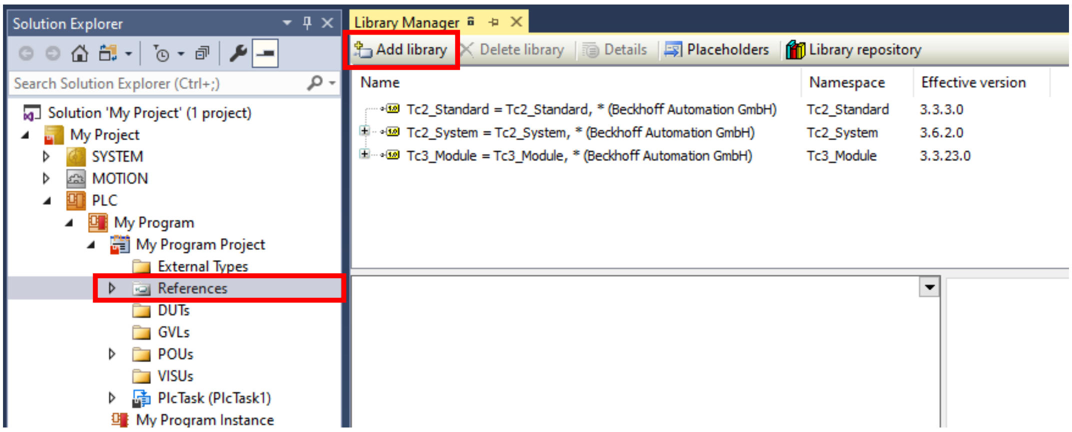
The following objects will now be visible under the “PLC” object in the project tree.



- PLC
  - My Program
    - My Program Project
      - External Types
      - References
      - DUTs
      - GVLs
      - POUs
      - VISUs
      - My Program.tmc
      - PlcTask (PlcTask1)
    - My Program Instance

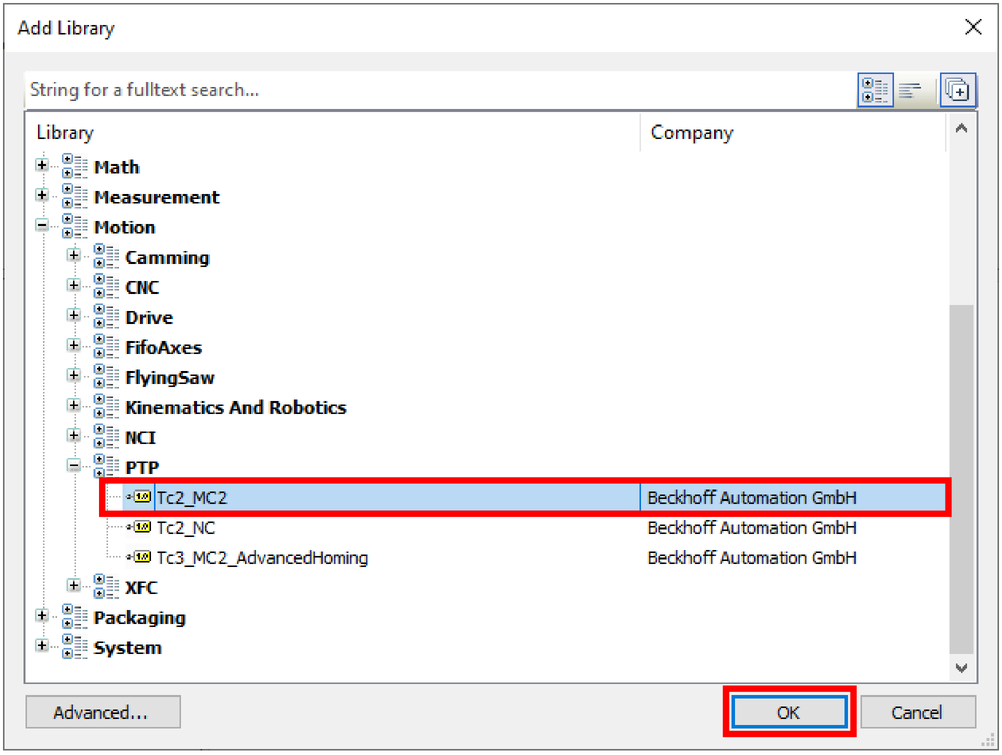
2.10 Adding the CT NC Axis Interface Library to the TwinCAT References

1. Double click on the “References” item from the project tree to open the library manager window. Select “Add Library”



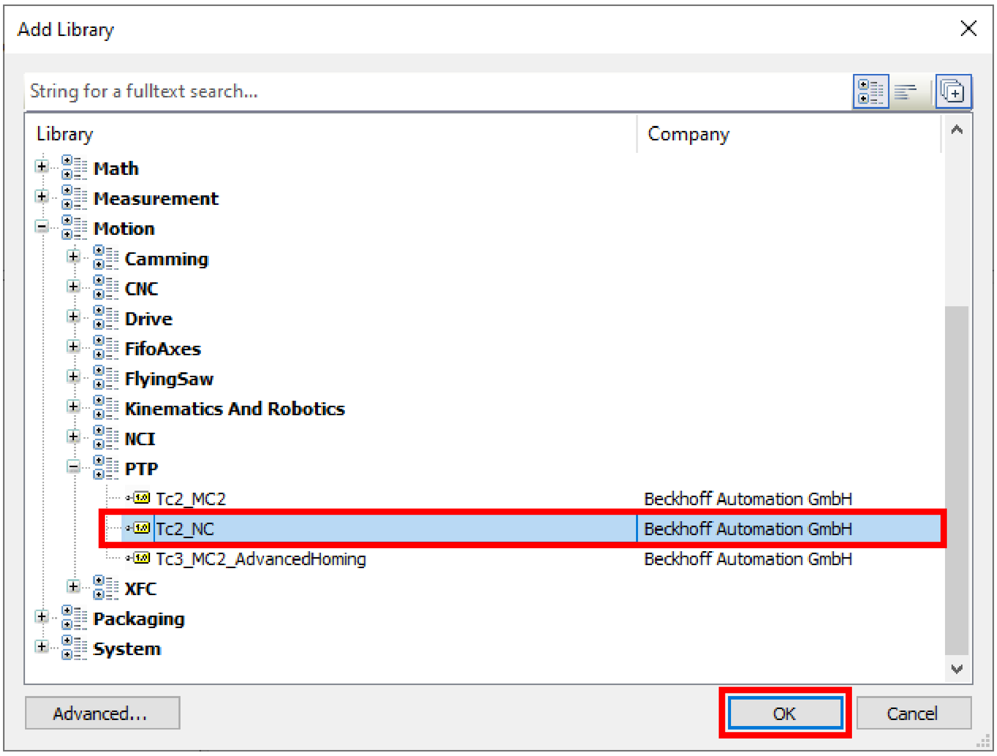
Name	Namespace	Effective version
Tc2_Standard = Tc2_Standard, * (Beckhoff Automation GmbH)	Tc2_Standard	3.3.3.0
Tc2_System = Tc2_System, * (Beckhoff Automation GmbH)	Tc2_System	3.6.2.0
Tc3_Module = Tc3_Module, * (Beckhoff Automation GmbH)	Tc3_Module	3.3.23.0

2. The “Add Library” dialog will be displayed. Navigate to “Motion > PTP” and add the “TC2_MC2 library.

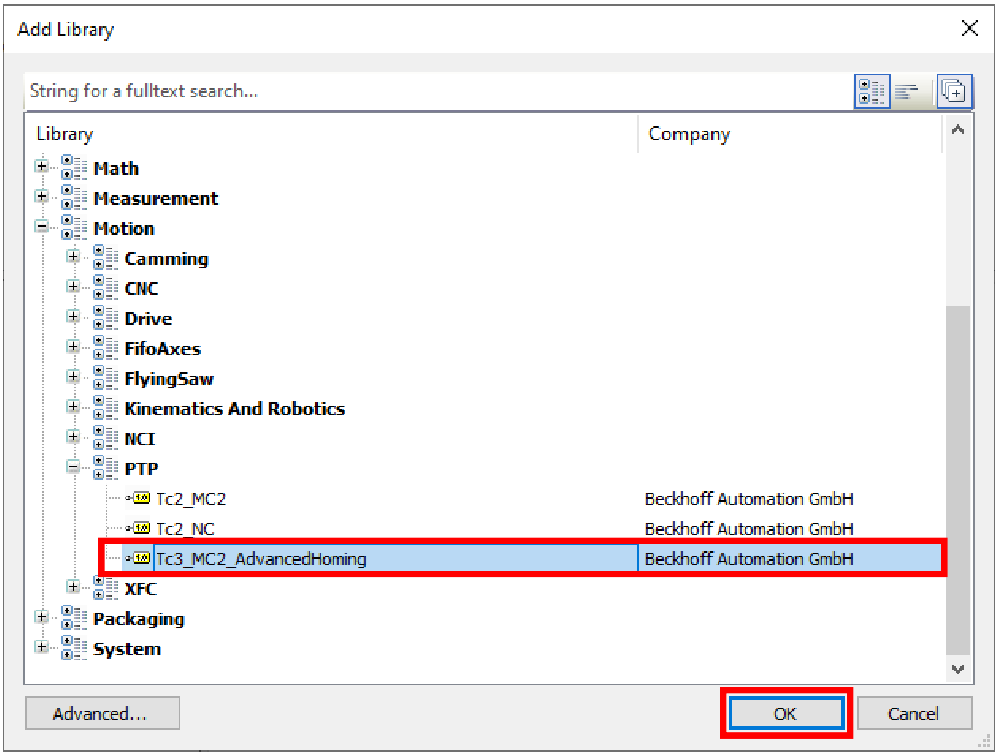


Library	Company
Tc2_MC2	Beckhoff Automation GmbH
Tc2_NC	Beckhoff Automation GmbH
Tc3_MC2_AdvancedHoming	Beckhoff Automation GmbH

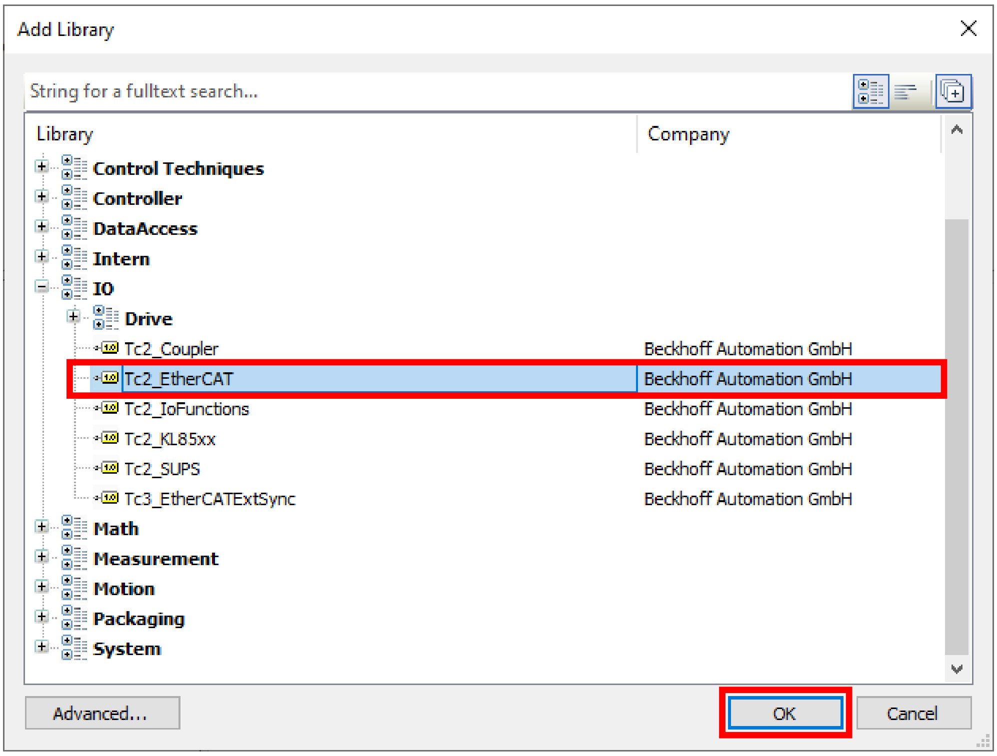
3. Select “Add Library” then navigate to “Motion > PTP” and add the “TC2_NC” library.



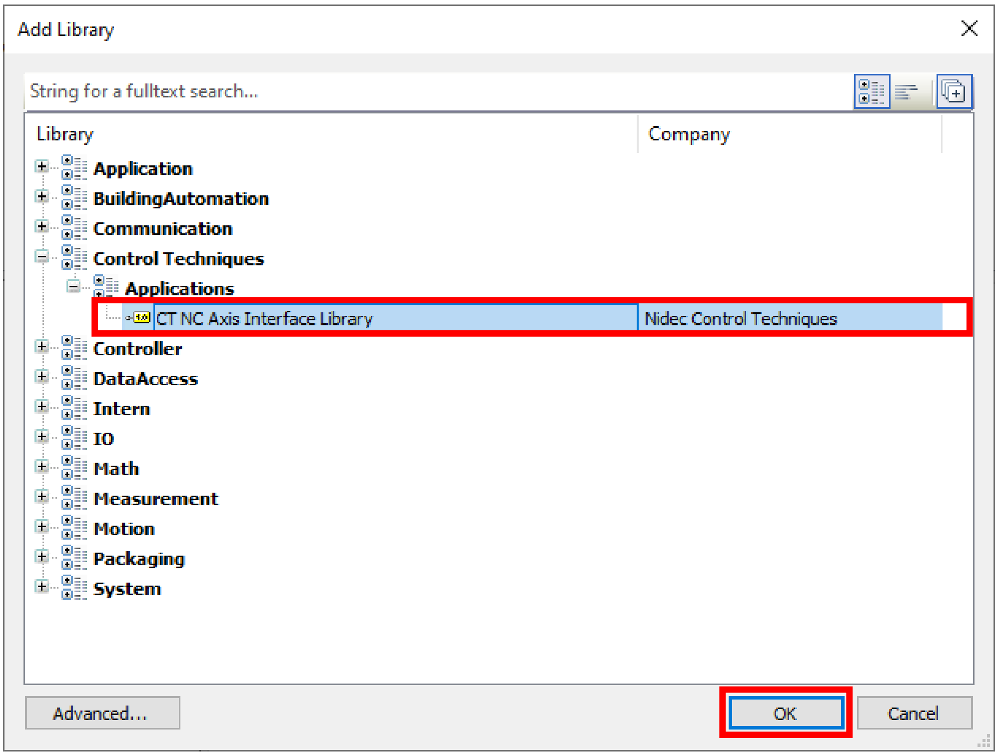
4. Select “Add Library” then navigate to “Motion > PTP” and add the “TC3_MC2_AdvancedHoming” library.



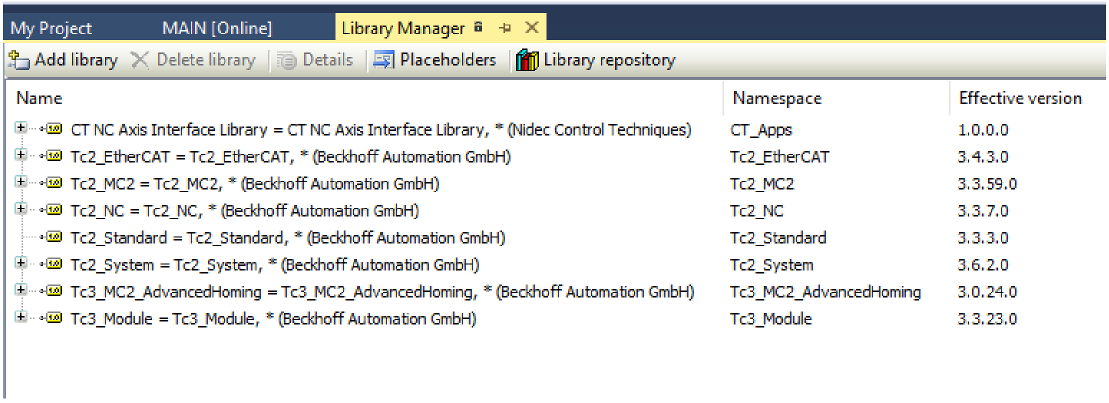
5. Select “Add Library” then navigate to “IO” and add the “TC2_EtherCAT” library.



6. Select “Add Library” then navigate to “Control Techniques > Applications” and add the “CT NC Axis Interface Library” library.



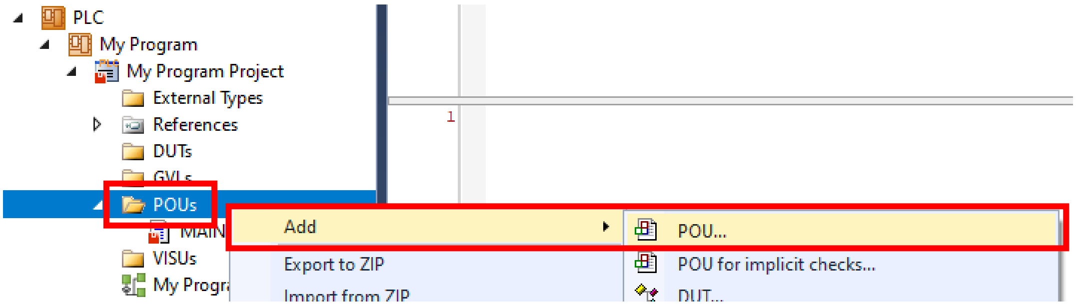
7. The Library Manager will now appear in the following way. Ensure that all of the libraries shown below are present in the project.



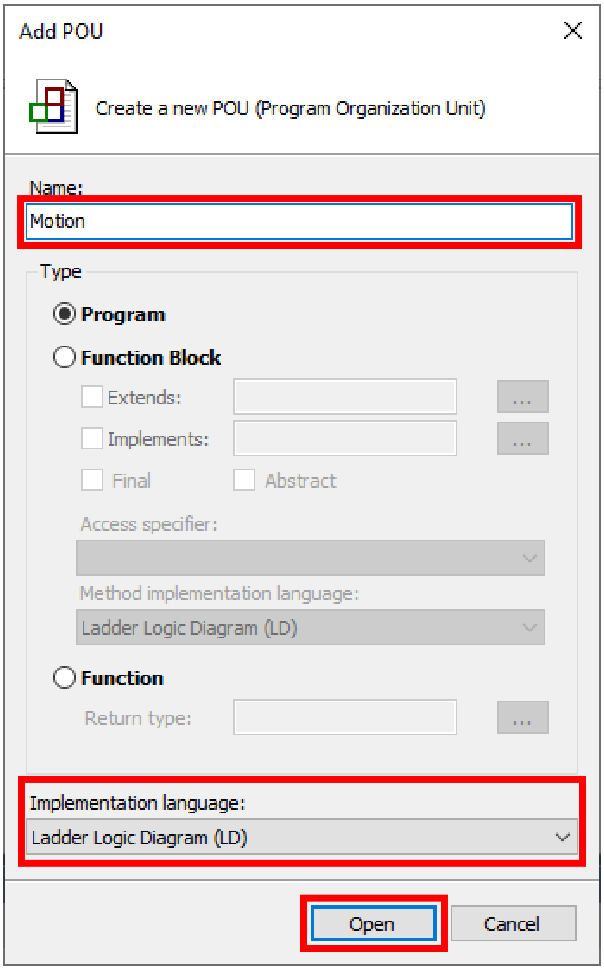
Name	Namespace	Effective version
CT NC Axis Interface Library = CT NC Axis Interface Library, * (Nidec Control Techniques)	CT_Apps	1.0.0.0
Tc2_EtherCAT = Tc2_EtherCAT, * (Beckhoff Automation GmbH)	Tc2_EtherCAT	3.4.3.0
Tc2_MC2 = Tc2_MC2, * (Beckhoff Automation GmbH)	Tc2_MC2	3.3.59.0
Tc2_NC = Tc2_NC, * (Beckhoff Automation GmbH)	Tc2_NC	3.3.7.0
Tc2_Standard = Tc2_Standard, * (Beckhoff Automation GmbH)	Tc2_Standard	3.3.3.0
Tc2_System = Tc2_System, * (Beckhoff Automation GmbH)	Tc2_System	3.6.2.0
Tc3_MC2_AdvancedHoming = Tc3_MC2_AdvancedHoming, * (Beckhoff Automation GmbH)	Tc3_MC2_AdvancedHoming	3.0.24.0
Tc3_Module = Tc3_Module, * (Beckhoff Automation GmbH)	Tc3_Module	3.3.23.0

2.11 Adding a call of the NC Axis Interface Library

1. From the project tree, navigate to “PLC > {PLC Program Name} > {Project Name}” and right click on “POUs” folder and select “Add > POU…”.



2. The “Add POU” window will be displayed, name the POU, select the “Program” radio key, choose “Ladder Logic Diagram (LD)” as the implementation language and click the “Open” button.



Name: Motion

Type

● Program

○ Function Block

☐ Extends:

☐ Implements:

☐ Final ☐ Abstract

Access specifier:

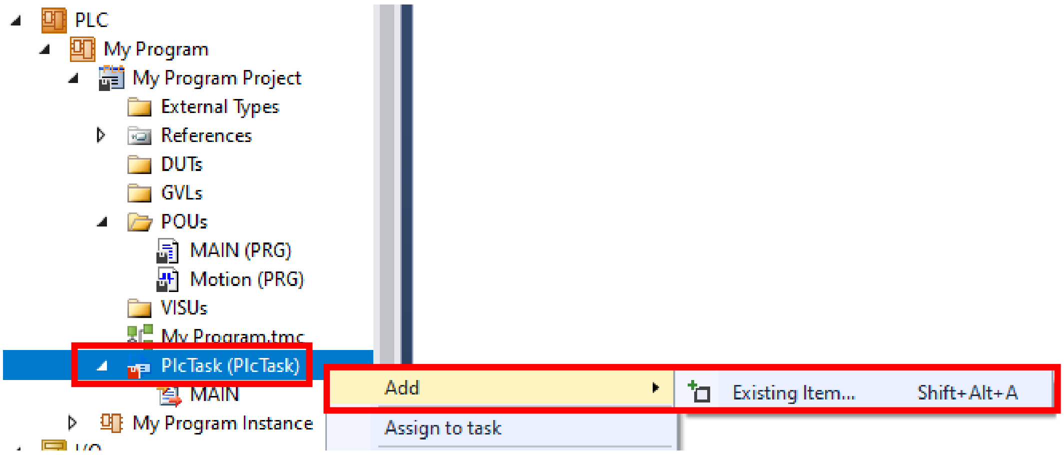
Method implementation language: Ladder Logic Diagram (LD)

○ Function

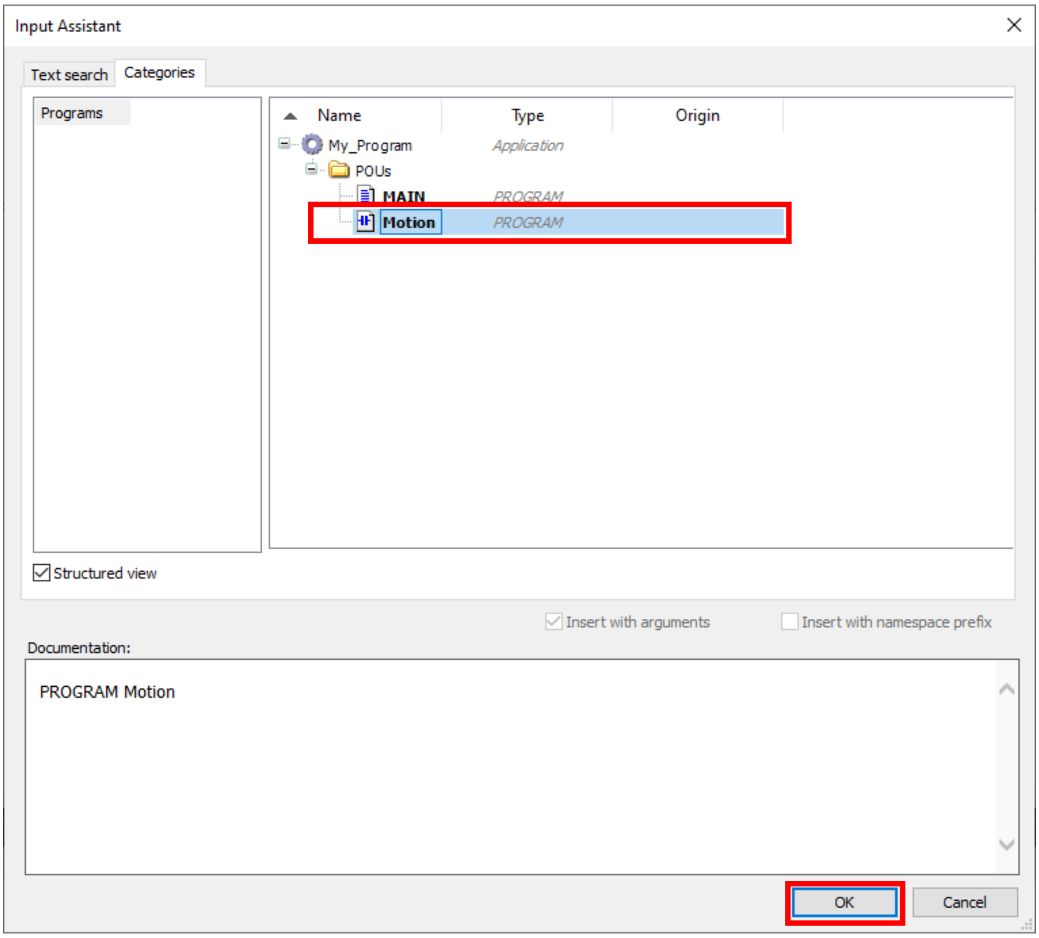
Return type:

Implementation language: Ladder Logic Diagram (LD)

3. The newly added program must be added to the Plc Task so that it is serviced at runtime. To do this, from the project tree, navigate to “PLC > {PLC Program Name} > {Project Name}” and right click on “Plc Task” item and select “Add >Existing Item…”.

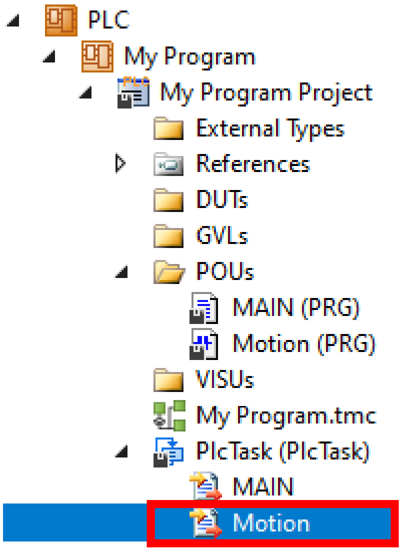


4. The “Input Assistant” dialog box will appear, from the “Categories” tab, navigate to “{Program Name} > POUs” and select the name of the new program and click “OK”.

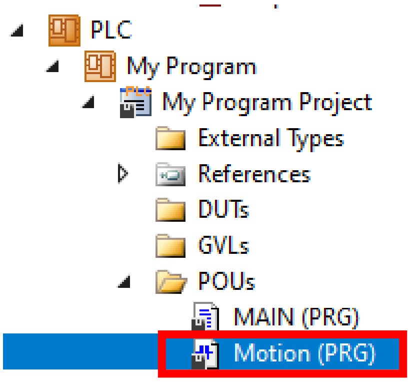


PROGRAM Motion

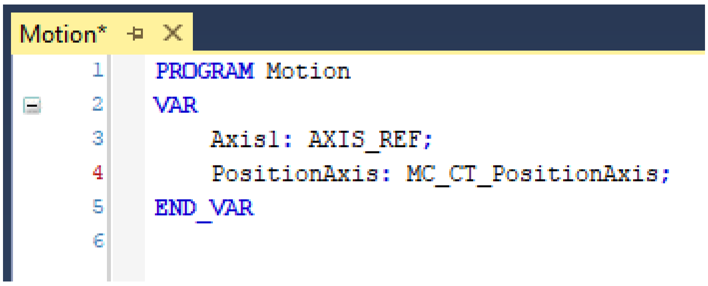
5. The program is now added to the Plc Task



6. In the project tree, navigate to “PLC > {PLC Program Name} > {Project Name} > POUs” double click on the newly added ladder program.

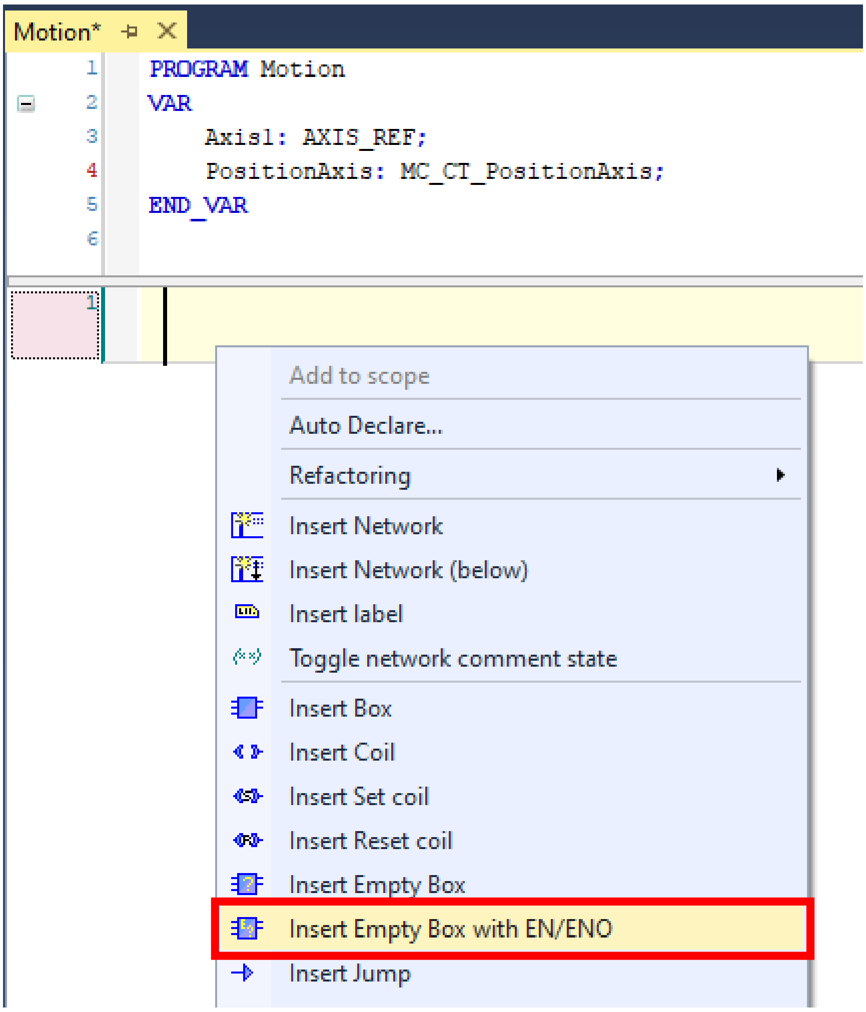


7. The program editor will open. In the declaration area, declare an instance of “AXIS_REF” and an instance of "MC_CT_PositionAxis”. Other axis types are available.



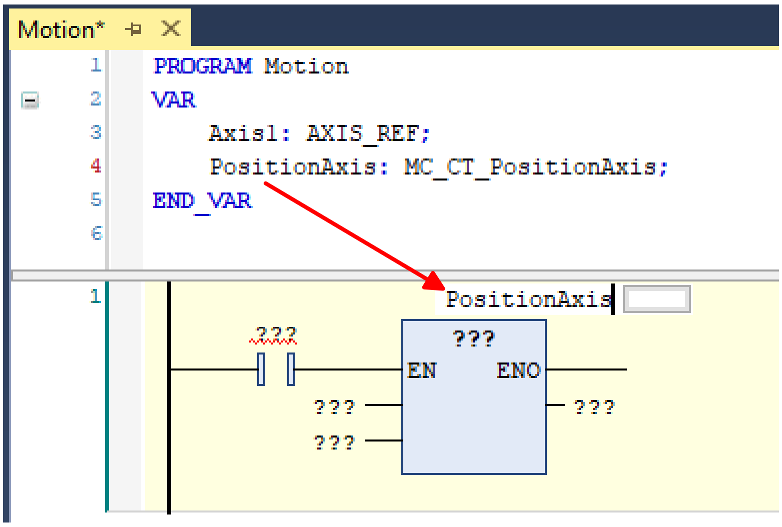
```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    PositionAxis: MC_CT_PositionAxis;
END_VAR
```

8. In the ladder editor, right click on an empty ladder network and select “Inset Empty Box with EN/ENO.



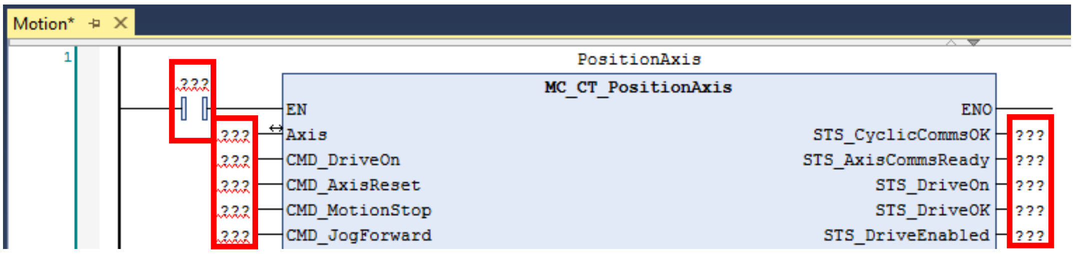
```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    PositionAxis: MC_CT_PositionAxis;
END_VAR
```

9. Copy the instance name of MC_CT_PositionAxis into the instance name of the new box and press enter.



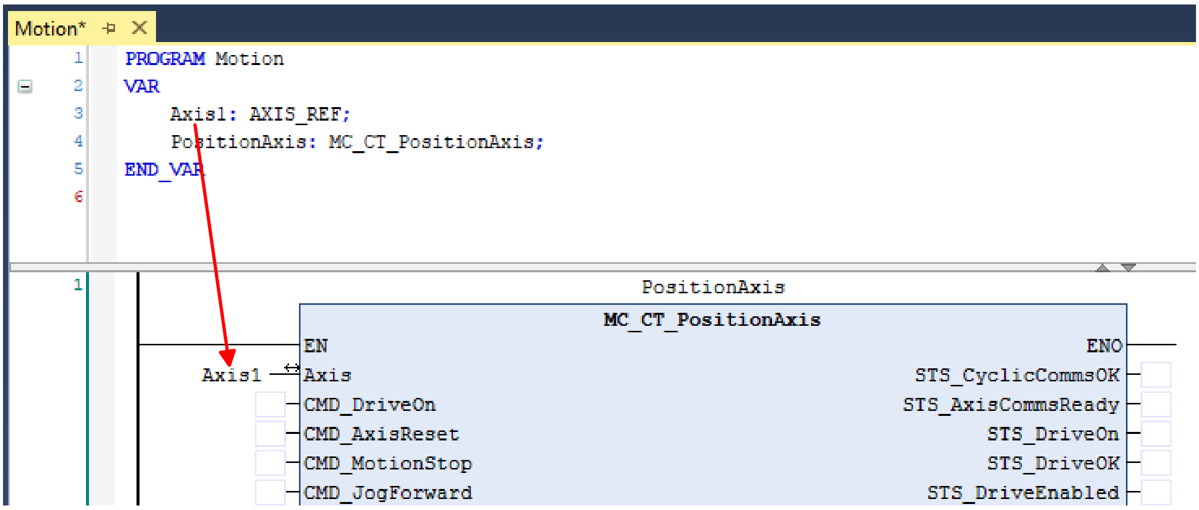
```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    PositionAxis: MC_CT_PositionAxis;
END_VAR
```

10. An instance of MC_CT_PositionAxis will be spawned. Delete the automatically generated contact at the “EN” input, along with the red “???” marks.



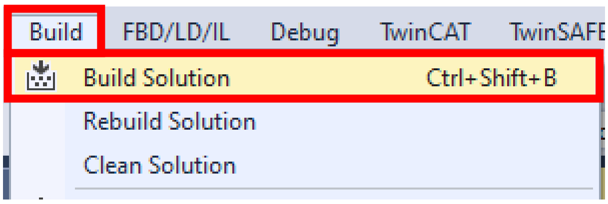
HINT: Automatic generation of these question marks is a feature that can be disabled in the TwinCAT environment itself, see section **3.15 How to Remove “???” from LD diagram POU inputs and outputs** for more information.

11. Copy the instance name of AXIS_REF into the Assignment box of the “Axis” In/Out parameter e.g. assign “Axis1” to the “Axis” input.

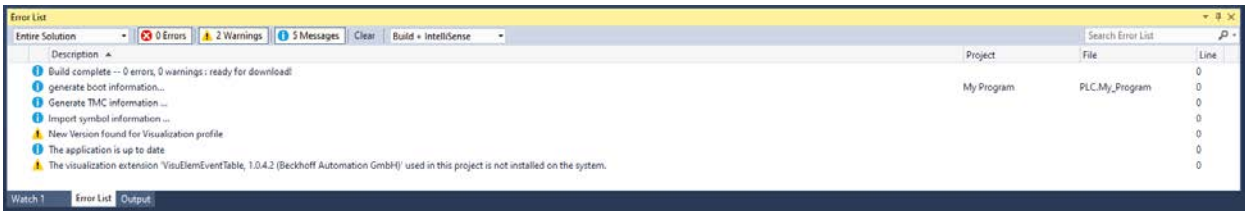


```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    PositionAxis: MC_CT_PositionAxis;
END_VAR
```

12. Perform a build by selecting “Build > Build Solution” from the main toolbar. And ensure the build completes successfully.



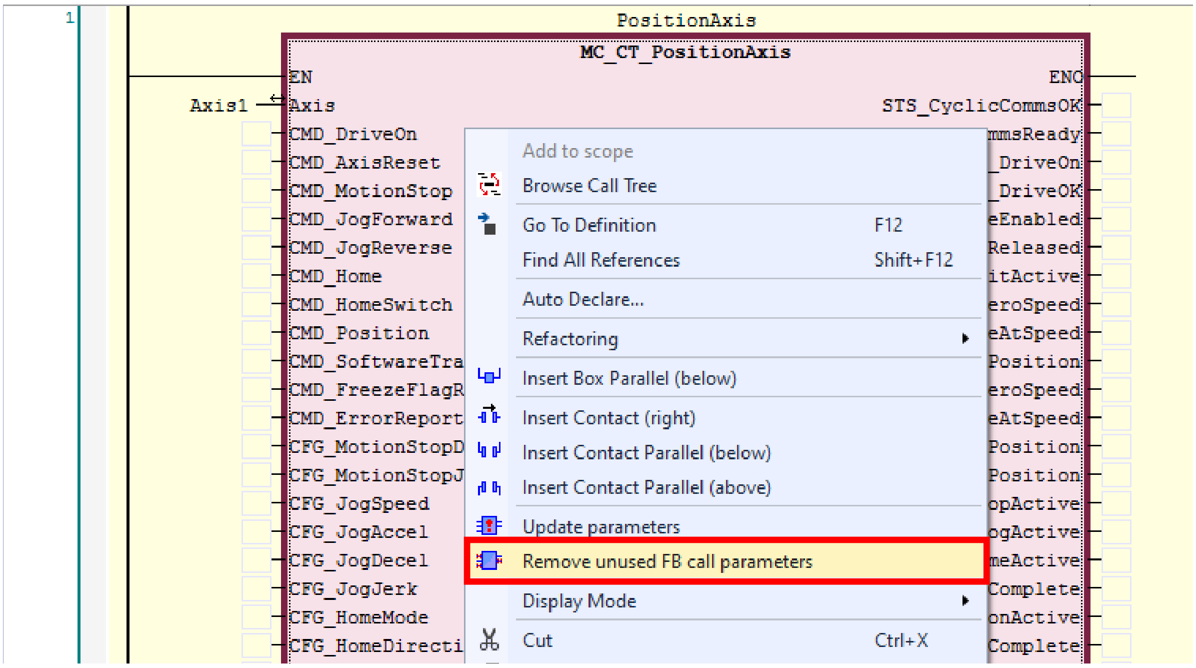
The build output is displayed in the “Error List” viewer, displayed at the bottom of the TwinCAT window by default. Ensure that the build has completed successfully with no errors.



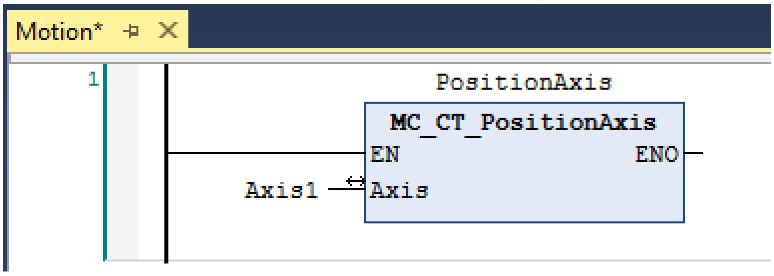
Description	Project	File	Line
Build complete -- 0 errors, 0 warnings : ready for download!			0
generate boot information...	My Program	PLC.My_Program	0
Generate TMC information ...			0
Import symbol information ...			0
New Version found for Visualization profile			0
The application is up to date			0
The visualization extension 'VisuElemEventTable, 1.0.4.2 (Beckhoff Automation GmbH)' used in this project is not installed on the system.			0

2.12 Removing unused function block parameters

In the TwinCAT 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”.

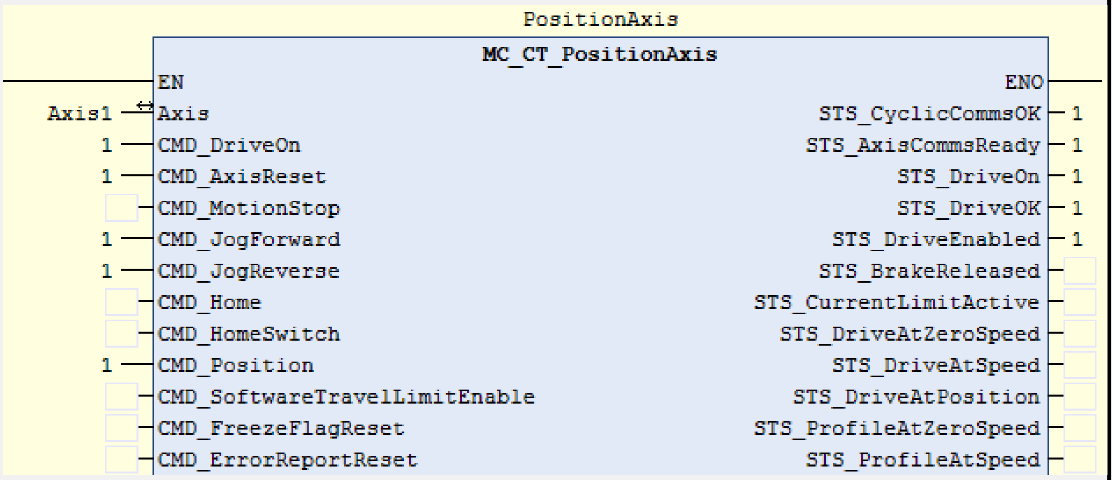


This results in the unused parameters being hidden from the graphical representation of the function block. It can be used to hide any inputs or outputs that have not been connected to a data type.



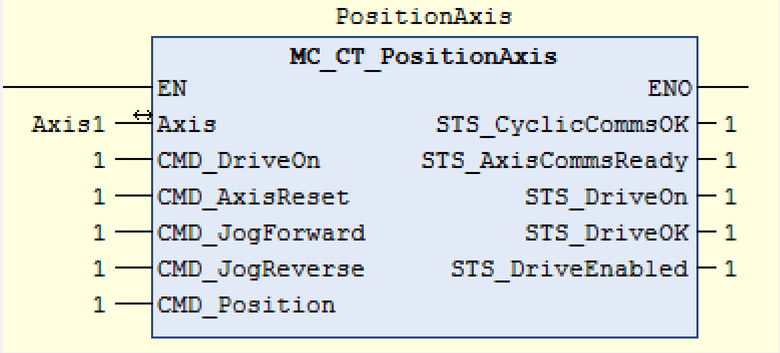
The parameters that have been removed are still accessible by user code, this is purely a visual change to alter the view of the block rather than an actual change of the function block interface and it's functionality.

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.



```
                          PositionAxis
              MC_CT_PositionAxis
      EN                                          ENO
Axis1 Axis                          STS_CyclicCommsOK 1
    1 CMD_DriveOn                  STS_AxisCommsReady 1
    1 CMD_AxisReset                       STS_DriveOn 1
      CMD_MotionStop                      STS_DriveOK 1
    1 CMD_JogForward                 STS_DriveEnabled 1
    1 CMD_JogReverse                STS_BrakeReleased
      CMD_Home                STS_CurrentLimitActive
      CMD_HomeSwitch            STS_DriveAtZeroSpeed
    1 CMD_Position                  STS_DriveAtSpeed
      CMD_SoftwareTravelLimitEnable STS_DriveAtPosition
      CMD_FreezeFlagReset     STS_ProfileAtZeroSpeed
      CMD_ErrorReportReset        STS_ProfileAtSpeed
```

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.



```
                 PositionAxis
            MC_CT_PositionAxis
      EN                               ENO
Axis1 Axis               STS_CyclicCommsOK 1
    1 CMD_DriveOn       STS_AxisCommsReady 1
    1 CMD_AxisReset            STS_DriveOn 1
    1 CMD_JogForward           STS_DriveOK 1
    1 CMD_JogReverse      STS_DriveEnabled 1
    1 CMD_Position
```

2.13 NC Axis Configuration

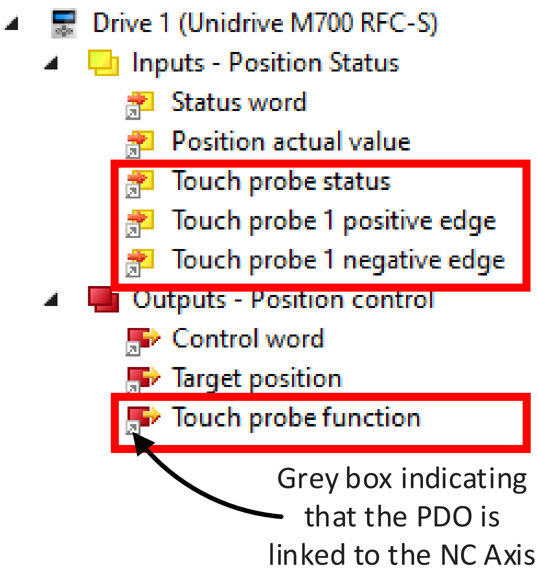
This section describes the basic setup of the NC Axis that was created in section **2.5 Setup an EtherCAT AoE connection from the PLC to the drive**.

2.13.1 Creating PDO mappings for Touch Probe functionality

When using the default ESI, the Touch probe PDOs are not mapped and will therefore result in MC_TouchProbe giving an error ID of 0x701 (1793).

The CT NC Axis Interface Library requires these touch probe mappings for homing and registration functionality to work correctly.

When Touch Probe 1 is mapped correctly it looks like this in the project tree:



- Drive 1 (Unidrive M700 RFC-S)
  - Inputs - Position Status
    - Status word
    - Position actual value
    - Touch probe status
    - Touch probe 1 positive edge
    - Touch probe 1 negative edge
  - Outputs - Position control
    - Control word
    - Target position
    - Touch probe function

Grey box indicating that the PDO is linked to the NC Axis

The following steps show how to create the touch probe PDO mappings:

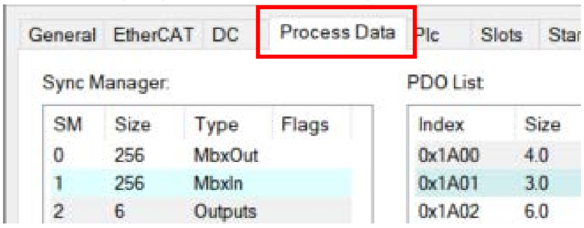
1. Put TwinCAT in Config mode by clicking on the button.

2. Locate the drive to add the links to under “I/O” in the solution explorer. Double click on the Drive name e.g. Drive 1:

   Drive 1 (Unidrive M700 All Modes)

3. When the properties browser opens, select the “Process Data” tab:

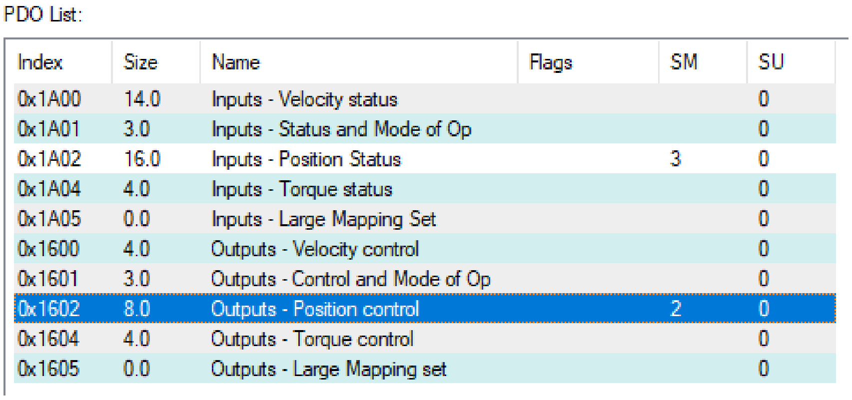
   Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	6	Outputs	

   PDO List:

Index	Size
0x1A00	4.0
0x1A01	3.0
0x1A02	6.0

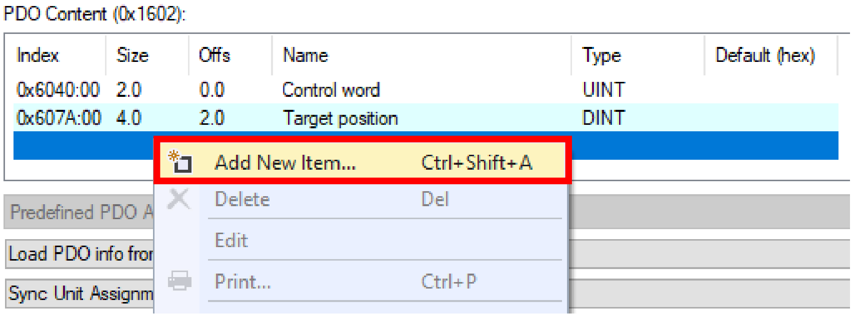
4. To set the PDO List find “Outputs - Position Control” box, and click on it to highlight the selection:



PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	14.0	Inputs - Velocity status			0
0x1A01	3.0	Inputs - Status and Mode of Op			0
0x1A02	16.0	Inputs - Position Status		3	0
0x1A04	4.0	Inputs - Torque status			0
0x1A05	0.0	Inputs - Large Mapping Set			0
0x1600	4.0	Outputs - Velocity control			0
0x1601	3.0	Outputs - Control and Mode of Op			0
0x1602	8.0	Outputs - Position control		2	0
0x1604	4.0	Outputs - Torque control			0
0x1605	0.0	Outputs - Large Mapping set			0

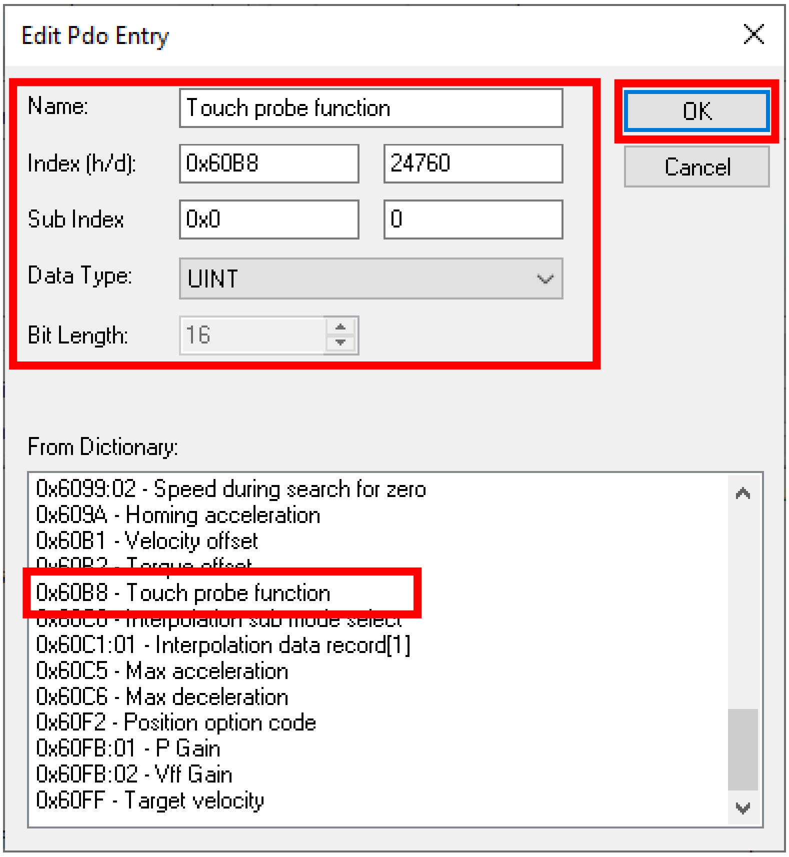
5. Right click in the “PDO Content” box, and right click, then select “Add New Item…”:



PDO Content (0x1602):

Index	Size	Offs	Name	Type	Default (hex)
0x6040:00	2.0	0.0	Control word	UINT	
0x607A:00	4.0	2.0	Target position	DINT	

6. When the “Pdo Entry dialog” appears, scroll through the “From Dictionary:” options and look for 0x60B8 or Touch Probe Function. Click on this entry when it has been located. The details of the mapping will be automatically added. Click “OK” when finished.



Edit Pdo Entry

Name: Touch probe function

Index (h/d): 0x60B8 24760

Sub Index 0x0 0

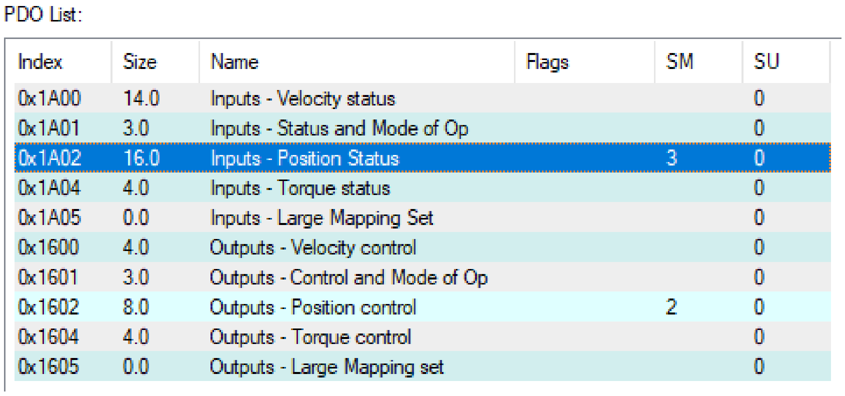
Data Type: UINT

Bit Length: 16

From Dictionary:

0x6099:02 - Speed during search for zero
0x609A - Homing acceleration
0x60B1 - Velocity offset
0x60B2 - Torque offset
0x60B8 - Touch probe function
0x60C0 - Interpolation sub mode select
0x60C1:01 - Interpolation data record[1]
0x60C5 - Max acceleration
0x60C6 - Max deceleration
0x60F2 - Position option code
0x60FB:01 - P Gain
0x60FB:02 - Vff Gain
0x60FF - Target velocity

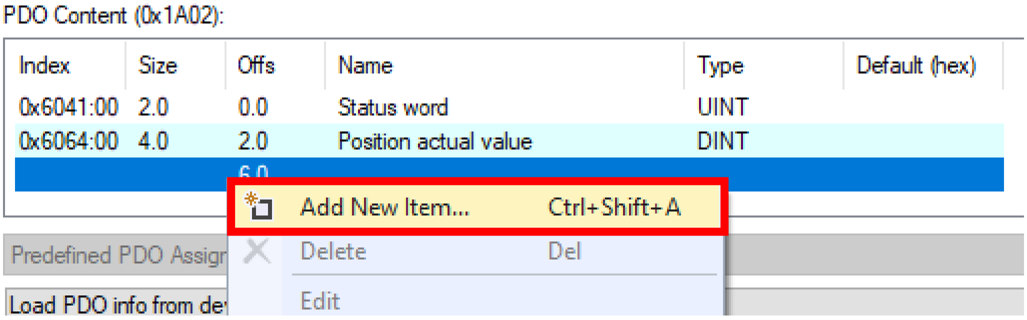
7. This process is repeated for the outputs. Select the PDO list item named “Inputs – Position Status”:



PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	14.0	Inputs - Velocity status			0
0x1A01	3.0	Inputs - Status and Mode of Op			0
0x1A02	16.0	Inputs - Position Status		3	0
0x1A04	4.0	Inputs - Torque status			0
0x1A05	0.0	Inputs - Large Mapping Set			0
0x1600	4.0	Outputs - Velocity control			0
0x1601	3.0	Outputs - Control and Mode of Op			0
0x1602	8.0	Outputs - Position control		2	0
0x1604	4.0	Outputs - Torque control			0
0x1605	0.0	Outputs - Large Mapping set			0

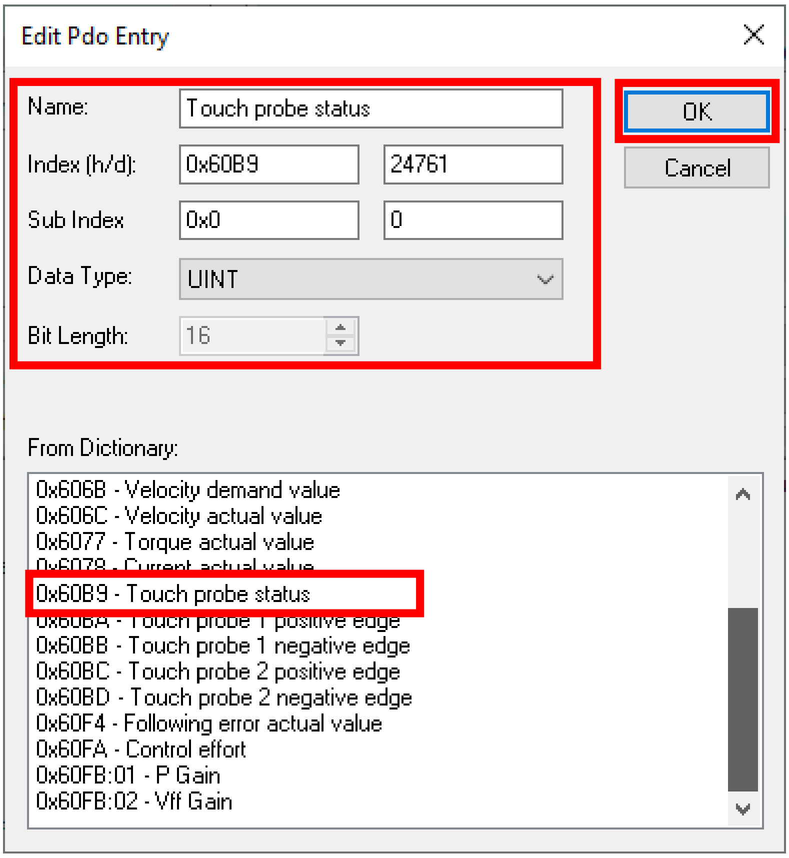
Then add a new item:



PDO Content (0x1A02):

Index	Size	Offs	Name	Type	Default (hex)
0x6041:00	2.0	0.0	Status word	UINT	
0x6064:00	4.0	2.0	Position actual value	DINT	
		6.0			

Scroll through the object dictionary and choose object 0x60B9 – Touch Probe Status and click “OK”:



Edit Pdo Entry

Name: Touch probe status

Index (h/d): 0x60B9 24761

Sub Index 0x0 0

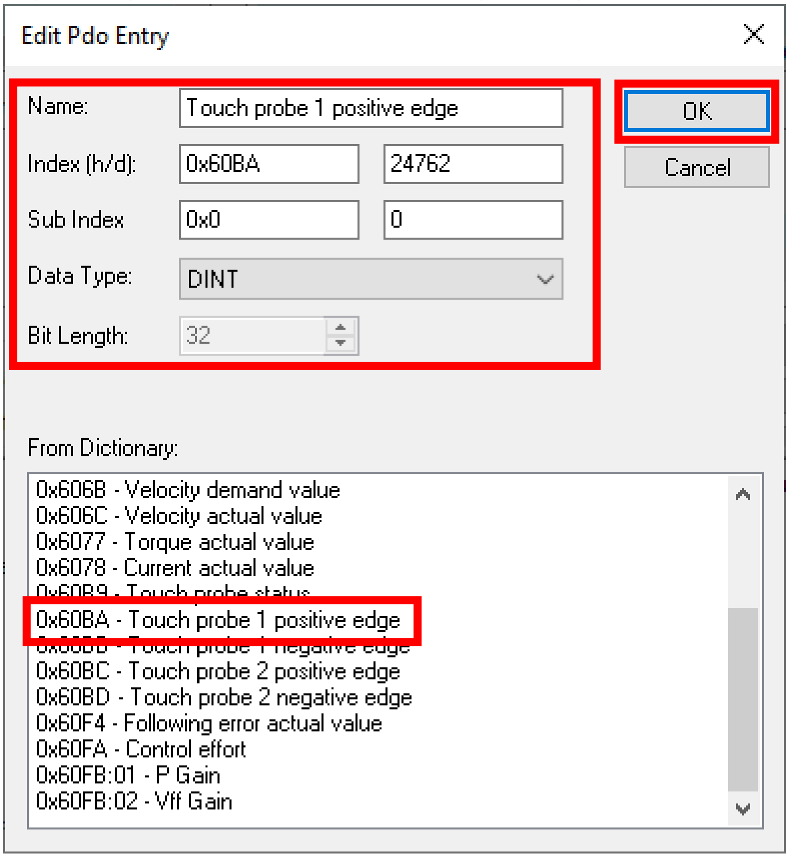
Data Type: UINT

Bit Length: 16

From Dictionary:

0x606B - Velocity demand value
0x606C - Velocity actual value
0x6077 - Torque actual value
0x6078 - Current actual value
0x60B9 - Touch probe status
0x60BA - Touch probe 1 positive edge
0x60BB - Touch probe 1 negative edge
0x60BC - Touch probe 2 positive edge
0x60BD - Touch probe 2 negative edge
0x60F4 - Following error actual value
0x60FA - Control effort
0x60FB:01 - P Gain
0x60FB:02 - Vff Gain

8. Repeat steps 7, this time adding “Touch probe 1 positive edge”.

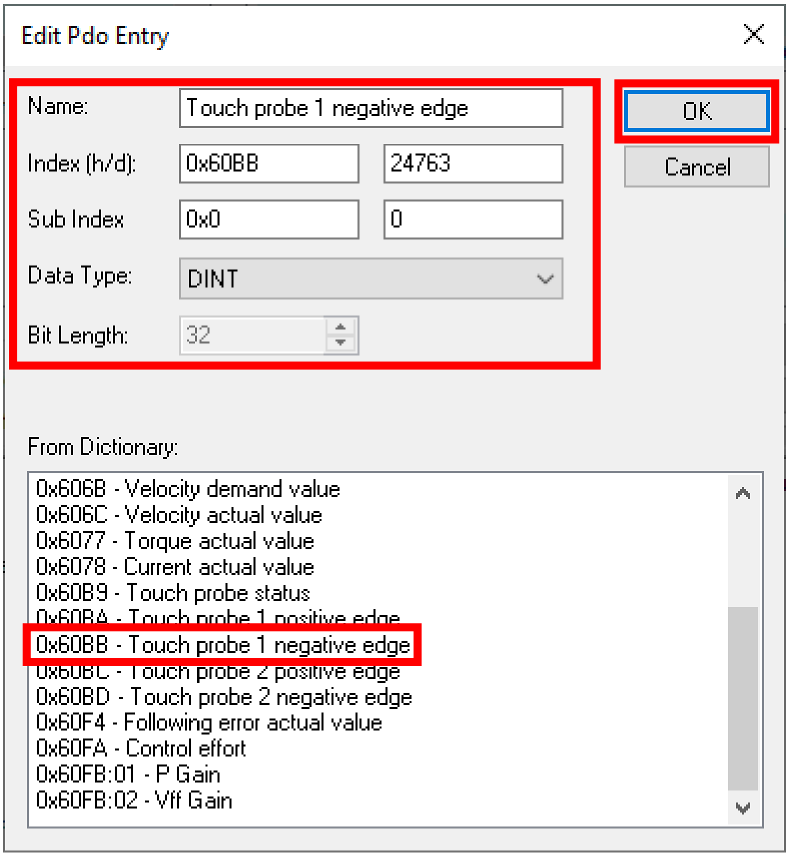


Name: Touch probe 1 positive edge
Index (h/d): 0x60BA 24762
Sub Index 0x0 0
Data Type: DINT
Bit Length: 32

From Dictionary:

0x606B - Velocity demand value
0x606C - Velocity actual value
0x6077 - Torque actual value
0x6078 - Current actual value
0x60B9 - Touch probe status
0x60BA - Touch probe 1 positive edge
0x60BB - Touch probe 1 negative edge
0x60BC - Touch probe 2 positive edge
0x60BD - Touch probe 2 negative edge
0x60F4 - Following error actual value
0x60FA - Control effort
0x60FB:01 - P Gain
0x60FB:02 - Vff Gain

9. Repeat step 7, this time adding “Touch probe 1 negative edge”.



Name: Touch probe 1 negative edge
Index (h/d): 0x60BB 24763
Sub Index 0x0 0
Data Type: DINT
Bit Length: 32

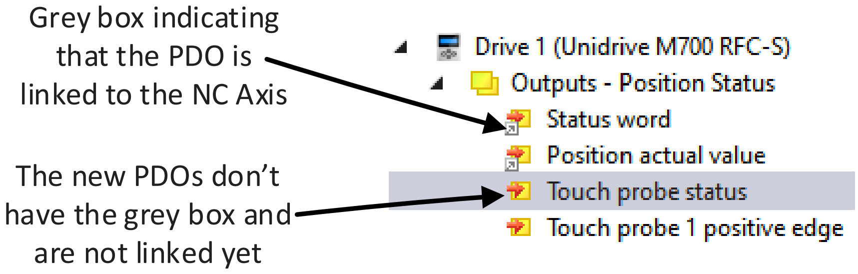
From Dictionary:

0x606B - Velocity demand value
0x606C - Velocity actual value
0x6077 - Torque actual value
0x6078 - Current actual value
0x60B9 - Touch probe status
0x60BA - Touch probe 1 positive edge
0x60BB - Touch probe 1 negative edge
0x60BC - Touch probe 2 positive edge
0x60BD - Touch probe 2 negative edge
0x60F4 - Following error actual value
0x60FA - Control effort
0x60FB:01 - P Gain
0x60FB:02 - Vff Gain

HINT: A minimum of firmware V01.61.00.00 drive firmware and EtherCAT firmware V01.12.00.24 with the associated ESI files are required to use this functionality.

For simplicity, this example continues assuming Touch probe 1 is used with a rising edge only.

10. When the PDO mappings are added they need a further step to link them to the axis.

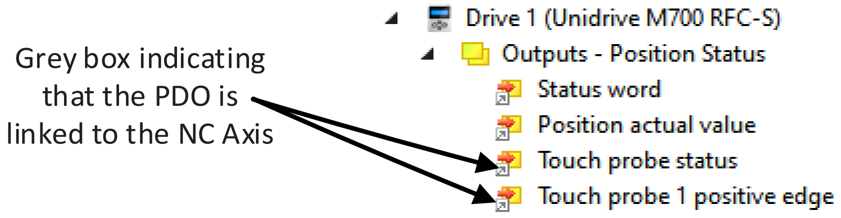


Grey box indicating that the PDO is linked to the NC Axis

The new PDOs don’t have the grey box and are not linked yet

HINT: In more modern versions of TwinCAT the axis linking process happens automatically.

11. The TwinCAT system will then automatically connect the Axis to the drive PDO mapping. The next section **2.13.2 Linking an axis to the dirve** describes how to perform the linking process:

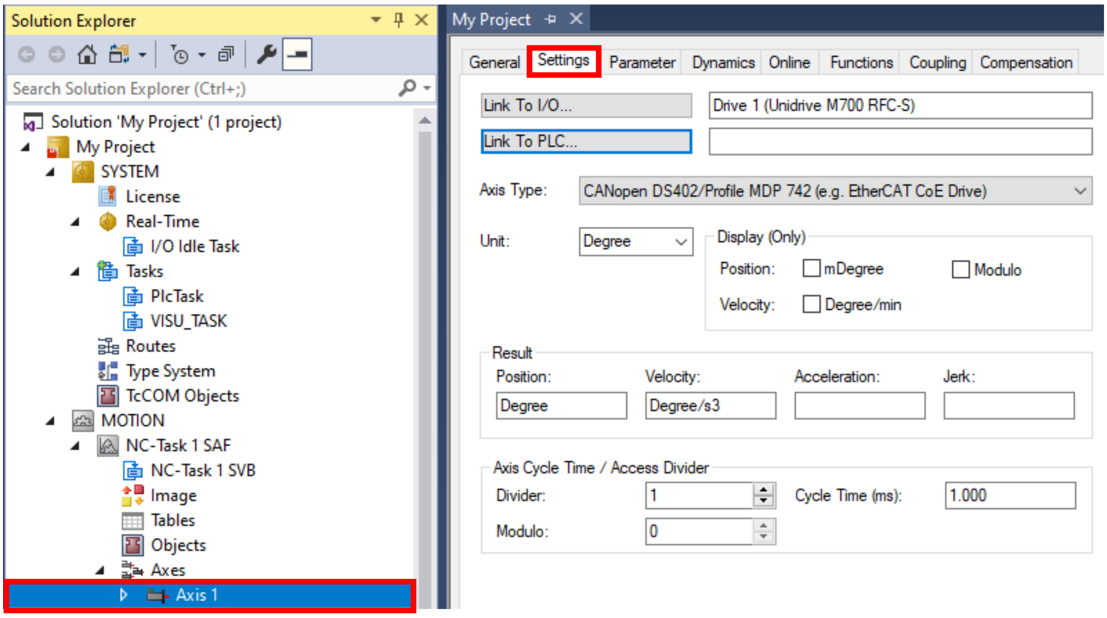


Grey box indicating that the PDO is linked to the NC Axis

2.13.2 Linking an axis to the dirve

An instance of AXIS_REF was declared in section **2.11 Adding a call of the NC Axis Interface Library.** This step describes how to link a motion axis to this PLC instance of AXIS_REF and how to link the motion axis to a physical axis (I/O).

1. From the project tree, navigate to “Project > MOTION > NC-Task 1 SVB > Axes” and double click on “Axis 1” and select the “Settings” tab.



Link To I/O...: Drive 1 (Unidrive M700 RFC-S)

Link To PLC...:

Axis Type: CANopen DS402/Profile MDP 742 (e.g. EtherCAT CoE Drive)

Unit: Degree

Display (Only)

Position: ☐ mDegree ☐ Modulo

Velocity: ☐ Degree/min

Result

Position:	Velocity:	Acceleration:	Jerk:
Degree	Degree/s3		

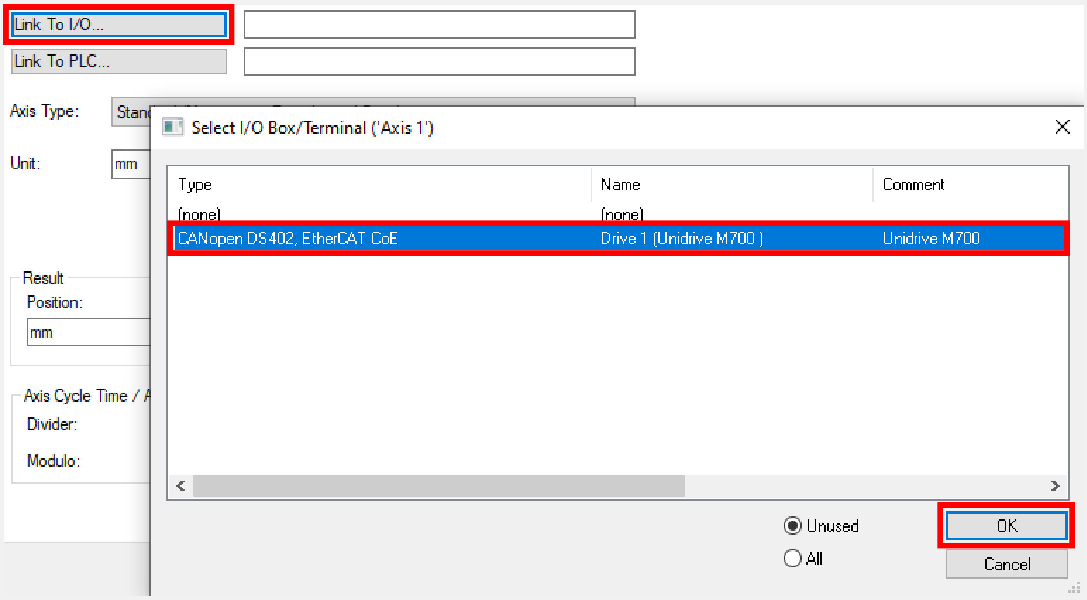
Axis Cycle Time / Access Divider

Divider: 1

Cycle Time (ms): 1.000

Modulo: 0

HINT: There should already be a link to I/O present, but if there isn’t, select the “Link To I/O…” button to link the axis to a physical drive. The “Select I/O Box/Terminal” window will appear. Select the physical drive to be linked to the axis and click “OK”.

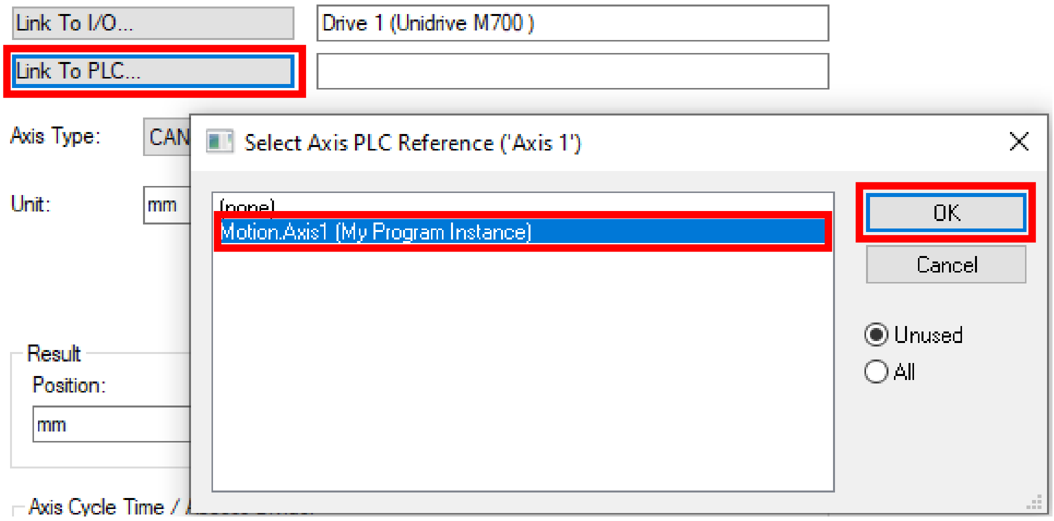


Type	Name	Comment
(none)	(none)	
CANopen DS402, EtherCAT CoE	Drive 1 (Unidrive M700)	Unidrive M700

● Unused

○ All

2. Select the “Link To PLC…” button to link the axis to an instance of AXIS_REF (declared in the PLC project). The “Select Axis PLC Reference” window will appear. Select the AXIS_REF instance to be linked to the axis and click “OK”.



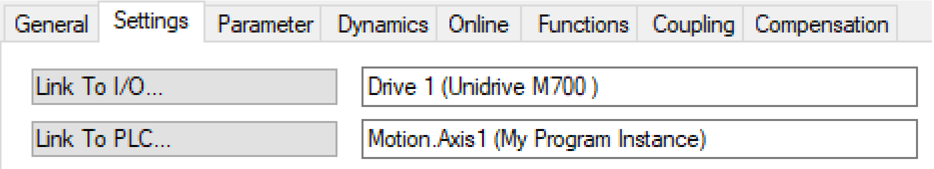
Link To I/O...: Drive 1 (Unidrive M700)

Unit: mm

Position: mm

(none)
Motion.Axis1 (My Program Instance)

3. This completes the axis linking steps. The settings tab of the axis object will now appear in the following way.



Link To I/O...: Drive 1 (Unidrive M700)

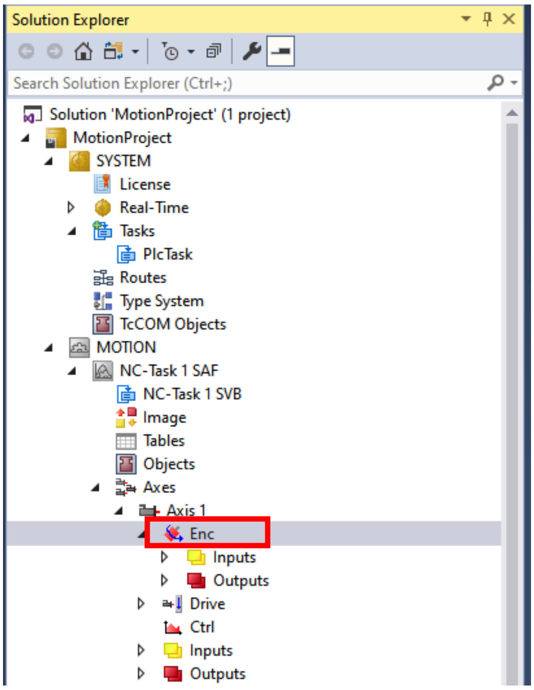
Link To PLC...: Motion.Axis1 (My Program Instance)

2.13.3 Axis configuration

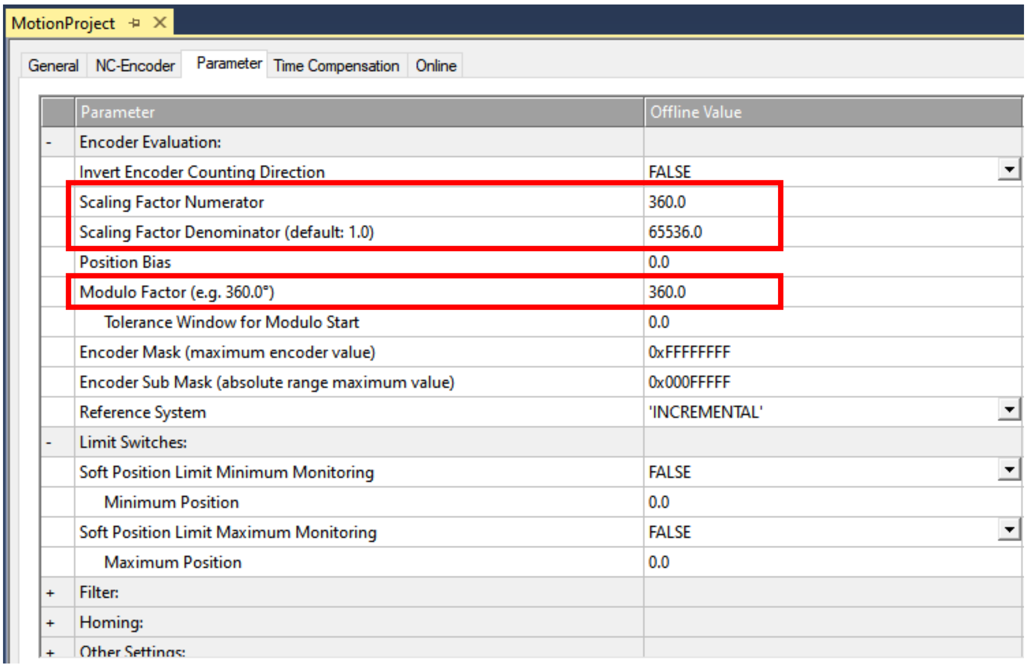
This section describes how to setup the basic properties of an axis including its unit scaling and monitoring. A detailed description of how scaling works is given in section **3.1 How to setup axis unit scaling and resolution**.

1. Put the IPC /PLC into configure mode by pressing the config mode button.

2. Set up the axis scaling. This can be found by selecting “MOTION” > “NC-Task 1 SAF” > “Axes” > “Axis 1” and then double click on “Enc”:



3. The Encoder setup appears. Select the “Parameter” tab. “Scaling Factor Numerator” defines the number of Units represented by the number of encoder counts defined by “Scaling Factor Denominator”. See section **3.1 How to setup axis unit scaling and resolution** for more information on how to scale the NC axis units.

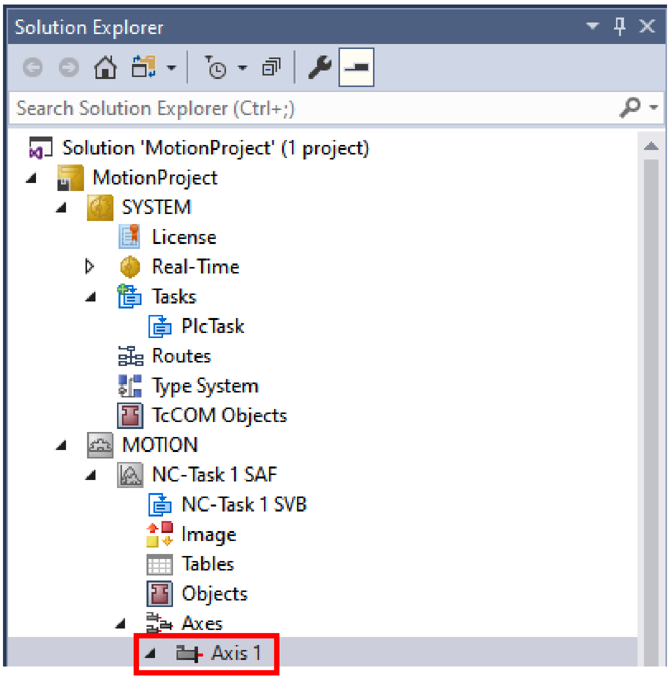


	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	360.0
	Scaling Factor Denominator (default: 1.0)	65536.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	360.0
	Tolerance Window for Modulo Start	0.0
	Encoder Mask (maximum encoder value)	0xFFFFFFFF
	Encoder Sub Mask (absolute range maximum value)	0x000FFFFF
	Reference System	'INCREMENTAL'
-	Limit Switches:	
	Soft Position Limit Minimum Monitoring	FALSE
	Minimum Position	0.0
	Soft Position Limit Maximum Monitoring	FALSE
	Maximum Position	0.0
+	Filter:	
+	Homing:	
+	Other Settings:	

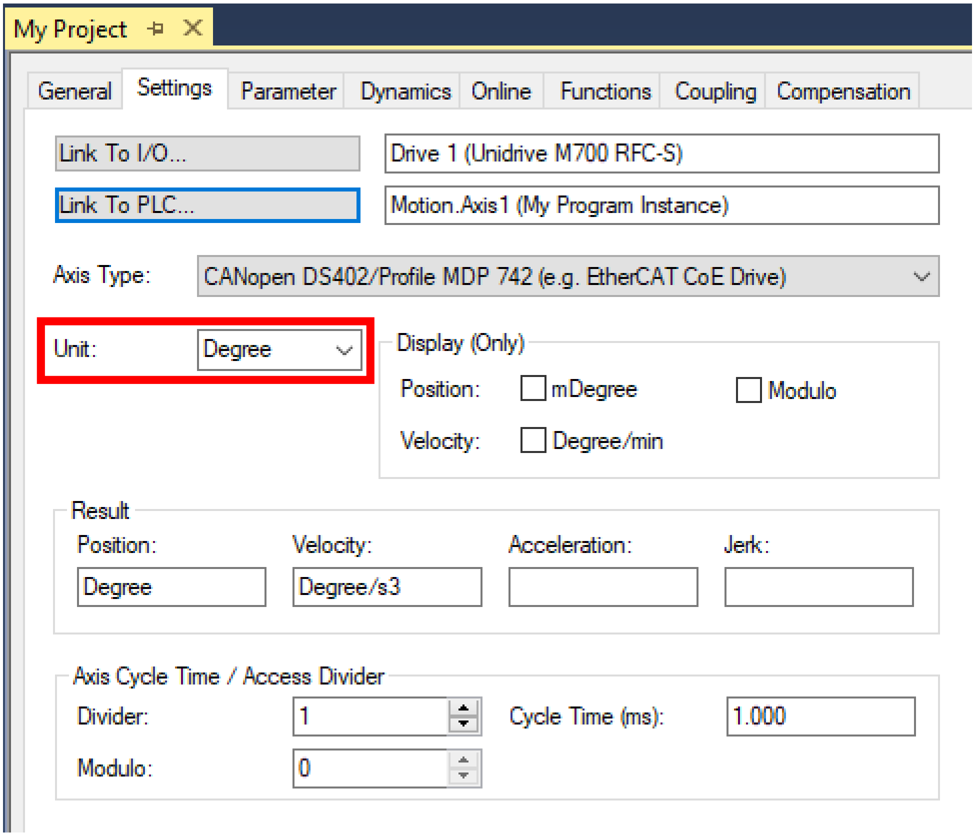
In the example above for 360° there are 65536 counts, so for 1 revolution in encoder counts we have 360°. Other configuration for things like software limits and DS402 Homing are on this tab if needed.

The modulo factor only needs to be set if the axis is going to do some modulo positioning using the postion reference where CFG_PositionMotionType = Modulo. With a 360° setting the position will change within a range of 0° to 359° as entererd in to CFG_PositionTarget when CFG_PositionMotionType = Modulo.

4. Next choose the units that the axis uses. Select “Motion” > “NC-Task 1 SAF” > “Axes” and then double click on “Axis 1”:



5. The axis setup appears. Select the “Settings” tab, and then change the unit type to something suitable e.g. “Degree”.



Link To I/O...: Drive 1 (Unidrive M700 RFC-S)

Link To PLC...: Motion.Axis1 (My Program Instance)

Axis Type: CANopen DS402/Profile MDP 742 (e.g. EtherCAT CoE Drive)

Unit: Degree

Position: ☐ mDegree ☐ Modulo

Velocity: ☐ Degree/min

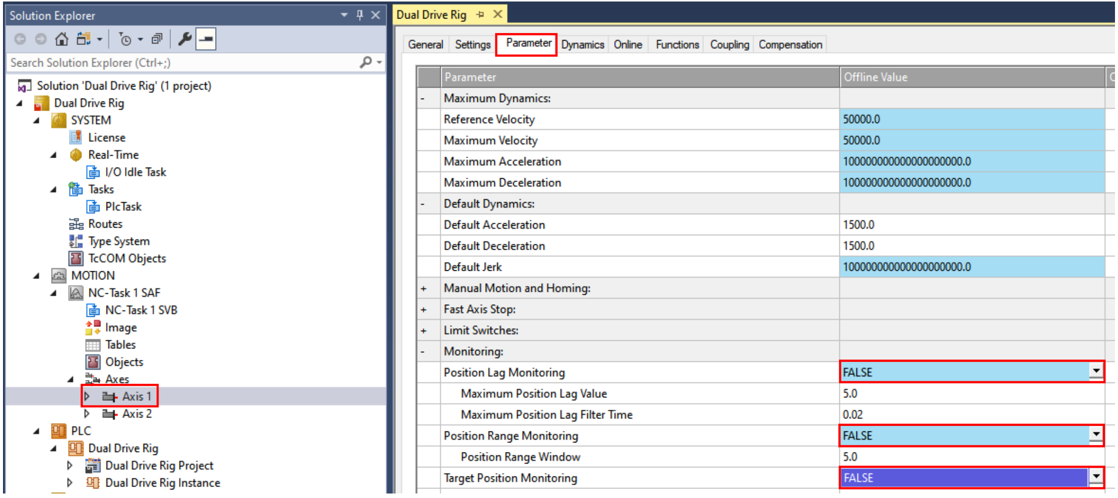
Position:	Velocity:	Acceleration:	Jerk:
Degree	Degree/s3		

Divider: 1

Cycle Time (ms): 1.000

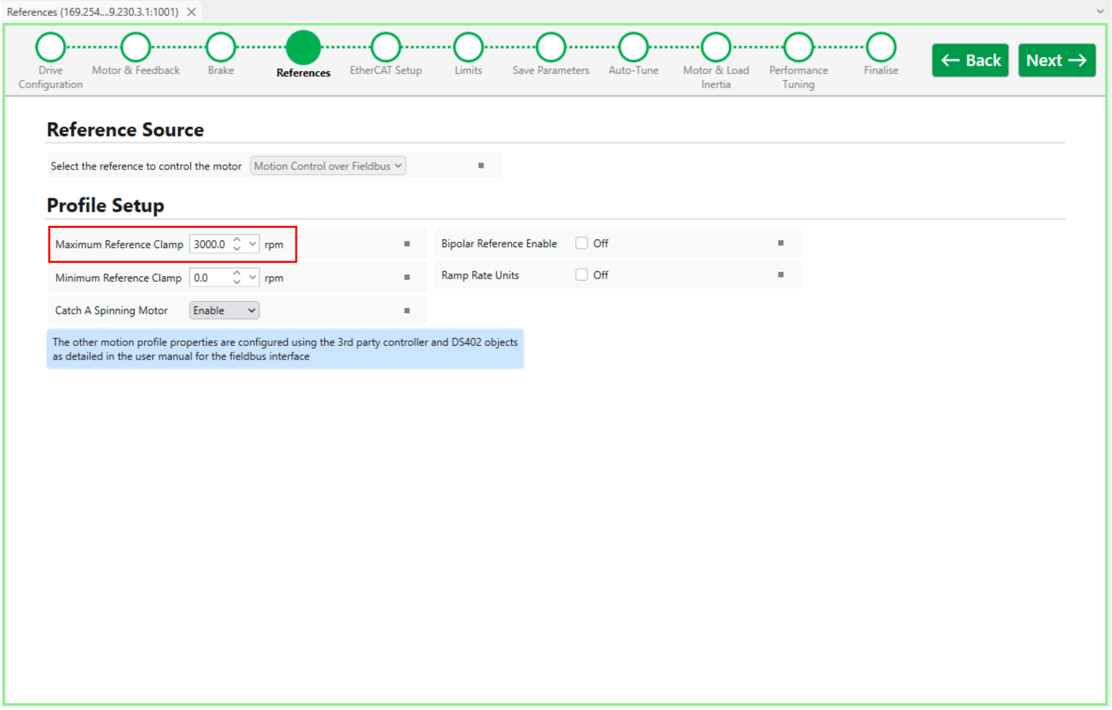
Modulo: 0

6. Turn off the encoder following error limits as these tend to cause nuisanse NC axis errors the first time the axis runs. Double click on the axis in the “Solution Explorer” tree, then select the “Parameter” tab. Scroll down to “Monitoring” and set “Position Lag Monitoring”, “Position Range Monitoring”, and “Target Position Monitoring” to FALSE. These checks can be reinstated later if required.



	Parameter	Offline Value
-	Maximum Dynamics:	
	Reference Velocity	50000.0
	Maximum Velocity	50000.0
	Maximum Acceleration	100000000000000000000.0
	Maximum Deceleration	100000000000000000000.0
-	Default Dynamics:	
	Default Acceleration	1500.0
	Default Deceleration	1500.0
	Default Jerk	100000000000000000000.0
+	Manual Motion and Homing:	
+	Fast Axis Stop:	
+	Limit Switches:	
-	Monitoring:	
	Position Lag Monitoring	FALSE
	Maximum Position Lag Value	5.0
	Maximum Position Lag Filter Time	0.02
	Position Range Monitoring	FALSE
	Position Range Window	5.0
	Target Position Monitoring	FALSE

7. Set the maximum dynamics. Typically the “Reference Velocity” and “Maximum Velocity” need to be set to the maximum reference that can be applied to the motor, as shown by Maximum Reference Clamp Pr1.006 or as configured on the “References” page in the Connect guided setup:



Reference Source

Select the reference to control the motor: Motion Control over Fieldbus

Profile Setup

Maximum Reference Clamp: 3000.0 rpm

Minimum Reference Clamp: 0.0 rpm

Catch A Spinning Motor: Enable

Bipolar Reference Enable: ☐ Off

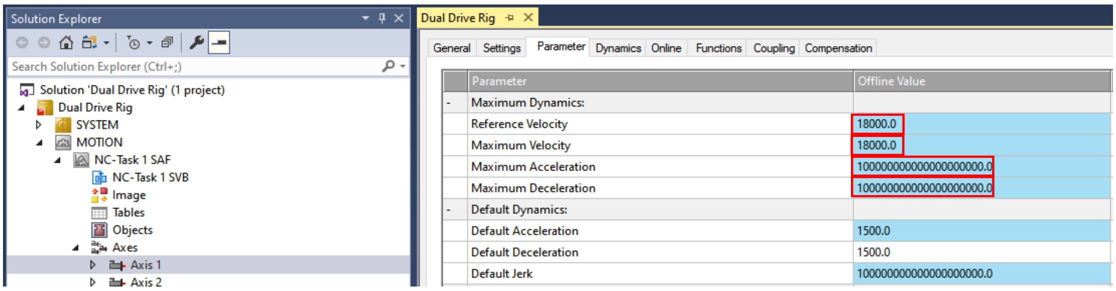
Ramp Rate Units: ☐ Off

The other motion profile properties are configured using the 3rd party controller and DS402 objects as detailed in the user manual for the fieldbus interface

Use the formula provided in section **3.1.2 How to scale between NC axis speed and rpm** to convert the maximum reference speed in rpm into units/s. An example is given below:

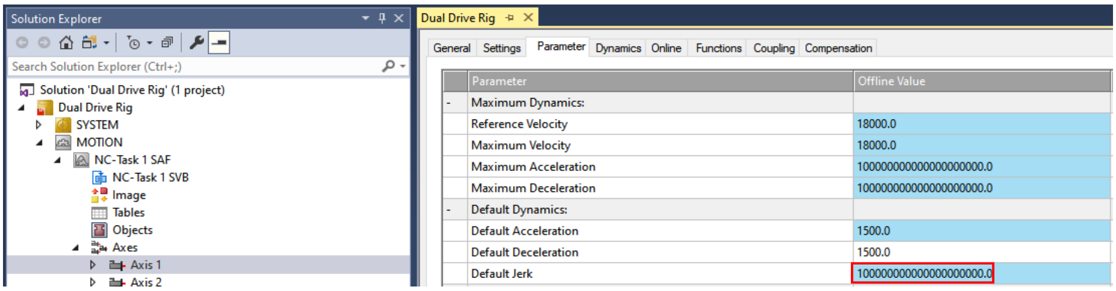
> An axis has a maximum speed of 3000rpm, and a scaling numerator of 360 units per revolution, the maximum speed in units per second is 3000 * 360 / 60 = 18000.

Unless the application has a specific reason to limit the acceleration and deceleration maximum, the best option is to set the acceleration and deceleration to a high value such as 1E+20, (100000000000000000000), so they don’t cause nuisance NC axis errors when the axis is run for the first time.



	Parameter	Offline Value
-	Maximum Dynamics:	
	Reference Velocity	18000.0
	Maximum Velocity	18000.0
	Maximum Acceleration	100000000000000000000.0
	Maximum Deceleration	100000000000000000000.0
-	Default Dynamics:	
	Default Acceleration	1500.0
	Default Deceleration	1500.0
	Default Jerk	100000000000000000000.0

8. Configure the default dynamics for the axis. These are entirely dependent on the specific application. It can be helpful to set the default Jerk to 1E+20, (100000000000000000000), which will invoke a linear acceleration profile if the jerk is left at 0 and will prevent nuisance NC axis errors.

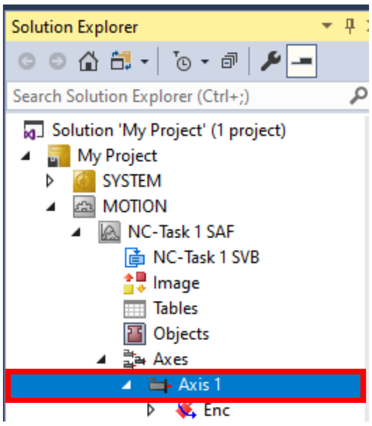


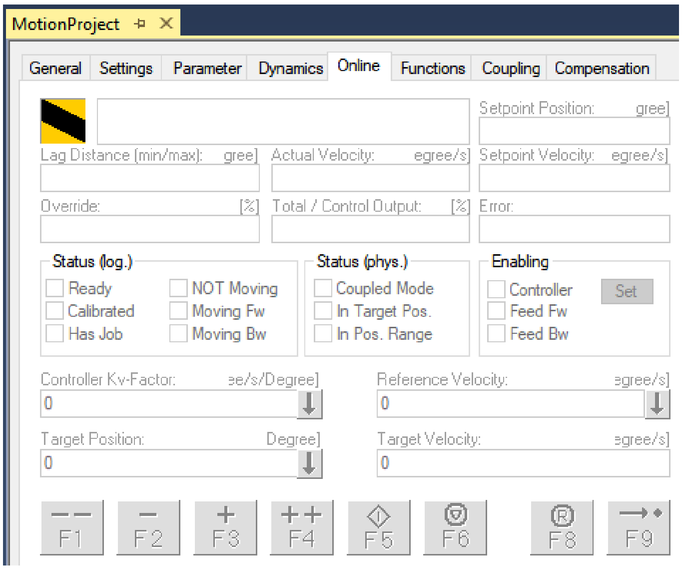
	Parameter	Offline Value
-	Maximum Dynamics:	
	Reference Velocity	18000.0
	Maximum Velocity	18000.0
	Maximum Acceleration	100000000000000000000.0
	Maximum Deceleration	100000000000000000000.0
-	Default Dynamics:	
	Default Acceleration	1500.0
	Default Deceleration	1500.0
	Default Jerk	100000000000000000000.0

2.13.4 Moving the axis using TwinCAT test moves

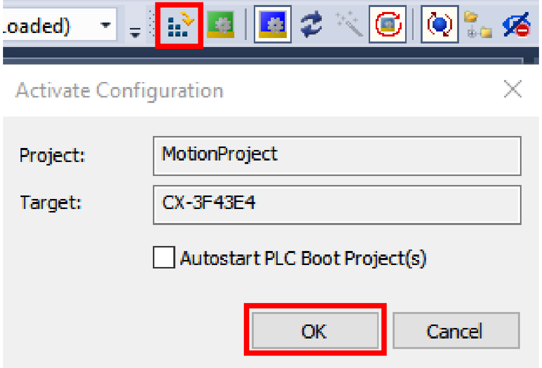
The TwinCAT test moves may be used to check that the axis is capable of running. This is not an essential step, but could prove useful if the axis doesn't run as a means to diagnose an NC Axis.

1. Double click on the axis in the "Solution Explorer" tree, then select the "Online" tab to reveal the test controls. At this point the system may be offline in "Config" mode where the controls will be greyed out.





2. Click the "Activate Configuration" button and then click "OK" if the "Activate Configuration" dialog appears:

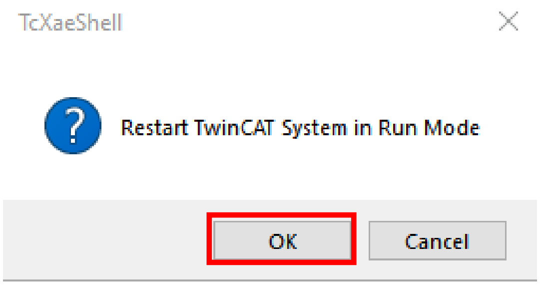


Project: MotionProject

Target: CX-3F43E4

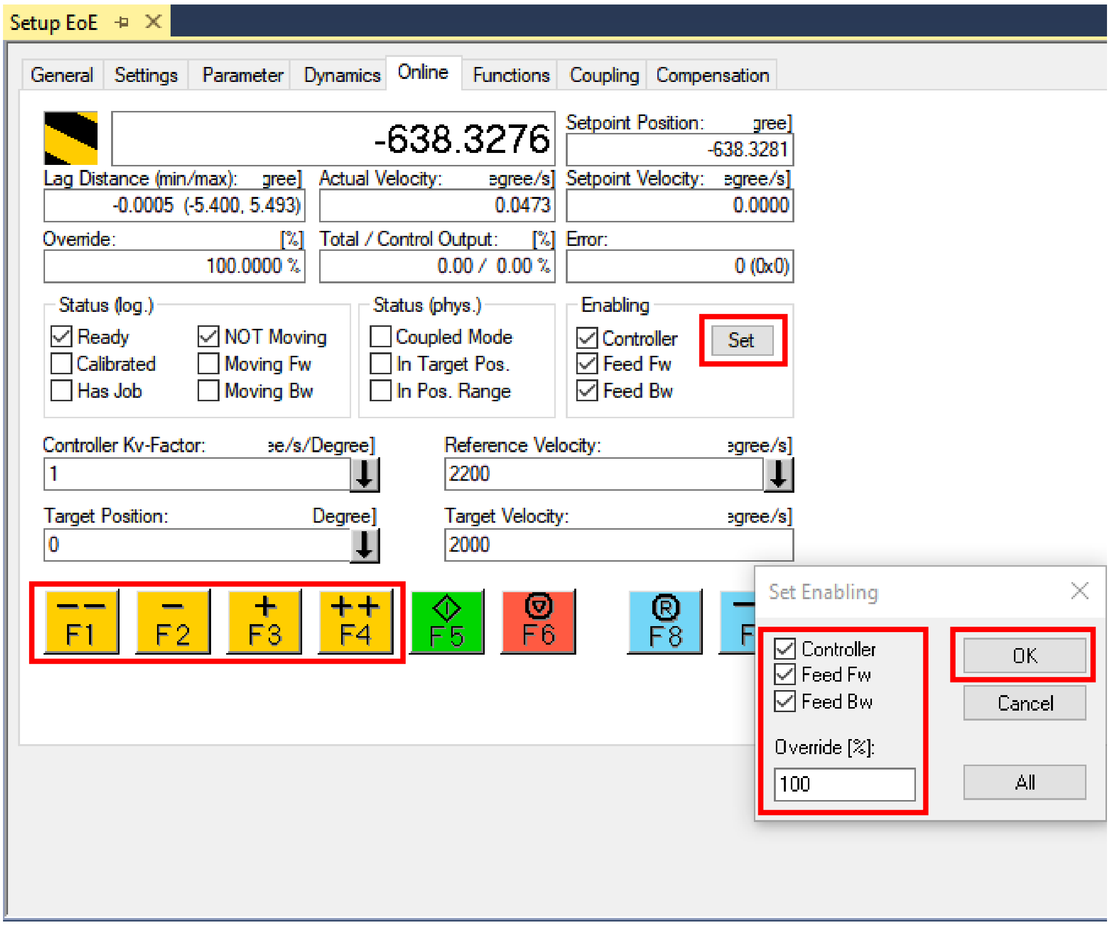
☐ Autostart PLC Boot Project(s)

3. Click "OK" on the "Restart TwinCAT in Run Mode" dialog:



Restart TwinCAT System in Run Mode

4. The “Online” tab test controls are now available. To run the drive the “Enabling” control must be configured. Press the “Set” button and when the “Set Enabling” dialog appears press the “All” button and set the override percentage to 100%. The axis will now move if for example F2 or F3 are pressed to jog the axis.



-638.3276

Setpoint Position: gree] -638.3281

Lag Distance (min/max): gree] -0.0005 (-5.400, 5.493)

Actual Velocity: egree/s] 0.0473

Setpoint Velocity: egree/s] 0.0000

Override: [%] 100.0000 %

Total / Control Output: [%] 0.00 / 0.00 %

Error: 0 (0x0)

Status (log.)
- ☑ Ready
- ☐ Calibrated
- ☐ Has Job
- ☑ NOT Moving
- ☐ Moving Fw
- ☐ Moving Bw

Status (phys.)
- ☐ Coupled Mode
- ☐ In Target Pos.
- ☐ In Pos. Range

Enabling
- ☑ Controller
- ☑ Feed Fw
- ☑ Feed Bw

Controller Kv-Factor: e/s/Degree] 1

Reference Velocity: egree/s] 2200

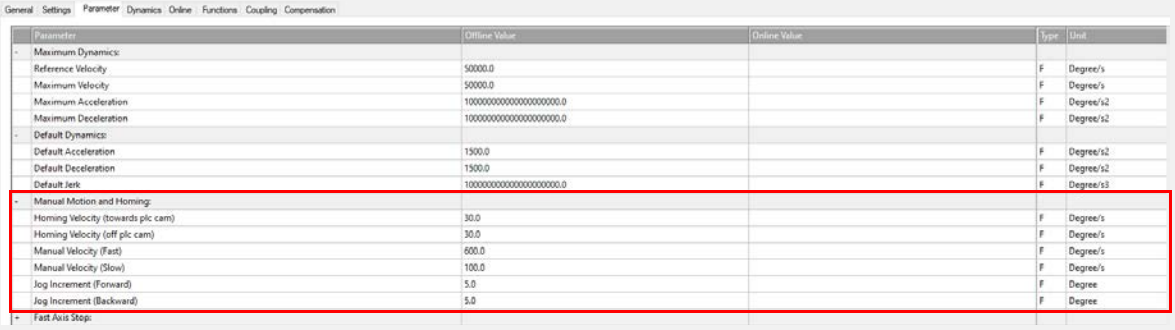
Target Position: Degree] 0

Target Velocity: egree/s] 2000

Set Enabling
- ☑ Controller
- ☑ Feed Fw
- ☑ Feed Bw

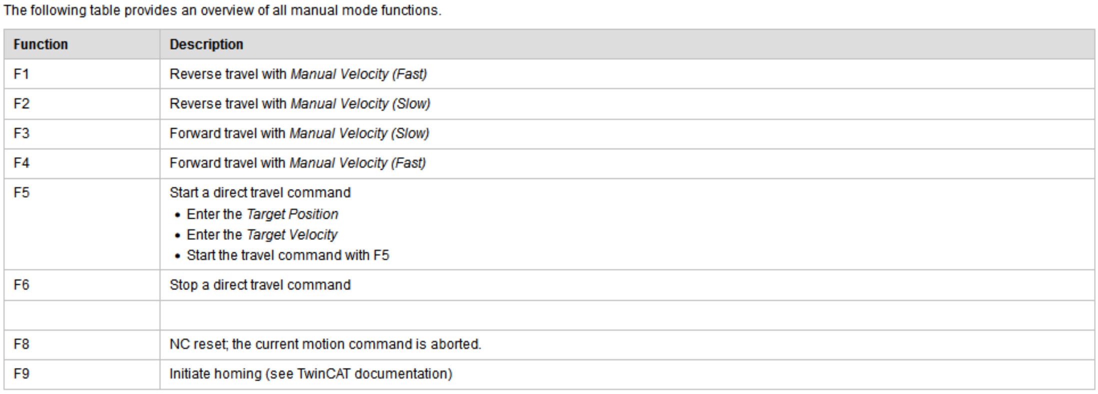
Override [%]: 100

HINT: The motion profile setup is configured on the “Parameter” tab under “Manual Motion and Homing:



Parameter	Offline Value	Online Value	Type	Unit
- Maximum Dynamics:				
Reference Velocity	50000.0		F	Degree/s
Maximum Velocity	50000.0		F	Degree/s
Maximum Acceleration	10000000000000000000000.0		F	Degree/s2
Maximum Deceleration	10000000000000000000000.0		F	Degree/s2
- Default Dynamics:				
Default Acceleration	1500.0		F	Degree/s2
Default Deceleration	1500.0		F	Degree/s2
Default Jerk	10000000000000000000000.0		F	Degree/s3
- Manual Motion and Homing:				
Homing Velocity (towards plc cam)	30.0		F	Degree/s
Homing Velocity (off plc cam)	30.0		F	Degree/s
Manual Velocity (Fast)	600.0		F	Degree/s
Manual Velocity (Slow)	100.0		F	Degree/s
Jog Increment (Forward)	5.0		F	Degree
Jog Increment (Backward)	5.0		F	Degree
+ Fast Axis Stop:				

5. The on screen buttons have the following functionality:



The following table provides an overview of all manual mode functions.

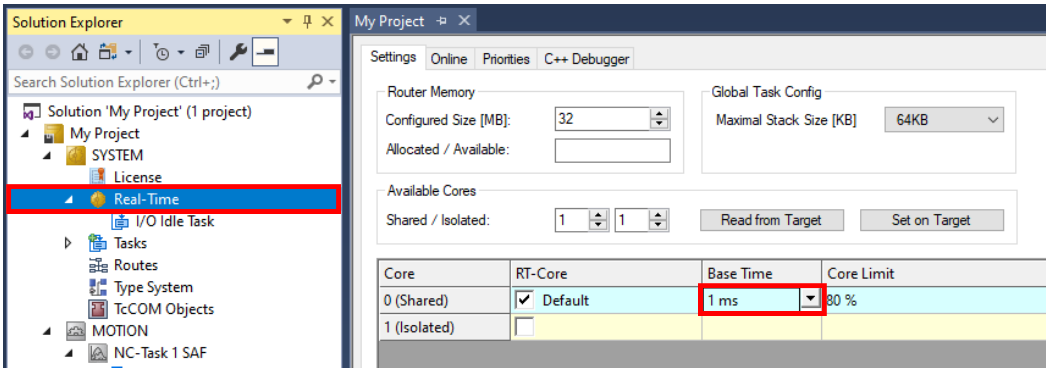
Function	Description
F1	Reverse travel with *Manual Velocity (Fast)*
F2	Reverse travel with *Manual Velocity (Slow)*
F3	Forward travel with *Manual Velocity (Slow)*
F4	Forward travel with *Manual Velocity (Fast)*
F5	Start a direct travel command • Enter the *Target Position* • Enter the *Target Velocity* • Start the travel command with F5
F6	Stop a direct travel command
F8	NC reset; the current motion command is aborted.
F9	Initiate homing (see TwinCAT documentation)

2.14 Setting the project task rates

For optimal operation of the Control Techniques NC Axis Interface Library, it is strongly recommended to configure all task rates to 1ms. This ensures that functionality such as electronic gearbox, filtering and any motion in general operates smoothly with the least amount of position granularity.

The following steps describe how to set all task rates to 1ms.

1. From the project tree, navigate to "Project > SYSTEM", double click on "Real-Time" and ensure that "Base Time" for the real time core is set to "1ms".



Configured Size [MB]: 32

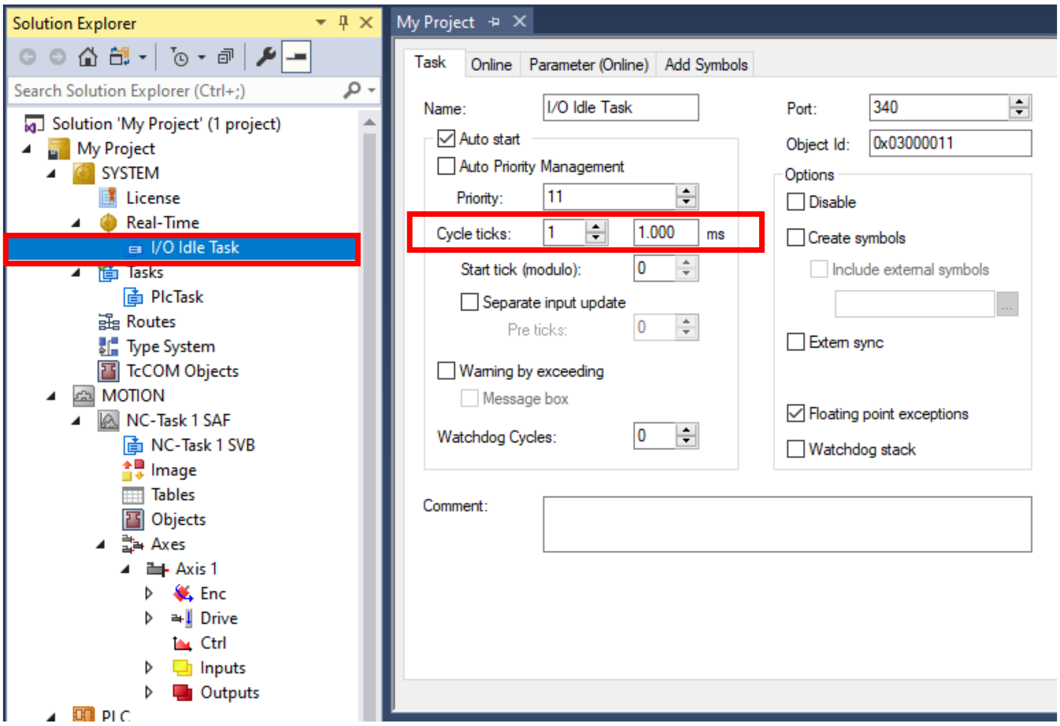
Allocated / Available:

Maximal Stack Size [KB]: 64KB

Shared / Isolated: 1 1

Core	RT-Core	Base Time	Core Limit
0 (Shared)	☑ Default	1 ms	80 %
1 (Isolated)	☐		

2. From the project tree, navigate to "Project > SYSTEM > Real-Time", double click on "I/O Idle Task" and ensure that "Cycle ticks" is set to "1" (resulting in a 1.000 ms cycle time).



Name: I/O Idle Task

Port: 340

Object Id: 0x03000011

☑ Auto start

☐ Auto Priority Management

Priority: 11

Cycle ticks: 1 1.000 ms

Start tick (modulo): 0

☐ Separate input update

Pre ticks: 0

☐ Warning by exceeding

☐ Message box

Watchdog Cycles: 0

☐ Disable

☐ Create symbols

☐ Include external symbols

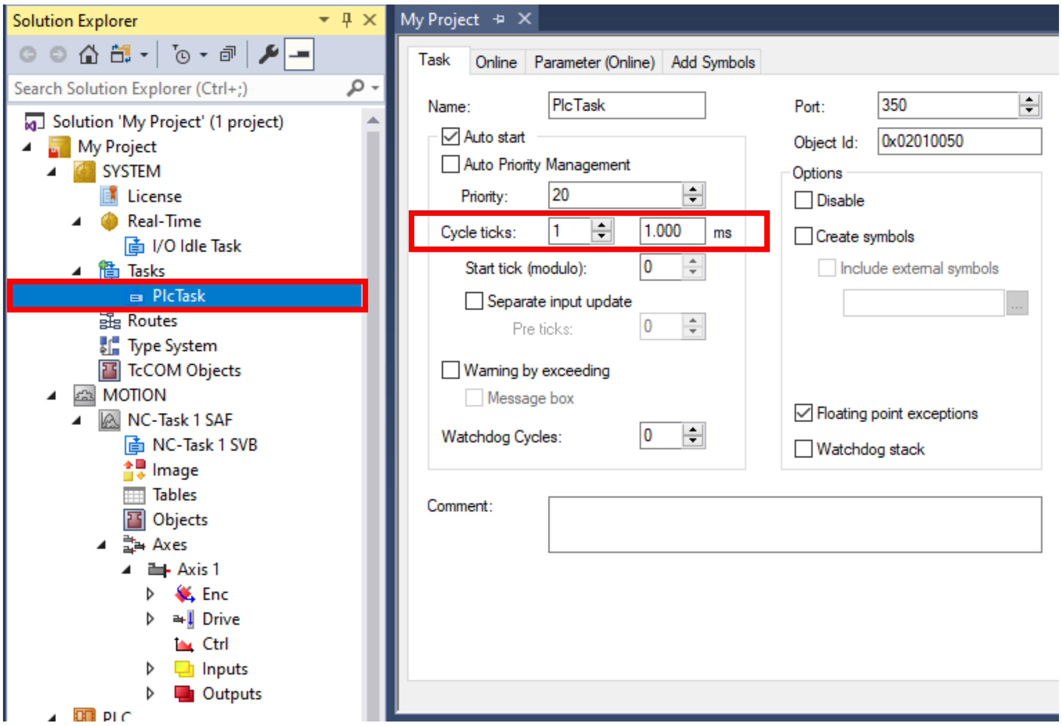
☐ Extern sync

☑ Floating point exceptions

☐ Watchdog stack

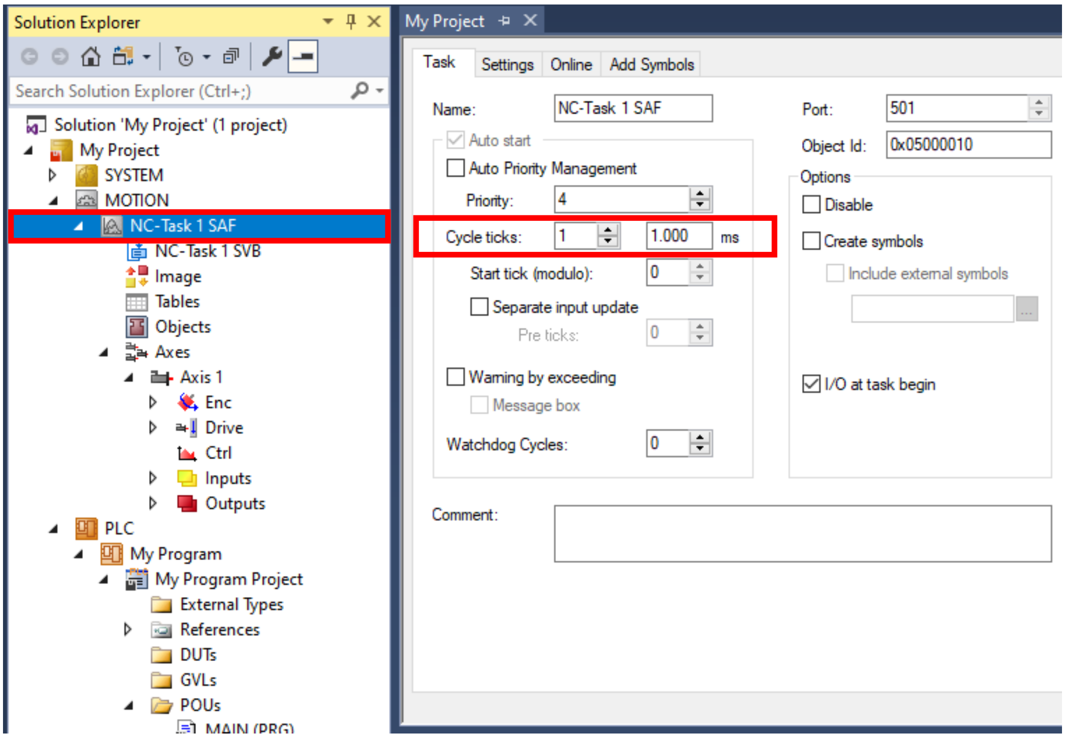
Comment:

3. From the project tree, navigate to “Project > SYSTEM > Tasks”, double click on “Plc Task” and ensure that “Cycle ticks” is set to “1” (resulting in a 1.000 ms cycle time).



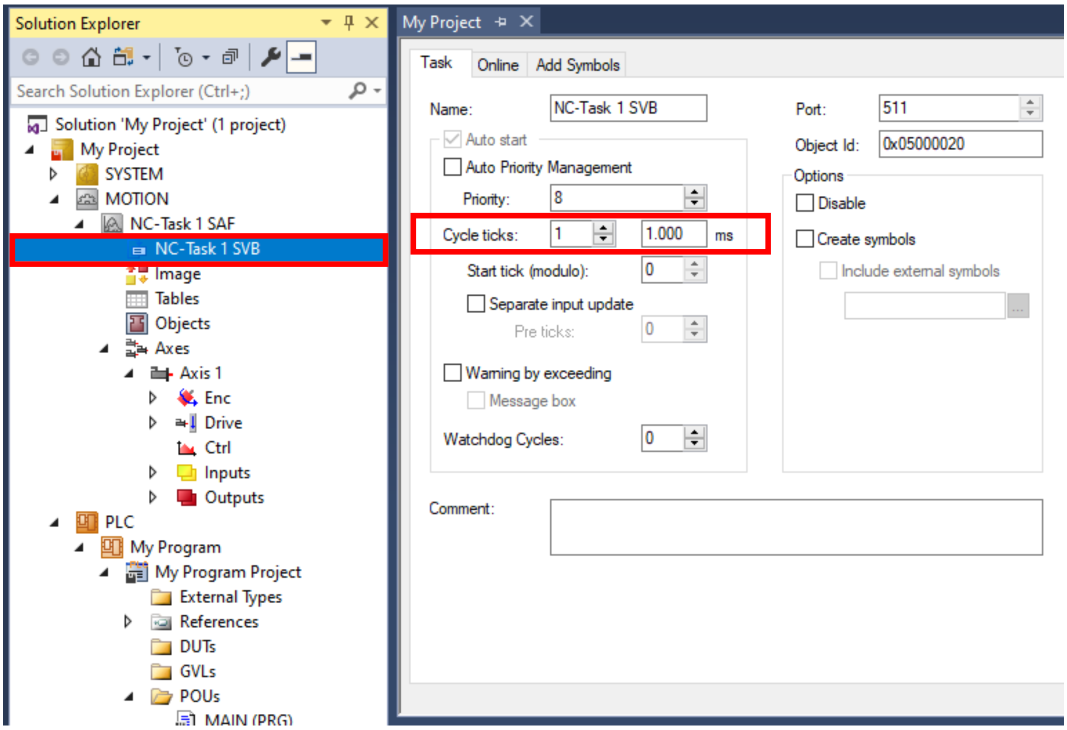
Name: PlcTask
Port: 350
Object Id: 0x02010050
☑ Auto start
☐ Auto Priority Management
Priority: 20
Cycle ticks: 1 1.000 ms
Start tick (modulo): 0
☐ Separate input update
Pre ticks: 0
☐ Warning by exceeding
☐ Message box
Watchdog Cycles: 0
Options
☐ Disable
☐ Create symbols
☐ Include external symbols
☑ Floating point exceptions
☐ Watchdog stack
Comment:

4. From the project tree, navigate to “Project > MOTION”, double click on “ NC-Task 1 SAF” and ensure that “Cycle ticks” is set to “1” (resulting in a 1.000 ms cycle time).



Name: NC-Task 1 SAF
Port: 501
Object Id: 0x05000010
☑ Auto start
☐ Auto Priority Management
Priority: 4
Cycle ticks: 1 1.000 ms
Start tick (modulo): 0
☐ Separate input update
Pre ticks: 0
☐ Warning by exceeding
☐ Message box
Watchdog Cycles: 0
Options
☐ Disable
☐ Create symbols
☐ Include external symbols
☑ I/O at task begin
Comment:

5. From the project tree, navigate to “Project > MOTION > NC-Task 1 SAF”, double click on “NC-Task 1 SVB” and ensure that “Cycle ticks” is set to “1” (resulting in a 1.000 ms cycle time).



Name: NC-Task 1 SVB

☑ Auto start

☐ Auto Priority Management

Priority: 8

Cycle ticks: 1 1.000 ms

Start tick (modulo): 0

☐ Separate input update

Pre ticks: 0

☐ Warning by exceeding

☐ Message box

Watchdog Cycles: 0

Port: 511

Object Id: 0x05000020

Options

☐ Disable

☐ Create symbols

☐ Include external symbols

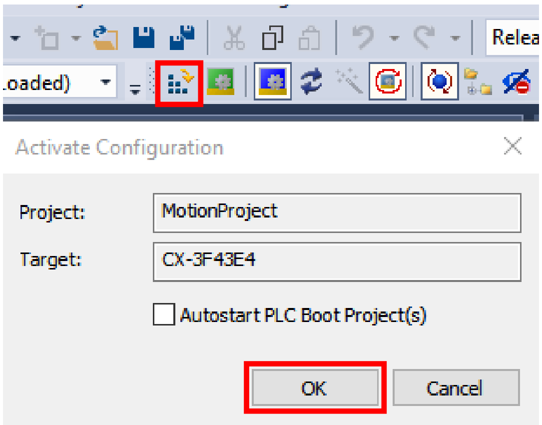
Comment:

6. This completes the task rate setup.

2.15 Verify the axis

The following steps should be followed to verify the operation of the axis:

1. Ensure that the project has been activated by clicking the “Activate Configuration” button and selecting “OK” if the “Activate Configuration” dialog appears:



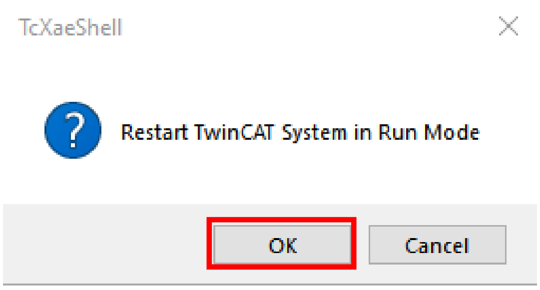
Activate Configuration

Project: MotionProject

Target: CX-3F43E4

☐ Autostart PLC Boot Project(s)

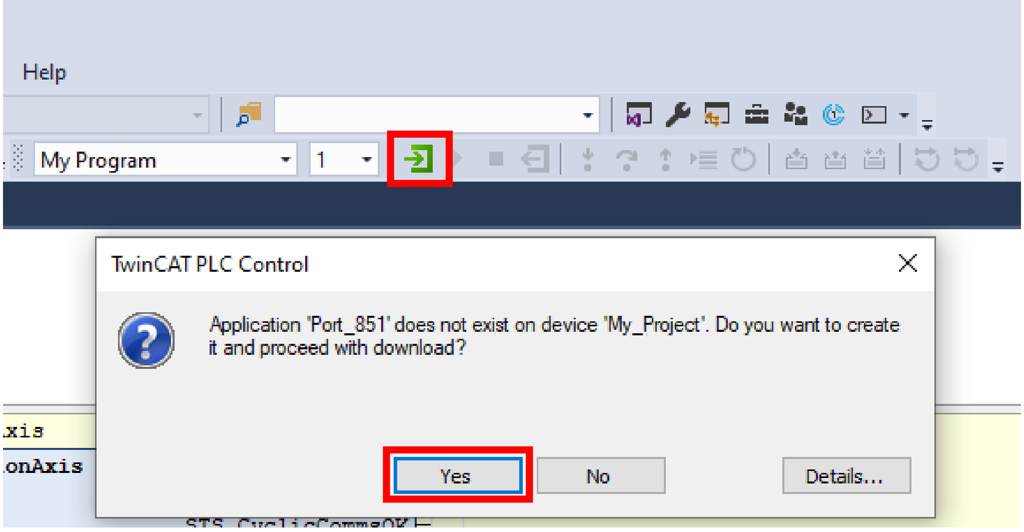
2. Click “OK” on the subsequent “Restart TwinCAT in Run Mode” dialog:



TcXaeShell

Restart TwinCAT System in Run Mode

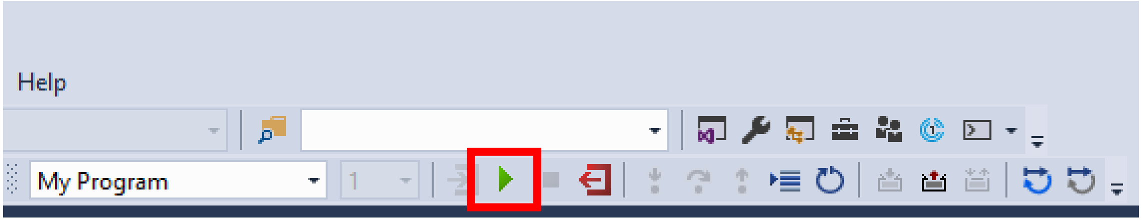
3. Click the “Login” button to download the PLC code to the controller. If the “TwinCAT PLC Control” dialog appears, click “Yes” to create the application on the controller.



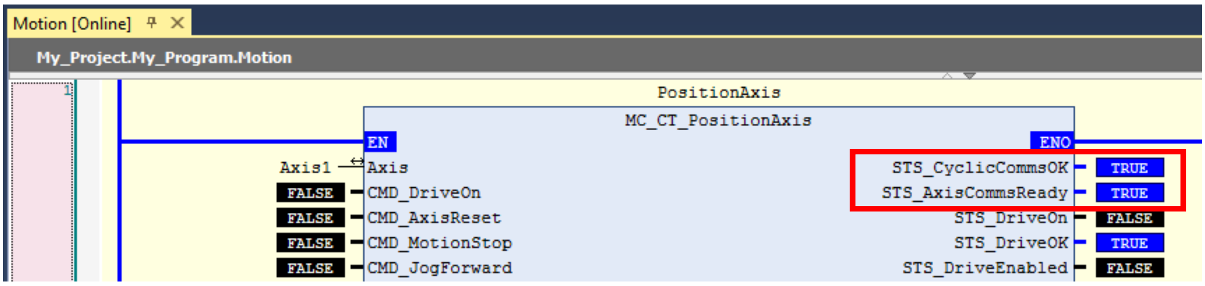
TwinCAT PLC Control

Application 'Port_851' does not exist on device 'My_Project'. Do you want to create it and proceed with download?

4. Click the “Start” button to begin running the PLC program.

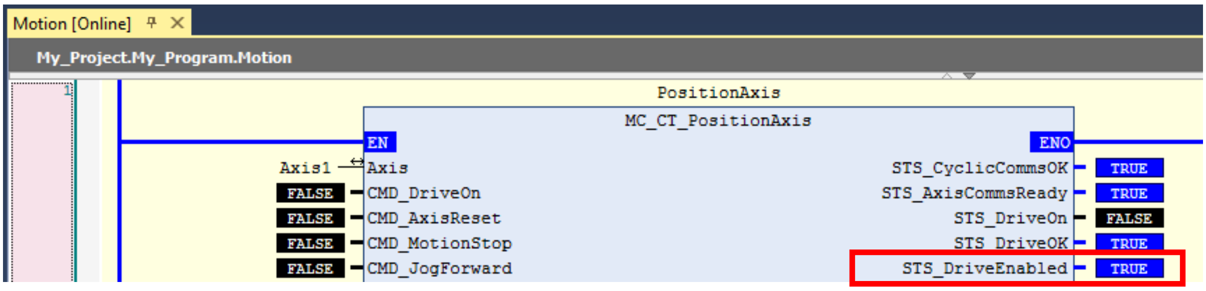


5. From the project tree navigate to “{Project Name} > PLC > {Program Name} > POUs” and double click on the program name where MC_CT_PositionAxis is being called. When the communications path is operating correctly STS_CyclicCommsOK and STS_AxisCommsReady are set to TRUE (blue).



Axis1 — Axis
FALSE — CMD_DriveOn
FALSE — CMD_AxisReset
FALSE — CMD_MotionStop
FALSE — CMD_JogForward
STS_CyclicCommsOK — TRUE
STS_AxisCommsReady — TRUE
STS_DriveOn — FALSE
STS_DriveOK — TRUE
STS_DriveEnabled — FALSE

6. Check that changes to the drives enable input can be seen at the function block output. Toggle the 24V signal applied to the STO input or for a dual STO input I/O module toggle the 24V signal applied to both STO1 and STO2.
The STS_DriveEnabled output are set to TRUE (blue) when the drive is enabled, or FALSE (black) when the drive is disabled.

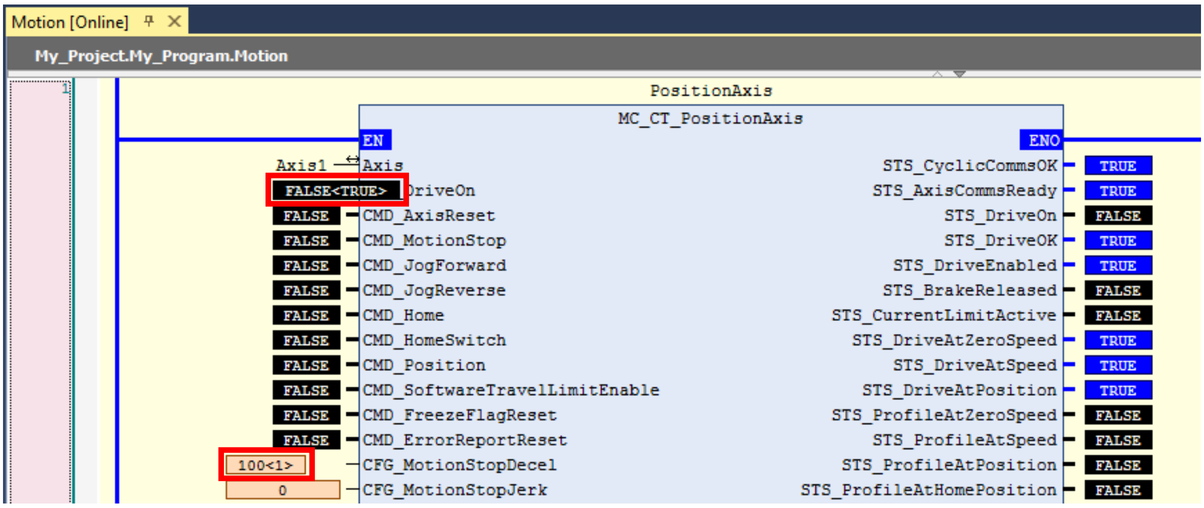


Axis1 — Axis
FALSE — CMD_DriveOn
FALSE — CMD_AxisReset
FALSE — CMD_MotionStop
FALSE — CMD_JogForward
STS_CyclicCommsOK — TRUE
STS_AxisCommsReady — TRUE
STS_DriveOn — FALSE
STS_DriveOK — TRUE
STS_DriveEnabled — TRUE

2.16 Moving the axis with the CT NC Axis Interface Library

This section gives a simple guide to moving the axis using the MC_CT_PositionAxis interface from the CT NC Axis Interface Library.

1. 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.

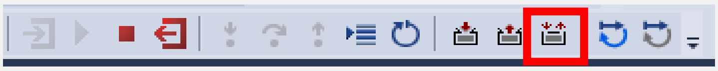


PositionAxis

MC_CT_PositionAxis

	EN	ENO	
Axis1	Axis	STS_CyclicCommsOK	TRUE
FALSE<TRUE>	DriveOn	STS_AxisCommsReady	TRUE
FALSE	CMD_AxisReset	STS_DriveOn	FALSE
FALSE	CMD_MotionStop	STS_DriveOK	TRUE
FALSE	CMD_JogForward	STS_DriveEnabled	TRUE
FALSE	CMD_JogReverse	STS_BrakeReleased	FALSE
FALSE	CMD_Home	STS_CurrentLimitActive	FALSE
FALSE	CMD_HomeSwitch	STS_DriveAtZeroSpeed	TRUE
FALSE	CMD_Position	STS_DriveAtSpeed	TRUE
FALSE	CMD_SoftwareTravelLimitEnable	STS_DriveAtPosition	TRUE
FALSE	CMD_FreezeFlagReset	STS_ProfileAtZeroSpeed	FALSE
FALSE	CMD_ErrorReportReset	STS_ProfileAtSpeed	FALSE
100<1>	CFG_MotionStopDecel	STS_ProfileAtPosition	FALSE
0	CFG_MotionStopJerk	STS_ProfileAtHomePosition	FALSE

HINT: Values in angle braces e.g. **“FALSE<TRUE>”** and **“100<1>”** in the above example, show that these values haven’t been applied yet.
These values can be applied by pressing Ctrl+F7. Alternatively, the “Write Values” button can be used, found on the main toolbar:



2. At this stage, **provided it is safe to do so**, jogging motion may be tested on the axis. To run the axis Set CMD_DriveOn to TRUE and apply 24V to the enable / STO input(s).
   The axis motor should remain stationary.
3. Set the jogging speed to nominal low value e.g. 10% of the motor rated speed in Application units in CFG_JogSpeed.
4. To run the jog reference set CMD_JogForward or CMD_JogReverse to TRUE, whichever is appropriate for the application’s mechanical arrangement. The axis should now begin to move.
5. If activating CMD_JogForward results in reversed physical motion, change the “Invert Motor Polarity” value. This can be found by navigating the following area via the project tree; “MOTION > NC-Task 1 SAF > Axes > Axis 1 > Drive” followed by the “Parameter” tab where the “Invert Motor Polarity” value will be displayed.
6. Please consult section **4.1 Motion Interface** for further information on the motion interfaces available.
7. Note that once the axis is under control of the NC Axis Interface Library, further tuning of the position loop and the use of inertia compensation may be required for the best performance.

   For more information see section **3.10 How and when to tune the Position Loop** and section **3.11 How and when to use inertia compensation**.

3 How to guides

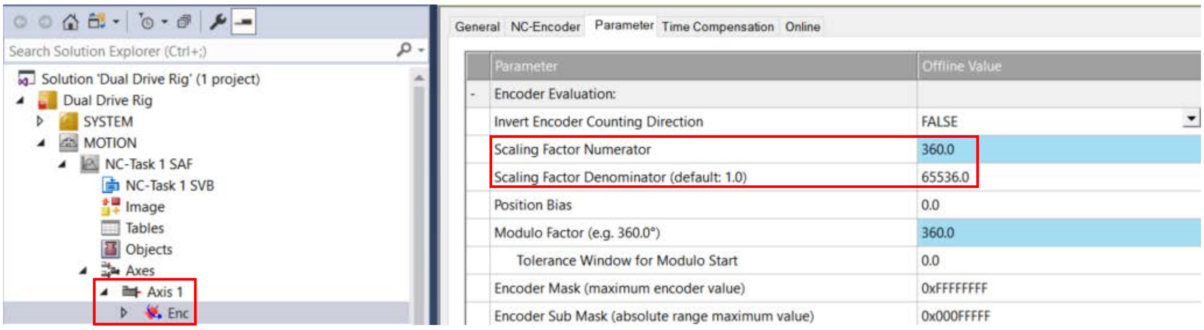
3.1 How to setup axis unit scaling and resolution

Unit scaling in most motion control systems, including a TwinCAT NC 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.
- The denominator is the number of position feedback counts that equals the numerator value.

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 TwinCAT it looks like this:



Search Solution Explorer (Ctrl+;)

Solution 'Dual Drive Rig' (1 project)
- Dual Drive Rig
  - SYSTEM
  - MOTION
    - NC-Task 1 SAF
      - NC-Task 1 SVB
      - Image
      - Tables
      - Objects
      - Axes
        - Axis 1
          - Enc

Parameter	Offline Value
Encoder Evaluation:	
Invert Encoder Counting Direction	FALSE
Scaling Factor Numerator	360.0
Scaling Factor Denominator (default: 1.0)	65536.0
Position Bias	0.0
Modulo Factor (e.g. 360.0°)	360.0
Tolerance Window for Modulo Start	0.0
Encoder Mask (maximum encoder value)	0xFFFFFFFF
Encoder Sub Mask (absolute range maximum value)	0x000FFFFF

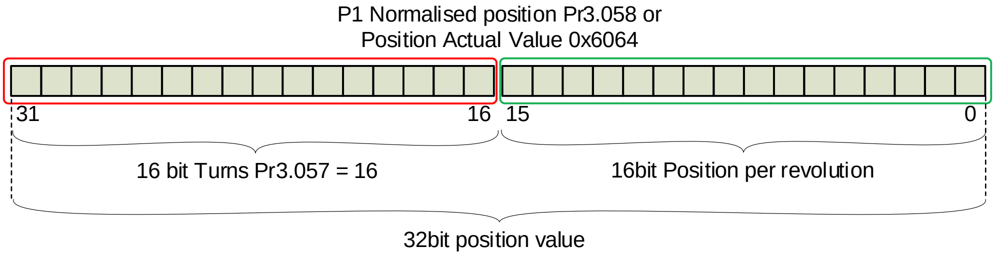
If we add a 4:1 reduction gearbox, the number of rotations at the output of the gearbox is reduced by the gearbox ratio so in this example the scaling numerator is 360 * 1 / 4 = 90.

The encoder resolution in feedback counts can be calculated by using the following formula:

Encoder Counts Per Revolution = 2^(32-**NormalisationTurns**)

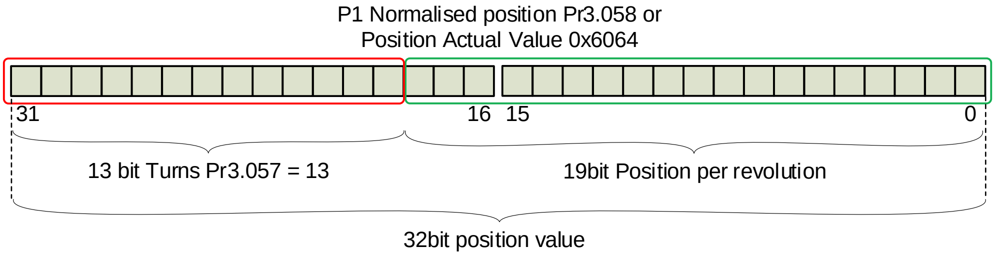
In the drive, the resolution in feedback counts is given by $2^{(32\text{-Pr3.057})}$.

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

See the next section **3.1.1 How to set the scaling using the startup list in TwinCAT** for more information on creating a start up list entry for Pr3.057 in TwinCAT.

See step 13 in section **2.7 Commission the drive and motor using Connect** for more information on exporting a startup list from Connect.

See section **3.2.1 How to import a startup list .xml file from Connect into TwinCAT** for more information on importing a startup list into TwinCAT.

3.1.1 How to set the scaling using the startup list in TwinCAT

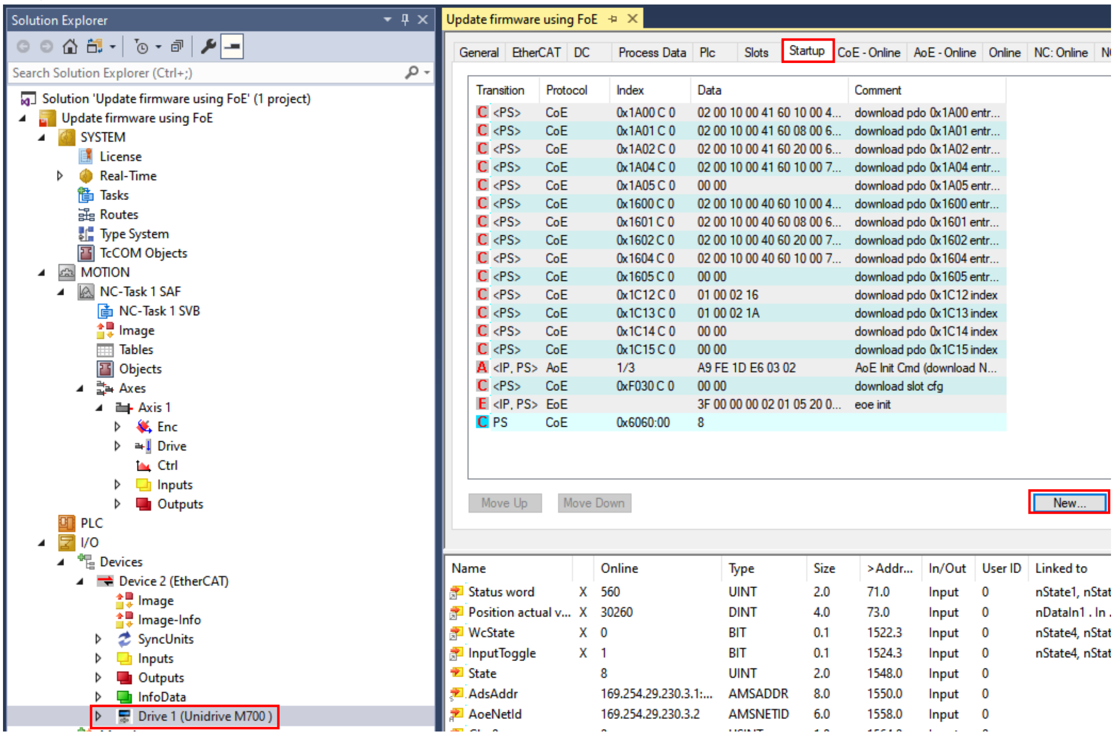
The resolution is configured by default in Pr3.057. See the previous table for the number that the turns bits should be set, (Hex value), to for a particular number of counts per revolution.

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:

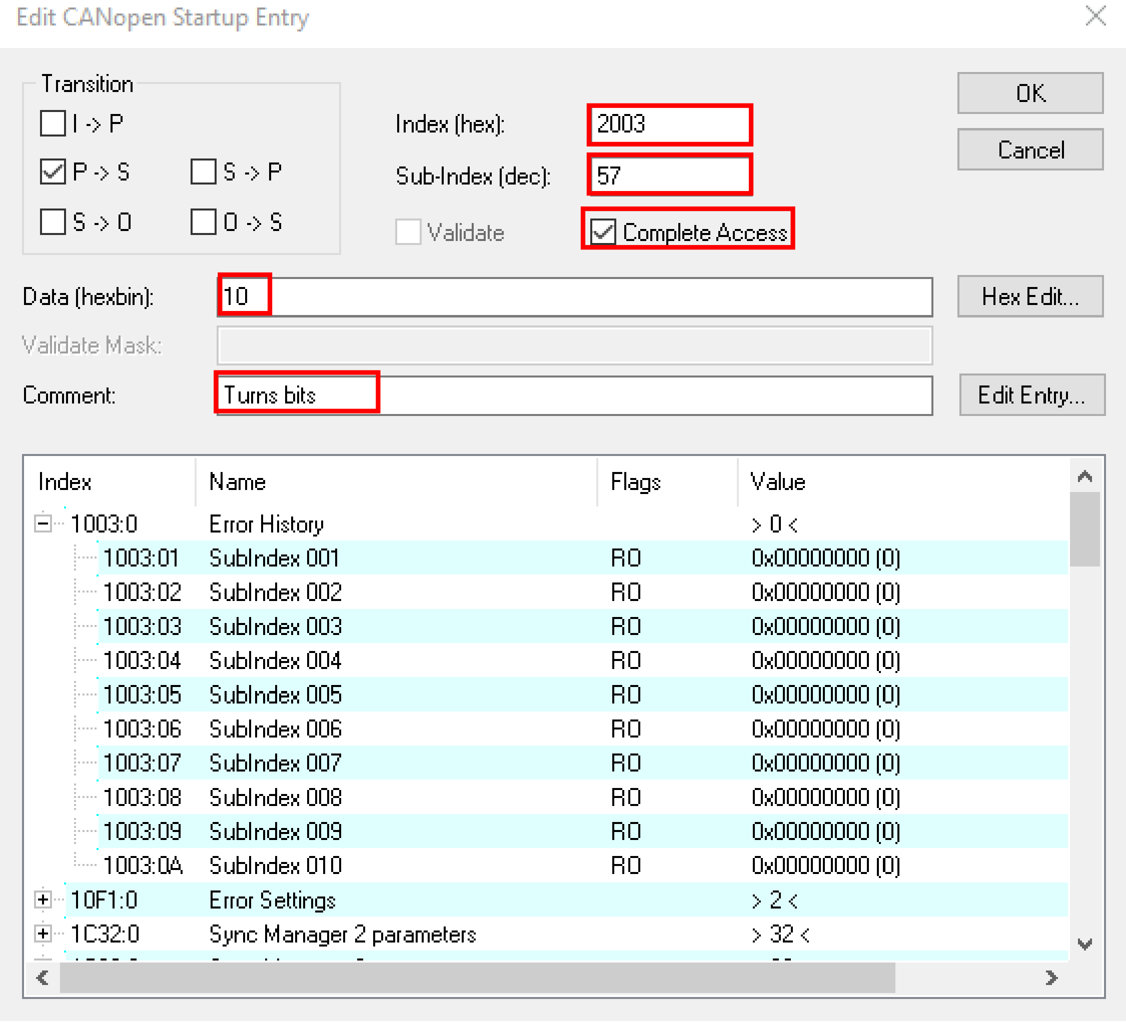
1. Double click on the drive to modify the startup list for, from the properties select the “Startup” tab, and then click “new”.



Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1A00 C 0	02 00 10 00 41 60 10 00 4...	download pdo 0x1A00 entr...
C <PS>	CoE	0x1A01 C 0	02 00 10 00 41 60 08 00 6...	download pdo 0x1A01 entr...
C <PS>	CoE	0x1A02 C 0	02 00 10 00 41 60 20 00 6...	download pdo 0x1A02 entr...
C <PS>	CoE	0x1A04 C 0	02 00 10 00 41 60 10 00 7...	download pdo 0x1A04 entr...
C <PS>	CoE	0x1A05 C 0	00 00	download pdo 0x1A05 entr...
C <PS>	CoE	0x1600 C 0	02 00 10 00 40 60 10 00 4...	download pdo 0x1600 entr...
C <PS>	CoE	0x1601 C 0	02 00 10 00 40 60 08 00 6...	download pdo 0x1601 entr...
C <PS>	CoE	0x1602 C 0	02 00 10 00 40 60 20 00 7...	download pdo 0x1602 entr...
C <PS>	CoE	0x1604 C 0	02 00 10 00 40 60 10 00 7...	download pdo 0x1604 entr...
C <PS>	CoE	0x1605 C 0	00 00	download pdo 0x1605 entr...
C <PS>	CoE	0x1C12 C 0	01 00 02 16	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	01 00 02 1A	download pdo 0x1C13 index
C <PS>	CoE	0x1C14 C 0	00 00	download pdo 0x1C14 index
C <PS>	CoE	0x1C15 C 0	00 00	download pdo 0x1C15 index
A <IP, PS>	AoE	1/3	A9 FE 1D E6 03 02	AoE Init Cmd (download N...
C <PS>	CoE	0xF030 C 0	00 00	download slot cfg
E <IP, PS>	EoE		3F 00 00 00 02 01 05 20 0...	eoe init
C PS	CoE	0x6060:00	8	

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Status word	X	560	UINT	2.0	71.0	Input	0	nState1, nStat
Position actual v...	X	30260	DINT	4.0	73.0	Input	0	nDataIn1 . In .
WcState	X	0	BIT	0.1	1522.3	Input	0	nState4, nStat
InputToggle	X	1	BIT	0.1	1524.3	Input	0	nState4, nStat
State		8	UINT	2.0	1548.0	Input	0	
AdsAddr		169.254.29.230.3.1:...	AMSADDR	8.0	1550.0	Input	0	
AoeNetId		169.254.29.230.3.2	AMSNETID	6.0	1558.0	Input	0	

2. When the “Edit CANopen Startup Entry” dialog opens, populate it as shown below and check the “Complete Access” checkbox to allow the Sub-Index to be configured:



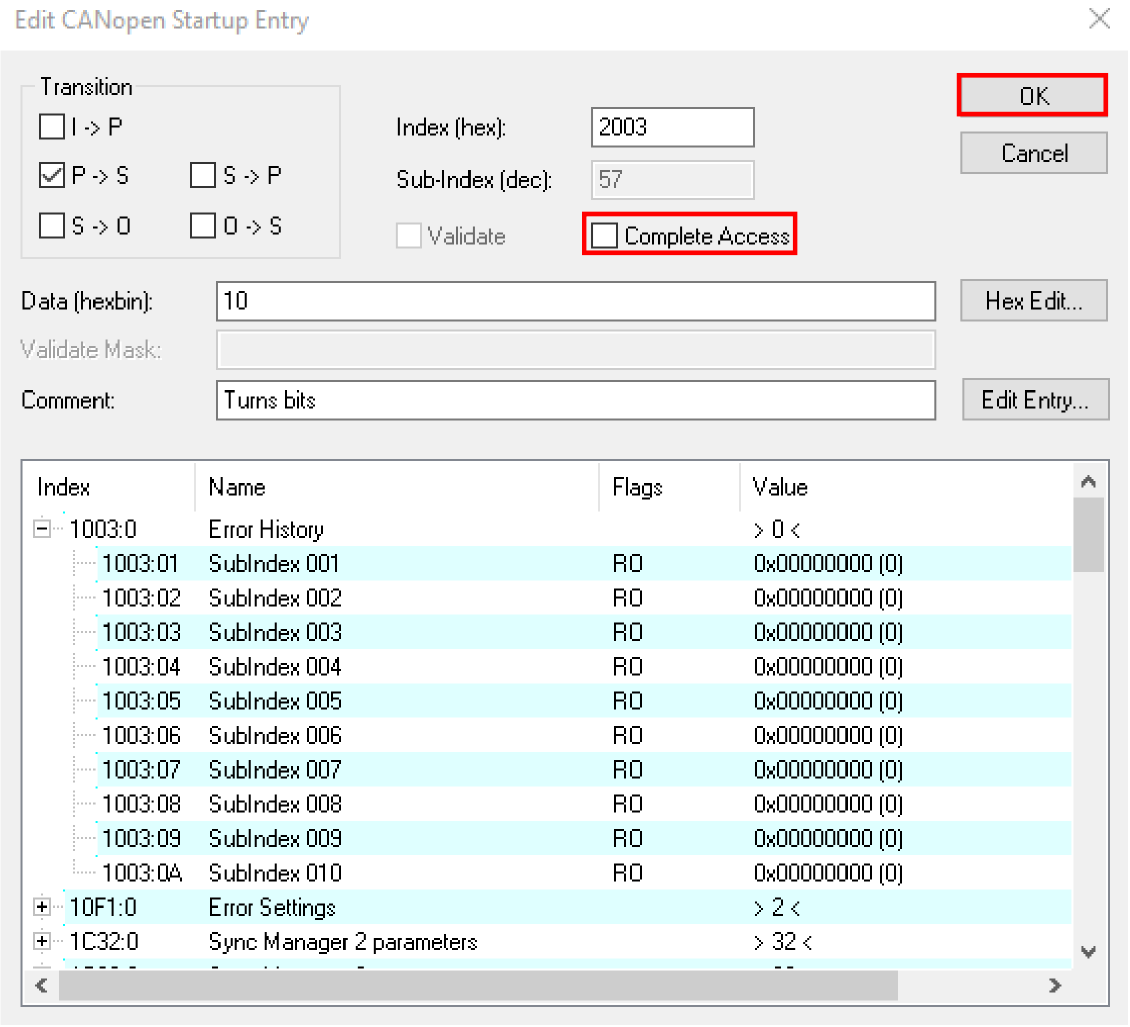
Edit CANopen Startup Entry

Transition
☐ I -> P
☑ P -> S
☐ S -> P
☐ S -> O
☐ O -> S

Index (hex): 2003
Sub-Index (dec): 57
☐ Validate
☑ Complete Access
Data (hexbin): 10
Validate Mask:
Comment: Turns bits

Index	Name	Flags	Value
1003:0	Error History		> 0 <
1003:01	SubIndex 001	RO	0x00000000 (0)
1003:02	SubIndex 002	RO	0x00000000 (0)
1003:03	SubIndex 003	RO	0x00000000 (0)
1003:04	SubIndex 004	RO	0x00000000 (0)
1003:05	SubIndex 005	RO	0x00000000 (0)
1003:06	SubIndex 006	RO	0x00000000 (0)
1003:07	SubIndex 007	RO	0x00000000 (0)
1003:08	SubIndex 008	RO	0x00000000 (0)
1003:09	SubIndex 009	RO	0x00000000 (0)
1003:0A	SubIndex 010	RO	0x00000000 (0)
10F1:0	Error Settings		> 2 <
1C32:0	Sync Manager 2 parameters		> 32 <

3. Uncheck the “Complete Access” checkbox and then click “OK”.



Edit CANopen Startup Entry

Transition
☐ I -> P
☑ P -> S
☐ S -> P
☐ S -> O
☐ O -> S

Index (hex): 2003
Sub-Index (dec): 57
☐ Validate
☐ Complete Access

Data (hexbin): 10
Validate Mask:
Comment: Turns bits

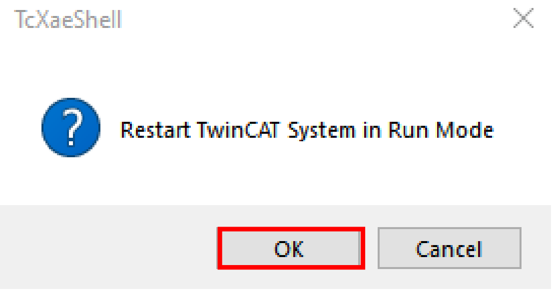
Index	Name	Flags	Value
1003:0	Error History		> 0 <
1003:01	SubIndex 001	RO	0x00000000 (0)
1003:02	SubIndex 002	RO	0x00000000 (0)
1003:03	SubIndex 003	RO	0x00000000 (0)
1003:04	SubIndex 004	RO	0x00000000 (0)
1003:05	SubIndex 005	RO	0x00000000 (0)
1003:06	SubIndex 006	RO	0x00000000 (0)
1003:07	SubIndex 007	RO	0x00000000 (0)
1003:08	SubIndex 008	RO	0x00000000 (0)
1003:09	SubIndex 009	RO	0x00000000 (0)
1003:0A	SubIndex 010	RO	0x00000000 (0)
10F1:0	Error Settings		> 2 <
1C32:0	Sync Manager 2 parameters		> 32 <

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

C PS CoE 0x2003:39 0x10 (16) Turns bits

5. Click the Activate Configuration button to apply the startup list.

6. Click OK on the “Restart TwinCAT System in Run Mode” dialog.



TcXaeShell

Restart TwinCAT System in Run Mode

7. After a few seconds the system will reset, and the new setting will be applied in the drive.

3.1.2 How to scale between NC axis speed and rpm

To convert an NC 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 1000°/s with a scaling numerator of 360 per rev. The resulting speed in rpm observed in Pr3.002 is 1000 * 60 / 360 = 166.6rpm

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

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

E.g. The axis is running at 1000rpm with a scaling numerator of 1234 mm per rev. The resulting speed in rpm observed in the NC axis ActVelo parameter is 1000 * 1234 / 60 = 20566.6rpm.

3.2 How to use a startup list in TwinCAT

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 couple of standard ways in which this feature is used:

1. 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.
2. 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.

The following sections detail how to use TwinCAT in these scenarios.

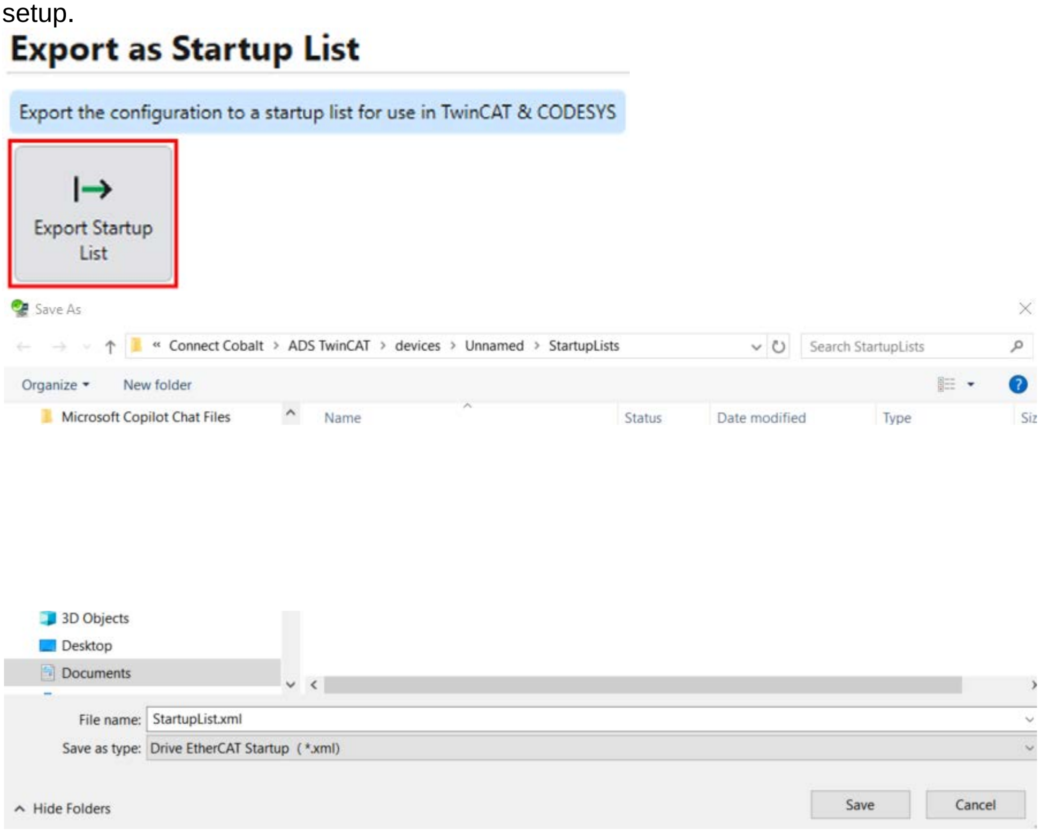
3.2.1 How to import a startup list .xml file from Connect into TwinCAT

A startup list .xml file can be generated by Control Techniques Connect after the axis has been commissioned. A startup list generated by Control Techniques Connect software holds all of the changes made to the drive in a format that TwinCAT can import, 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.

This startup list should be generated after the speed and position loops have been tuned so that all of the setup is retained.

The following steps show how to import the file into TwinCAT:

1. An EtherCAT startup list may be generated to preserve the configuration in the TwinCAT PLC/IPC. The startup list is in the form of an startup list .xml file that may be imported into TwinCAT. The export start-up list button can be found on the "Finalize" step in the guided setup.



Export as Startup List

Export the configuration to a startup list for use in TwinCAT & CODESYS

File name: StartupList.xml

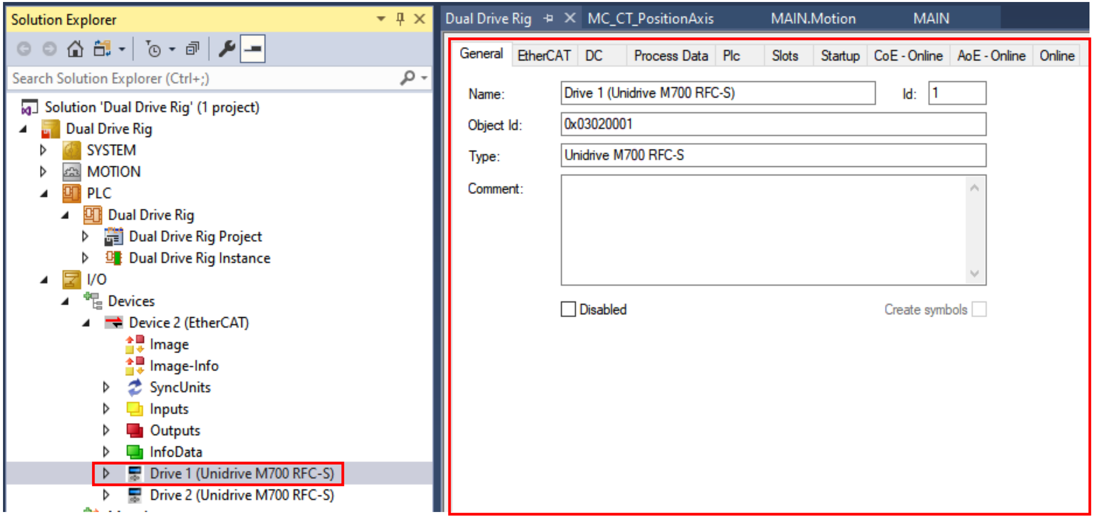
Save as type: Drive EtherCAT Startup (*.xml)

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

2. Open the TwinCAT project.

3. Make sure config mode is selected as indicated by the blue config button being highlighted. Click the blue Config button if it is not highlighted.

4. Double click on the drive node in the Solution explorer pane, that the startup list .xml file will be applied to. This opens the properties for the drive node.



Name: Drive 1 (Unidrive M700 RFC-S) Id: 1

Object Id: 0x03020001

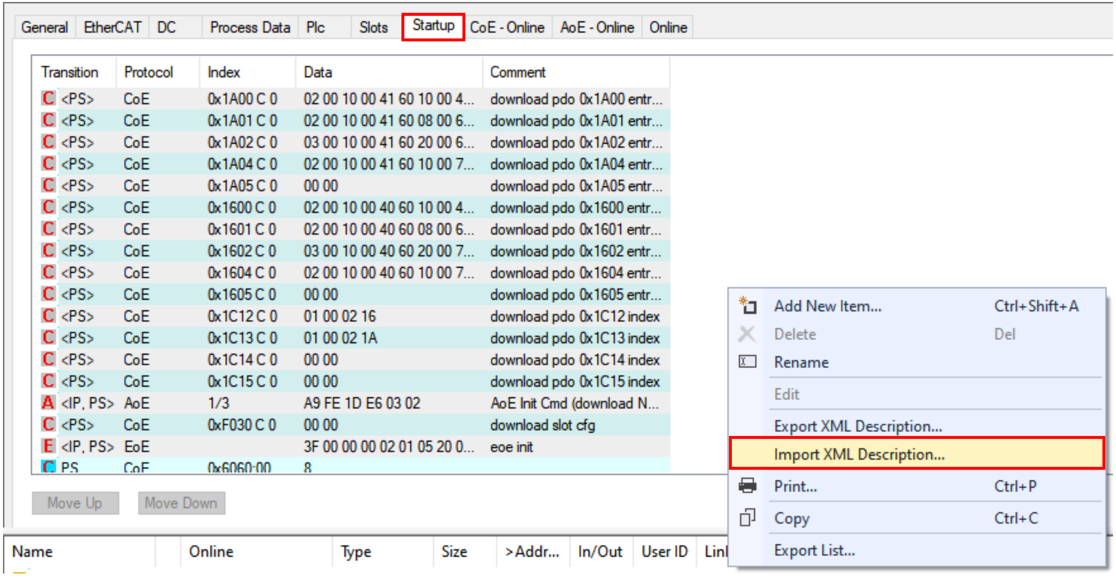
Type: Unidrive M700 RFC-S

Comment:

☐ Disabled

Create symbols ☐

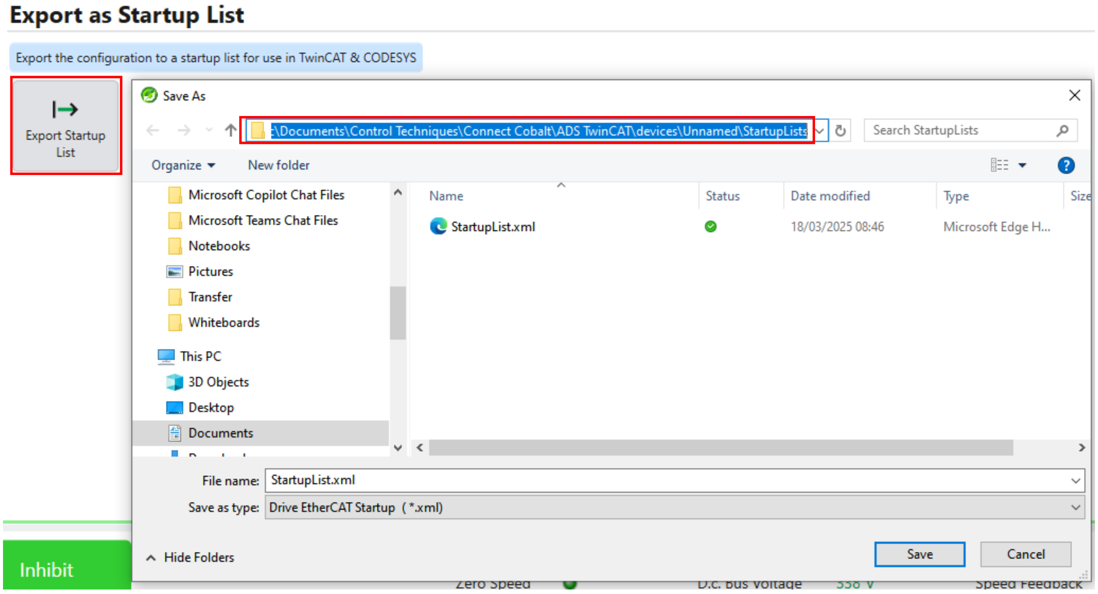
5. Select the Startup tab. Right click anywhere on the startup tab, and select “Import XML Description…”



Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1A00 C 0	02 00 10 00 41 60 10 00 4...	download pdo 0x1A00 entr...
C <PS>	CoE	0x1A01 C 0	02 00 10 00 41 60 08 00 6...	download pdo 0x1A01 entr...
C <PS>	CoE	0x1A02 C 0	03 00 10 00 41 60 20 00 6...	download pdo 0x1A02 entr...
C <PS>	CoE	0x1A04 C 0	02 00 10 00 41 60 10 00 7...	download pdo 0x1A04 entr...
C <PS>	CoE	0x1A05 C 0	00 00	download pdo 0x1A05 entr...
C <PS>	CoE	0x1600 C 0	02 00 10 00 40 60 10 00 4...	download pdo 0x1600 entr...
C <PS>	CoE	0x1601 C 0	02 00 10 00 40 60 08 00 6...	download pdo 0x1601 entr...
C <PS>	CoE	0x1602 C 0	03 00 10 00 40 60 20 00 7...	download pdo 0x1602 entr...
C <PS>	CoE	0x1604 C 0	02 00 10 00 40 60 10 00 7...	download pdo 0x1604 entr...
C <PS>	CoE	0x1605 C 0	00 00	download pdo 0x1605 entr...
C <PS>	CoE	0x1C12 C 0	01 00 02 16	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	01 00 02 1A	download pdo 0x1C13 index
C <PS>	CoE	0x1C14 C 0	00 00	download pdo 0x1C14 index
C <PS>	CoE	0x1C15 C 0	00 00	download pdo 0x1C15 index
A <IP, PS>	AoE	1/3	A9 FE 1D E6 03 02	AoE Init Cmd (download N...
C <PS>	CoE	0xF030 C 0	00 00	download slot cfg
E <IP, PS>	EoE		3F 00 00 00 02 01 05 20 0...	eoe init
C PS	CoE	0x6060:00	8	

6. Navigate to the location of the startup list generated by Connect. The general form for the path to the .xml file is “C:\Users\[USER NAME]\OneDrive\Documents\Control Techniques\Connect Cobalt\[PROJECT NAME]\devices\[DEVICE NAME]\StartupLists”

   Alternatively the path for the .xml file can be copied by opening the original Connect project, go to “Setup” > “Finalise” page and then click on “Export Startup List” and then copy the path from the explorer window.



Export as Startup List

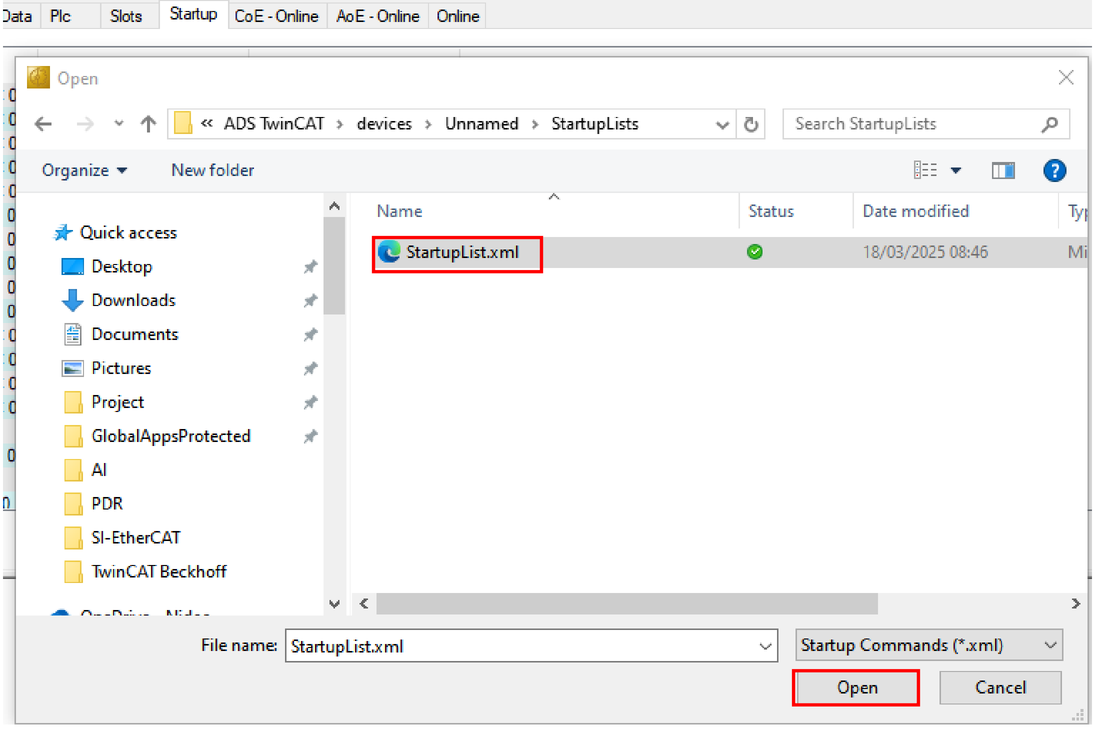
Export the configuration to a startup list for use in TwinCAT & CODESYS

Path: :\Documents\Control Techniques\Connect Cobalt\ADS TwinCAT\devices\Unnamed\StartupLists

File name: StartupList.xml

Save as type: Drive EtherCAT Startup (*.xml)

   Paste the path from Connect into the TwinCAT import .XML window path and click “Open” when finished.

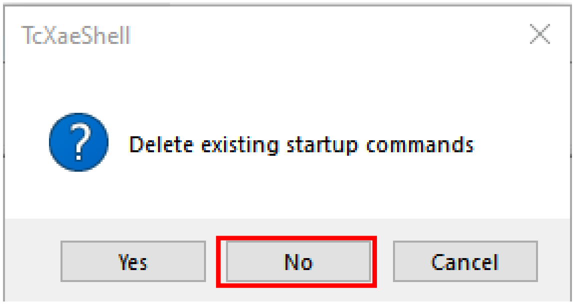


Path: « ADS TwinCAT › devices › Unnamed › StartupLists

File name: StartupList.xml

Startup Commands (*.xml)

7. A pop up will ask if the original start up list should be deleted? Select “No” which instead of deleting the original start up list and replacing it with the one from Connect, it will append / add to the existing list with the extra setup imported from Connect.



TcXaeShell

Delete existing startup commands

8. Apply the startup list by clicking the “Activate Configuration” button, and click “OK” or “Yes” on any following pop-up messages:

3.2.2 How to add a single CAN object to the TwinCAT Startup List & Cyclic comms loss action

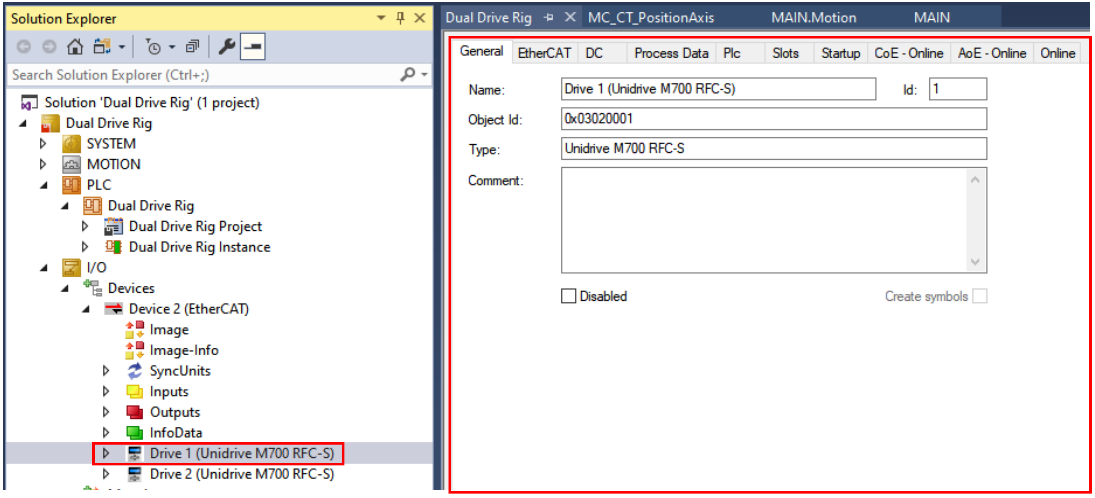
This section describes how to add single CAN objects to the startup list. The object may be ones specified by the CiA402 standard, custom ones as specified in the SI-EtherCAT manual e.g. 0x3005 Cyclic data loss behaviour, or drive parameters arranged in CAN object format - see section **3.3.1 How to convert parameter numbers to CANopen object references**.

Use the following steps to add ESI specified CANopen objects to the startup list in TwinCAT:

1. Open the TwinCAT project.

2. Make sure config mode is selected as indicated by the blue config button being highlighted. Click the blue Config button if it is not highlighted. Click "OK" on any following po-up messages.

3. Double click on the drive node in the Solution explorer pane, that the startup list .xml file will be applied to. This opens the properties for the drive node.



Name: Drive 1 (Unidrive M700 RFC-S)

Id: 1

Object Id: 0x03020001

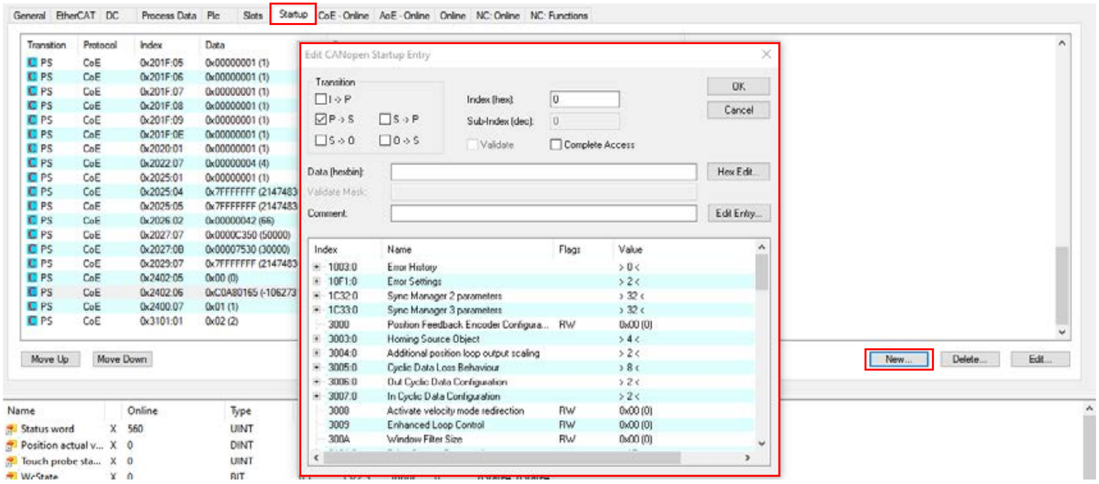
Type: Unidrive M700 RFC-S

Comment:

☐ Disabled

Create symbols ☐

4. Select the Startup tab. Then click on "New":



Transition	Protocol	Index	Data
PS	CoE	0x201F:05	0x00000001 (1)
PS	CoE	0x201F:06	0x00000001 (1)
PS	CoE	0x201F:07	0x00000001 (1)
PS	CoE	0x201F:08	0x00000001 (1)
PS	CoE	0x201F:09	0x00000001 (1)
PS	CoE	0x201F:0E	0x00000001 (1)
PS	CoE	0x2020:01	0x00000001 (1)
PS	CoE	0x2022:07	0x00000004 (4)
PS	CoE	0x2025:01	0x00000001 (1)
PS	CoE	0x2025:04	0x7FFFFFFF (2147483
PS	CoE	0x2025:05	0x7FFFFFFF (2147483
PS	CoE	0x2026:02	0x00000042 (66)
PS	CoE	0x2027:07	0x0000C350 (50000)
PS	CoE	0x2027:08	0x00007530 (30000)
PS	CoE	0x2029:07	0x7FFFFFFF (2147483
PS	CoE	0x2402:05	0x00 (0)
PS	CoE	0x2402:06	0xC0A80165 (-106273
PS	CoE	0x2400:07	0x01 (1)
PS	CoE	0x3101:01	0x02 (2)

Name		Online	Type
Status word	X	560	UINT
Position actual v...	X	0	DINT
Touch probe sta...	X	0	UINT
WcState	X	0	BIT

Transition

☐ I -> P

☑ P -> S ☐ S -> P

☐ S -> 0 ☐ 0 -> S

Index (hex): 0

Sub-Index (dec): 0

☐ Validate ☐ Complete Access

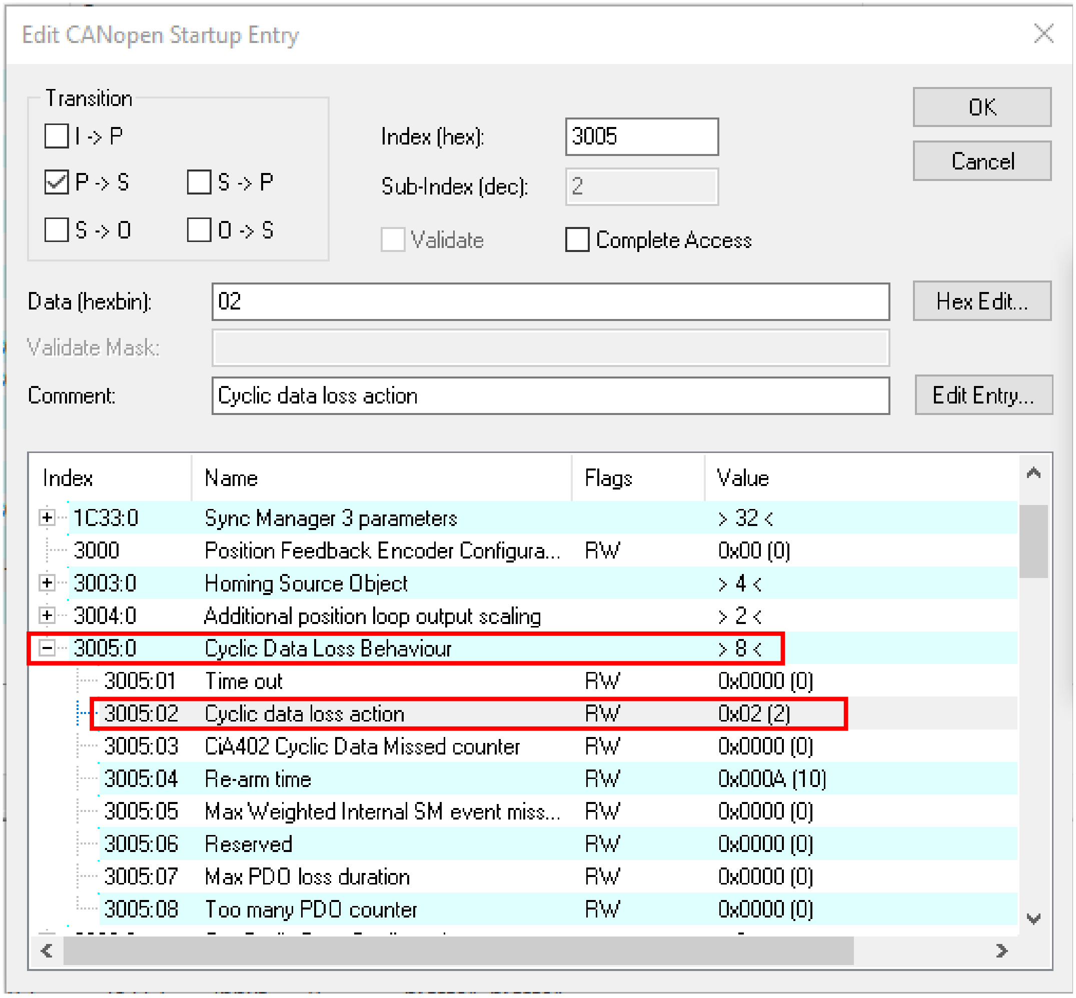
Data (hexbin):

Validate Mask:

Comment:

Index	Name	Flags	Value
1003:0	Error History		> 0 <
10F1:0	Error Settings		> 2 <
1C32:0	Sync Manager 2 parameters		> 32 <
1C33:0	Sync Manager 3 parameters		> 32 <
3000	Position Feedback Encoder Configura...	RW	0x00 (0)
3003:0	Homing Source Object		> 4 <
3004:0	Additional position loop output scaling		> 2 <
3005:0	Cyclic Data Loss Behaviour		> 8 <
3006:0	Out Cyclic Data Configuration		> 2 <
3007:0	In Cyclic Data Configuration		> 2 <
3000	Activate velocity mode redirection	RW	0x00 (0)
3009	Enhanced Loop Control	RW	0x00 (0)
300A	Window Filter Size	RW	0x00 (0)

5. Select the CANopen object index and subindex to edit from the list. In this example Index 3005 subindex 2 is selected.



Transition
☐ I -> P
☑ P -> S ☐ S -> P
☐ S -> O ☐ O -> S

Index (hex): 3005

Sub-Index (dec): 2

☐ Validate ☐ Complete Access

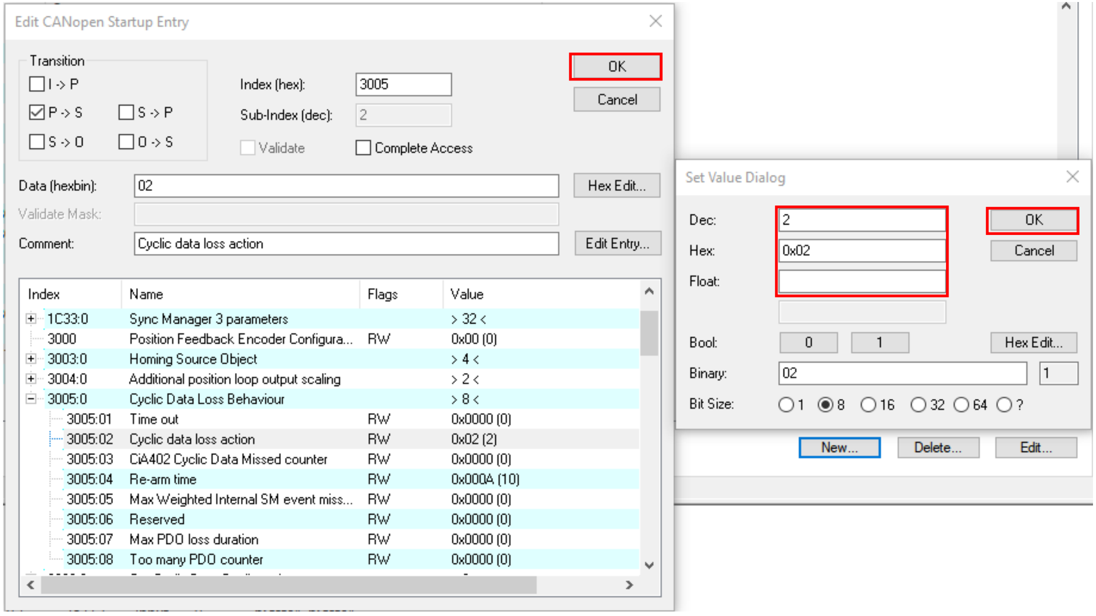
Data (hexbin): 02

Validate Mask:

Comment: Cyclic data loss action

Index	Name	Flags	Value
1C33:0	Sync Manager 3 parameters		> 32 <
3000	Position Feedback Encoder Configura...	RW	0x00 (0)
3003:0	Homing Source Object		> 4 <
3004:0	Additional position loop output scaling		> 2 <
3005:0	Cyclic Data Loss Behaviour		> 8 <
3005:01	Time out	RW	0x0000 (0)
3005:02	Cyclic data loss action	RW	0x02 (2)
3005:03	CiA402 Cyclic Data Missed counter	RW	0x0000 (0)
3005:04	Re-arm time	RW	0x000A (10)
3005:05	Max Weighted Internal SM event miss...	RW	0x0000 (0)
3005:06	Reserved	RW	0x0000 (0)
3005:07	Max PDO loss duration	RW	0x0000 (0)
3005:08	Too many PDO counter	RW	0x0000 (0)

6. Double click on the subindex to open the “Set Value Dialog”. Enter the value as either a decimal or hex value. When finished click “OK” in the “Set Value Dialog” and the “Edit CANopen Startup Entry” dialog.



Transition
☐ I -> P
☑ P -> S ☐ S -> P
☐ S -> O ☐ O -> S

Index (hex): 3005
Sub-Index (dec): 2
☐ Validate ☐ Complete Access

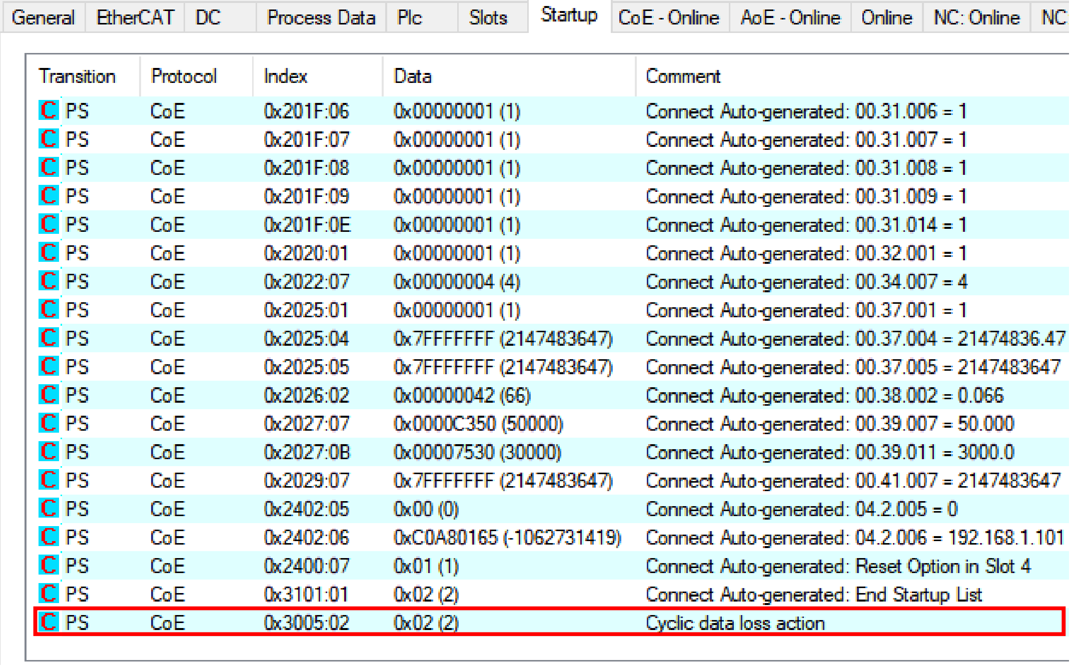
Data (hexbin): 02
Validate Mask:
Comment: Cyclic data loss action

Index	Name	Flags	Value
1C33:0	Sync Manager 3 parameters		> 32 <
3000	Position Feedback Encoder Configura...	RW	0x00 (0)
3003:0	Homing Source Object		> 4 <
3004:0	Additional position loop output scaling		> 2 <
3005:0	Cyclic Data Loss Behaviour		> 8 <
3005:01	Time out	RW	0x0000 (0)
3005:02	Cyclic data loss action	RW	0x02 (2)
3005:03	CiA402 Cyclic Data Missed counter	RW	0x0000 (0)
3005:04	Re-arm time	RW	0x000A (10)
3005:05	Max Weighted Internal SM event miss...	RW	0x0000 (0)
3005:06	Reserved	RW	0x0000 (0)
3005:07	Max PDO loss duration	RW	0x0000 (0)
3005:08	Too many PDO counter	RW	0x0000 (0)

Set Value Dialog
Dec: 2
Hex: 0x02
Float:
Bool: 0 1
Binary: 02 1
Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

HINT: Data is entered in this tool in hex with the byte pair values entered in reverse endian format. This means the least significant byte pair is entered first followed by the other byte pairs with increasing significance. E.g. a 32bit value in conventional hex format is written as 0x 12 34 56 78, but in reverse endian format it is entered as 0x 78 56 34 12.

7. The new entry is added to the startup list.



Transition	Protocol	Index	Data	Comment
C PS	CoE	0x201F:06	0x00000001 (1)	Connect Auto-generated: 00.31.006 = 1
C PS	CoE	0x201F:07	0x00000001 (1)	Connect Auto-generated: 00.31.007 = 1
C PS	CoE	0x201F:08	0x00000001 (1)	Connect Auto-generated: 00.31.008 = 1
C PS	CoE	0x201F:09	0x00000001 (1)	Connect Auto-generated: 00.31.009 = 1
C PS	CoE	0x201F:0E	0x00000001 (1)	Connect Auto-generated: 00.31.014 = 1
C PS	CoE	0x2020:01	0x00000001 (1)	Connect Auto-generated: 00.32.001 = 1
C PS	CoE	0x2022:07	0x00000004 (4)	Connect Auto-generated: 00.34.007 = 4
C PS	CoE	0x2025:01	0x00000001 (1)	Connect Auto-generated: 00.37.001 = 1
C PS	CoE	0x2025:04	0x7FFFFFFF (2147483647)	Connect Auto-generated: 00.37.004 = 21474836.47
C PS	CoE	0x2025:05	0x7FFFFFFF (2147483647)	Connect Auto-generated: 00.37.005 = 2147483647
C PS	CoE	0x2026:02	0x00000042 (66)	Connect Auto-generated: 00.38.002 = 0.066
C PS	CoE	0x2027:07	0x0000C350 (50000)	Connect Auto-generated: 00.39.007 = 50.000
C PS	CoE	0x2027:0B	0x00007530 (30000)	Connect Auto-generated: 00.39.011 = 3000.0
C PS	CoE	0x2029:07	0x7FFFFFFF (2147483647)	Connect Auto-generated: 00.41.007 = 2147483647
C PS	CoE	0x2402:05	0x00 (0)	Connect Auto-generated: 04.2.005 = 0
C PS	CoE	0x2402:06	0xC0A80165 (-1062731419)	Connect Auto-generated: 04.2.006 = 192.168.1.101
C PS	CoE	0x2400:07	0x01 (1)	Connect Auto-generated: Reset Option in Slot 4
C PS	CoE	0x3101:01	0x02 (2)	Connect Auto-generated: End Startup List
C PS	CoE	0x3005:02	0x02 (2)	Cyclic data loss action

Apply the startup list by clicking the “Activate Configuration” button. Click “OK” on any following pop-up messages.

3.3 How to configure PDO mappings

3.3.1 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)

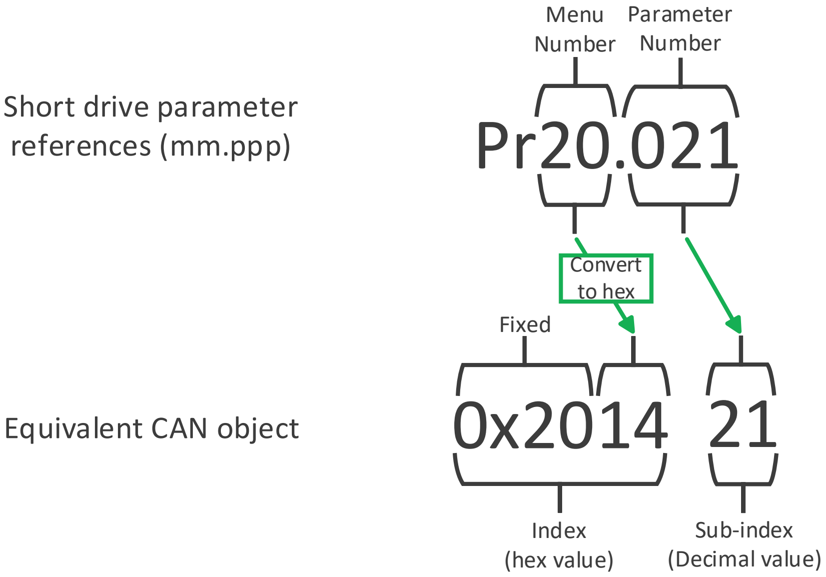
Subindex = Parameter number

E.g. Drive parameter Pr3.017 is referenced as Index = 2003, Subindex = 17

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

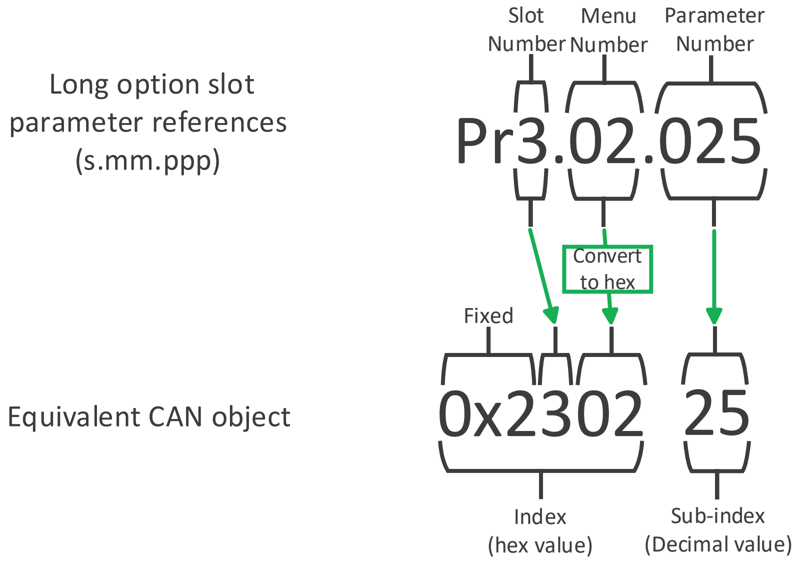
3.3.1.1 Short drive parameter references

Pr20.021 → Index 0x2014 Sub-index 21



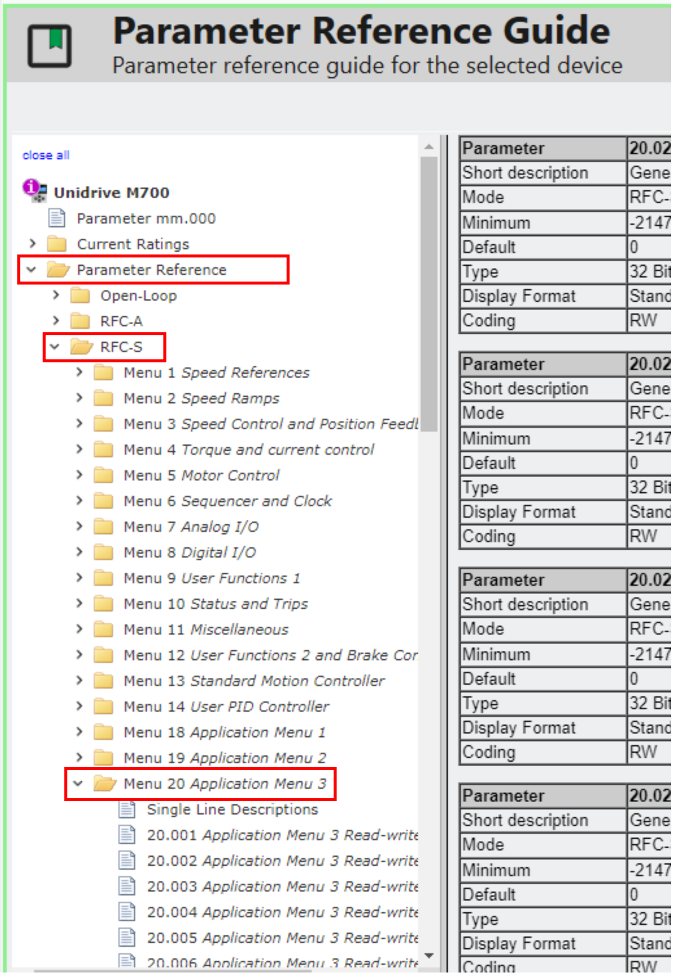
3.3.1.2 Long option parameter references

Pr3.02.025 → Index 0x2302 Sub-index 25



3.3.1.3 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.



Parameter Reference Guide

Parameter reference guide for the selected device

close all

- **Unidrive M700**
  - Parameter mm.000
  - Current Ratings
  - Parameter Reference
    - Open-Loop
    - RFC-A
    - RFC-S
      - Menu 1 *Speed References*
      - Menu 2 *Speed Ramps*
      - Menu 3 *Speed Control and Position Feedl*
      - Menu 4 *Torque and current control*
      - Menu 5 *Motor Control*
      - Menu 6 *Sequencer and Clock*
      - Menu 7 *Analog I/O*
      - Menu 8 *Digital I/O*
      - Menu 9 *User Functions 1*
      - Menu 10 *Status and Trips*
      - Menu 11 *Miscellaneous*
      - Menu 12 *User Functions 2 and Brake Cor*
      - Menu 13 *Standard Motion Controller*
      - Menu 14 *User PID Controller*
      - Menu 18 *Application Menu 1*
      - Menu 19 *Application Menu 2*
      - Menu 20 *Application Menu 3*
        - Single Line Descriptions
        - 20.001 *Application Menu 3 Read-write*
        - 20.002 *Application Menu 3 Read-write*
        - 20.003 *Application Menu 3 Read-write*
        - 20.004 *Application Menu 3 Read-write*
        - 20.005 *Application Menu 3 Read-write*
        - 20.006 *Application Menu 3 Read-write*

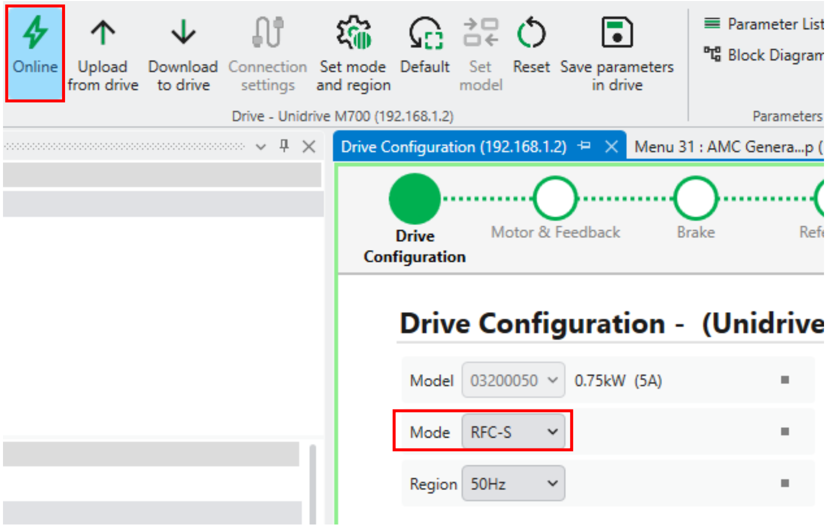
Parameter	20.02
Short description	Gene
Mode	RFC-
Minimum	-2147
Default	0
Type	32 Bit
Display Format	Stand
Coding	RW

Parameter	20.02
Short description	Gene
Mode	RFC-
Minimum	-2147
Default	0
Type	32 Bit
Display Format	Stand
Coding	RW

Parameter	20.02
Short description	Gene
Mode	RFC-
Minimum	-2147
Default	0
Type	32 Bit
Display Format	Stand
Coding	RW

Parameter	20.02
Short description	Gene
Mode	RFC-
Minimum	-2147
Default	0
Type	32 Bit
Display Format	Stand
Coding	RW

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”.



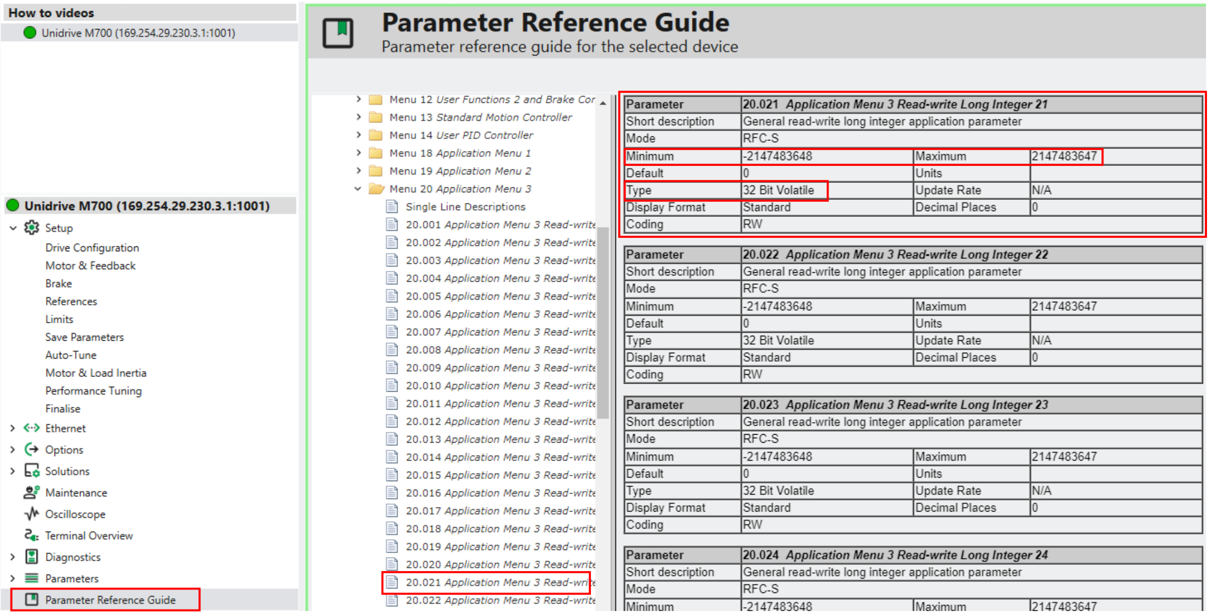
Drive Configuration - (Unidrive

Model 03200050 0.75kW (5A)

Mode RFC-S

Region 50Hz

Select the parameter to get the data type for:



Parameter Reference Guide

Parameter reference guide for the selected device

Parameter	**20.021 *Application Menu 3 Read-write Long Integer 21***		
Short description	General read-write long integer application parameter		
Mode	RFC-S		
Minimum	-2147483648	Maximum	2147483647
Default	0	Units	
Type	32 Bit Volatile	Update Rate	N/A
Display Format	Standard	Decimal Places	0
Coding	RW		

Parameter	**20.022 *Application Menu 3 Read-write Long Integer 22***		
Short description	General read-write long integer application parameter		
Mode	RFC-S		
Minimum	-2147483648	Maximum	2147483647
Default	0	Units	
Type	32 Bit Volatile	Update Rate	N/A
Display Format	Standard	Decimal Places	0
Coding	RW		

Parameter	**20.023 *Application Menu 3 Read-write Long Integer 23***		
Short description	General read-write long integer application parameter		
Mode	RFC-S		
Minimum	-2147483648	Maximum	2147483647
Default	0	Units	
Type	32 Bit Volatile	Update Rate	N/A
Display Format	Standard	Decimal Places	0
Coding	RW		

Parameter	**20.024 *Application Menu 3 Read-write Long Integer 24***		
Short description	General read-write long integer application parameter		
Mode	RFC-S		
Minimum	-2147483648	Maximum	2147483647

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 TwinCAT:

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

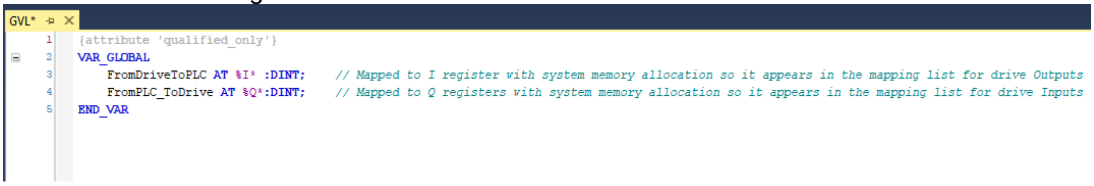
E.g. Pr20.021 is a 32bit value and the range is -2147483648 to 2147483647 which shows that the value is signed, therefore the IEC data type used in TwinCAT is DINT.

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

3.3.2 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 GVL (Global Variable List) making the data available to any POU. Right click on the GVL folder and select “Add” > “Global Variable List”. Name the list and then in declaration area of the GVL add the following:



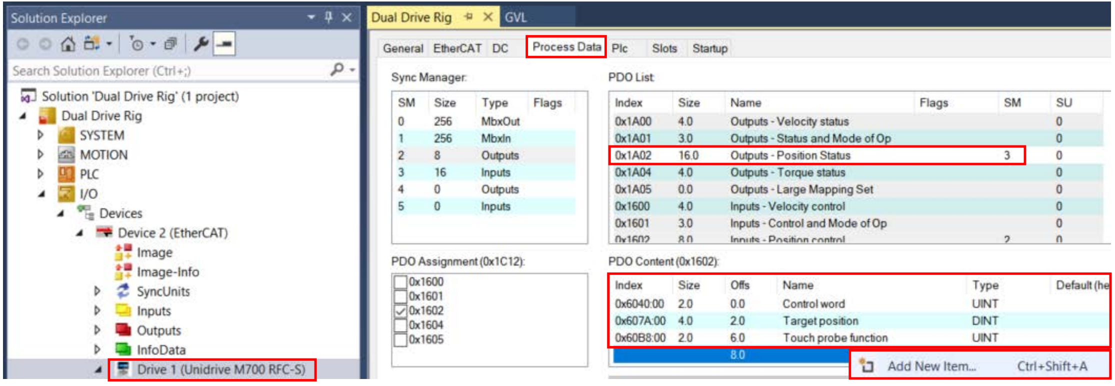
```
{attribute 'qualified_only'}
VAR_GLOBAL
    FromDriveToPLC AT %I* :DINT;    // Mapped to I register with system memory allocation so it appears in the mapping list for drive Outputs
    FromPLC_ToDrive AT %Q*:DINT;    // Mapped to Q registers with system memory allocation so it appears in the mapping list for drive Inputs
END_VAR
```

HINT: The variables in the list have been allocated to the I and Q memory areas. This is a key step that allows the variables to be “seen” in the in the PDO linking list. Take care to use the correct memory area where I registers are for data going from the drive to the PLC, and Q registers are for data going from the PLC to the drive.

HINT: The “*” used in the memory allocation e.g. %I* allows the TwinCAT system to select which area of the registers should be used. This is recommended.

2. When the GVL is created build the application by pressing Ctrl+Shift+b. This will make the GVL elements available in later steps. Failure to do this will mean that the variables such as FromDriveToPLC will not appear in the data mapping selector.

3. In the “Solution Explorer” tree, double click on the drive node to exchange PDO data with to open the drive properties. Next click on the “Process Data” tab, then select the 0x1A02 index from the PDO list, then right click in the PDO Content box and select “Add New Item…”



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	8	Outputs	
3	16	Inputs	
4	0	Outputs	
5	0	Inputs	

PDO Assignment (0x1C12):

☐ 0x1600
☐ 0x1601
☑ 0x1602
☐ 0x1604
☐ 0x1605

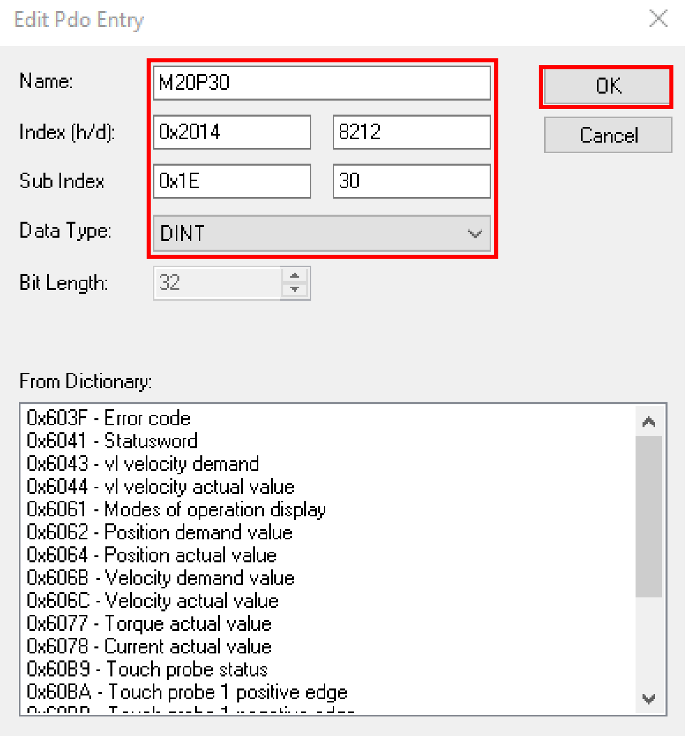
PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	4.0	Outputs - Velocity status			0
0x1A01	3.0	Outputs - Status and Mode of Op			0
0x1A02	16.0	Outputs - Position Status		3	0
0x1A04	4.0	Outputs - Torque status			0
0x1A05	0.0	Outputs - Large Mapping Set			0
0x1600	4.0	Inputs - Velocity control			0
0x1601	3.0	Inputs - Control and Mode of Op			0
0x1602	8.0	Inputs - Position control		2	0

PDO Content (0x1602):

Index	Size	Offs	Name	Type	Default (he
0x6040:00	2.0	0.0	Control word	UINT	
0x607A:00	4.0	2.0	Target position	DINT	
0x60B8:00	2.0	6.0	Touch probe function	UINT	
		8.0			

4. Using the information in section **3.3.1 How to convert parameter numbers to CANopen object references**, add the reference for the parameter to receive data from. For this example Pr20.030 will be used. Click “OK” when the parameter reference has been added.



Name: M20P30

Index (h/d): 0x2014 8212

Sub Index 0x1E 30

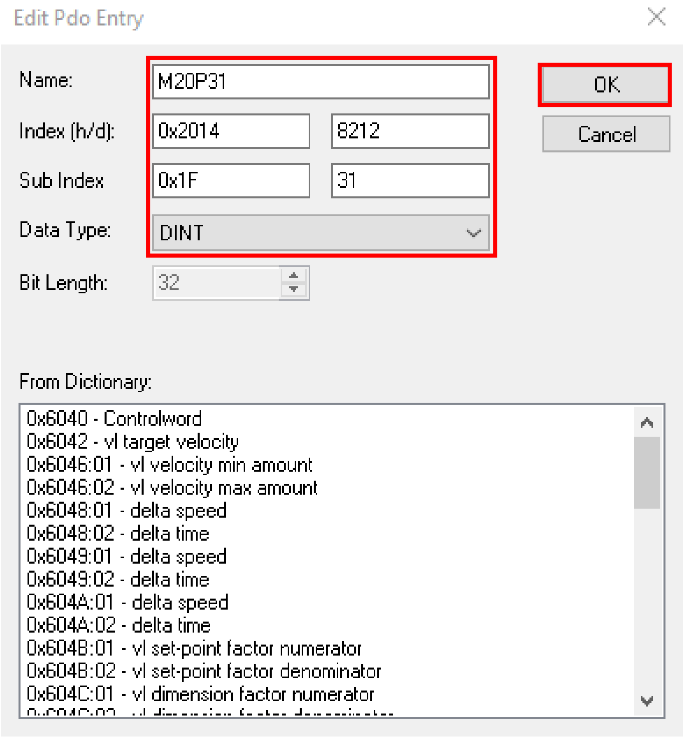
Data Type: DINT

Bit Length: 32

From Dictionary:

0x603F - Error code
0x6041 - Statusword
0x6043 - vl velocity demand
0x6044 - vl velocity actual value
0x6061 - Modes of operation display
0x6062 - Position demand value
0x6064 - Position actual value
0x606B - Velocity demand value
0x606C - Velocity actual value
0x6077 - Torque actual value
0x6078 - Current actual value
0x60B9 - Touch probe status
0x60BA - Touch probe 1 positive edge

5. Select “0x1602” in the PDO list, then right click in the PDO Content box and select “Add New Item…”. Using the information in section **3.3.1 How to convert parameter numbers to CANopen object references**, add the reference for the parameter to send data to. For this example Pr20.031 will be used. Click “OK” when the parameter reference has been added. add the reference for the parameter to send data to.



Name: M20P31

Index (h/d): 0x2014 8212

Sub Index 0x1F 31

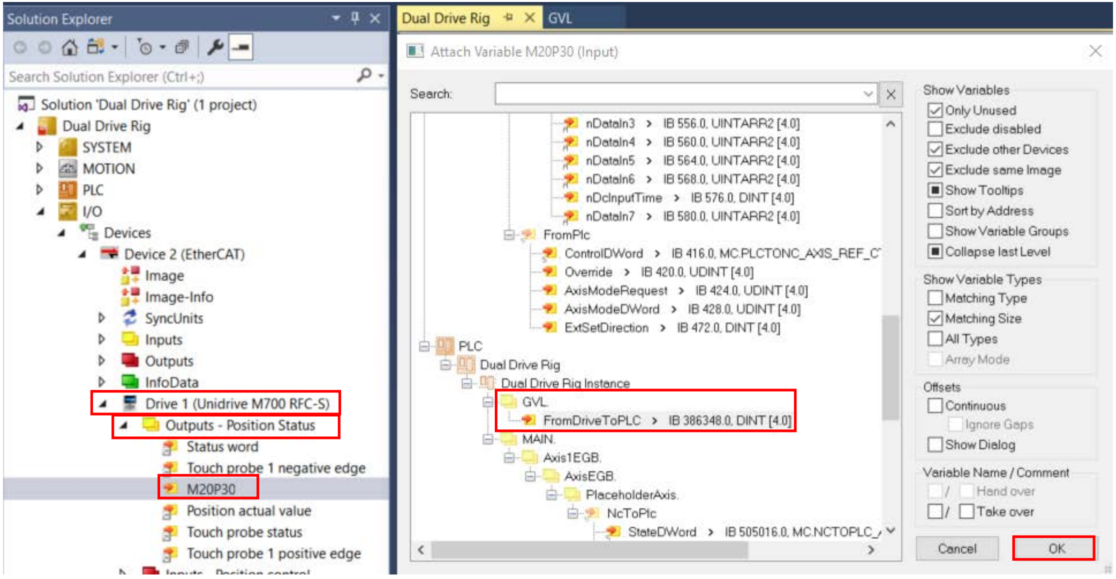
Data Type: DINT

Bit Length: 32

From Dictionary:

0x6040 - Controlword
0x6042 - vl target velocity
0x6046:01 - vl velocity min amount
0x6046:02 - vl velocity max amount
0x6048:01 - delta speed
0x6048:02 - delta time
0x6049:01 - delta speed
0x6049:02 - delta time
0x604A:01 - delta speed
0x604A:02 - delta time
0x604B:01 - vl set-point factor numerator
0x604B:02 - vl set-point factor denominator
0x604C:01 - vl dimension factor numerator

6. Map the Output PDO “M20.P30” to “GVL.FromDriveToPLC”. Expand the tree under the drive to reveal the Output mappings, then double click on the mapping to link. When the “Attach Variable” dialog opens, scroll down the list until “GVL.FromDriveToPLC” is seen. Click on “GVL.FromDriveToPLC” and then click “OK” to link the mapping to the variable.



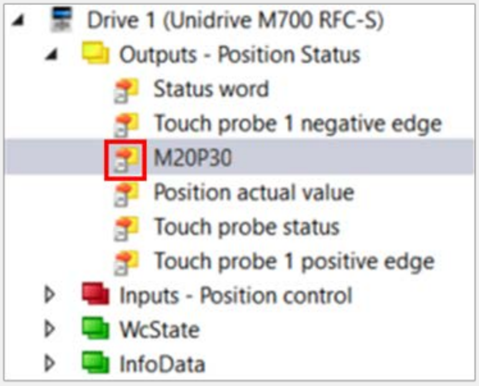
Show Variables
☑ Only Unused
☐ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
■ Show Tooltips
☐ Sort by Address
☐ Show Variable Groups
■ Collapse last Level

Show Variable Types
☐ Matching Type
☑ Matching Size
☐ All Types
☐ Array Mode

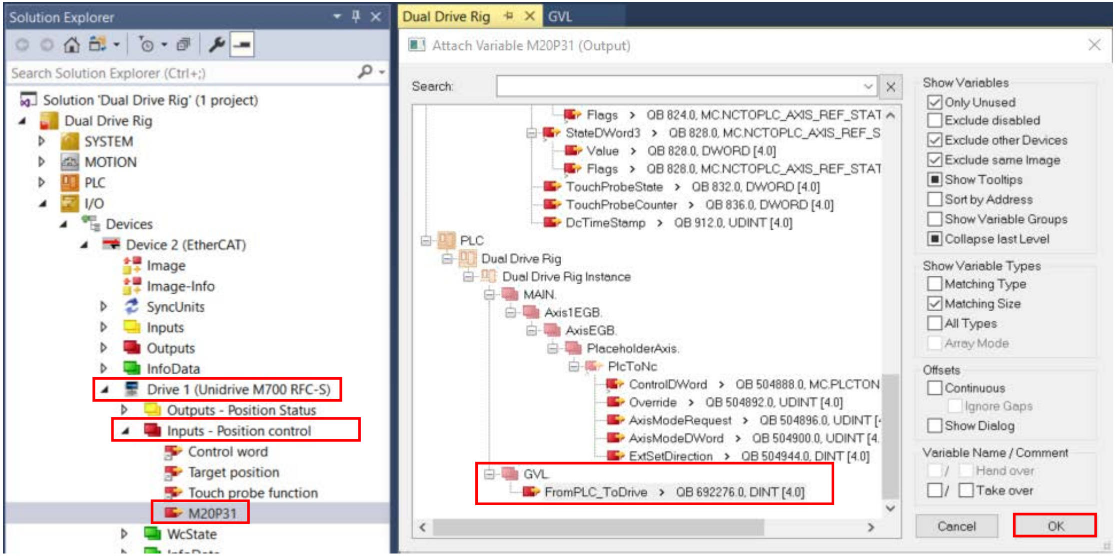
Offsets
☐ Continuous
☐ Ignore Gaps
☐ Show Dialog

Variable Name / Comment
☐ / ☐ Hand over
☐ / ☐ Take over

HINT: The link icon is now visible, (small grey box with an arrow inside), indicating that the PDO mapping has been linked.



7. Map the Input PDO “M20.P31” to “GVL.FromPLC_ToDrive”. Expand the tree under the drive to reveal the Input mappings, then double click on the mapping to link. When the attach variable dialog opens, scroll down the list until “ GVL.FromPLC_ToDrive” is seen. Click on “ GVL.FromPLC_ToDrive” and then click “OK” to link the mapping to the variable.



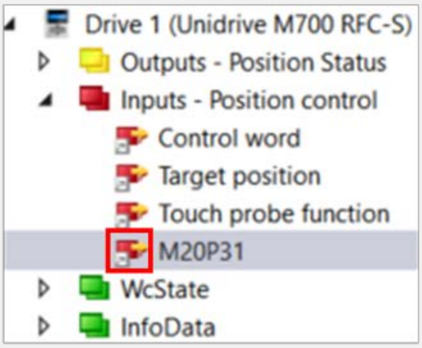
Show Variables
☑ Only Unused
☐ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
☒ Show Tooltips
☐ Sort by Address
☐ Show Variable Groups
☒ Collapse last Level

Show Variable Types
☐ Matching Type
☑ Matching Size
☐ All Types
☐ Array Mode

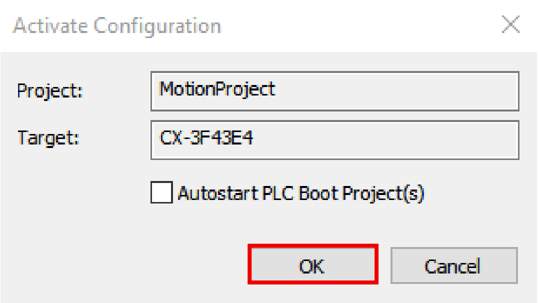
Offsets
☐ Continuous
☐ Ignore Gaps
☐ Show Dialog

Variable Name / Comment
☐ / ☐ Hand over
☐ / ☐ Take over

HINT: The link icon is now visible, (small grey box with an arrow inside), indicating that the PDO mapping has been linked.



8. Click the “Activate Configuration” button and then click “OK” if the “Activate Configuration” dialog appears:



Project: MotionProject

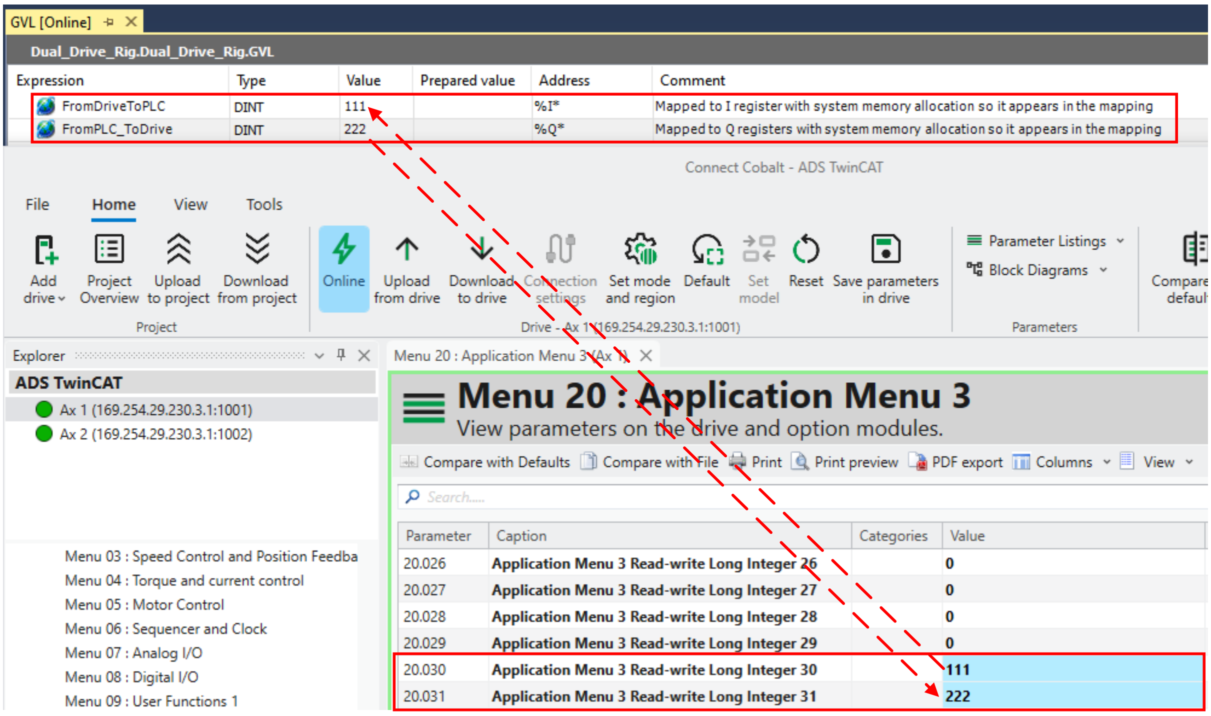
Target: CX-3F43E4

☐ Autostart PLC Boot Project(s)

9. Click the login icon to view the code while it runs:

10. Run the application (if it is not already running) by pressing the button.

Double click on the GVL in “Solution Explorer”. Values set in Pr20.030 in the drive are now passed to “GVL.FromDriveToPLC” and values set in “GVL.FromPLC_ToDrive” are passed to Pr20.031.



Expression	Type	Value	Prepared value	Address	Comment
FromDriveToPLC	DINT	111		%I*	Mapped to I register with system memory allocation so it appears in the mapping
FromPLC_ToDrive	DINT	222		%Q*	Mapped to Q registers with system memory allocation so it appears in the mapping

Parameter	Caption	Categories	Value
20.026	Application Menu 3 Read-write Long Integer 26		0
20.027	Application Menu 3 Read-write Long Integer 27		0
20.028	Application Menu 3 Read-write Long Integer 28		0
20.029	Application Menu 3 Read-write Long Integer 29		0
20.030	Application Menu 3 Read-write Long Integer 30		111
20.031	Application Menu 3 Read-write Long Integer 31		222

In the previous example it can be seen that a value of “111” was passed from Pr20.030 to GVL.FromDriveToPLC, and that a value of “222” was passed from GVL.FromPLC_ToDrive to Pr20.031. The screen shot shows the GVL values from TwinCAT and the drive parameter values in Connect.

3.3.3 How to route the drive P2 encoder interface to an encoder axis

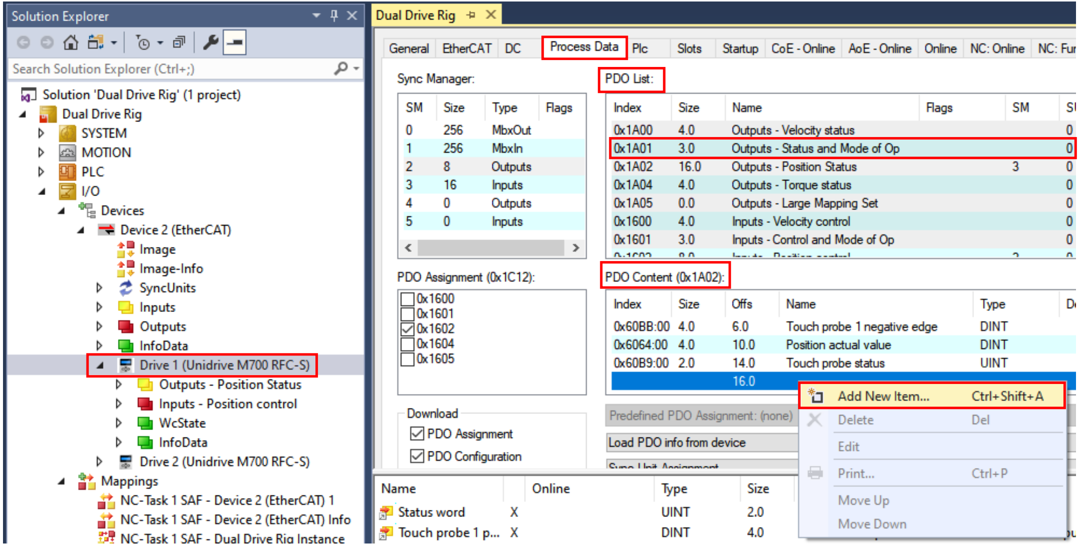
It is possible to map a drive encoder e.g. a line encoder connected to the drive P2 interface or an SI-Universal Encoder module, so that it could be used as a master position reference for an EGB axis. The following steps show how to configure the mapping and setup the Encoder axis.

1. It is assumed that the encoder to map to has already been connected to a spare encoder port on the drive e.g. Drive P2 interface on the D-type encoder port, or an encoder module such as an SI-Universal Encoder fitted in one of the drive option slots. Find the Normalised Position parameter. The table below shows the parameters for each possible encoder port and the equivalent Index and Subindex to use in the TwinCAT Process Data mapping :

Encoder port	Parameter	Index	Subindex	Data Type
Drive P1	Pr3.058	0x2003	0x3A	DINT (32Bit)
Drive P2	Pr3.158	0x2003	0x9E	DINT (32Bit)
Slot 1 P1	Pr15.058	0x200F	0x3A	DINT (32Bit)
Slot 1 P2	Pr26.058	0x201A	0x3A	DINT (32Bit)
Slot 2 P1	Pr16.058	0x2010	0x3A	DINT (32Bit)
Slot 2 P2	Pr27.058	0x201B	0x3A	DINT (32Bit)
Slot 3 P1	Pr17.058	0x2011	0x3A	DINT (32Bit)
Slot 3 P2	Pr28.058	0x201C	0x3A	DINT (32Bit)

2. Ensure that the encoder is working by moving the encoder and at the same time check that the position in the relevant encoder interface parameter, as detailed in the previous table is changing with the movement using Connect or the drives keypad.

3. In TwinCAT, double click on the EtherCAT slave drive to add the encoder PDO mapping to. When the properties for the slave drive open, select the “Process Data” tab, and then in the “PDO List” select object “0x1A02”. In the “PDO content” box scroll to the bottom of the list and then right click and then select “Add New Item…”



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	8	Outputs	
3	16	Inputs	
4	0	Outputs	
5	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	S
0x1A00	4.0	Outputs - Velocity status			0
0x1A01	3.0	Outputs - Status and Mode of Op			0
0x1A02	16.0	Outputs - Position Status		3	0
0x1A04	4.0	Outputs - Torque status			0
0x1A05	0.0	Outputs - Large Mapping Set			0
0x1600	4.0	Inputs - Velocity control			0
0x1601	3.0	Inputs - Control and Mode of Op			0

PDO Assignment (0x1C12):

☐ 0x1600
☐ 0x1601
☑ 0x1602
☐ 0x1604
☐ 0x1605

PDO Content (0x1A02):

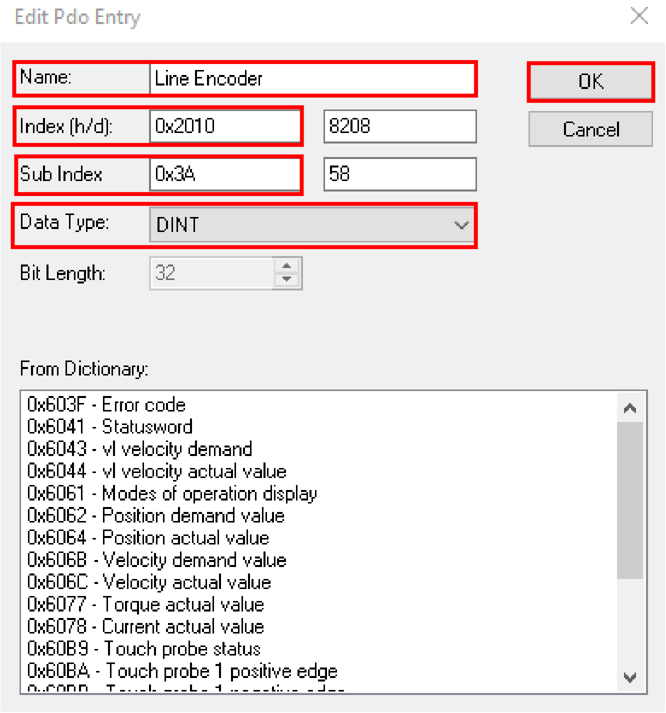
Index	Size	Offs	Name	Type
0x60BB:00	4.0	6.0	Touch probe 1 negative edge	DINT
0x6064:00	4.0	10.0	Position actual value	DINT
0x60B9:00	2.0	14.0	Touch probe status	UINT
		16.0		

Download

☑ PDO Assignment
☑ PDO Configuration

Name		Online	Type	Size
Status word	X		UINT	2.0
Touch probe 1 p...	X		DINT	4.0

4. When the “Edit Pdo Entry” dialog opens, enter the Name, Index, Subindex and Data Type as listed in the previous table. For this example, Slot 2 P1 is the location of the second encoder, and the mapping has been named “Line Encoder”. Click “OK” when finished.



Edit Pdo Entry

Name: Line Encoder

Index (h/d): 0x2010 8208

Sub Index 0x3A 58

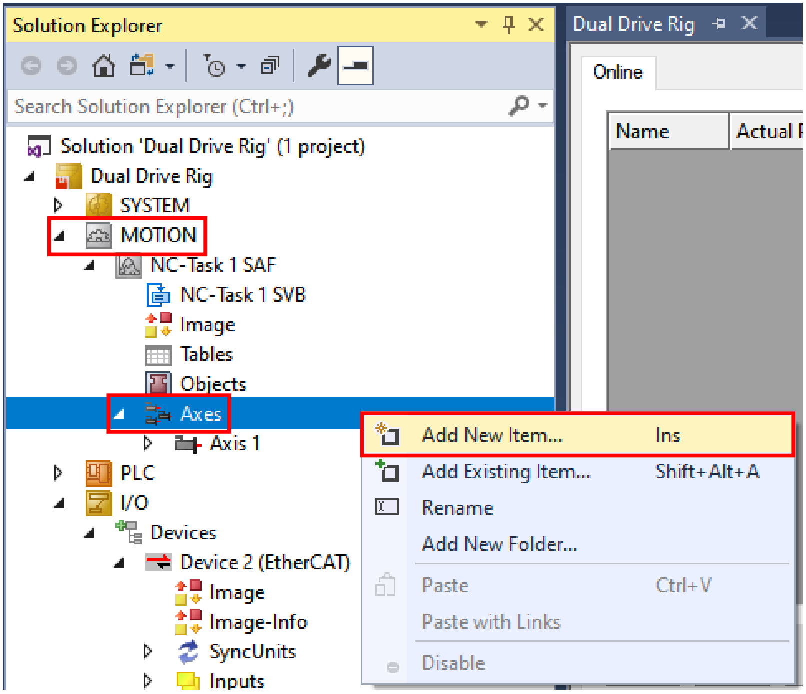
Data Type: DINT

Bit Length: 32

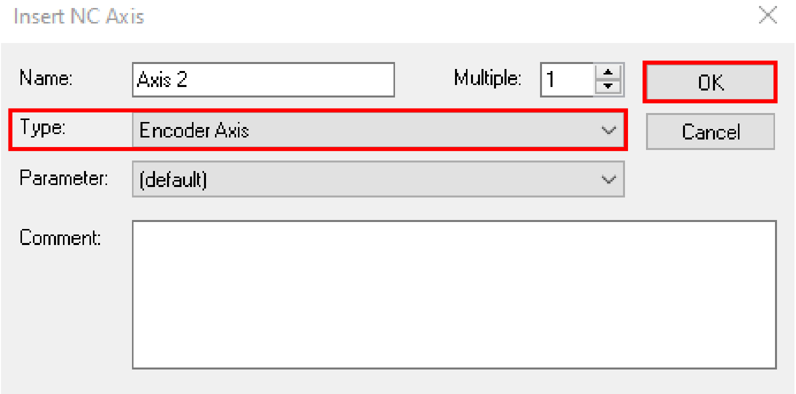
From Dictionary:

0x603F - Error code
0x6041 - Statusword
0x6043 - vl velocity demand
0x6044 - vl velocity actual value
0x6061 - Modes of operation display
0x6062 - Position demand value
0x6064 - Position actual value
0x606B - Velocity demand value
0x606C - Velocity actual value
0x6077 - Torque actual value
0x6078 - Current actual value
0x60B9 - Touch probe status
0x60BA - Touch probe 1 positive edge

5. In the solution explorer under “Motion”, right click on “Axes” and select “Add New Item…”.



6. When the “Insert NC Axis” dialog opens, change the type to “Encoder Axis”. Click “OK” when finished.



Name: Axis 2

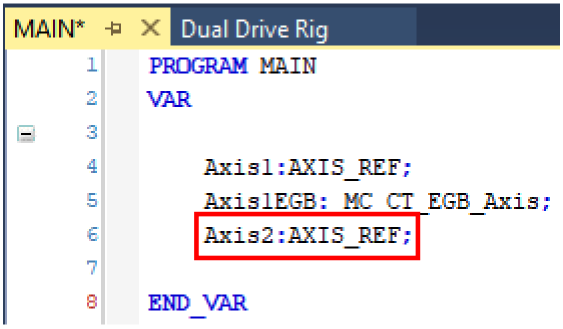
Multiple: 1

Type: Encoder Axis

Parameter: (default)

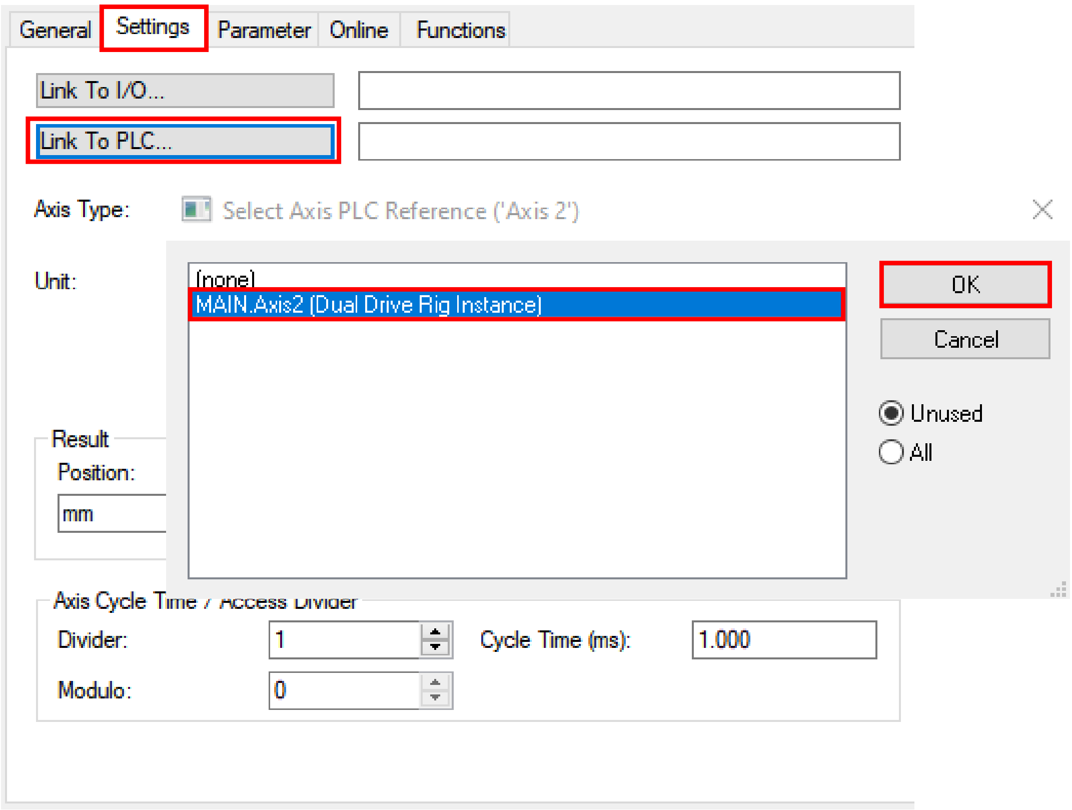
Comment:

7. In the motion program POU add an instance of AXIS_REF for the Encoder Axis.



```
PROGRAM MAIN
VAR

    Axis1:AXIS_REF;
    Axis1EGB: MC_CT_EGB_Axis;
    Axis2:AXIS_REF;

END_VAR
```

8. Double click on the encoder axis, (Axis 2), when the properties for the axis open, select the “Settings” tab. Click “Link To PLC”, and then select the new instance of AXIS_REF, e.g. “Axis2”.



(none)

MAIN.Axis2 (Dual Drive Rig Instance)

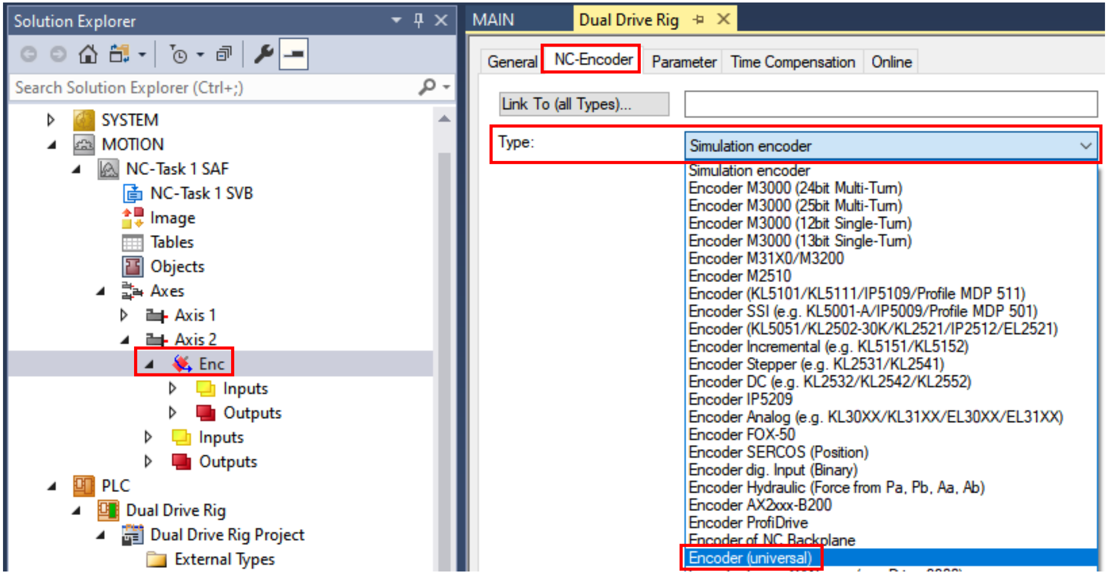
Axis Cycle Time / Access Divider

Divider: 1

Cycle Time (ms): 1.000

Modulo: 0

9. Expand the encoder axis and double click on “Enc”, when the properties for the encoder open, select the “NC-Encoder” tab. Change the “Type:” to “Encoder (universal)”

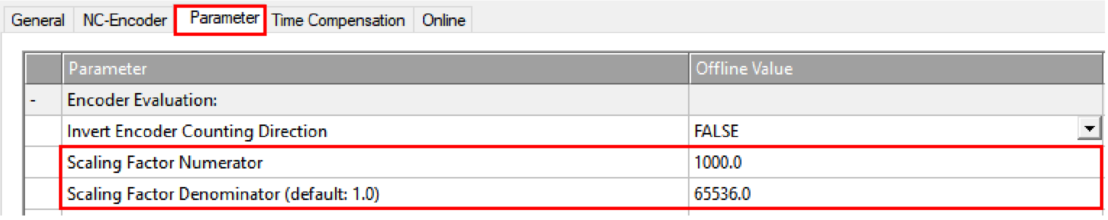


- SYSTEM
- MOTION
  - NC-Task 1 SAF
    - NC-Task 1 SVB
    - Image
    - Tables
    - Objects
    - Axes
      - Axis 1
      - Axis 2
        - Enc
          - Inputs
          - Outputs
        - Inputs
        - Outputs
- PLC
  - Dual Drive Rig
    - Dual Drive Rig Project
      - External Types

Type: Simulation encoder

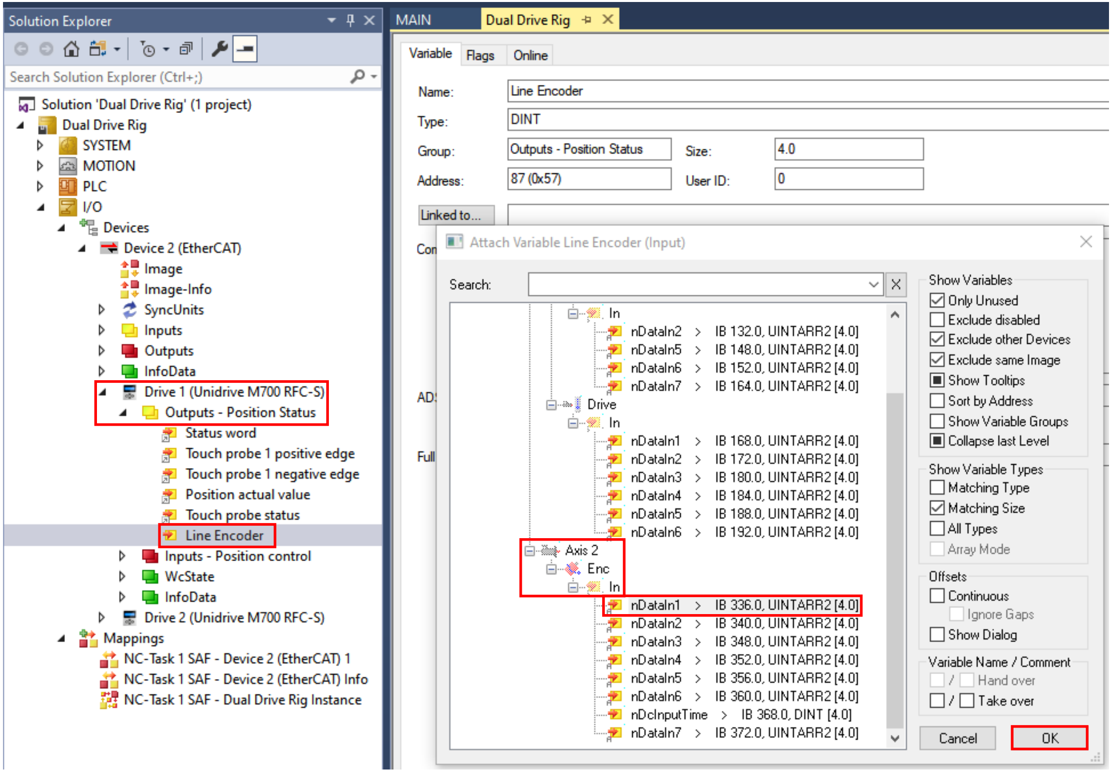
Simulation encoder
Encoder M3000 (24bit Multi-Turn)
Encoder M3000 (25bit Multi-Turn)
Encoder M3000 (12bit Single-Turn)
Encoder M3000 (13bit Single-Turn)
Encoder M31X0/M3200
Encoder M2510
Encoder (KL5101/KL5111/IP5109/Profile MDP 511)
Encoder SSI (e.g. KL5001-A/IP5009/Profile MDP 501)
Encoder (KL5051/KL2502-30K/KL2521/IP2512/EL2521)
Encoder Incremental (e.g. KL5151/KL5152)
Encoder Stepper (e.g. KL2531/KL2541)
Encoder DC (e.g. KL2532/KL2542/KL2552)
Encoder IP5209
Encoder Analog (e.g. KL30XX/KL31XX/EL30XX/EL31XX)
Encoder FOX-50
Encoder SERCOS (Position)
Encoder dig. Input (Binary)
Encoder Hydraulic (Force from Pa, Pb, Aa, Ab)
Encoder AX2xxx-B200
Encoder ProfiDrive
Encoder of NC Backplane
Encoder (universal)

10. Select the “Parameter” tab and then set the “Scaling Factor Numerator” to the number of units per revolution of the encoder, e.g.1000. Set the “Scaling Factor Denominator” to the number of encoder counts per revolution – by default this is 65536.



	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	1000.0
	Scaling Factor Denominator (default: 1.0)	65536.0

11. Expand the tree for the drive that the encoder is connected to, then open “Outputs”, and double click on the name given to the encoder PDO link created in step 4 e.g. “Line Encoder”.



Name: Line Encoder
Type: DINT
Group: Outputs - Position Status Size: 4.0
Address: 87 (0x57) User ID: 0

```
In
  nDataIn2  >  IB 132.0, UINTARR2 [4.0]
  nDataIn5  >  IB 148.0, UINTARR2 [4.0]
  nDataIn6  >  IB 152.0, UINTARR2 [4.0]
  nDataIn7  >  IB 164.0, UINTARR2 [4.0]
Drive
 In
  nDataIn1  >  IB 168.0, UINTARR2 [4.0]
  nDataIn2  >  IB 172.0, UINTARR2 [4.0]
  nDataIn3  >  IB 180.0, UINTARR2 [4.0]
  nDataIn4  >  IB 184.0, UINTARR2 [4.0]
  nDataIn5  >  IB 188.0, UINTARR2 [4.0]
  nDataIn6  >  IB 192.0, UINTARR2 [4.0]
Axis 2
 Enc
  In
  nDataIn1  >  IB 336.0, UINTARR2 [4.0]
  nDataIn2  >  IB 340.0, UINTARR2 [4.0]
  nDataIn3  >  IB 348.0, UINTARR2 [4.0]
  nDataIn4  >  IB 352.0, UINTARR2 [4.0]
  nDataIn5  >  IB 356.0, UINTARR2 [4.0]
  nDataIn6  >  IB 360.0, UINTARR2 [4.0]
  nDcInputTime  >  IB 368.0, DINT [4.0]
  nDataIn7  >  IB 372.0, UINTARR2 [4.0]
```

Show Variables
- ☑ Only Unused
- ☐ Exclude disabled
- ☑ Exclude other Devices
- ☑ Exclude same Image
- ■ Show Tooltips
- ☐ Sort by Address
- ☐ Show Variable Groups
- ■ Collapse last Level

Show Variable Types
- ☐ Matching Type
- ☑ Matching Size
- ☐ All Types
- ☐ Array Mode

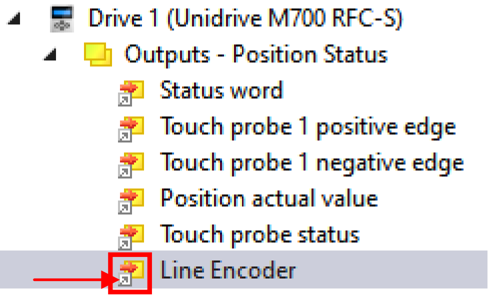
Offsets
- ☐ Continuous
- ☐ Ignore Gaps
- ☐ Show Dialog

Variable Name / Comment
- ☐ / ☐ Hand over
- ☐ / ☐ Take over

When the “Attach Variable Line Encoder (Input)” dialog opens scroll down to the axis associated with the encoder, e.g. Axis 2 and select “Axis 2” > “Enc” > “In” > “nDataIn1” and then click “OK”.

> **HINT:** “nDataIn1” is always where the encoder data is mapped to, regardless of the axis number.

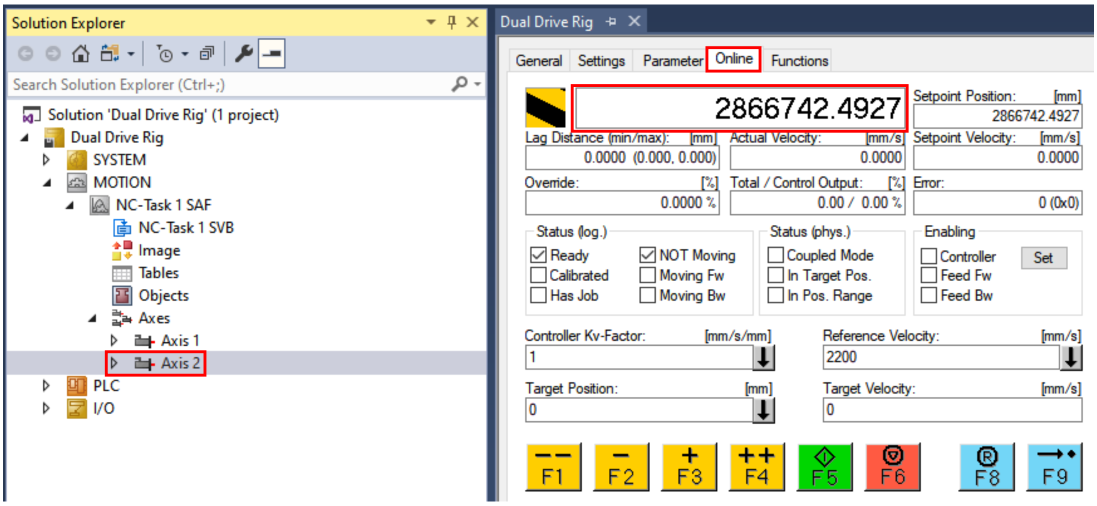
The PDO for the encoder is now linked to the encoder axis as indicated by the small grey box with an arrow in it:



- Drive 1 (Unidrive M700 RFC-S)
  - Outputs - Position Status
    - Status word
    - Touch probe 1 positive edge
    - Touch probe 1 negative edge
    - Position actual value
    - Touch probe status
    - Line Encoder

12. Apply the new settings to the project by clicking the activate configuration button .

13. Verify the encoder is working by double clicking on the encoder axis e.g. Axis 2, and when the properties for the axis open, select the online tab to view the position. This will change as the encoder is moved.



2866742.4927

Setpoint Position: [mm] 2866742.4927

Lag Distance (min/max): [mm] 0.0000 (0.000, 0.000)

Actual Velocity: [mm/s] 0.0000

Setpoint Velocity: [mm/s] 0.0000

Override: [%] 0.0000 %

Total / Control Output: [%] 0.00 / 0.00 %

Error: 0 (0x0)

Status (log.): ☑ Ready ☐ Calibrated ☐ Has Job ☑ NOT Moving ☐ Moving Fw ☐ Moving Bw

Status (phys.): ☐ Coupled Mode ☐ In Target Pos. ☐ In Pos. Range

Enabling: ☐ Controller ☐ Feed Fw ☐ Feed Bw

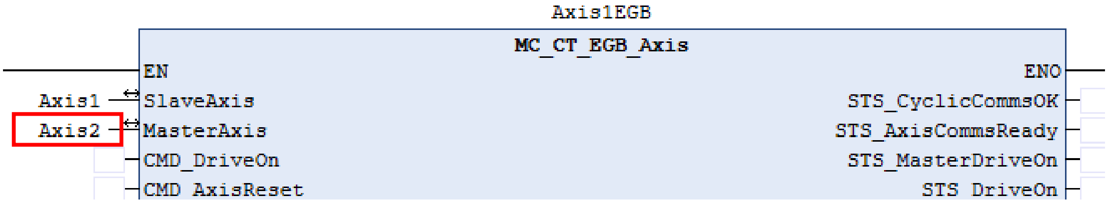
Controller Kv-Factor: [mm/s/mm] 1

Reference Velocity: [mm/s] 2200

Target Position: [mm] 0

Target Velocity: [mm/s] 0

14. The encoder reference can now be used as the master for an EGB axis. Connect the new encoder axis to an EGB axis master input, compile the code and download it to the target IPC/PLC by clicking and then :



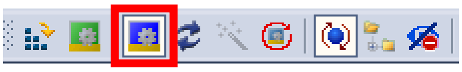
Axis1 — SlaveAxis

Axis2 — MasterAxis

3.4 How to use a virtual master axis to control two slave drives using the EGB interface

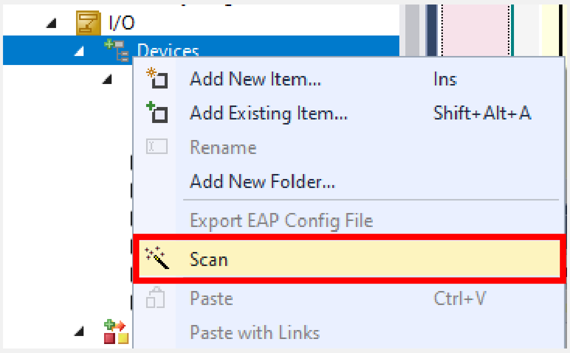
This section describes how to setup and use a system comprising of two physical slave drives following a single virtual master drive, coupled together using the “MC_CT_EGB_Axis” interface.

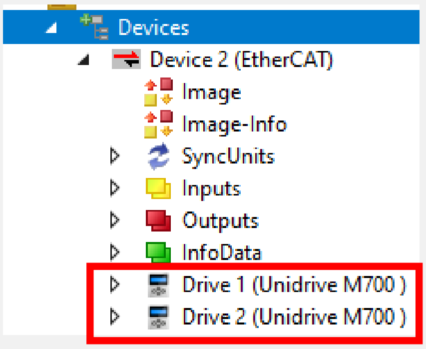
1. Ensure that the controller has been put into “Config Mode” by clicking the “Config Mode” icon.

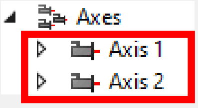


HINT: **It is assumed that:**

- A second Control Techniques drive has been connected to the EtherCAT network.
- A device “Scan” has been performed, that a second drive has been found and added to the project along with a second NC Axis configuration.
- The second NC axis has been configured using the relevant sections from section **2.13 NC Axis Configuration** so that there are now two axes in the project.

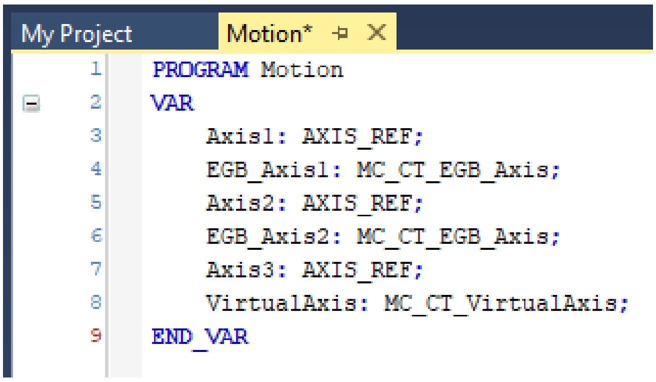






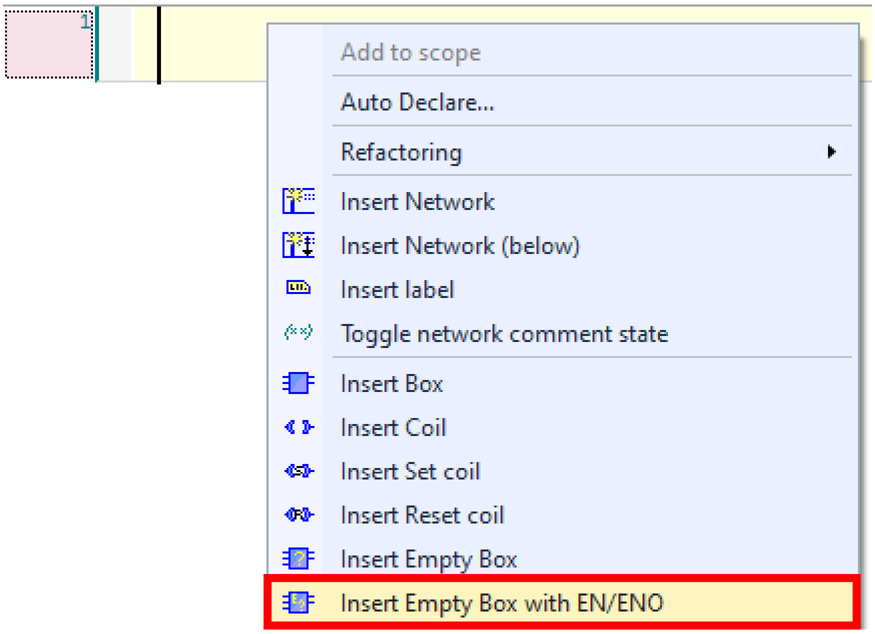
2. From the project tree navigate to “{Project Name}” > “PLC” > “{Program Name}” > “{Program Project Name}” > “POUs” > “Motion” and modify the declaration area so that it includes:
   - Two instances of MC_CT_EGB_Axis.
   - Three instances of “AXIS_REF”.
   - One instance of MC_CT_VirtualAxis.

   See the example below:

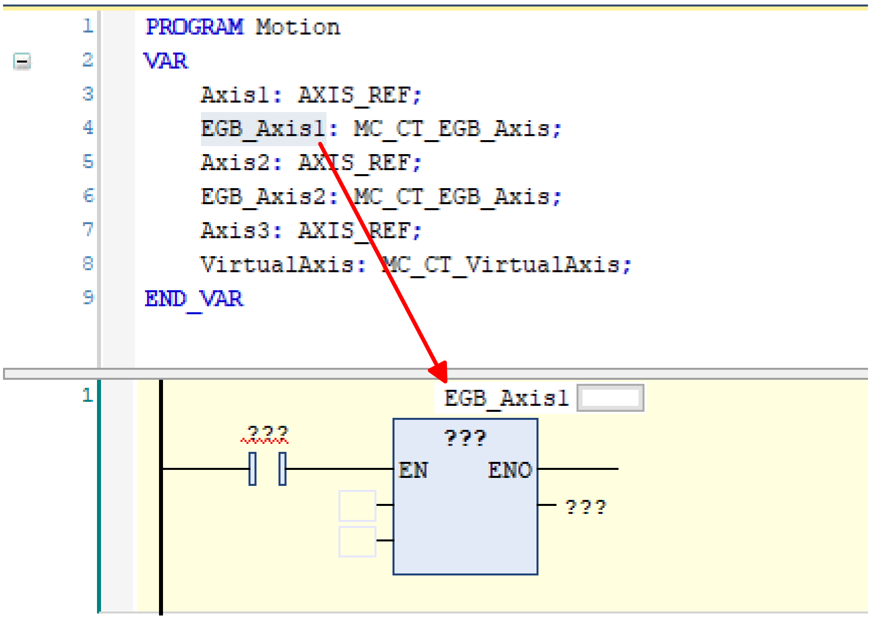


```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    Axis2: AXIS_REF;
    EGB_Axis2: MC_CT_EGB_Axis;
    Axis3: AXIS_REF;
    VirtualAxis: MC_CT_VirtualAxis;
END_VAR
```

3. In the ladder editor, right click on an empty ladder network and select “Inset Empty Box with EN/ENO.

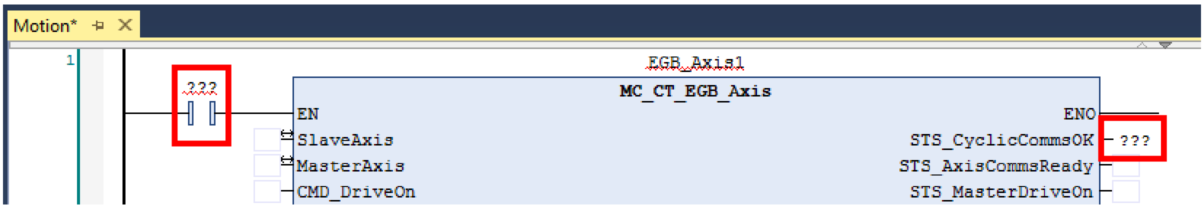


4. Copy the first instance name of MC_CT_EGB_Axis into the instance name of the new box and press enter.

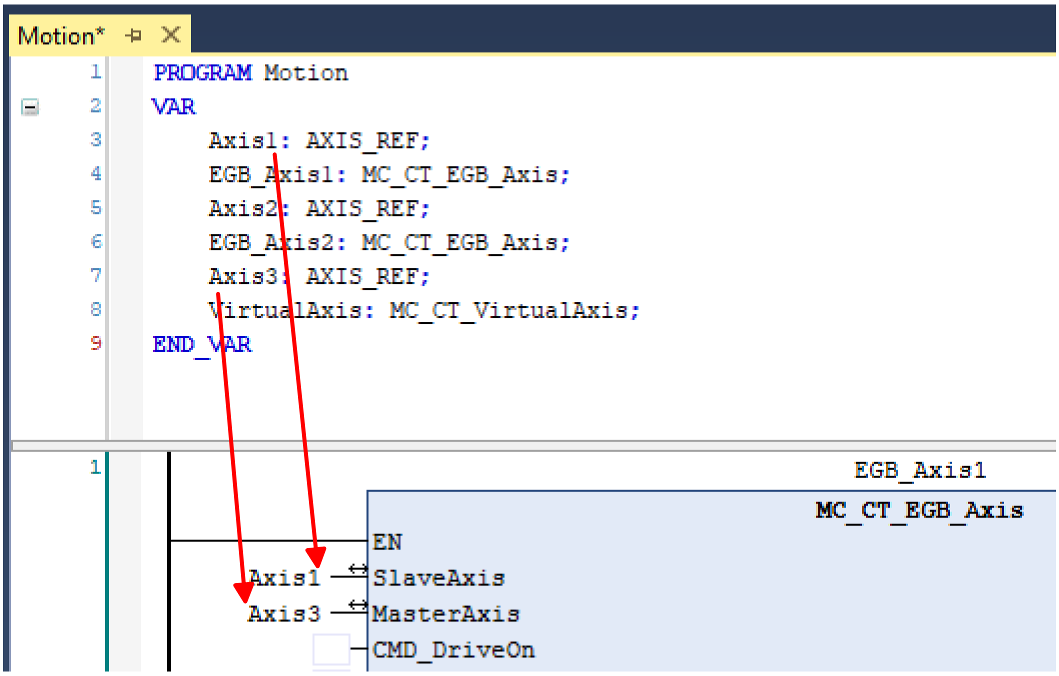


```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    Axis2: AXIS_REF;
    EGB_Axis2: MC_CT_EGB_Axis;
    Axis3: AXIS_REF;
    VirtualAxis: MC_CT_VirtualAxis;
END_VAR
```

5. An instance of MC_CT_EGB_Axis will be spawned. Delete the automatically generated contact at the “EN” input, along with the red “???” marks

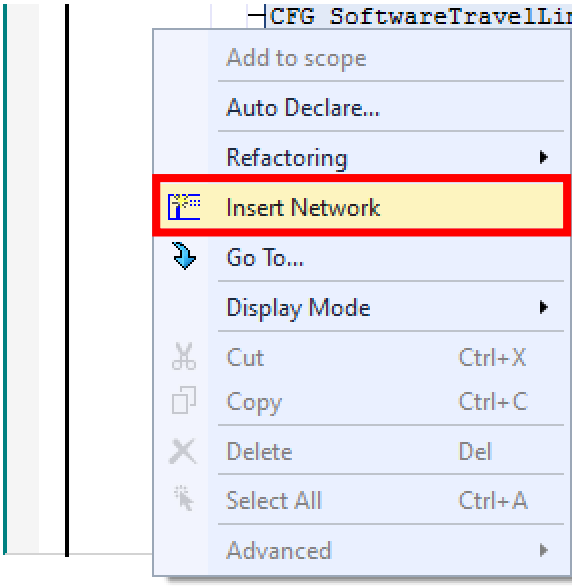


6. Connect the “Axis1” instance of “AXIS_REF” to “SlaveAxis” and the “Axis3” instance of “AXIS_REF” to “MasterAxis”.

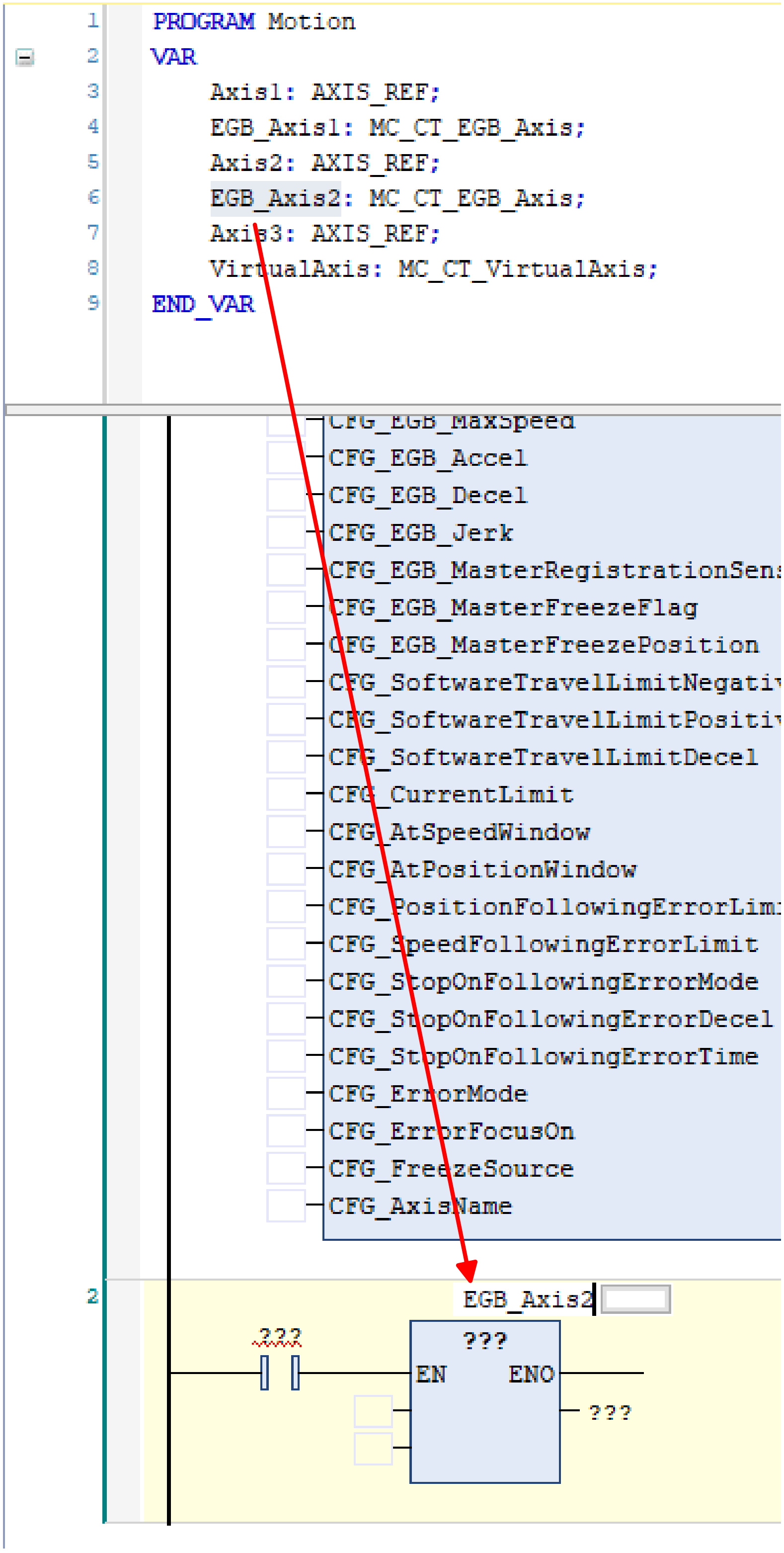


```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    Axis2: AXIS_REF;
    EGB_Axis2: MC_CT_EGB_Axis;
    Axis3: AXIS_REF;
    VirtualAxis: MC_CT_VirtualAxis;
END_VAR
```

7. In the ladder editor, create an empty network below the existing position interface by right clicking in the white space and selecting “Insert Network”.

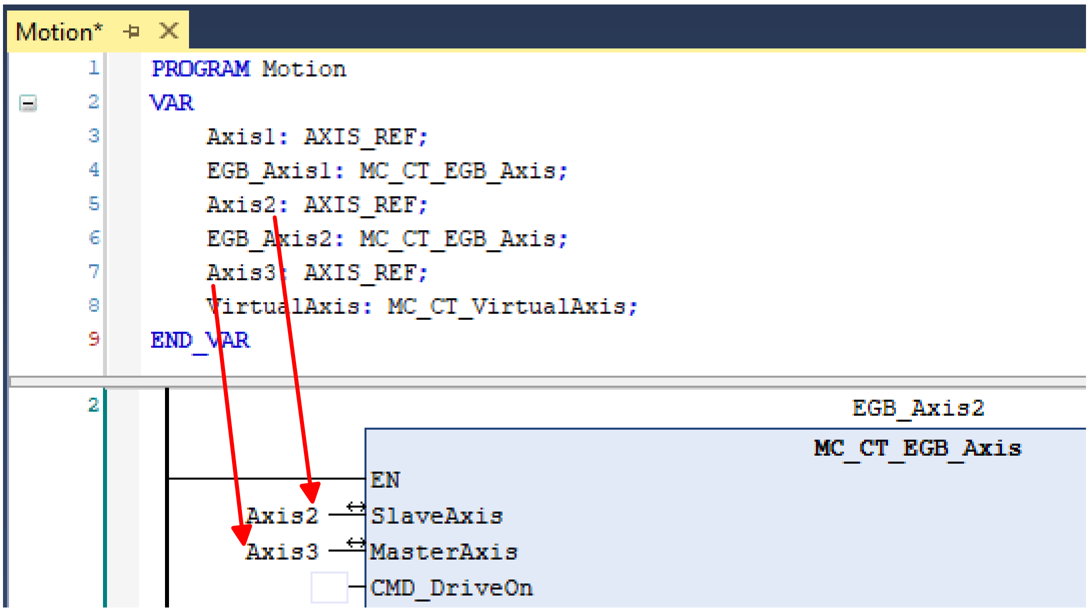


8. Repeat step 3, 4 and 5, this time copying the second instance name of MC_CT_EGB_Axis into the instance name of the new box and press enter.



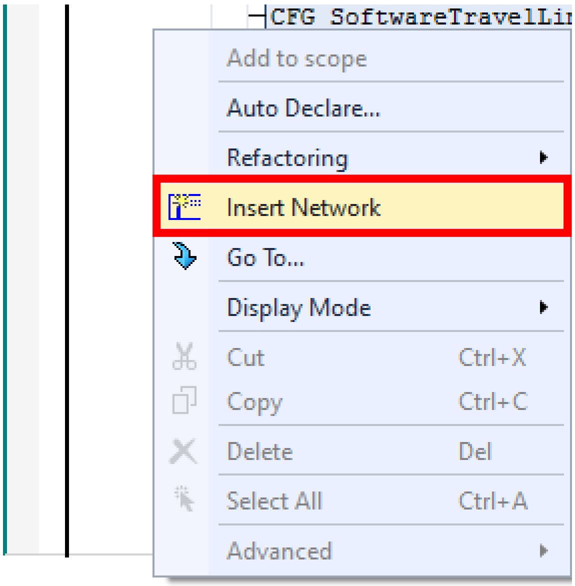
```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    Axis2: AXIS_REF;
    EGB_Axis2: MC_CT_EGB_Axis;
    Axis3: AXIS_REF;
    VirtualAxis: MC_CT_VirtualAxis;
END_VAR
```

9. Connect the “Axis2” instance of “AXIS_REF” to “SlaveAxis” and the “Axis3” instance of “AXIS_REF” to “MasterAxis”.

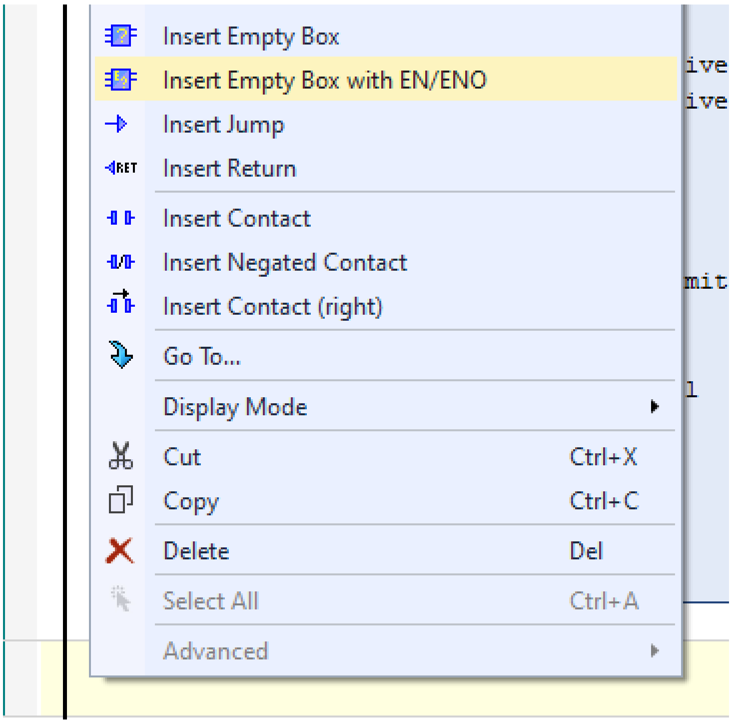


```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    Axis2: AXIS_REF;
    EGB_Axis2: MC_CT_EGB_Axis;
    Axis3: AXIS_REF;
    VirtualAxis: MC_CT_VirtualAxis;
END_VAR
```

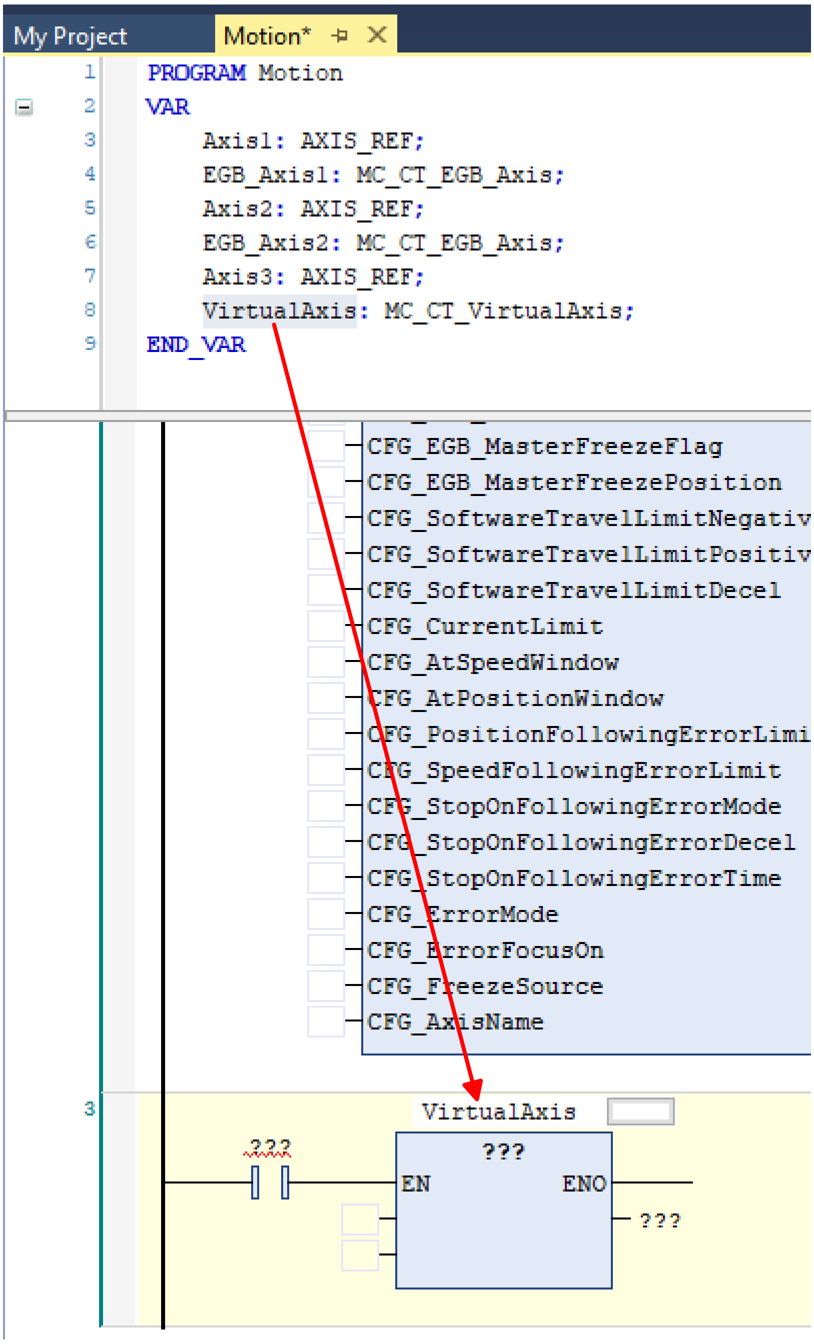
10. In the ladder editor, create an empty network below the existing position interface by right clicking in the white space and selecting “Insert Network”.



11. In the ladder editor, right click on the empty ladder network and select “Inset Empty Box with EN/ENO.

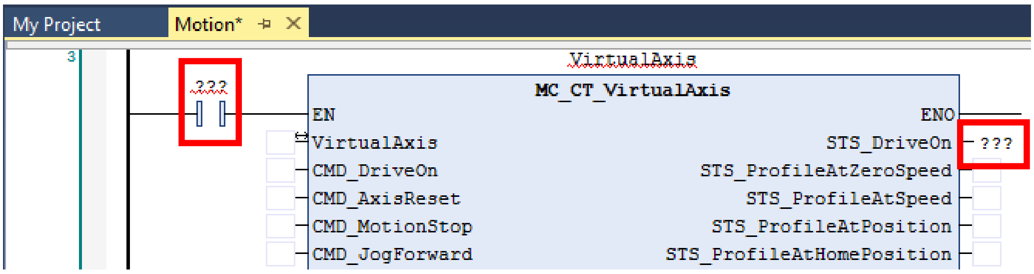


12. Copy the instance name of MC_CT_VirtualAxis into the instance name of the new box and press enter.

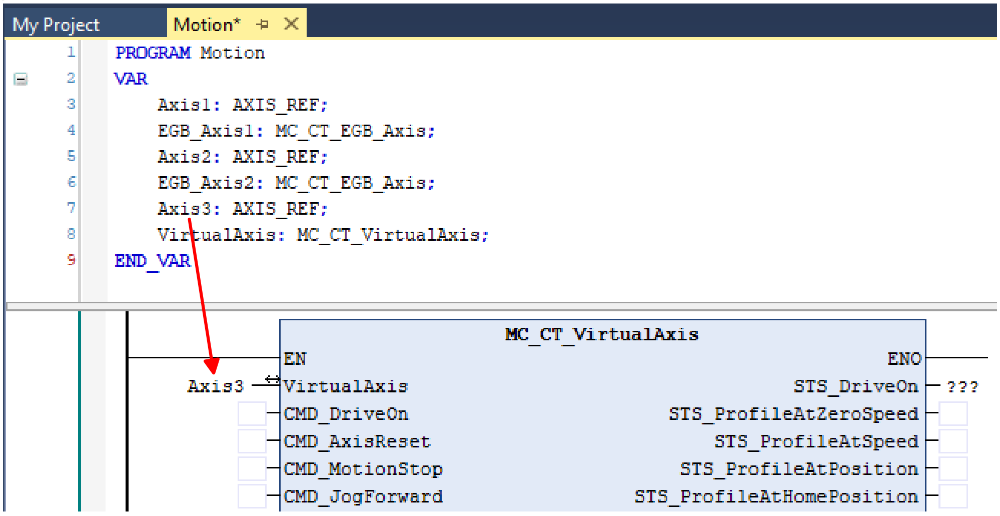


```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    Axis2: AXIS_REF;
    EGB_Axis2: MC_CT_EGB_Axis;
    Axis3: AXIS_REF;
    VirtualAxis: MC_CT_VirtualAxis;
END_VAR
```

13. An instance of MC_CT_VirtualAxis will be spawned. Delete the automatically generated contact at the "EN" input, along with the red "???" marks.

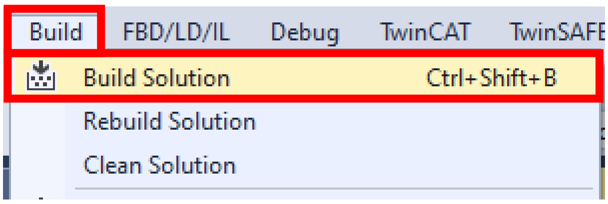


14. Copy the instance name of AXIS_REF into the Assignment box of the "Axis" In/Out parameter.

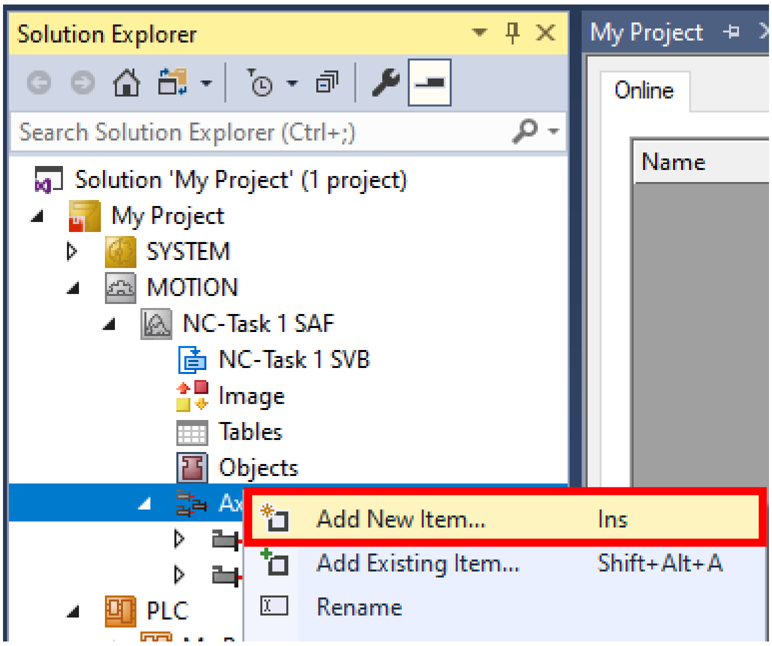


```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    Axis2: AXIS_REF;
    EGB_Axis2: MC_CT_EGB_Axis;
    Axis3: AXIS_REF;
    VirtualAxis: MC_CT_VirtualAxis;
END_VAR
```

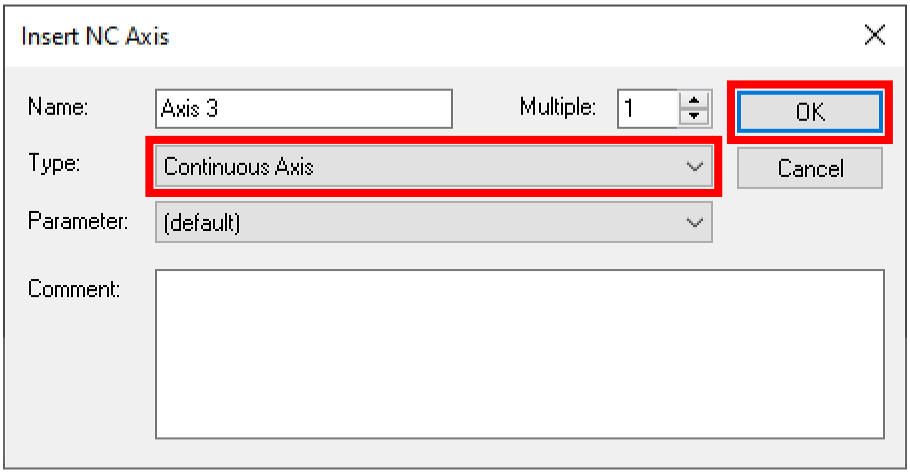
15. Perform a build by selecting "Build" > "Build Solution" from the main toolbar.



16. From the project tree, navigate to “{Project Name}” > “MOTION” > “NC-Task 1 SAF” > “Axes”, right click on “Axis 1” and select “Add New Item…”.



17. The “Insert NC Axis” dialog will appear. Ensure the “Type” is set to “Continuous Axis” and click “OK”.


Insert NC Axis

Name: Axis 3

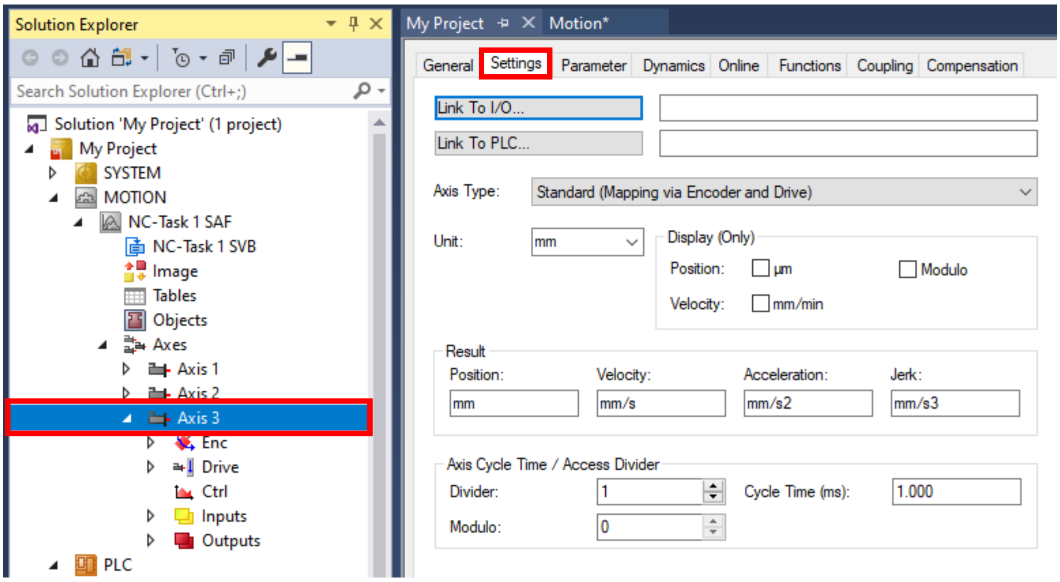
Multiple: 1

Type: Continuous Axis

Parameter: (default)

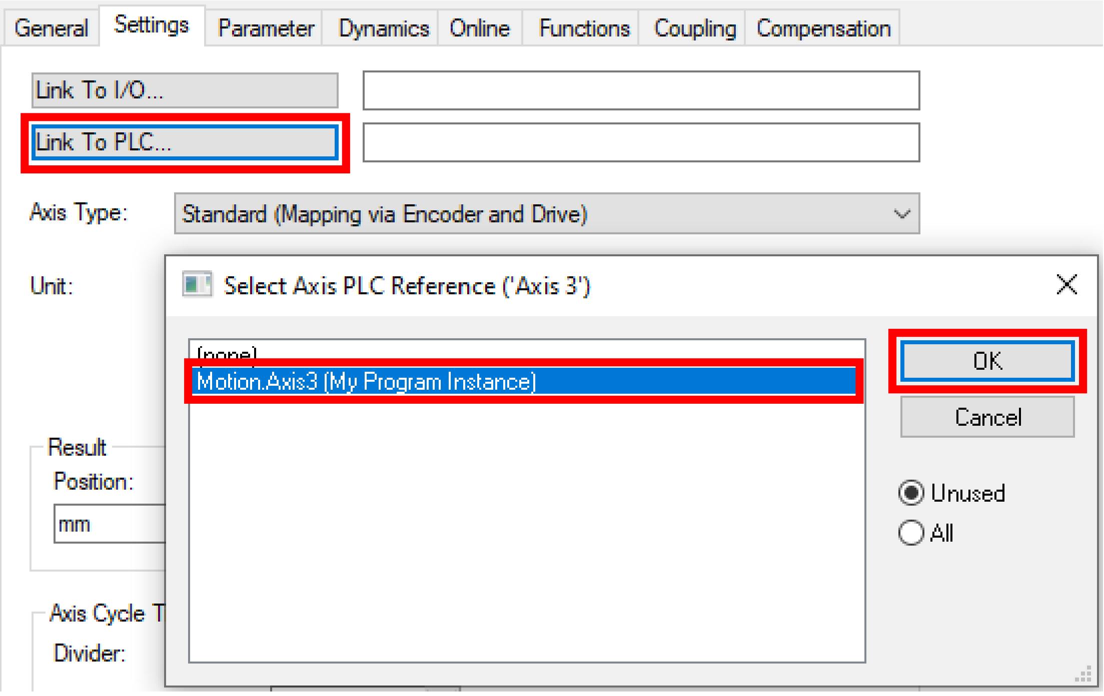
Comment:

18. Axis 3 will now appear in the project tree. Double click on “Axis 3” and navigate to the “Settings” tab.



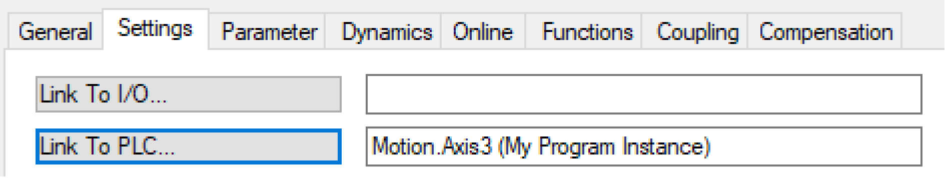
Axis Type: Standard (Mapping via Encoder and Drive)
Unit: mm
Position: ☐ µm ☐ Modulo
Velocity: ☐ mm/min
Position: mm
Velocity: mm/s
Acceleration: mm/s2
Jerk: mm/s3
Divider: 1
Cycle Time (ms): 1.000
Modulo: 0

19. Select the “Link To PLC…” button to link the axis to the new instance of AXIS_REF (declared in the PLC project). The “Select Axis PLC Reference” window will appear. Select the AXIS_REF instance to be linked to the axis and click “OK”.



Axis Type: Standard (Mapping via Encoder and Drive)
(none)
Motion.Axis3 (My Program Instance)
● Unused
○ All

20. This completes the axis linking steps. The settings tab of the axis object will now appear in the following way.



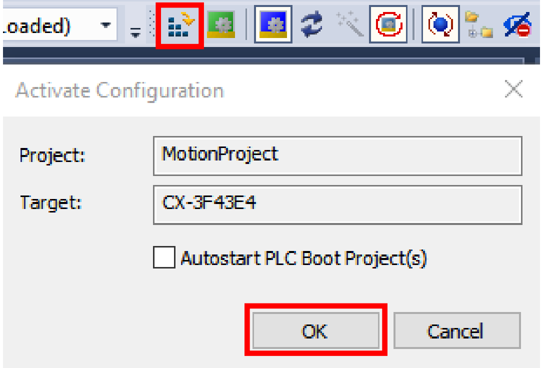
Link To I/O...

Link To PLC... Motion.Axis3 (My Program Instance)

21. The virtual axis (Axis3) may be configured using the following sections:

- **2.13.3 Axis configuration**.
- **2.15 Verify the axis**.

22. Click the “Activate Configuration” button and then click “OK” if the “Activate Configuration” dialog appears:



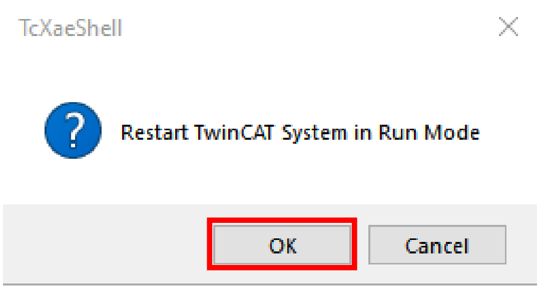
Activate Configuration

Project: MotionProject

Target: CX-3F43E4

☐ Autostart PLC Boot Project(s)

23. Click “OK” on the “Restart TwinCAT in Run Mode” dialog:



TcXaeShell

Restart TwinCAT System in Run Mode

24. The following sequence is a very simplified use case that will result in both EGB slave axes following the virtual master axis.

    a. Set the “CMD_DriveOn” input of the “VirtualAxis”, “EGB_Axis1” and “EGB_Axis2” function blocks to “TRUE”.

    b. Set appropriate values at the “CFG_EGB_MaxSpeed”, “CFG_EGB_Accel” And “CFG_EGB_Decel” inputs of the “EGB_Axis1” and “EGB_Axis2” function blocks.

    c. Set the “CMD_EGB” inputs of the “EGB_Axis1” and “EGB_Axis2” function blocks to “TRUE”.

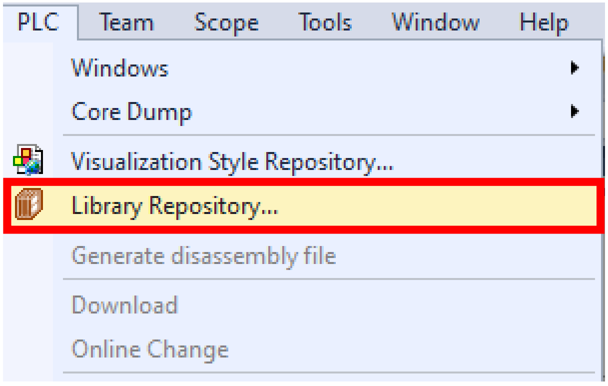
    d. Set appropriate values at the “CFG_JogSpeed”, “CFG_JogAccel” and “CFG_JogDecel” inputs of the “VirtualAxis” function block.

    e. Jog the Virtual axis associated with the “VirtualAxis” function block using the “CMD_JogForward” and “CMD_JogReverse” inputs – both slave axes should follow the virtual master axis.

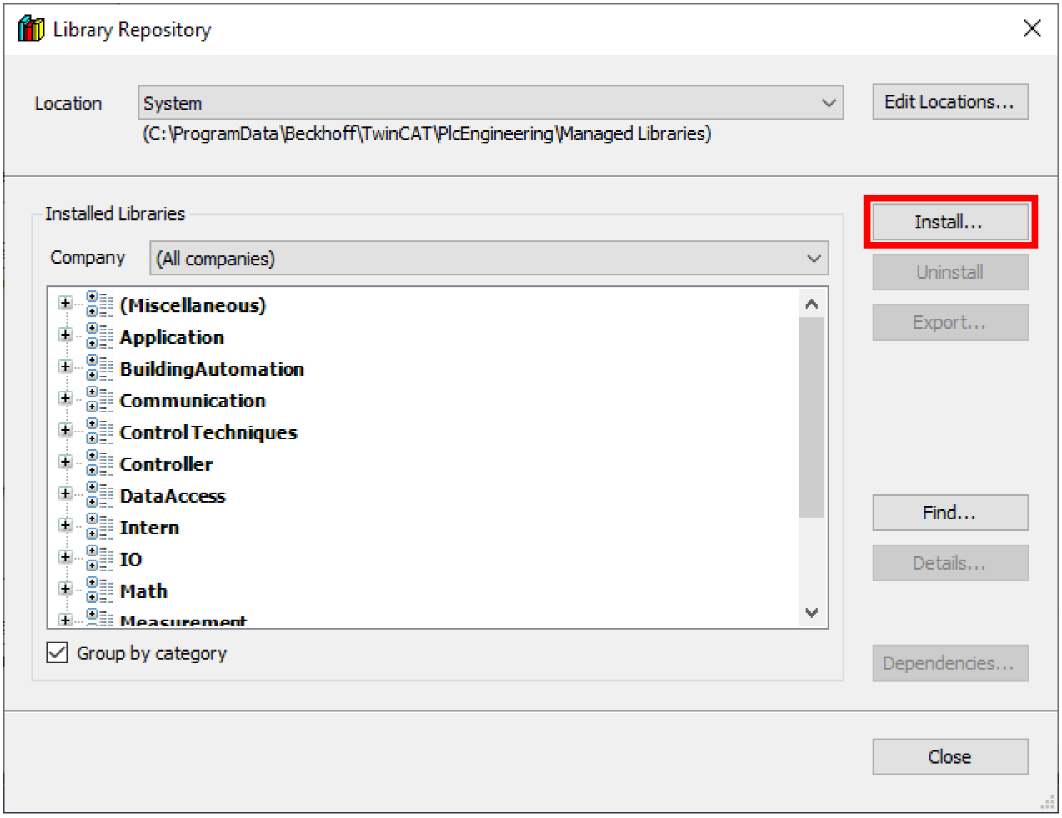
3.5 How to update the CT NC Axis Interface library

The following steps describe how to add a new version of the CT NC Axis Interface Library into TwinCAT:

1. From the TwinCAT tool bar select “PLC” > “Library Repository…”.



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

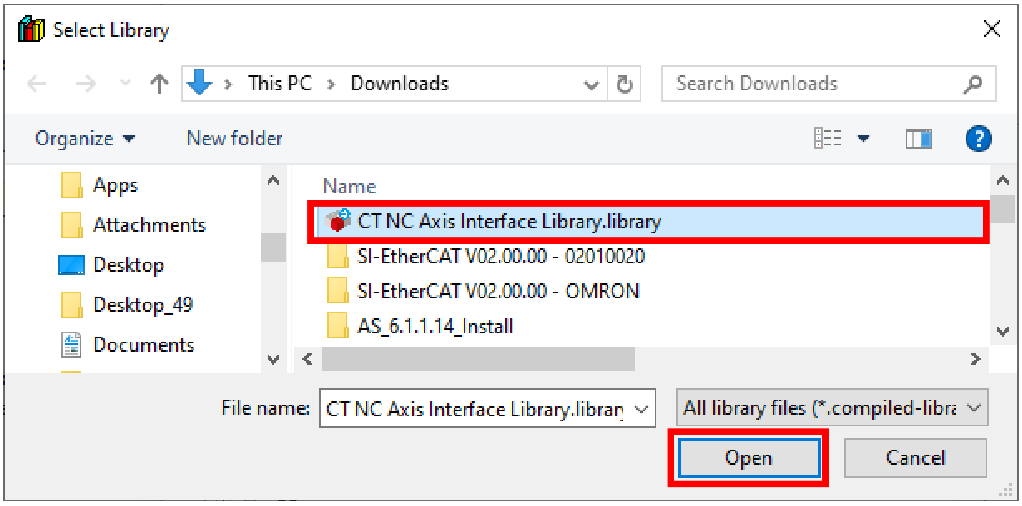


Location: System
(C:\ProgramData\Beckhoff\TwinCAT\PlcEngineering\Managed Libraries)

Company: (All companies)

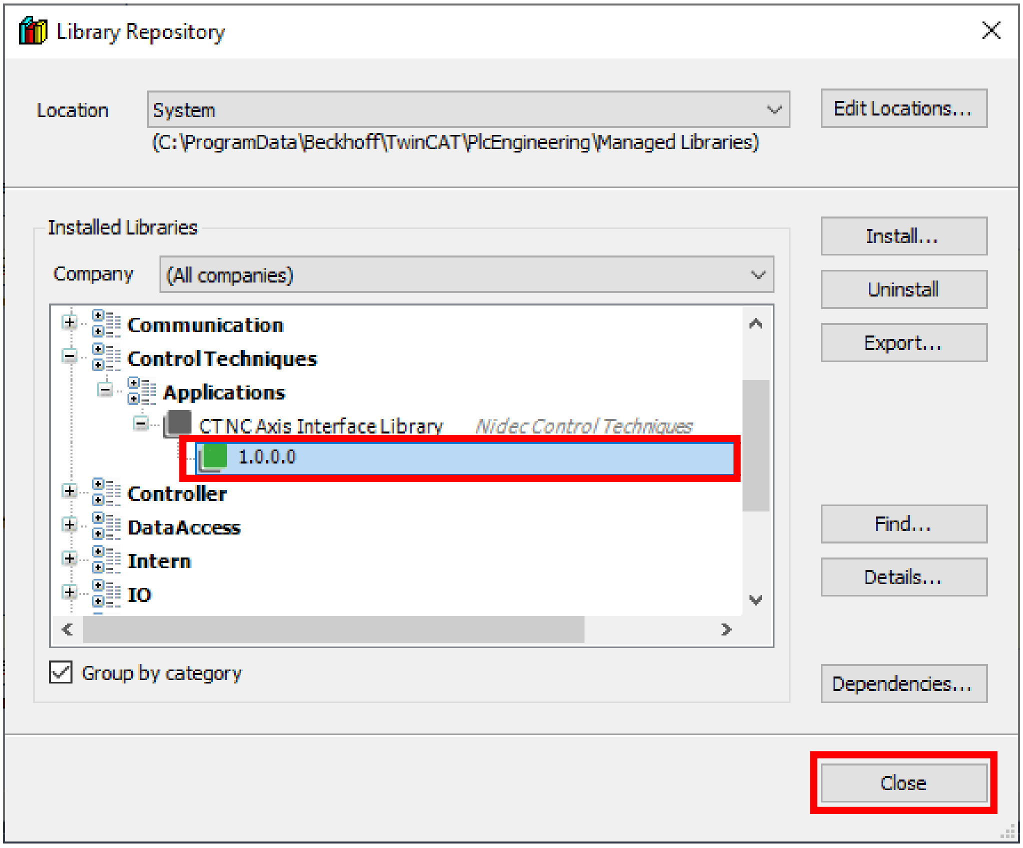
☑ Group by category

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



File name: CT NC Axis Interface Library.library

4. The library is now added to the “Library Repository” and may be added to a TwinCAT project. Click close when finished adding libraries.

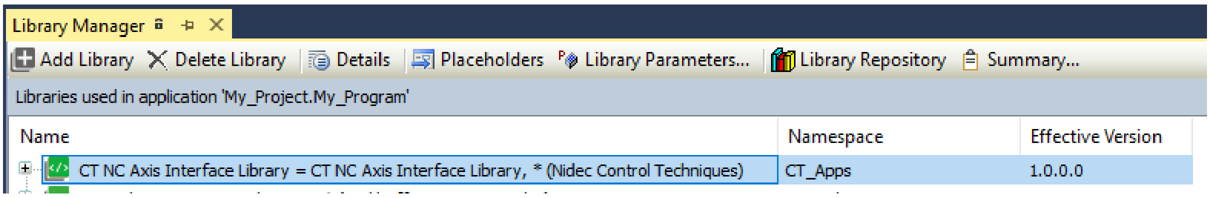


Location: System
(C:\ProgramData\Beckhoff\TwinCAT\PlcEngineering\Managed Libraries)

Company: (All companies)

☑ Group by category

5. Provided that the CT NC Axis Interface Library is already present in the Library Manager, the newest version of the library is now used by the project.



Libraries used in application 'My_Project.My_Program'

Name	Namespace	Effective Version
CT NC Axis Interface Library = CT NC Axis Interface Library, * (Nidec Control Techniques)	CT_Apps	1.0.0.0

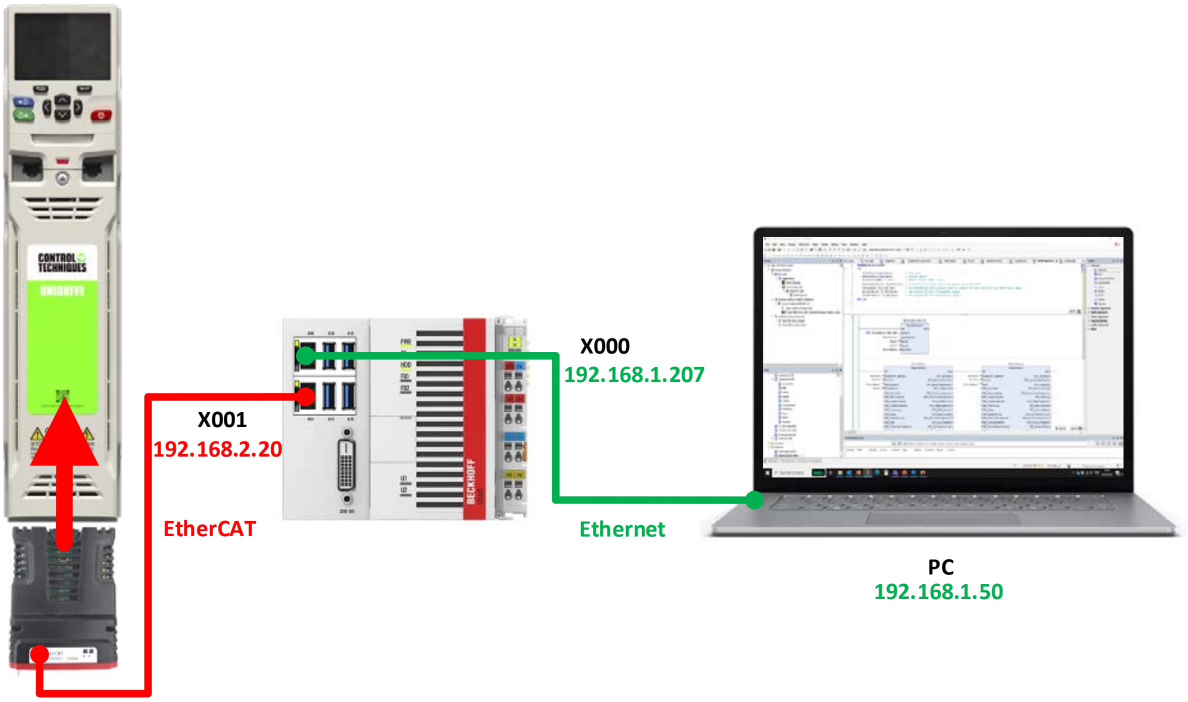
3.6 How to set up an EtherCAT EoE connection from the PLC to the drive

To allow Connect or any other Ethernet based PC tool such as CT Scope to connect to the slave drives in the EtherCAT network, an EoE (Ethernet over EtherCAT) connection may be configured.

Before continuing, an Ethernet cable should be connected between the Ethernet port of the programming PC and the Ethernet port of the PLC / IPC / Controller (X000 in the diagram below).

A second Ethernet cable should be connected between the EtherCAT port of the PLC / IPC / Controller (X001 in the diagram below) and the EtherCAT post of the Control Techniques Drive / SI-EtherCAT module in use.

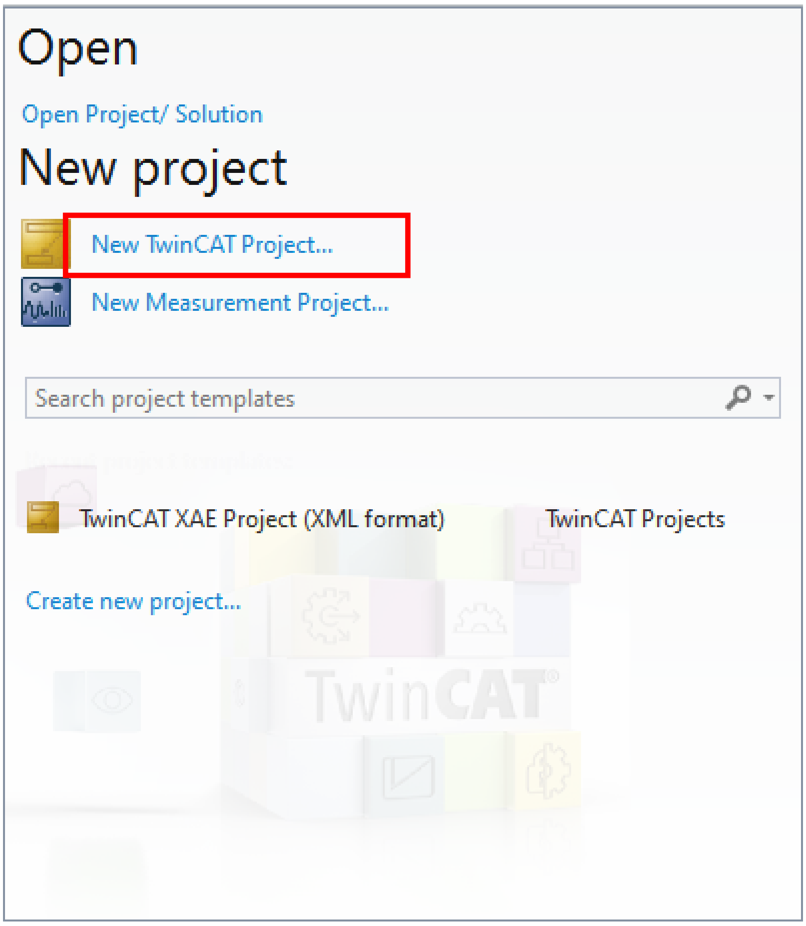
It is assumed that the Ethernet addresses of the programming PC and PLC / IPC / Controller are in the same range e.g.192.168.1.x, and that the range of addresses for the EtherCAT port are in the range 192.168.2.x.



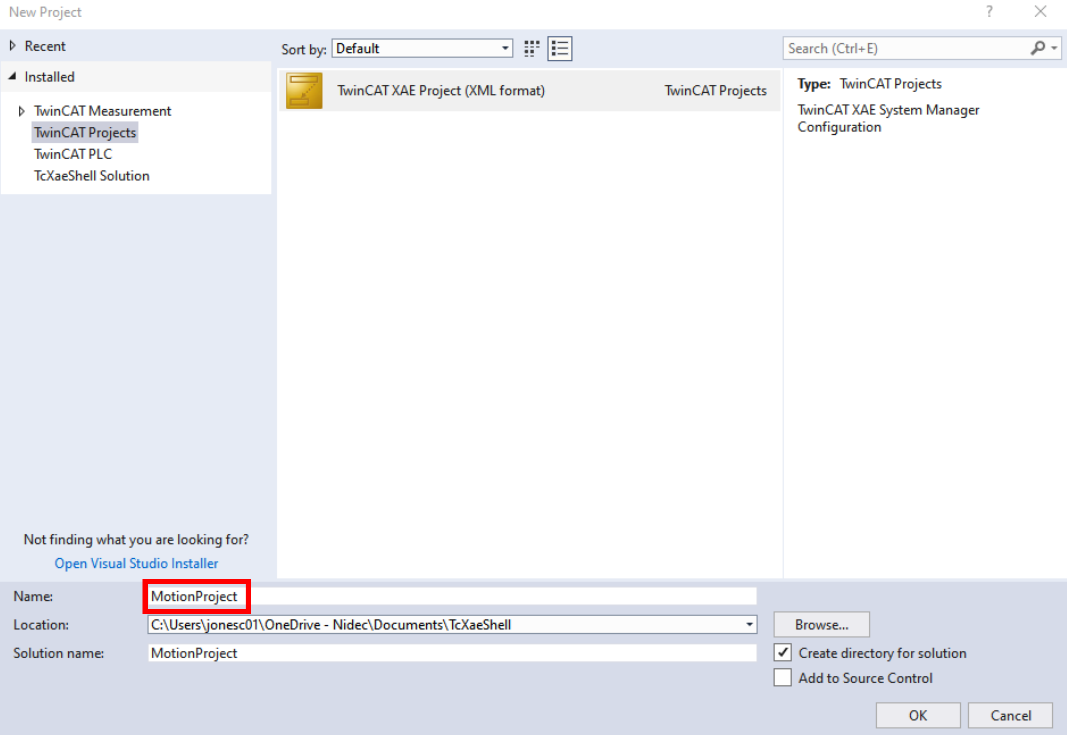
Use the following steps to configure an EoE network:

1. Start TwinCAT. For the purposes of these examples the TwinCAT XAE Shell is used to run the TwinCAT IDE. This is available even if the PC doesn't have Visual Studio installed.

2. Select “New TwinCAT Project”:



3. Give the project a sensible name e.g. “MotionProject” and then click “OK”.



Sort by: Default

Type: TwinCAT Projects

TwinCAT XAE System Manager Configuration

Name: MotionProject

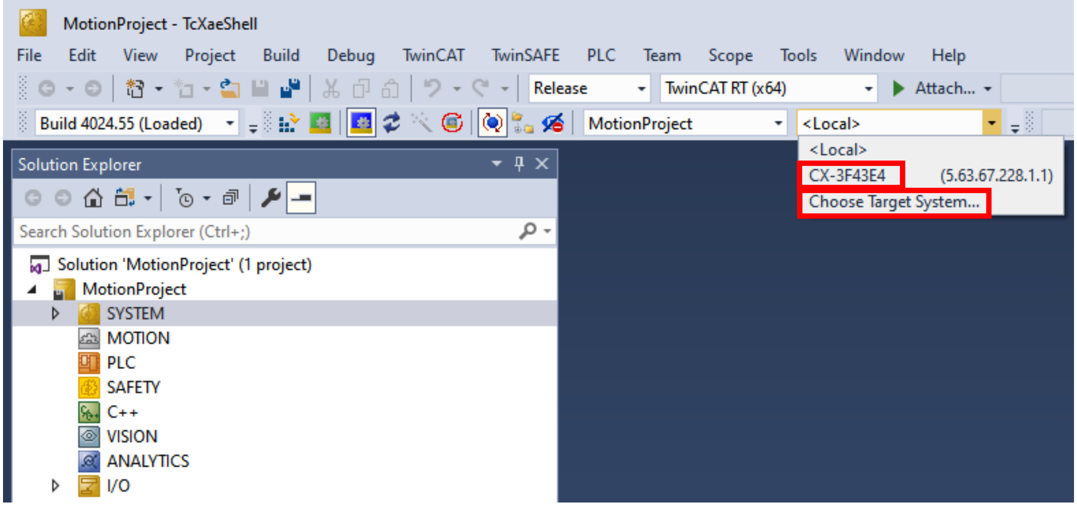
Location: C:\Users\jonesc01\OneDrive - Nidec\Documents\TcXaeShell

Solution name: MotionProject

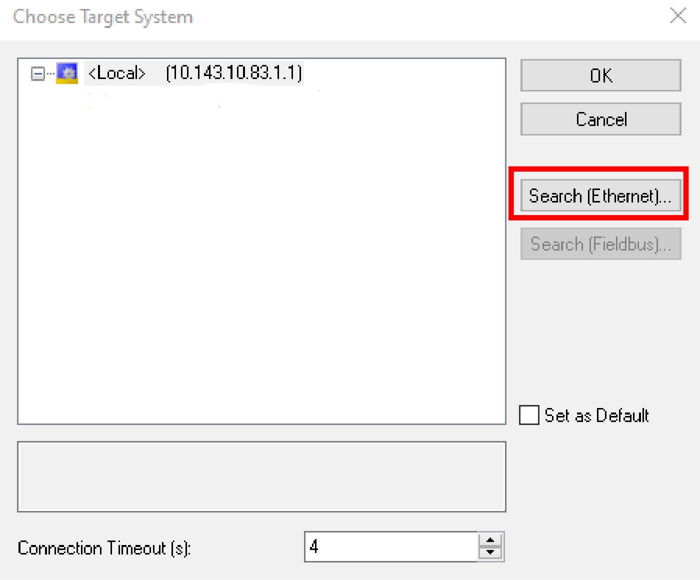
☑ Create directory for solution

☐ Add to Source Control

4. Select the target PLC. If your PLC has previously been connected to a TwinCAT project, then the name and ADS Net ID of the PLC is shown.

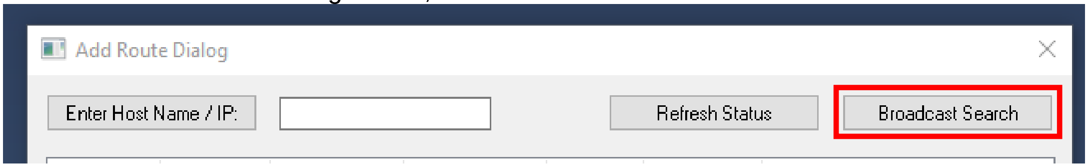


If the PLC hasn't previously been connected to a TwinCAT project, then select "Choose Target System…" and then "Search (Ethernet)" from the "Choose Target System" dialog.

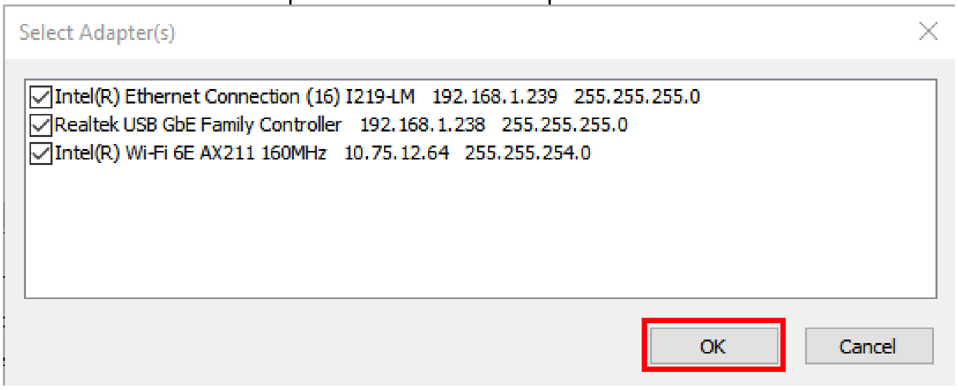


Connection Timeout (s): 4

5. When the “Add Route Dialog” starts, click “Broadcast search”.

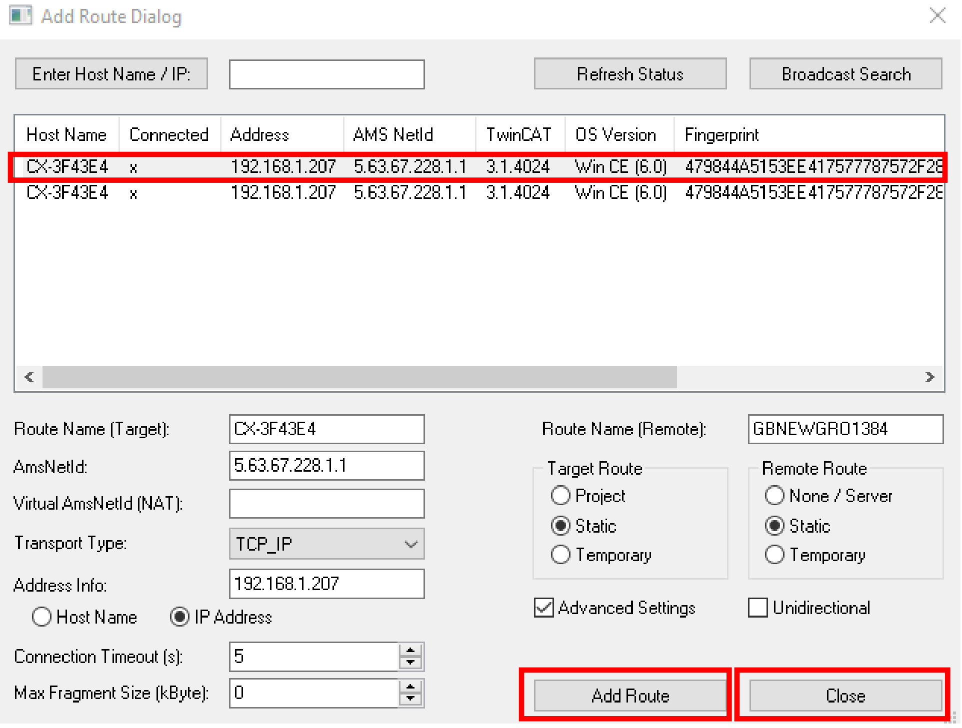


and then select the adapter to search for compatible devices:



☑ Intel(R) Ethernet Connection (16) I219-LM 192.168.1.239 255.255.255.0
☑ Realtek USB GbE Family Controller 192.168.1.238 255.255.255.0
☑ Intel(R) Wi-Fi 6E AX211 160MHz 10.75.12.64 255.255.254.0

6. Select the PLC to connect to and then click “Add Route” and then “Close”.



Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Fingerprint
CX-3F43E4	x	192.168.1.207	5.63.67.228.1.1	3.1.4024	Win CE (6.0)	479844A5153EE417577787572F2E
CX-3F43E4	x	192.168.1.207	5.63.67.228.1.1	3.1.4024	Win CE (6.0)	479844A5153EE417577787572F2E

Route Name (Target): CX-3F43E4
AmsNetId: 5.63.67.228.1.1
Virtual AmsNetId (NAT):
Transport Type: TCP_IP
Address Info: 192.168.1.207
○ Host Name ● IP Address
Connection Timeout (s): 5
Max Fragment Size (kByte): 0
Route Name (Remote): GBNEWGR01384
Target Route: ○ Project ● Static ○ Temporary
Remote Route: ○ None / Server ● Static ○ Temporary
☑ Advanced Settings ☐ Unidirectional

HINT: If there is an Administrator password challenge, the default password is “1”

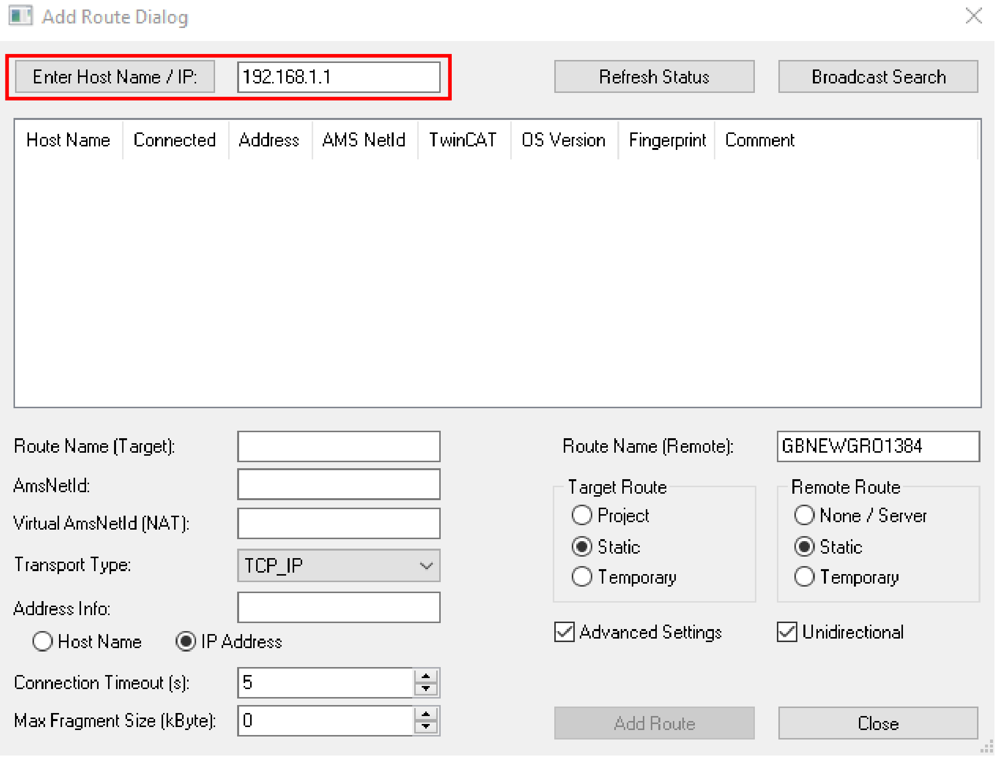
Click “Yes” if the message below appears:

Active solution platform 'TwinCAT RT (x64)' differs from new target platform 'TwinCAT RT (x86)'!

Change solution platform?

HINT: For brand new IPCs it is likely that the runtime hasn’t been installed, and that the EtherCAT port hasn’t been assigned. See section **3.14 How to install TwinCAT Runtime and configure Ethernet ports for EtherCAT use on a BECKHOFF IPC.**

If this doesn’t allow connection, Ping the IP address for the device or connect via Remote Desktop to ensure the connection to the PC is good. Then enter the IP address into the “Enter Host Name / IP:” input box and press enter.



Enter Host Name / IP: 192.168.1.1

Route Name (Target):

AmsNetId:

Virtual AmsNetId (NAT):

Transport Type: TCP_IP

Address Info:

○ Host Name ● IP Address

Connection Timeout (s): 5

Max Fragment Size (kByte): 0

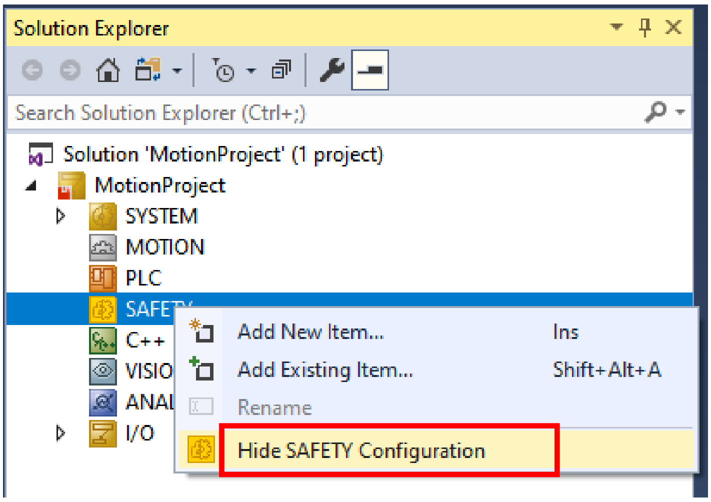
Route Name (Remote): GBNEWGR01384

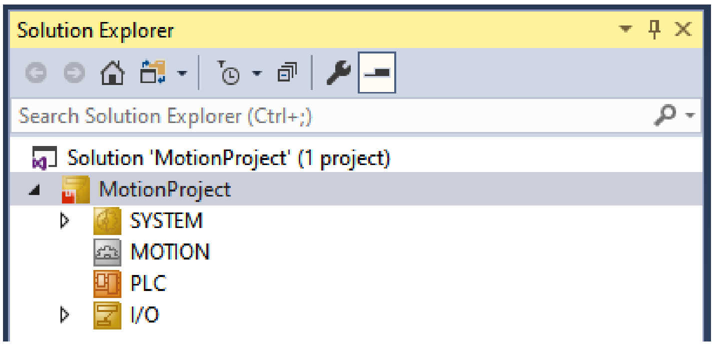
Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None / Server ● Static ○ Temporary

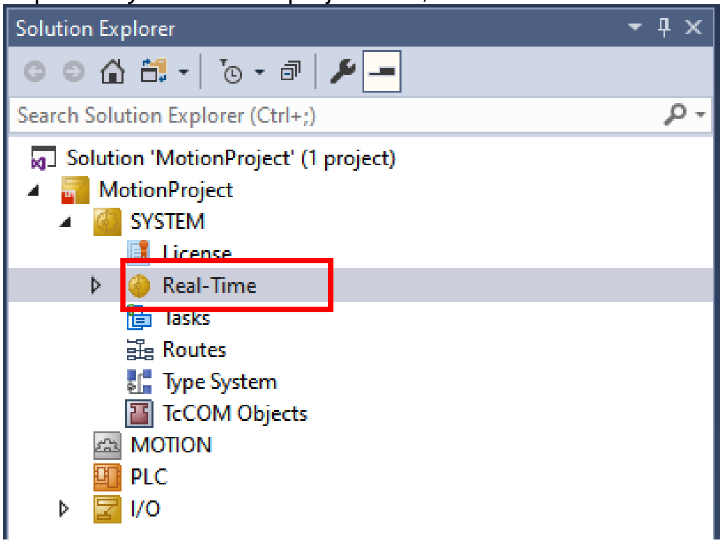
☑ Advanced Settings ☑ Unidirectional

7. An optional step is to clean up any unrequired items from the project tree e.g. this example doesn't require Safety, C++, or Vision etc. To do this right click on the item and select "Hide xxxx".

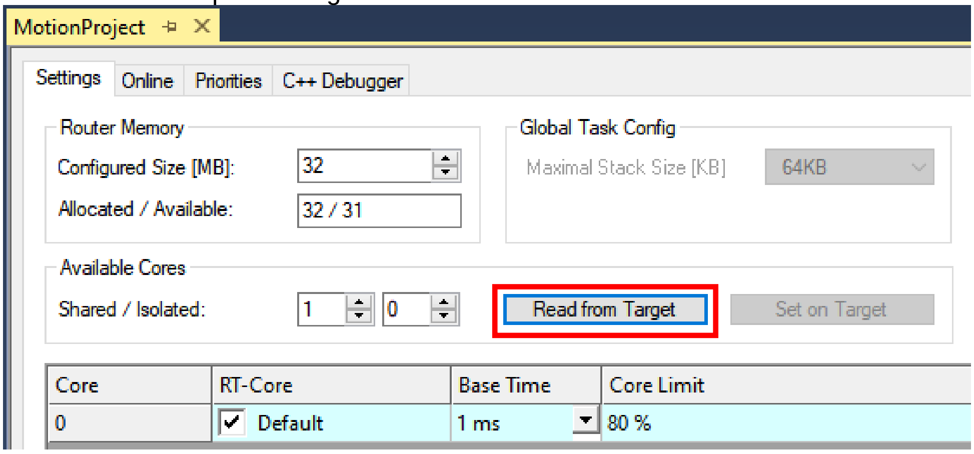




8. Expand “System” in the project tree, the double click on “Real-Time”:



When the project settings tab appears, Select “Read from Target” to update the information on the number of processing cores.



Router Memory

Configured Size [MB]: 32

Allocated / Available: 32 / 31

Global Task Config

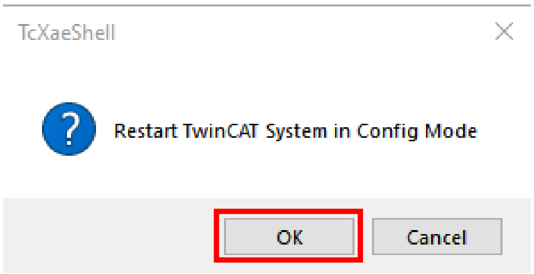
Maximal Stack Size [KB] 64KB

Available Cores

Shared / Isolated: 1 0

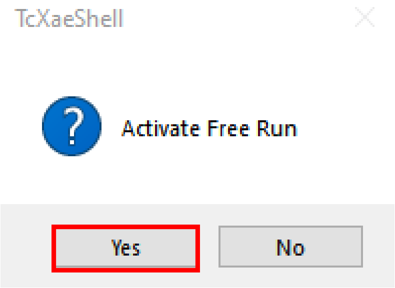
Core	RT-Core	Base Time	Core Limit
0	☑ Default	1 ms	80 %

9. Make sure the TwinCAT is in Config Mode by clicking on in the tool bar and then click “OK” when prompted with the “Restart TwinCAT system in Config Mode” message:



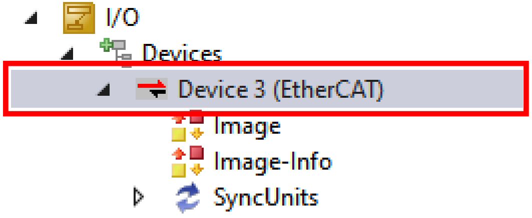
Restart TwinCAT System in Config Mode

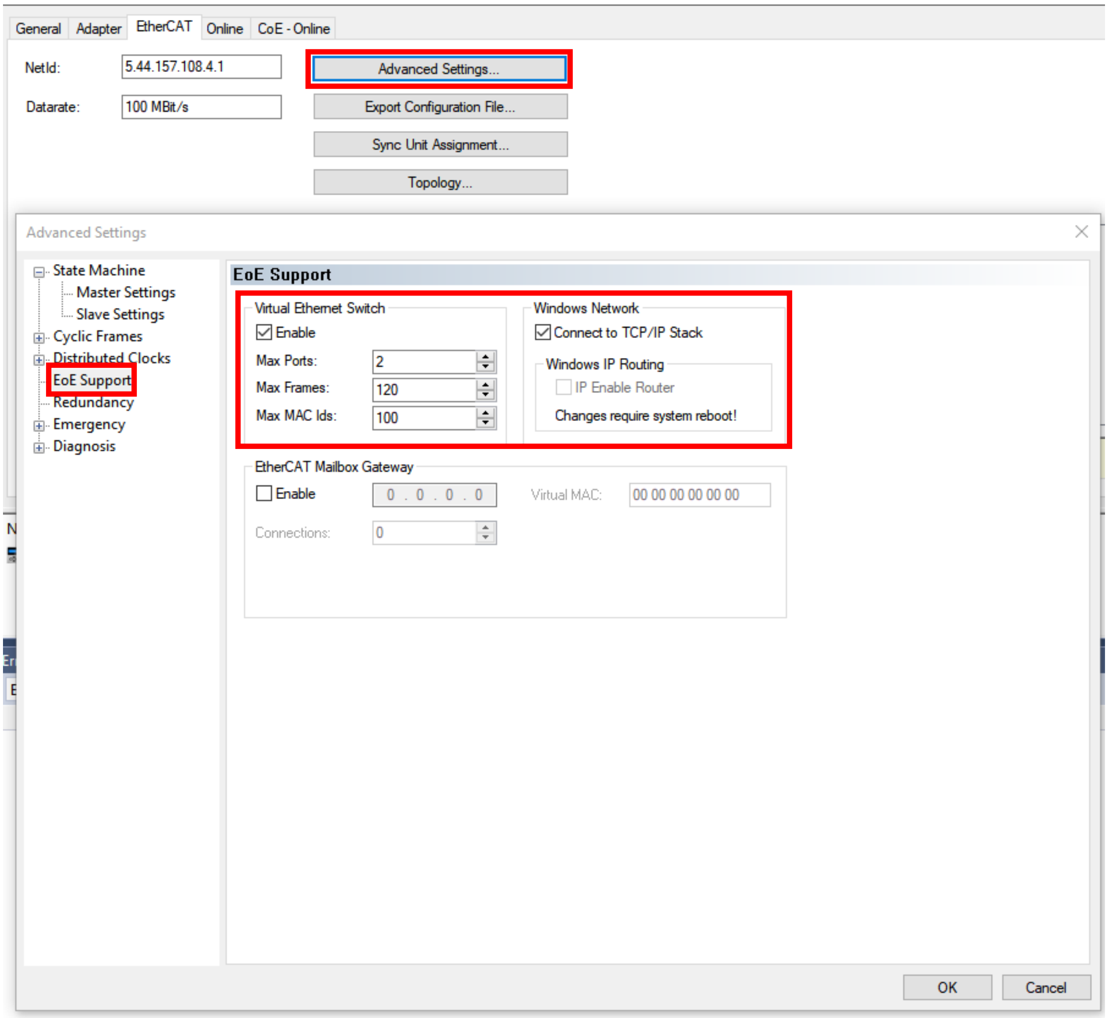
Failure to do this may result in the “Scan” feature being greyed out. Click “Yes” to the “Activate Free Run” pop-up.



Activate Free Run

10. Double click on the EtherCAT Device under “I/O” > ”Devices”, then select the EtherCAT tab, and then “Advanced Settings…”





NetId: 5.44.157.108.4.1

Datarate: 100 MBit/s

EoE Support

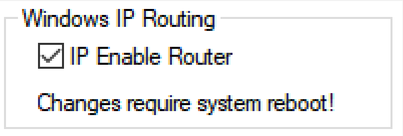
Virtual Ethernet Switch
☑ Enable
Max Ports: 2
Max Frames: 120
Max MAC Ids: 100

Windows Network
☑ Connect to TCP/IP Stack

Windows IP Routing
☐ IP Enable Router
Changes require system reboot!

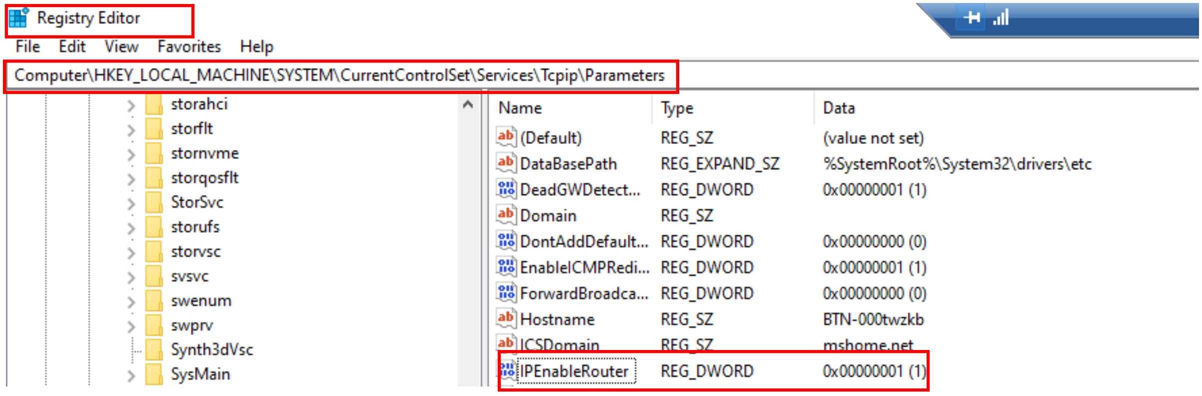
EtherCAT Mailbox Gateway
☐ Enable 0 . 0 . 0 . 0
Virtual MAC: 00 00 00 00 00 00
Connections: 0

HINT: for IPCs / PLCs, the “Windows IP Routing” checkbox must be checked to allow the IPC to act as a router for the EoE messages.



Windows IP Routing
☑ IP Enable Router
Changes require system reboot!

In the registry for the IPC/PLC this sets IPEnableRouter to 0x00000001(1)

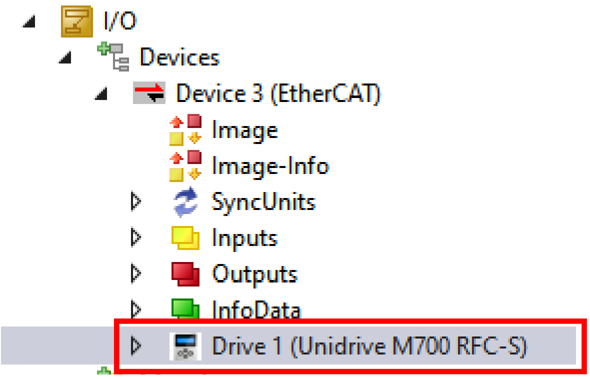


Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

- storahci
- storflt
- stornvme
- storqosflt
- StorSvc
- storufs
- storvsc
- svsvc
- swenum
- swprv
- Synth3dVsc
- SysMain

Name	Type	Data
(Default)	REG_SZ	(value not set)
DataBasePath	REG_EXPAND_SZ	%SystemRoot%\System32\drivers\etc
DeadGWDetect...	REG_DWORD	0x00000001 (1)
Domain	REG_SZ	
DontAddDefault...	REG_DWORD	0x00000000 (0)
EnableICMPRedi...	REG_DWORD	0x00000001 (1)
ForwardBroadca...	REG_DWORD	0x00000000 (0)
Hostname	REG_SZ	BTN-000twzkb
ICSDomain	REG_SZ	mshome.net
IPEnableRouter	REG_DWORD	0x00000001 (1)

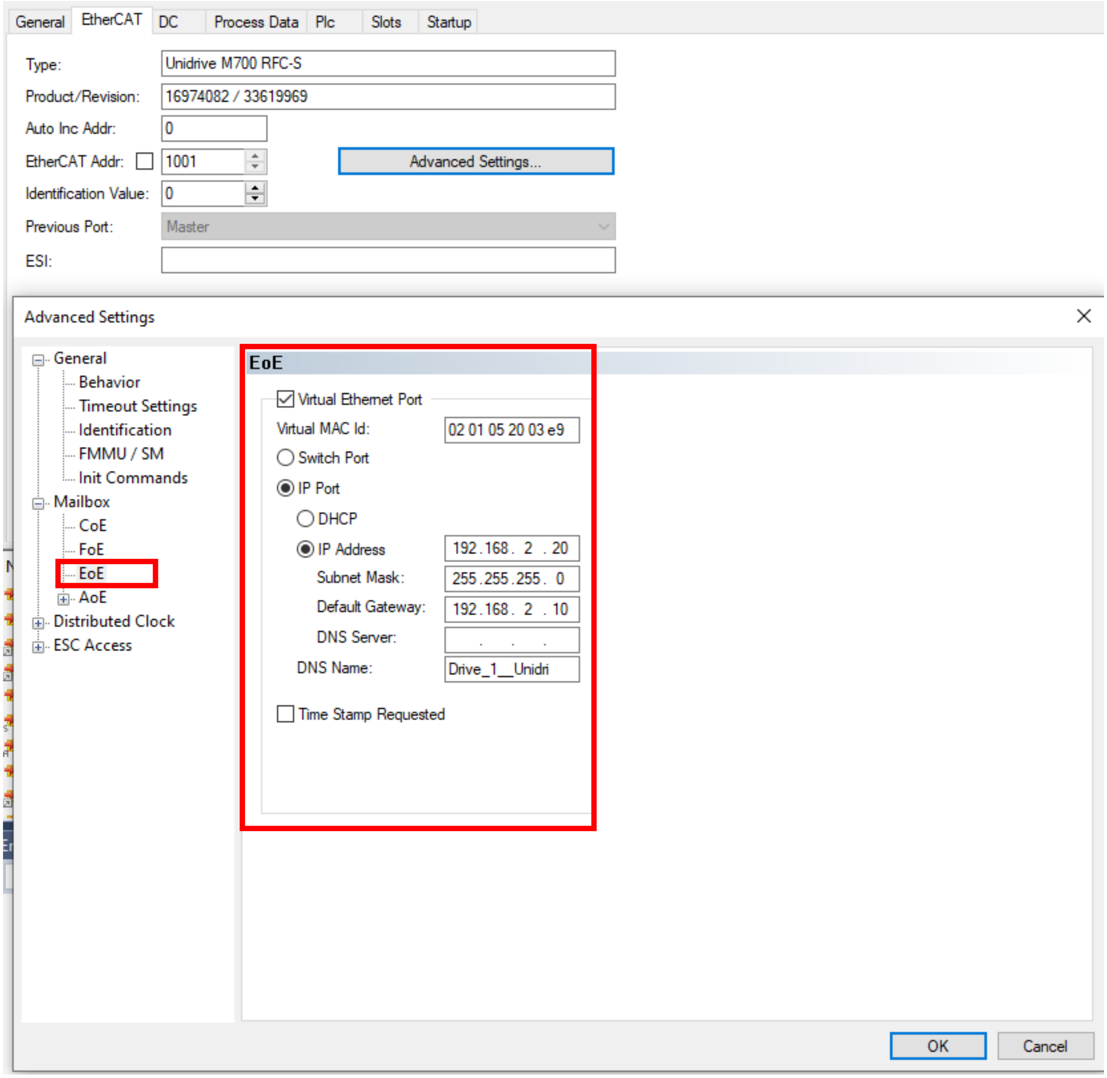
11. Next, click on the drive to connect to, “I/O” > “Devices” > “Device (n)” > “Drive(n)”:



- I/O
  - Devices
    - Device 3 (EtherCAT)
      - Image
      - Image-Info
      - SyncUnits
      - Inputs
      - Outputs
      - InfoData
      - Drive 1 (Unidrive M700 RFC-S)

When the device information opens, select the “EtherCAT” tab and then “Advanced Settings…”

When the “Advanced Settings” tab opens select “Mailbox” > “EoE”:



Type: Unidrive M700 RFC-S

Product/Revision: 16974082 / 33619969

Auto Inc Addr: 0

EtherCAT Addr: ☐ 1001

Identification Value: 0

Previous Port: Master

ESI:

EoE

☑ Virtual Ethernet Port

Virtual MAC Id: 02 01 05 20 03 e9

○ Switch Port

● IP Port

○ DHCP

● IP Address 192.168. 2 . 20

Subnet Mask: 255.255.255. 0

Default Gateway: 192.168. 2 . 10

DNS Server: . . .

DNS Name: Drive_1__Unidri

☐ Time Stamp Requested

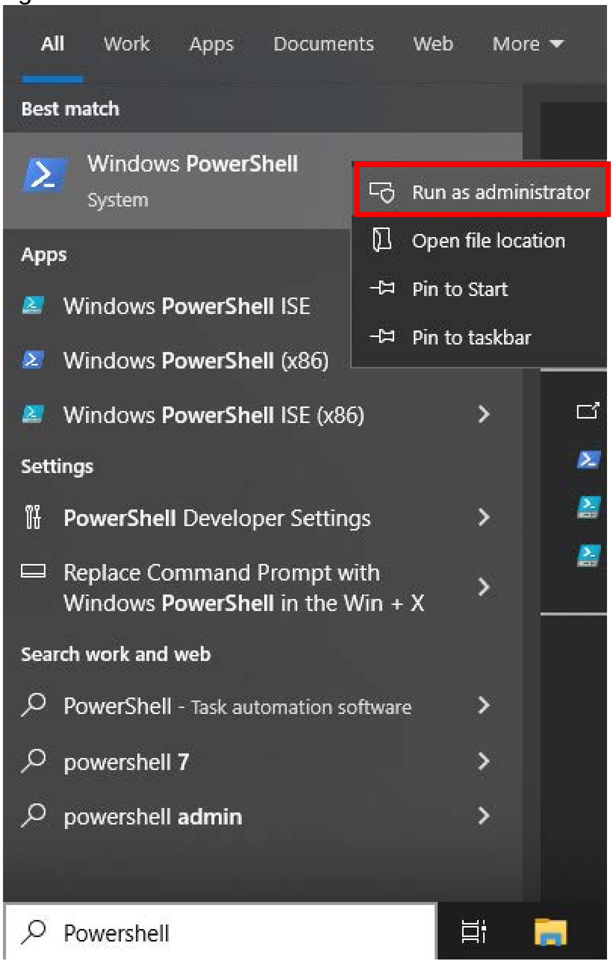
Check the “Virtual Ethernet Port” box, then set the “IP Port” and “IP Address” radio buttons.

The “IP Address” is set to the IP address that you would like the drive to have, which must not clash with the PLC’s EtherCAT port IP address, e.g. 192.168.2.20.

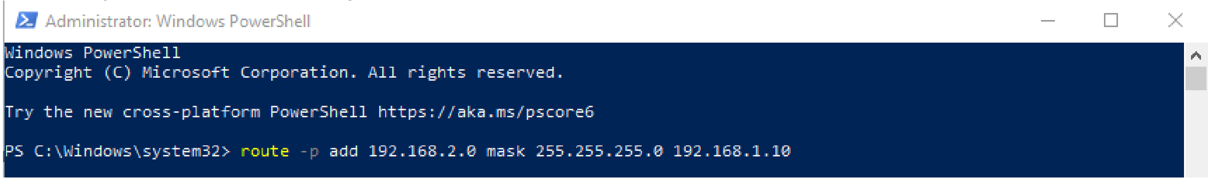
The subnet mask is set to 255.255.255.0. the Default gateway is set to the IP Address of the port that connects the drive to the PLC e.g. 192.168.2.10. Click “OK” when completed.

12. Download the configuration to the PLC by clicking on Activate Configuration .

13. 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”.



14. Add the persistent route using the “route” command:

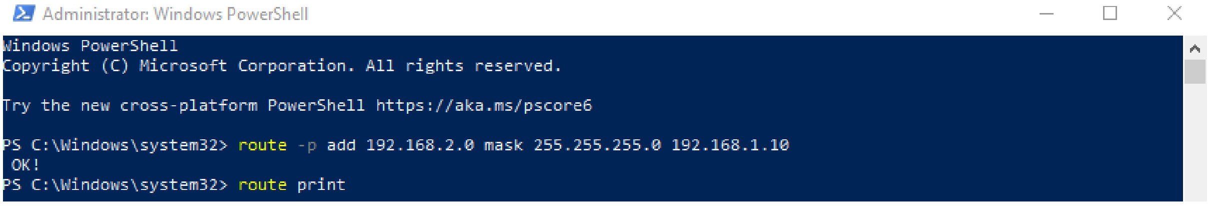


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> route -p add 192.168.2.0 mask 255.255.255.0 192.168.1.10
```

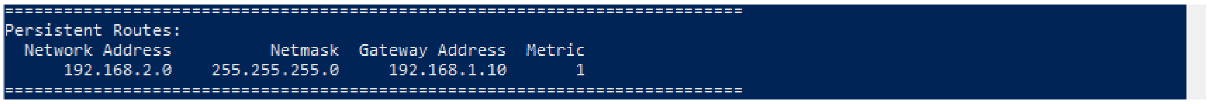
In this example:

a. “route -p add” gives a permanent route that will persist even when the power is cycled on the PC.
b. “192.168.2.0” allows routing to any IP address in the range of 192.168.2.1 to 192.168.2.255. The drive’s virtual Ethernet port was set to 192.168.2.20 in step 9 and falls within this range, therefore it will be reachable.
c. Subnet mask is “255.255.255.0”
d. The gateway is set to the IP address of the Ethernet port on the PLC/IPC, used to connect to the PC.

15. Use Route print to check the route is now listed:

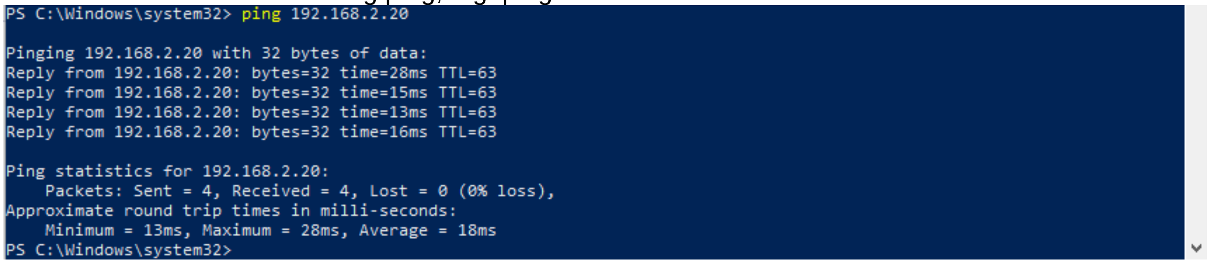


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> route -p add 192.168.2.0 mask 255.255.255.0 192.168.1.10
 OK!
PS C:\Windows\system32> route print
```



```
===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
      192.168.2.0    255.255.255.0     192.168.1.10       1
===========================================================================
```

16. Test the route to the drive using ping, e.g. ping 192.168.2.20:



```
PS C:\Windows\system32> ping 192.168.2.20

Pinging 192.168.2.20 with 32 bytes of data:
Reply from 192.168.2.20: bytes=32 time=28ms TTL=63
Reply from 192.168.2.20: bytes=32 time=15ms TTL=63
Reply from 192.168.2.20: bytes=32 time=13ms TTL=63
Reply from 192.168.2.20: bytes=32 time=16ms TTL=63

Ping statistics for 192.168.2.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 13ms, Maximum = 28ms, Average = 18ms
PS C:\Windows\system32>
```

17. PC tools such as Connect or CT Scope may now be used with the drive using the EoE IP Address e.g. 192.168.2.20.

3.7 How to create a Connect project for the application (EoE)

This section shows how to set up a Connect project for connections made using an EoE virtual port. If the EoE virtual port hasn't been configured, please see section **3.6 How to set up an EtherCAT EoE connection from the PLC to the drive**.

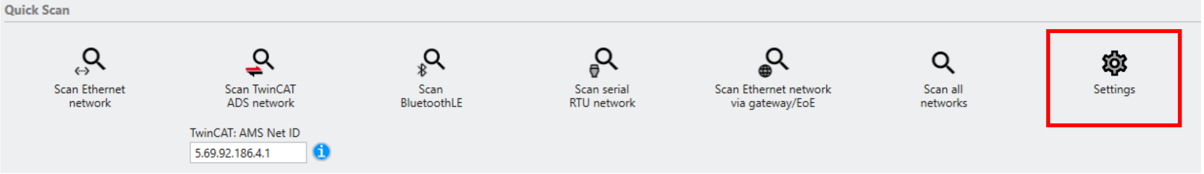
Create a new Connect project to match the requirement of the application:

1. Open the Connect PC software by double clicking the "Connect" icon.

2. Ensure the version of Connect is a minimum V03.00.00. If an earlier version is installed please upgrade to V03.00.00. The software file may be obtained from your local Control Techniques Drive Centre / Distributor or the Control Techniques website.

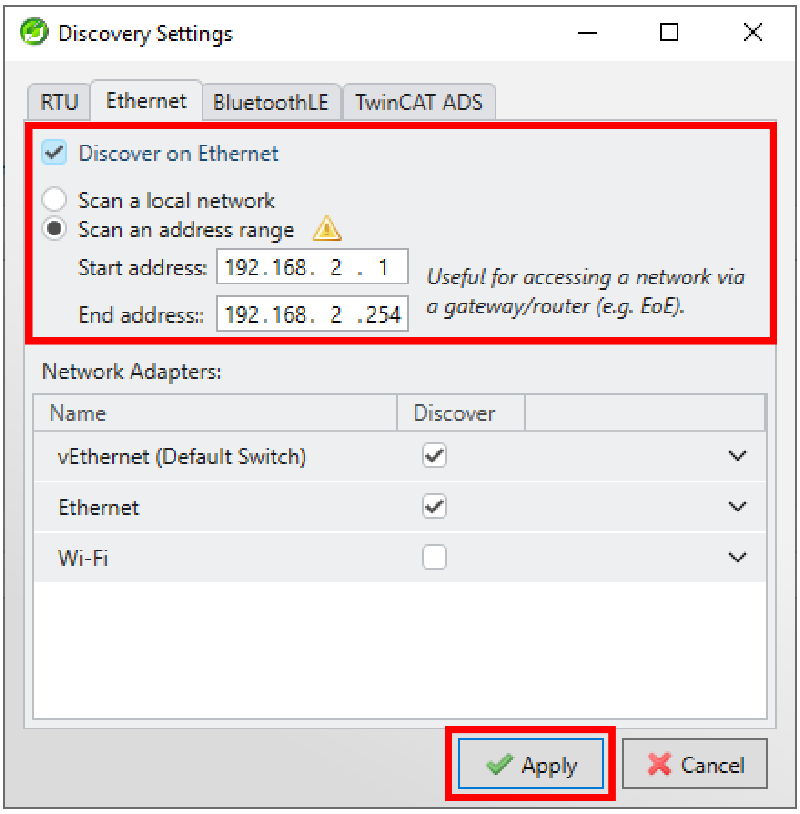
3. When Connect opens select "New project from network scan".

4. If using EoE, select “Settings”.



TwinCAT: AMS Net ID 5.69.92.186.4.1

5. The “Discovery Settings” dialog will appear, ensure the “Discover on Ethernet” and “Scan an address range” are selected and that the drive EoE address falls within the “Start address” and “End address” range.



☑ Discover on Ethernet

○ Scan a local network

● Scan an address range

Start address: 192.168. 2 . 1

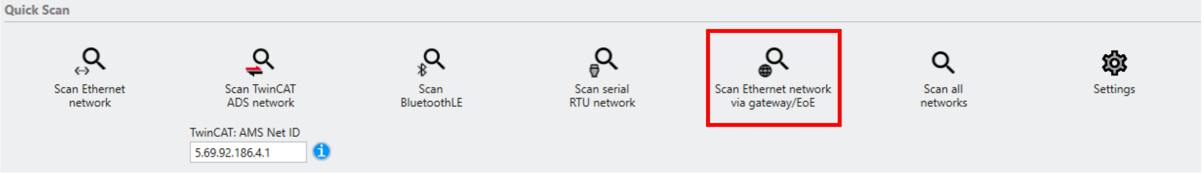
End address:: 192.168. 2 .254

Useful for accessing a network via a gateway/router (e.g. EoE).

Network Adapters:

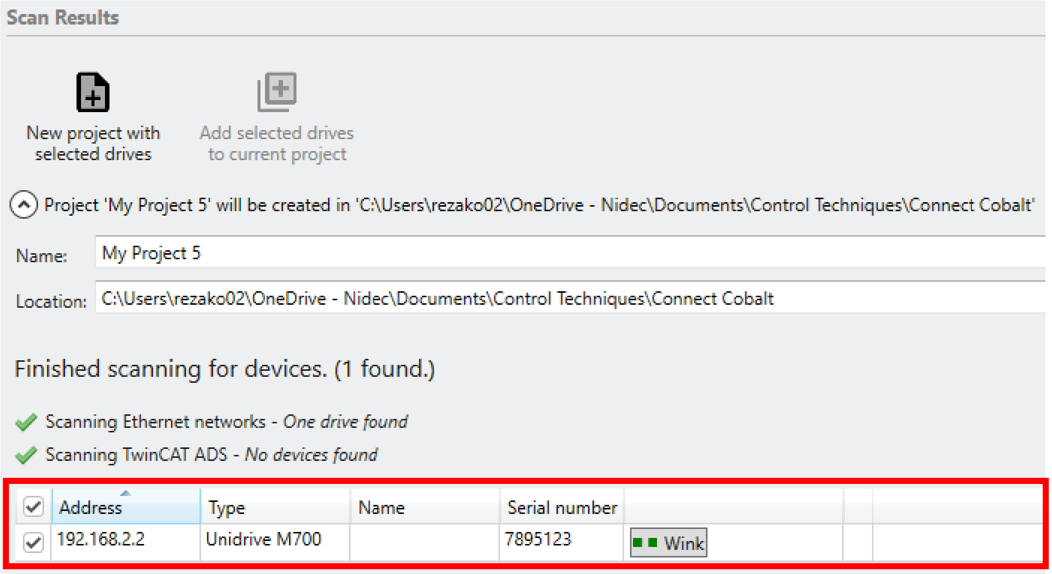
Name	Discover
vEthernet (Default Switch)	☑
Ethernet	☑
Wi-Fi	☐

6. Select the communications type, such as EoE, to scan that network and locate the drives to be configured:



TwinCAT: AMS Net ID 5.69.92.186.4.1

7. When the scan completes all of the available nodes on the network are found.



Project 'My Project 5' will be created in 'C:\Users\rezako02\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt'

Name: My Project 5

Location: C:\Users\rezako02\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt

Finished scanning for devices. (1 found.)

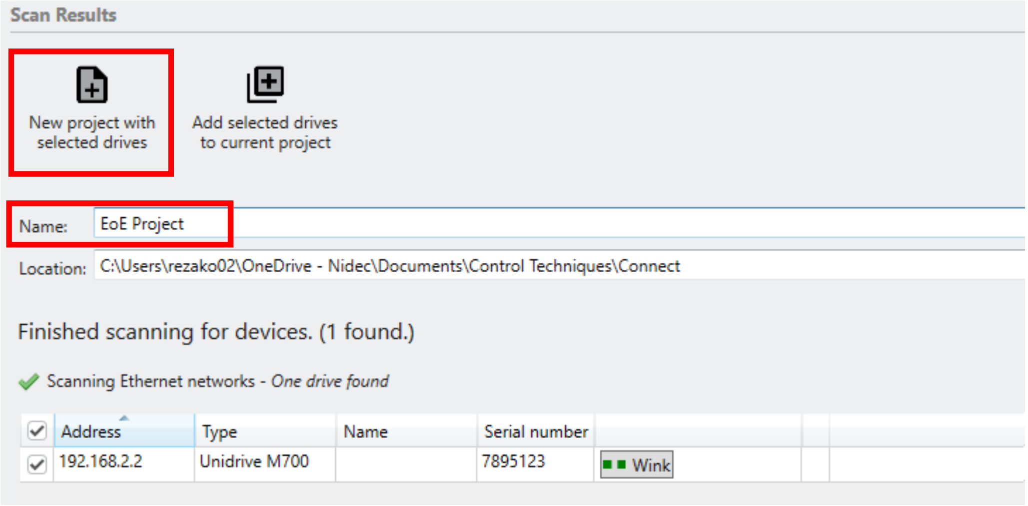
✓ Scanning Ethernet networks - *One drive found*

✓ Scanning TwinCAT ADS - *No devices found*

☑	Address	Type	Name	Serial number	
☑	192.168.2.2	Unidrive M700		7895123	Wink

HINT: For this to be successful on an EtherCAT network, the TwinCAT PLC/IPC and it's EtherCAT network must be running.

8. Give the project a meaningful name and then click "New project with selected drives".



Name: EoE Project

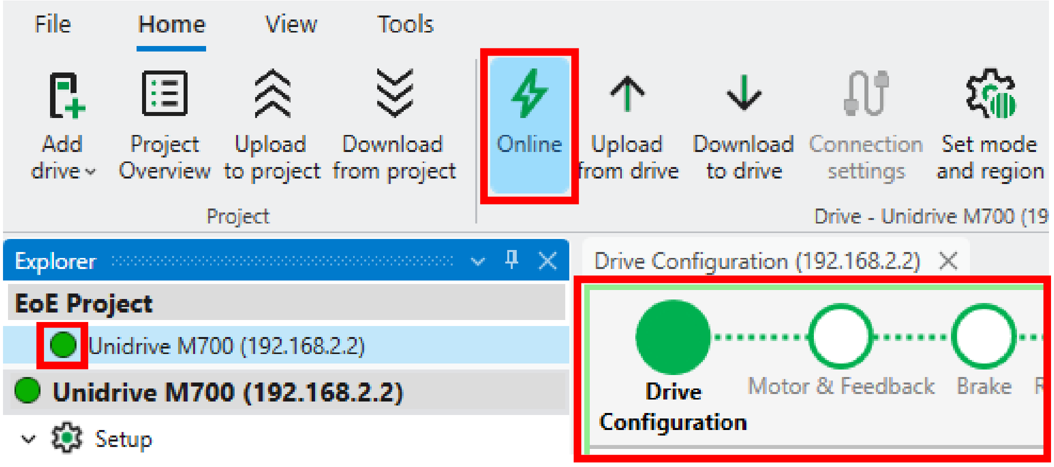
Location: C:\Users\rezako02\OneDrive - Nidec\Documents\Control Techniques\Connect

Finished scanning for devices. (1 found.)

✓ Scanning Ethernet networks - *One drive found*

☑	Address	Type	Name	Serial number	
☑	192.168.2.2	Unidrive M700		7895123	Wink

9. When the project opens, all drives on the network are “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.

10. The steps in section **2.6 Configure the drive using Connect** can now be followed to commission the drive.

3.8 How to Create a Connect project for the application (Ethernet & 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.

   Connect

2. Ensure the version of Connect is a minimum V03.00.00. If an earlier version is installed please upgrade to V03.00.00. The software file may be obtained from your local Control Techniques Drive Centre / Distributor or the Control Techniques website.

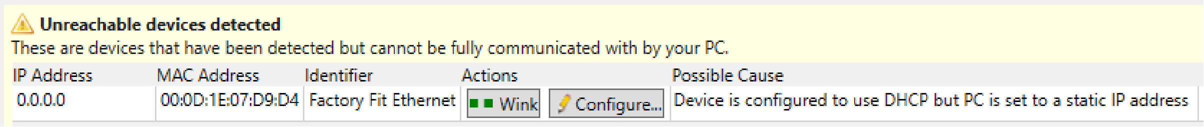
3. When Connect starts, select “New project from network scan”.

   New project from network scan

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

   Scan Ethernet network

5. After scanning, any drive with an IP address 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....

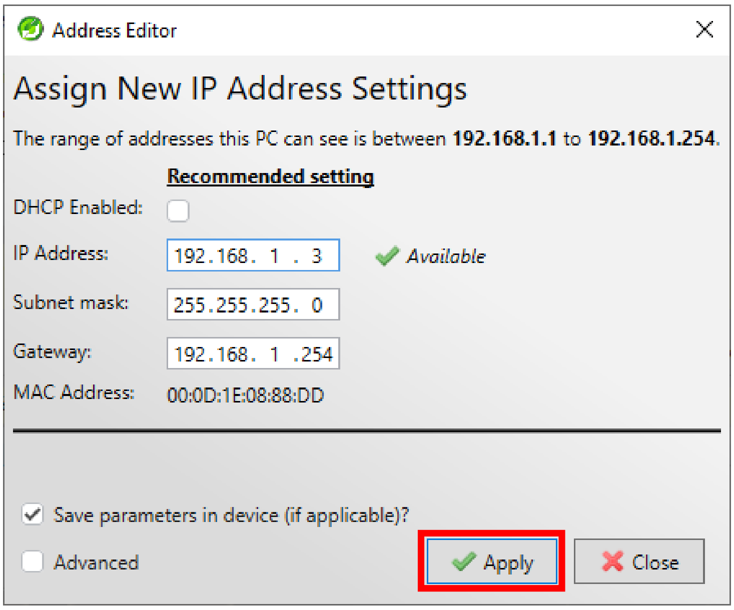


Unreachable devices detected

These are devices that have been detected but cannot be fully communicated with by your PC.

IP Address	MAC Address	Identifier	Actions	Possible Cause
0.0.0.0	00:0D:1E:07:D9:D4	Factory Fit Ethernet	Wink Configure...	Device is configured to use DHCP but PC is set to a static IP address

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



Assign New IP Address Settings

The range of addresses this PC can see is between **192.168.1.1** to **192.168.1.254**.

Recommended setting

DHCP Enabled: ☐

IP Address: 192.168. 1 . 3 ✓ *Available*

Subnet mask: 255.255.255. 0

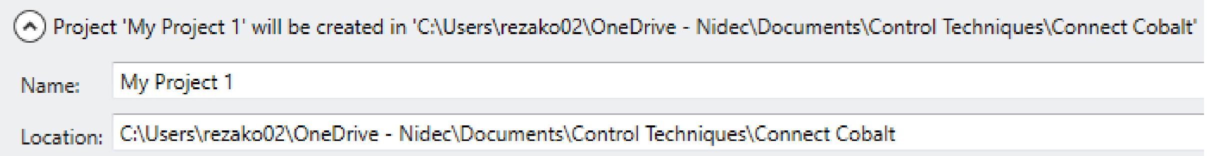
Gateway: 192.168. 1 .254

MAC Address: 00:0D:1E:08:88:DD

☑ Save parameters in device (if applicable)?

☐ Advanced

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



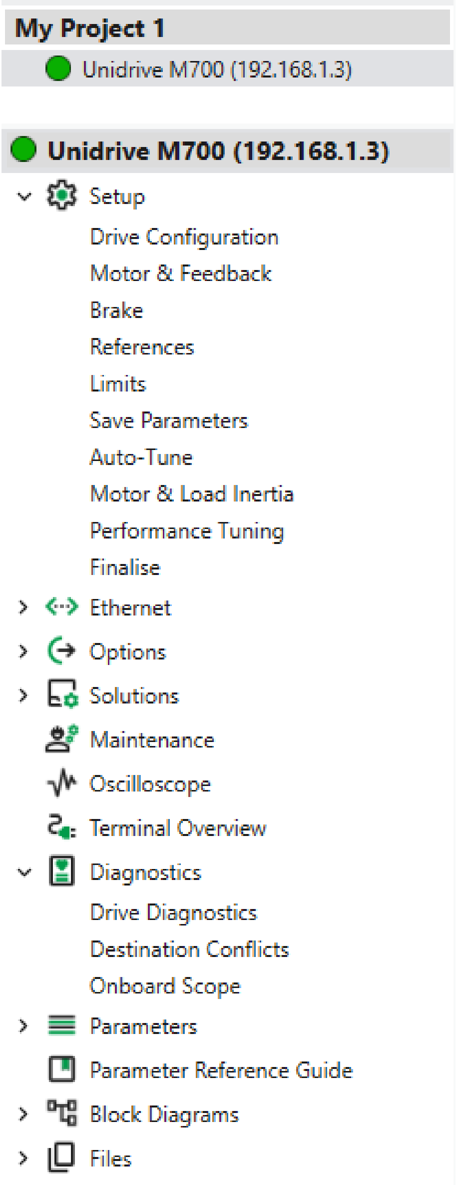
Project 'My Project 1' will be created in 'C:\Users\rezako02\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt'

Name: My Project 1

Location: C:\Users\rezako02\OneDrive - Nidec\Documents\Control Techniques\Connect Cobalt

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

8. The project is created with the selected drives.



Explorer

My Project 1

● Unidrive M700 (192.168.1.3)

● Unidrive M700 (192.168.1.3)

- Setup
  - Drive Configuration
  - Motor & Feedback
  - Brake
  - References
  - Limits
  - Save Parameters
  - Auto-Tune
  - Motor & Load Inertia
  - Performance Tuning
  - Finalise
- Ethernet
- Options
- Solutions
- Maintenance
- Oscilloscope
- Terminal Overview
- Diagnostics
  - Drive Diagnostics
  - Destination Conflicts
  - Onboard Scope
- Parameters
- Parameter Reference Guide
- Block Diagrams
- Files

9. The steps in section **2.6 Configure the drive using Connect** can now be followed to commission the drive.

3.9 How to upgrade the EtherCAT interface firmware

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.

There are 2 main ways to update the firmware and they are each used in different situations.

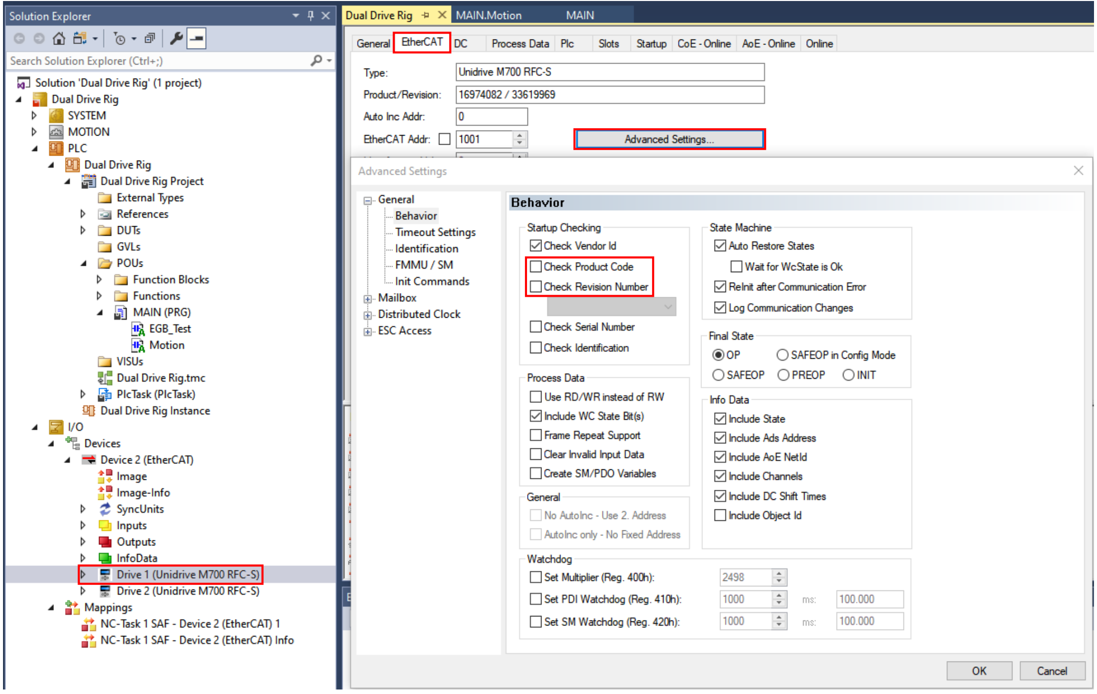
The most common way as described in section **3.9.1 How to upgrade the EtherCAT interface firmware using Connect** shows how to update the firmware using Connect which us used when there is an existing, working connection to the drive, either by ADS or AoE or serial / ethernet communications.

The second method is for situations where communications are by EtherCAT and it hasn’t been possible to establish an ADS or AOE connection. In this scenario it is recommended to use FoE in TwinCAT to update the firmware. This is described in section **3.9.2 How to use FoE to update the EtherCAT interface firmware.**

3.9.1 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. The following steps assume that there is a working EtherCAT connection already – if a connection can’t be established, use the steps in section **3.9.2 How to use FoE to update the EtherCAT interface firmware**.

1. Before updating using the EtherCAT interface using Connect, in the TwinCAT project, uncheck the “Check Product Code” and “Check Revision Number” check boxes in the EtherCAT drive properties, and then activate the configuration . This is so that the changes to the product code that may happen when the firmware is updated don’t cause ADS / AoE communications to stop part way through the firmware update.



Type: Unidrive M700 RFC-S
Product/Revision: 16974082 / 33619969
Auto Inc Addr: 0
EtherCAT Addr: ☐ 1001

Behavior

Startup Checking
☑ Check Vendor Id
☐ Check Product Code
☐ Check Revision Number
☐ Check Serial Number
☐ Check Identification

State Machine
☑ Auto Restore States
☐ Wait for WcState is Ok
☑ ReInit after Communication Error
☑ Log Communication Changes

Final State
● OP ○ SAFEOP in Config Mode
○ SAFEOP ○ PREOP ○ INIT

Process Data
☐ Use RD/WR instead of RW
☑ Include WC State Bit(s)
☐ Frame Repeat Support
☐ Clear Invalid Input Data
☐ Create SM/PDO Variables

Info Data
☑ Include State
☑ Include Ads Address
☑ Include AoE NetId
☑ Include Channels
☑ Include DC Shift Times
☐ Include Object Id

General
☐ No AutoInc - Use 2. Address
☐ AutoInc only - No Fixed Address

Watchdog
☐ Set Multiplier (Reg. 400h): 2498
☐ Set PDI Watchdog (Reg. 410h): 1000 ms: 100.000
☐ Set SM Watchdog (Reg. 420h): 1000 ms: 100.000

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

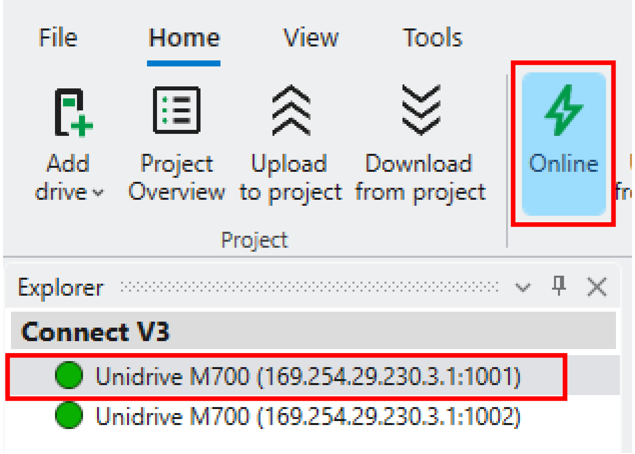
3. Establish a connection between the PC being used for Connect and the Control Techniques drive, this connection can be achieved via ADS, EoE, Ethernet or serial communications.

   See sections **2.4** and **2.5** for details on how to establish the connection over an EtherCAT network using AoE, sections **3.6** and **3.7** for details on how to establish the connection over an EtherCAT network using EoE, or section **3.8** for details on how to establish the connection over Ethernet or serial communications.

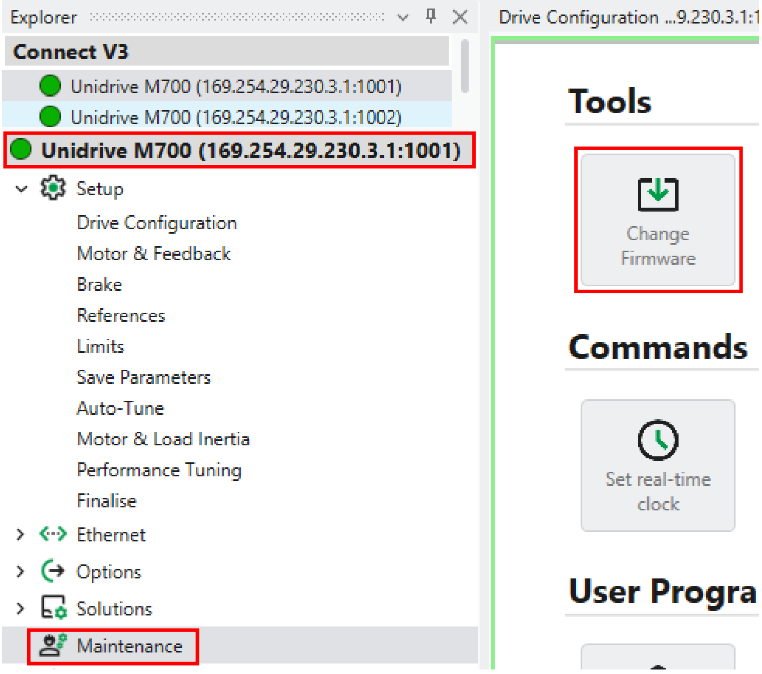
   The remaining firmware upgrade steps are shown using ADS.

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

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

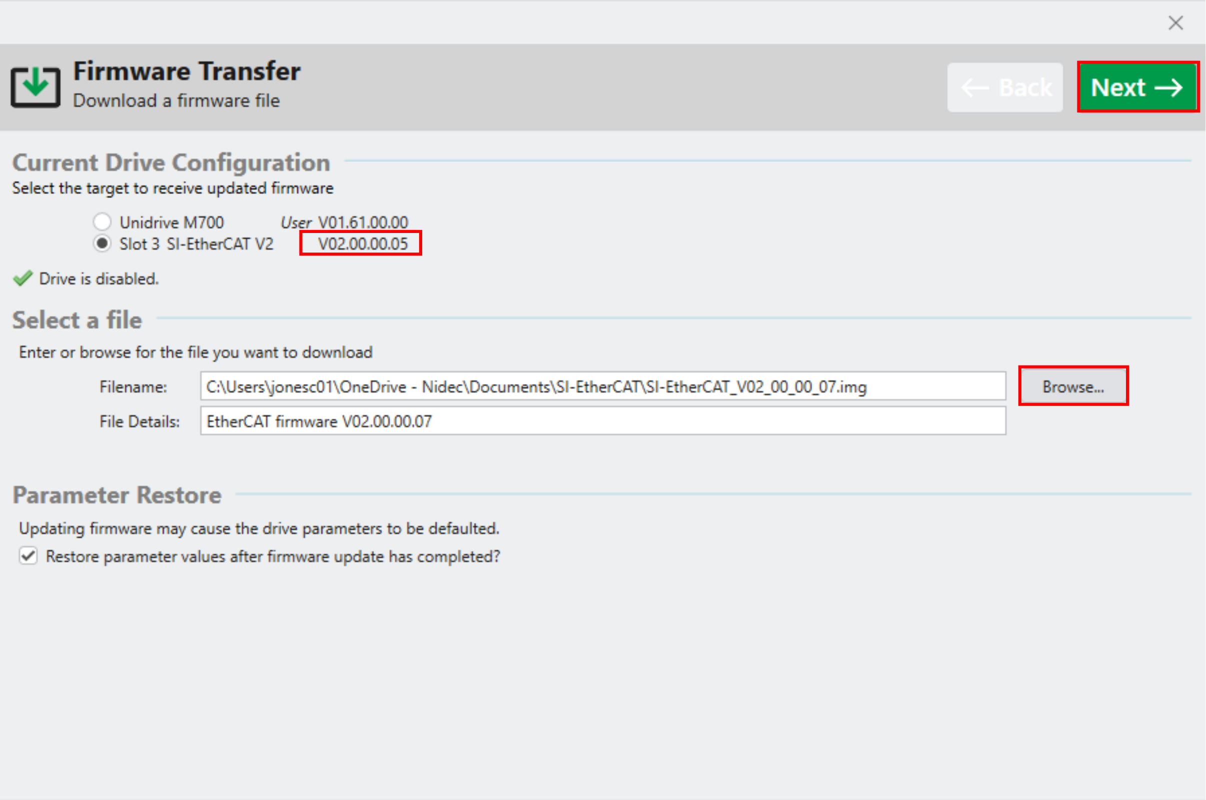


5. From the drive explorer tree double click on “Maintenance” and then “Change Firmware”:



6. The “Firmware Transfer” tool opens. 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:



Current Drive Configuration

Select the target to receive updated firmware

○ Unidrive M700 *User* V01.61.00.00
● Slot 3 SI-EtherCAT V2 V02.00.00.05

✓ Drive is disabled.

Select a file

Enter or browse for the file you want to download

Filename: C:\Users\jonesc01\OneDrive - Nidec\Documents\SI-EtherCAT\SI-EtherCAT_V02_00_00_07.img

File Details: EtherCAT firmware V02.00.00.07

Parameter Restore

Updating firmware may cause the drive parameters to be defaulted.

☑ Restore parameter values after firmware update has completed?

   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 **Next →** to continue.

7. 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.

8. When the firmware update has completed, power down the PLC/IPC and drives, then re-apply power to re-establish the network.

3.9.2 How to use FoE to update the EtherCAT interface firmware

In situations where it hasn't been possible to establish communications to the drive and only EtherCAT communications are available, instead of using Connect over ADS / AoE, it is possible to load the firmware directly using TwinCAT provided the EtherCAT interface firmware is greater than V01.08.00.14.

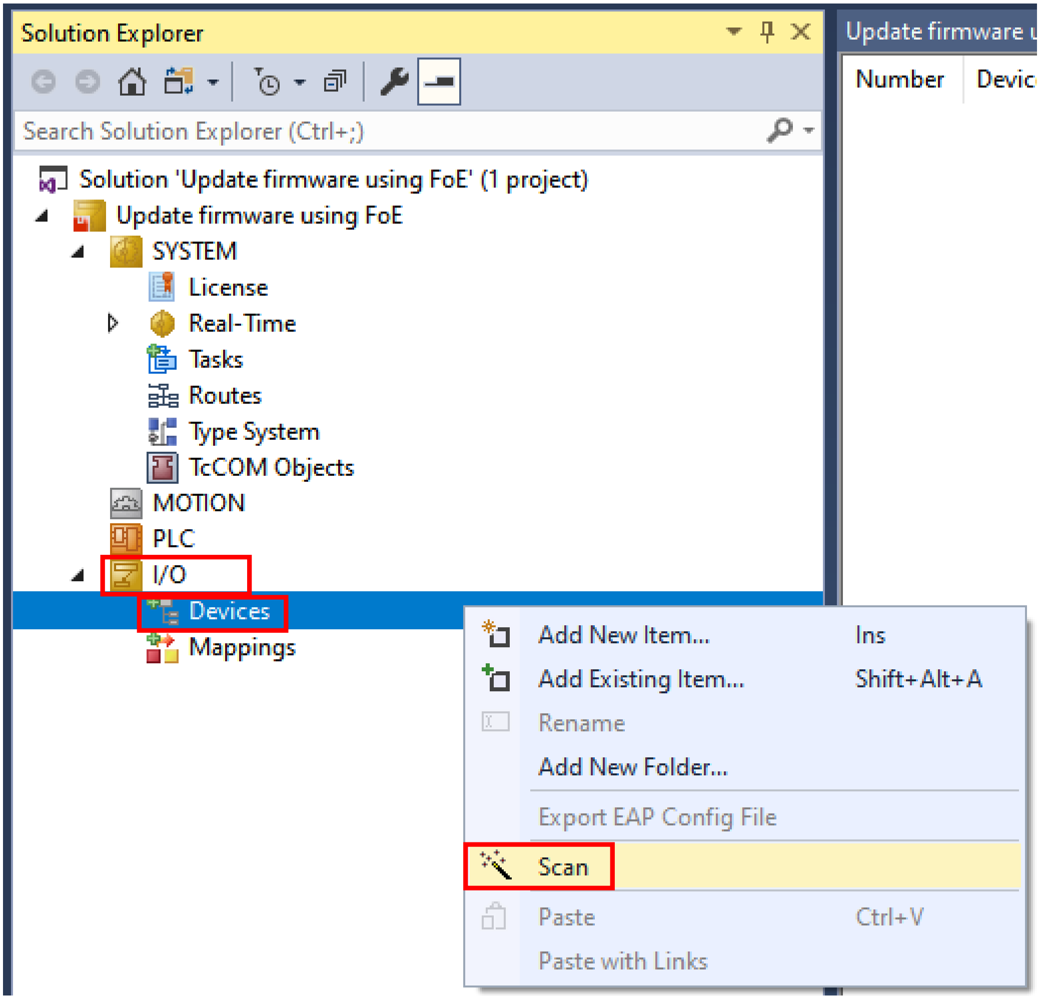
For firmware older than V01.08.00.14 an alternative connection must be made to the drive using Control Techniques RS485 adapter, or using Ethernet and an SI-Ethernet / Factory Fitted Ethernet option, or a KI-Keypad Plus.

This situation might occur if the EDS files installed in TwinCAT don't match the firmware in the module. Before using this method, make sure that the EDS files in TwinCAT match the new firmware that will be loaded in the following steps.

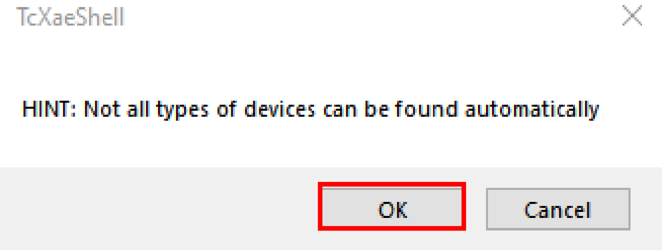
It is assumed that the steps in section **2.4 Initial TwinCAT project setup** have been followed to create a project before using these instructions.

For a brand new project it is **strongly** recommended to update to EtherCAT interface V2 firmware and ESI files. Use the following steps to update the firmware:

1. Expand "I/O" in the Solution Explorer and then right click on "Devices" and then select "Scan":



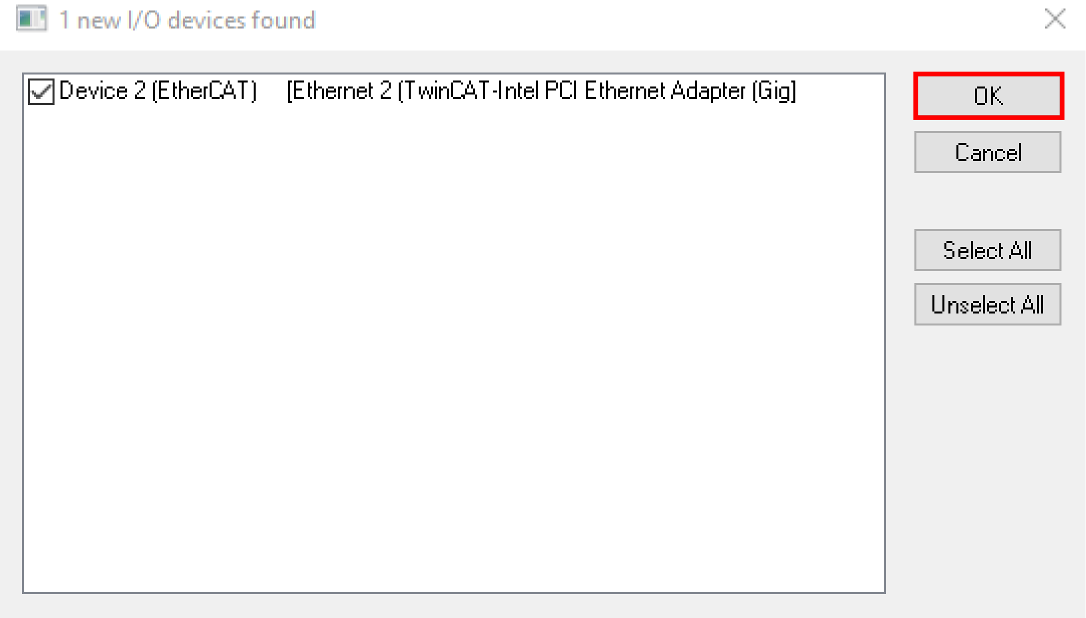
2. Click “OK” to the “HINT: Not all types of devices can be found automatically” pop-up.



TcXaeShell

HINT: Not all types of devices can be found automatically

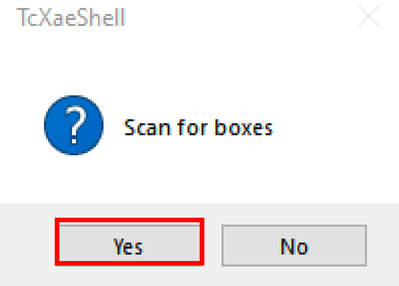
3. After the device search finishes, select the EtherCAT interface and then select “OK”.



1 new I/O devices found

☑ Device 2 (EtherCAT) [Ethernet 2 (TwinCAT-Intel PCI Ethernet Adapter (Gig)]

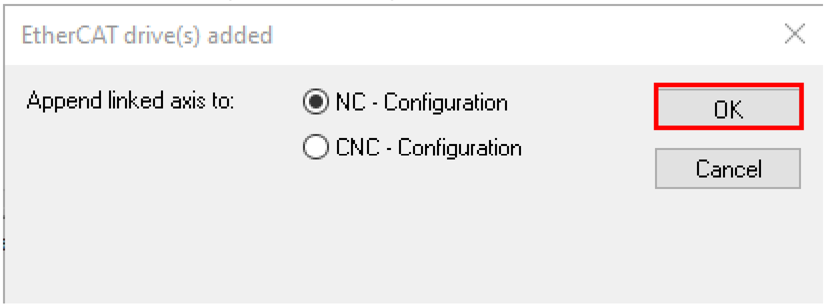
4. Select “Yes” when the “Scan for boxes” pop-up appears:



TcXaeShell

Scan for boxes

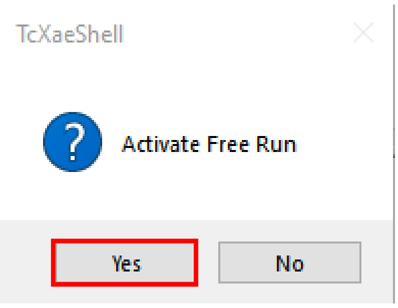
5. Click “OK” to adding “NC – Configuration”.



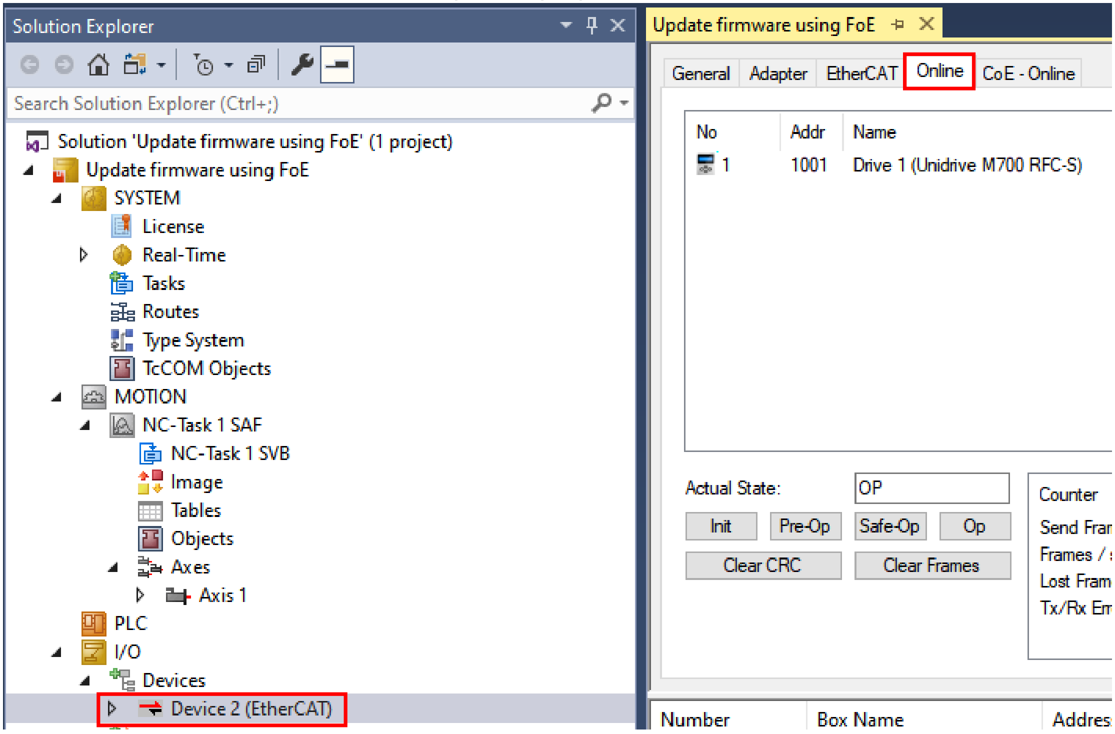
EtherCAT drive(s) added

Append linked axis to: ● NC - Configuration ○ CNC - Configuration

6. Click “Yes” on the “Activate Free Run” pop-up.



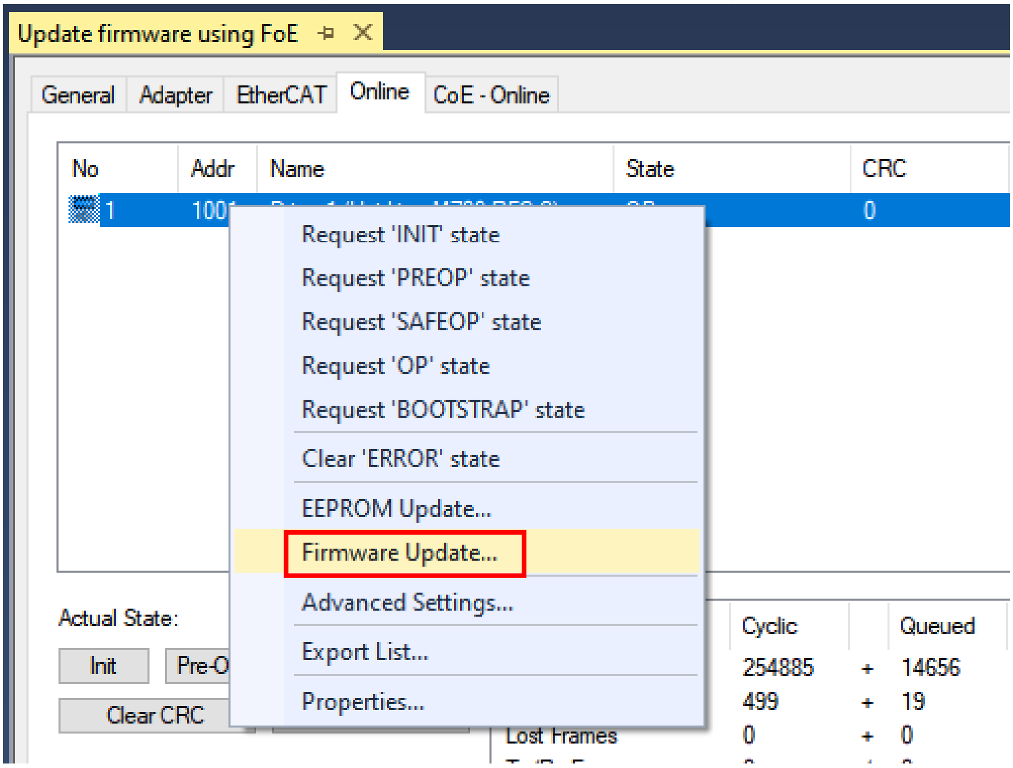
7. Double click the EtherCAT master to open its properties and then select the “Online” tab:



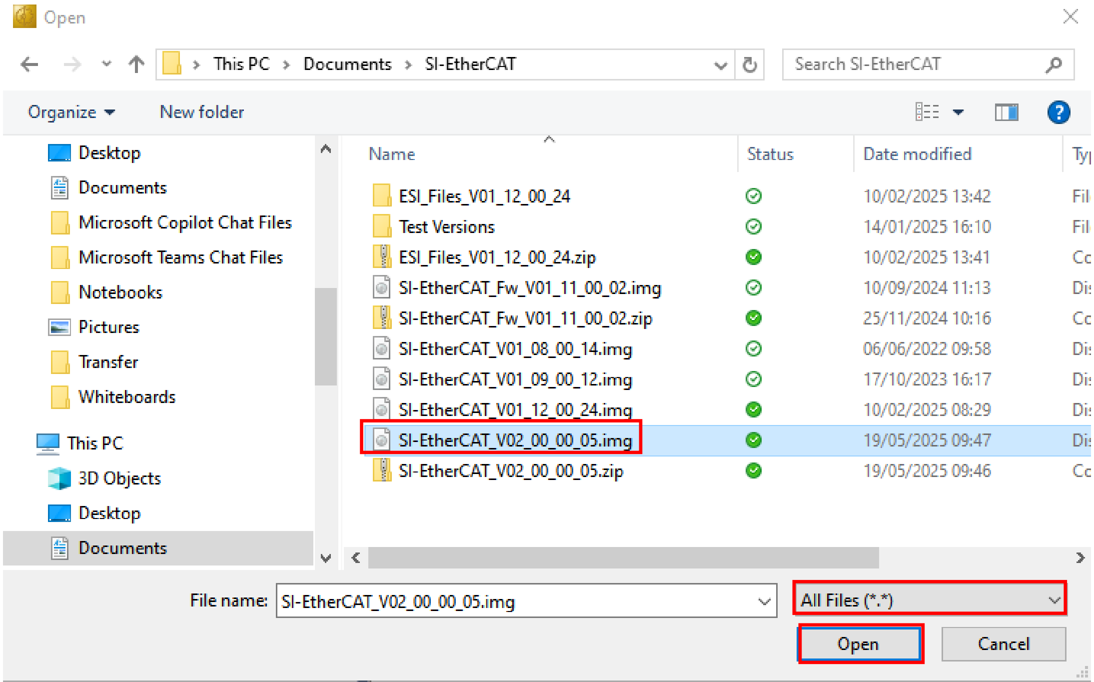
No	Addr	Name
1	1001	Drive 1 (Unidrive M700 RFC-S)

Actual State: OP

8. Right click on the drive to update and then select “Firmware Update…”



9. The file explorer opens. Set the file type to “All Files (*.*)”, then select the firmware file, and then select “Open”.

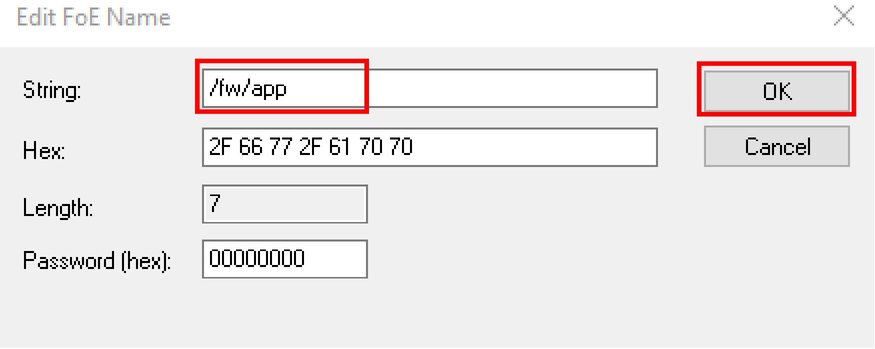


Name	Status	Date modified	Ty
ESI_Files_V01_12_00_24	✓	10/02/2025 13:42	Fil
Test Versions	✓	14/01/2025 16:10	Fil
ESI_Files_V01_12_00_24.zip	✓	10/02/2025 13:41	Cc
SI-EtherCAT_Fw_V01_11_00_02.img	✓	10/09/2024 11:13	Di
SI-EtherCAT_Fw_V01_11_00_02.zip	✓	25/11/2024 10:16	Cc
SI-EtherCAT_V01_08_00_14.img	✓	06/06/2022 09:58	Di
SI-EtherCAT_V01_09_00_12.img	✓	17/10/2023 16:17	Di
SI-EtherCAT_V01_12_00_24.img	✓	10/02/2025 08:29	Di
SI-EtherCAT_V02_00_00_05.img	✓	19/05/2025 09:47	Di
SI-EtherCAT_V02_00_00_05.zip	✓	19/05/2025 09:46	Cc

File name: SI-EtherCAT_V02_00_00_05.img

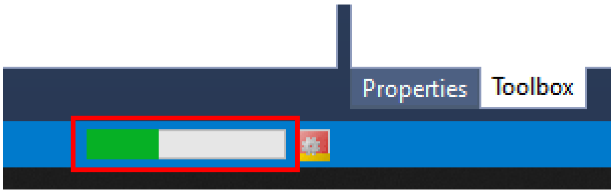
All Files (*.*)

10. When the “Edit FoE Name” pop-up appears change “String:” to the correct file path for the firmware image file which is “/fw/app”, and then select “OK”.

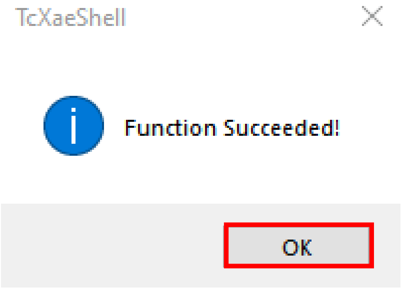


String: /fw/app
Hex: 2F 66 77 2F 61 70 70
Length: 7
Password (hex): 00000000

11. The file download runs. Wait until it finishes.

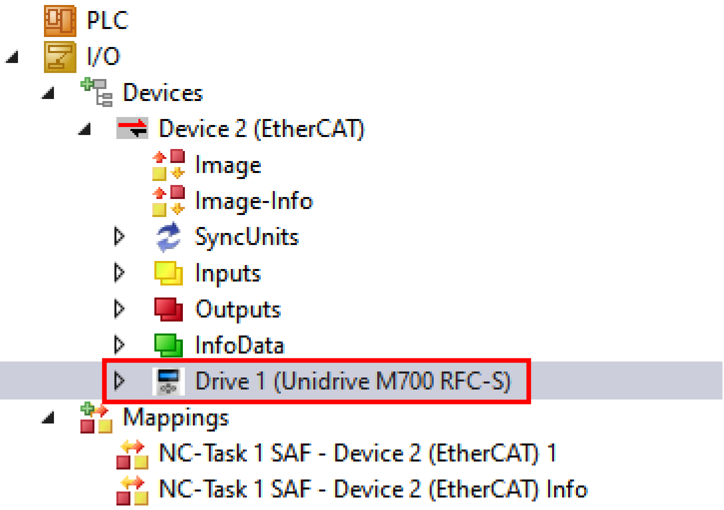


12. When the download completes the “Function Succeeded!” message appears. Click “OK”.

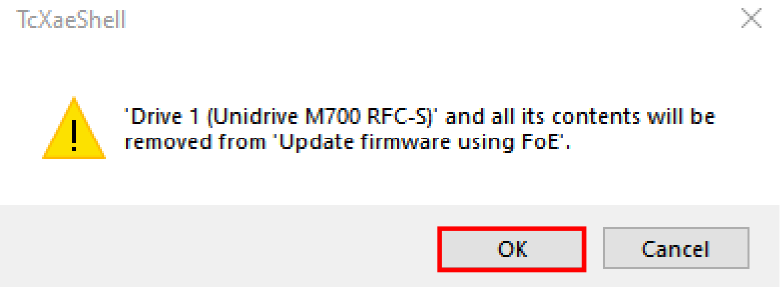


Function Succeeded!

13. Delete the drive that has been updated in the Solution Explorer tree.

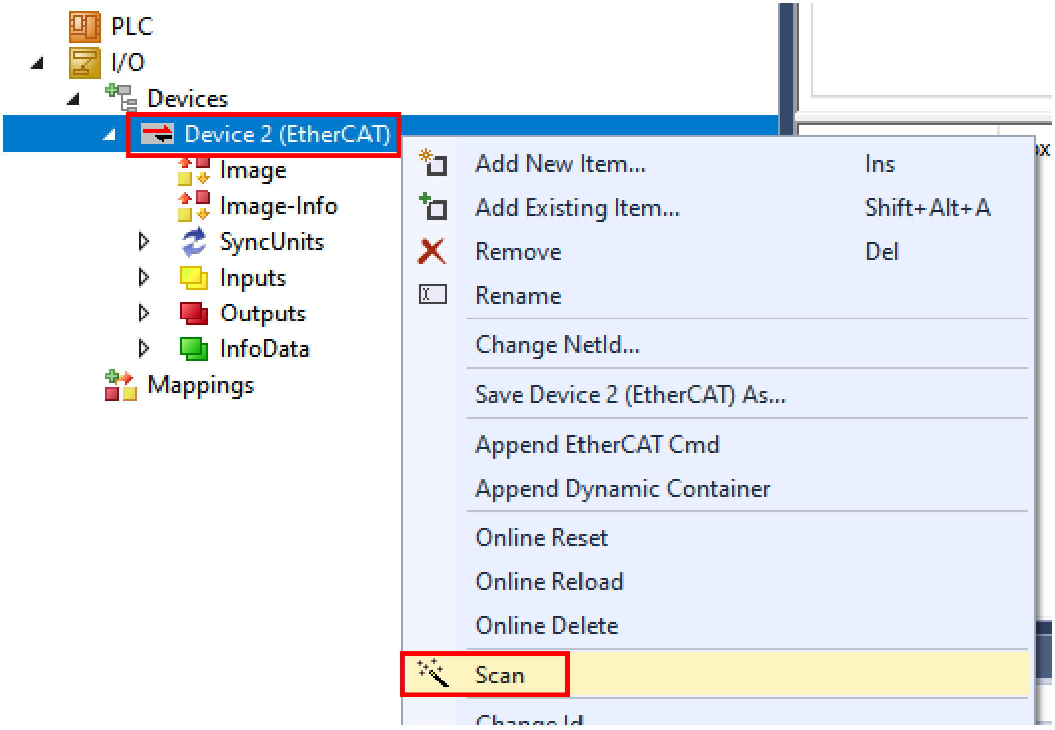


Click “OK” to the following message:

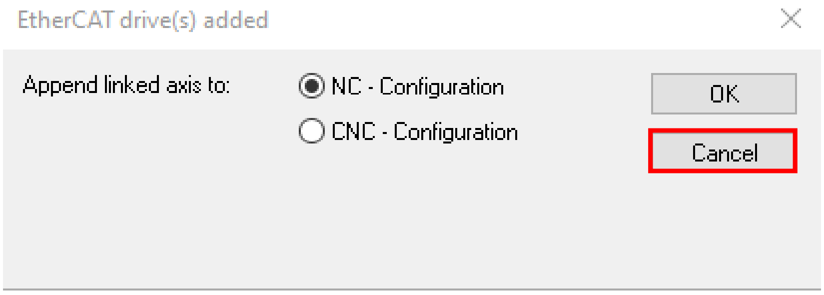


'Drive 1 (Unidrive M700 RFC-S)' and all its contents will be removed from 'Update firmware using FoE'.

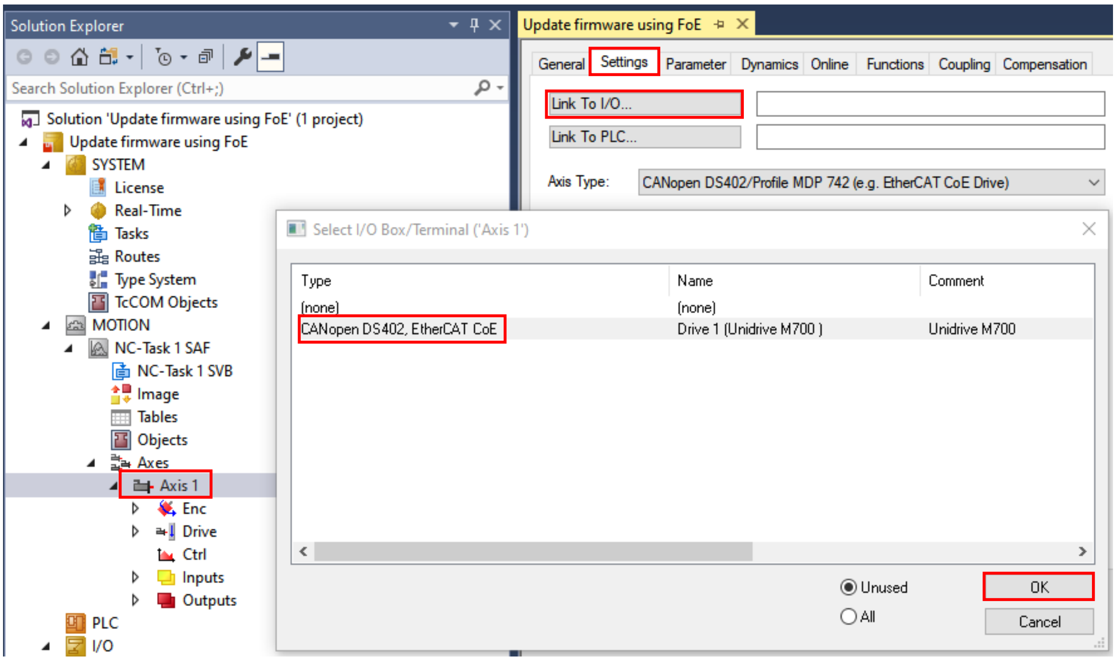
14. Right click on the EtherCAT master in the solution explorer tree and then select “Scan”.



15. When the “EtherCAT drive(s) added” dialog appears select “Cancel” so that another NC axis isn’t added to the project.

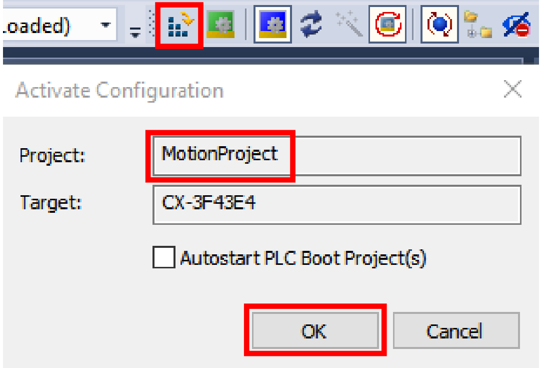


16. Link the NC that was created for the drive axis originally, back to the drive EtherCAT comms by double clicking on the axis e.g. Axis 1 to open it properties and select the “Settings” tab. Click on “Link To I/O…”, select the Drive from the list to link to the axis, e.g. Drive 1, and then select “OK”.



Type	Name	Comment
(none)	(none)	
CANopen DS402, EtherCAT CoE	Drive 1 (Unidrive M700)	Unidrive M700

17. Click the “Activate Configuration” button and then click “OK” if the “Activate Configuration” dialog appears:



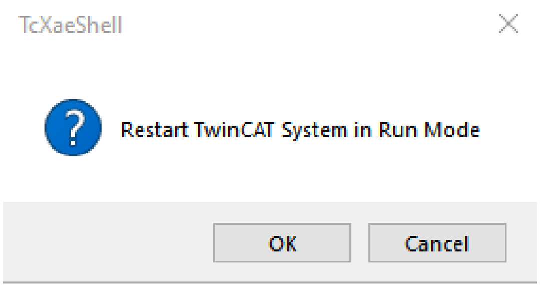
Activate Configuration

Project: MotionProject

Target: CX-3F43E4

☐ Autostart PLC Boot Project(s)

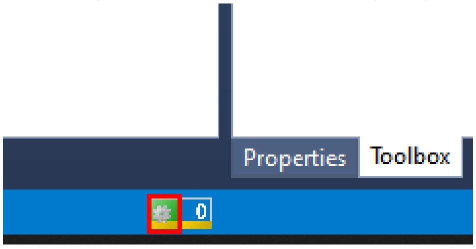
Click “OK” on the “Restart TwinCAT in Run Mode” dialog:



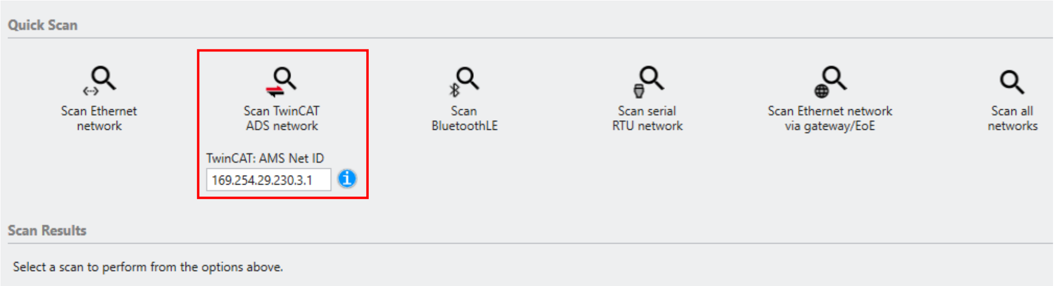
TcXaeShell

Restart TwinCAT System in Run Mode

18. If the system is operating properly the status Icon in the bottom right corner of TwinCAT will show a green box with a rotating cog:



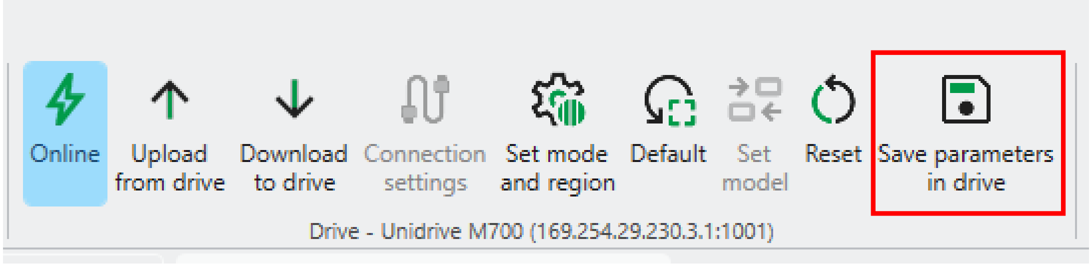
19. Communications will now be running and Connect will be able to search for the drive(s) on the EtherCAT network using the “NetId:” and the “Scan TwinCAT ADS network” Quick Scan function.



TwinCAT: AMS Net ID 169.254.29.230.3.1

Select a scan to perform from the options above.

20. If the EtherCAT firmware has been upgraded from V1 to V2 a save is required to acknowledge the update. Start a project as described in section **2.6 Configure the drive using Connect** and once the project is open, select the drive to save and then click “Save parameters in drive” to complete the process.



3.10 How and when to tune the Position Loop

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

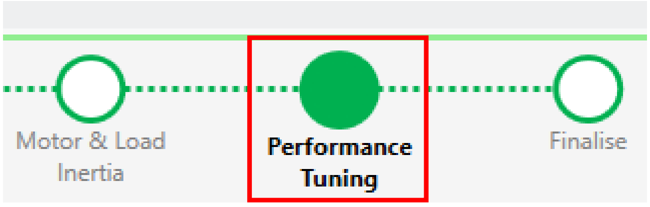
- The drive is enabled by applying 24V to the STO terminal.
- The drive is healthy i.e. no trips.
- MC_Power has been used, successfully, to power on the axis.
- If a keypad is fitted, the display indicates “Run”.
- If Connect is online, the status in the bottom left shows “Run”.

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.

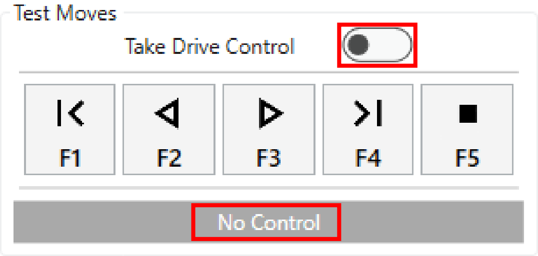
In many cases the default position loop p gain of 25.00 will give reasonable performance, but in cases where there are tight tolerances, the position loop performance can be increased to suit the requirement.

Before attempting to tune the position loop the speed loop must be tuned first as described in section **2.7 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



<u>Don’t</u> 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.

Time/ Div(s): 1

M03.002 Speed Feedback
M03.003 Speed Error
M03.001 Final Speed Reference
M39.008 AMC Position Error
M04.020 Percentage Load
M05.005 D.c. Bus Voltage
Click to select a parameter

0Hz — 100 Hz — 200 Hz

Speed loop P gain 0.0141 s/rad
Speed loop I gain 1.79 1/rad
Motor Inertia 0.00013 kgm²
Enable inertia compensation? ☐ Off

0 — 100

AMC Position Control Loop Kp Gain 25.000
AMC Position Loop Speed Clamp 300 rpm

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

Time/ Div(s): 1

M03.002 Speed Feedback
M03.003 Speed Error
M03.001 Final Speed Reference
M39.008 AMC Position Error
M04.020 Percentage Load
M05.005 D.c. Bus Voltage
Click to select a parameter

0Hz — 100 Hz — 200 Hz

Speed loop P gain 0.0141 s/rad
Speed loop I gain 1.79 1/rad
Motor Inertia 0.00013 kgm²
Enable inertia compensation? ☐ Off

0 — 100

AMC Position Control Loop Kp Gain 25.000
AMC Position Loop Speed Clamp 300 rpm

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.

> **HINT:** When observing the AMC Position Following Error (UU) note that the error shown is in encoder counts.

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

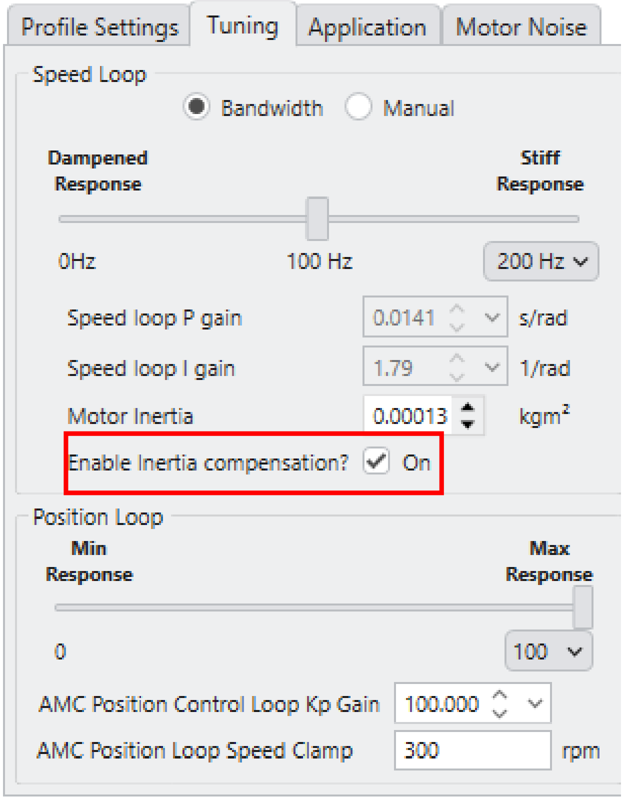
3.11 How and when to use inertia compensation

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 it is best not to use it.

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 **2.7 Commission the drive and motor using Connect** and **3.10 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.



Speed Loop

● Bandwidth ○ Manual

200 Hz

Speed loop P gain 0.0141 s/rad

Speed loop I gain 1.79 1/rad

Motor Inertia 0.00013 kgm^2

Enable Inertia compensation? ☑ On

Position Loop

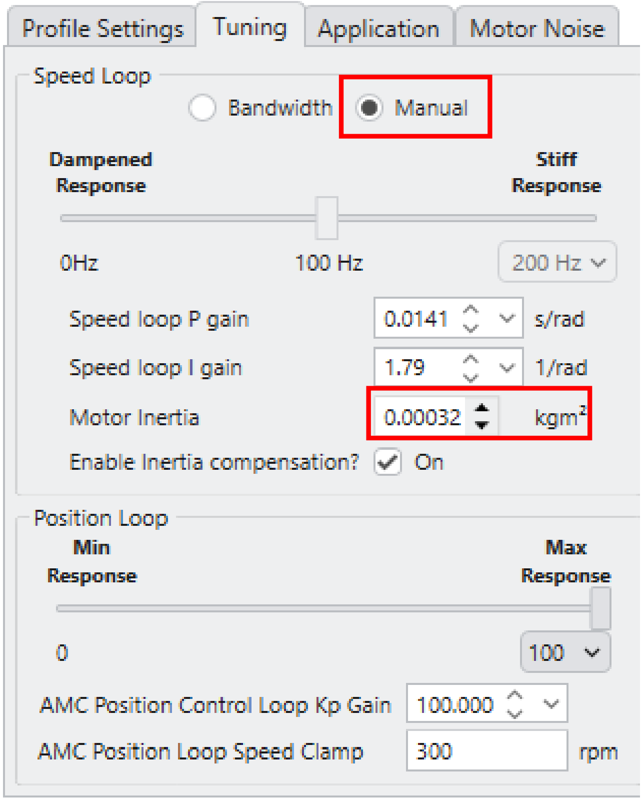
100

AMC Position Control Loop Kp Gain 100.000

AMC Position Loop Speed Clamp 300 rpm

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.



Speed Loop

○ Bandwidth ● Manual

Speed loop P gain 0.0141 s/rad

Speed loop I gain 1.79 1/rad

Motor Inertia 0.00032 kgm²

Enable Inertia compensation? ☑ On

Position Loop

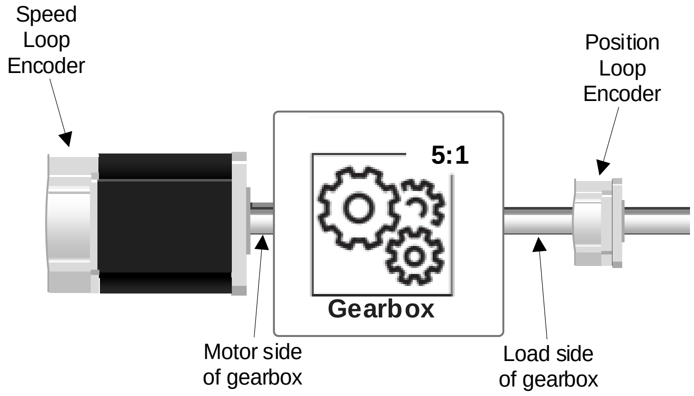
AMC Position Control Loop Kp Gain 100.000

AMC Position Loop Speed Clamp 300 rpm

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 will be calculated incorrectly.

3.12 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 mechanics driving / attached to the load as opposed to the motor side encoder.



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

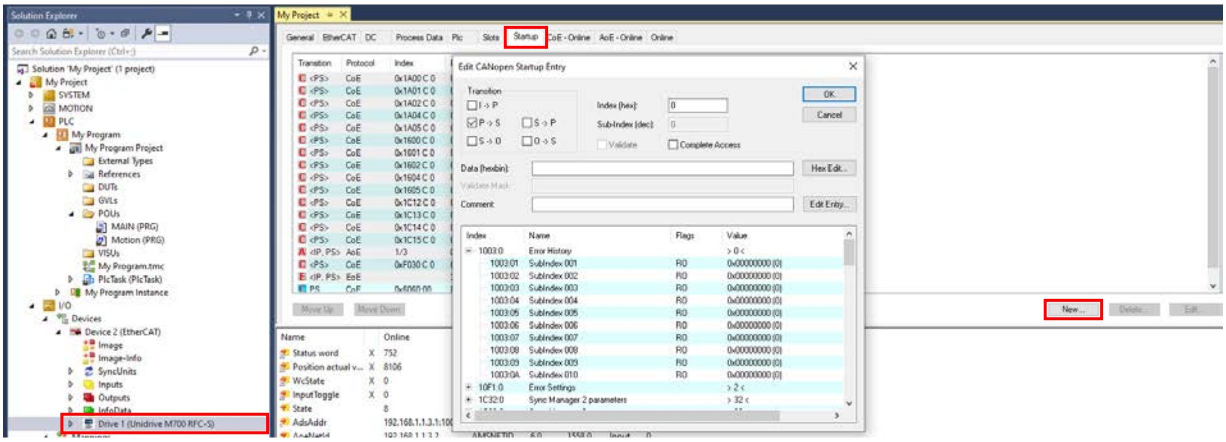
For this example, it is assumed that the motor encoder will be connected the drives P1 interface, whilst the position feedback encoder will be connected to the drives P2 interface. Therefore, we need to configure the following CANopen over EtherCAT (CoE) objects:

- 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 cause the configuration of these CoE objects to happen when the PLC / IPC / Controller is first powered on.

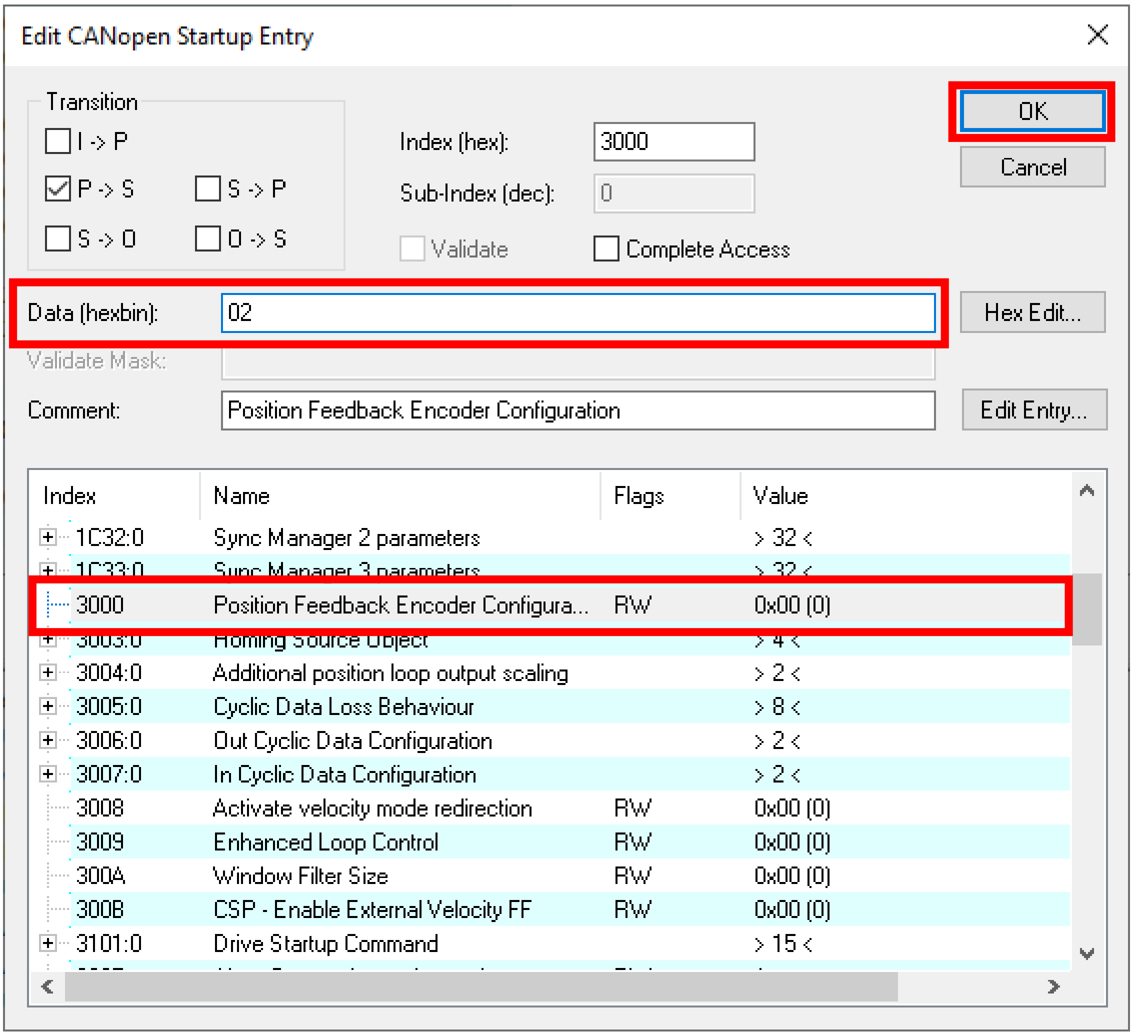
1. Follow the relevant steps outlined in section **2 Getting Started** to configure a TwinCAT project that contains the CT NC Axis Interface Library, an EtherCAT network configured with a single Control Techniques drive and a call / instance of MC_CT_PositionAxis linked to the associated NC Axis.
2. Ensure config mode is selected as indicated by the blue config button being highlighted. Click the blue Config button if it is not highlighted.

3. From the project tree, double click on “Drive1” and select the “Startup” tab followed by the “New…” button



4. The “Edit CANopen Startup Entry” dialog will be displayed.

   From the list of available mappings, select “3000 Position Feedback Encoder Configuration”, enter “02” into the Data field and click “OK”.



Edit CANopen Startup Entry

Transition

☐ I -> P

☑ P -> S ☐ S -> P

☐ S -> O ☐ O -> S

Index (hex): 3000

Sub-Index (dec): 0

☐ Validate ☐ Complete Access

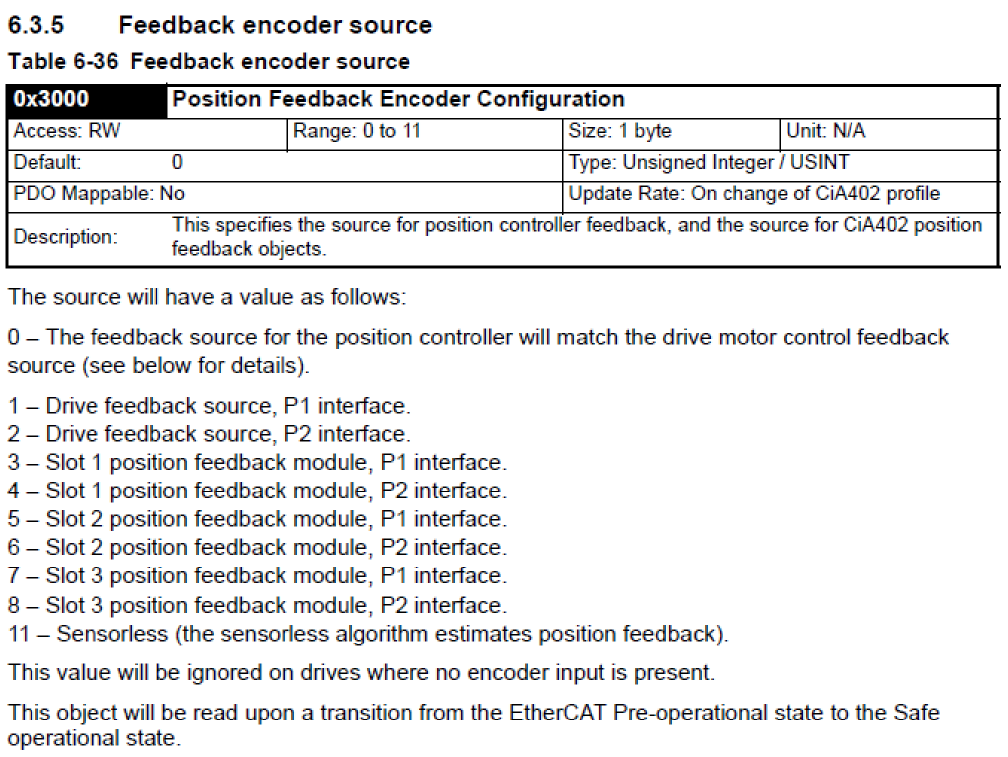
Data (hexbin): 02

Validate Mask:

Comment: Position Feedback Encoder Configuration

Index	Name	Flags	Value
1C32:0	Sync Manager 2 parameters		> 32 <
1C33:0	Sync Manager 3 parameters		> 32 <
3000	Position Feedback Encoder Configura...	RW	0x00 (0)
3003:0	Homing Source Object		> 4 <
3004:0	Additional position loop output scaling		> 2 <
3005:0	Cyclic Data Loss Behaviour		> 8 <
3006:0	Out Cyclic Data Configuration		> 2 <
3007:0	In Cyclic Data Configuration		> 2 <
3008	Activate velocity mode redirection	RW	0x00 (0)
3009	Enhanced Loop Control	RW	0x00 (0)
300A	Window Filter Size	RW	0x00 (0)
300B	CSP - Enable External Velocity FF	RW	0x00 (0)
3101:0	Drive Startup Command		> 15 <

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.



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).

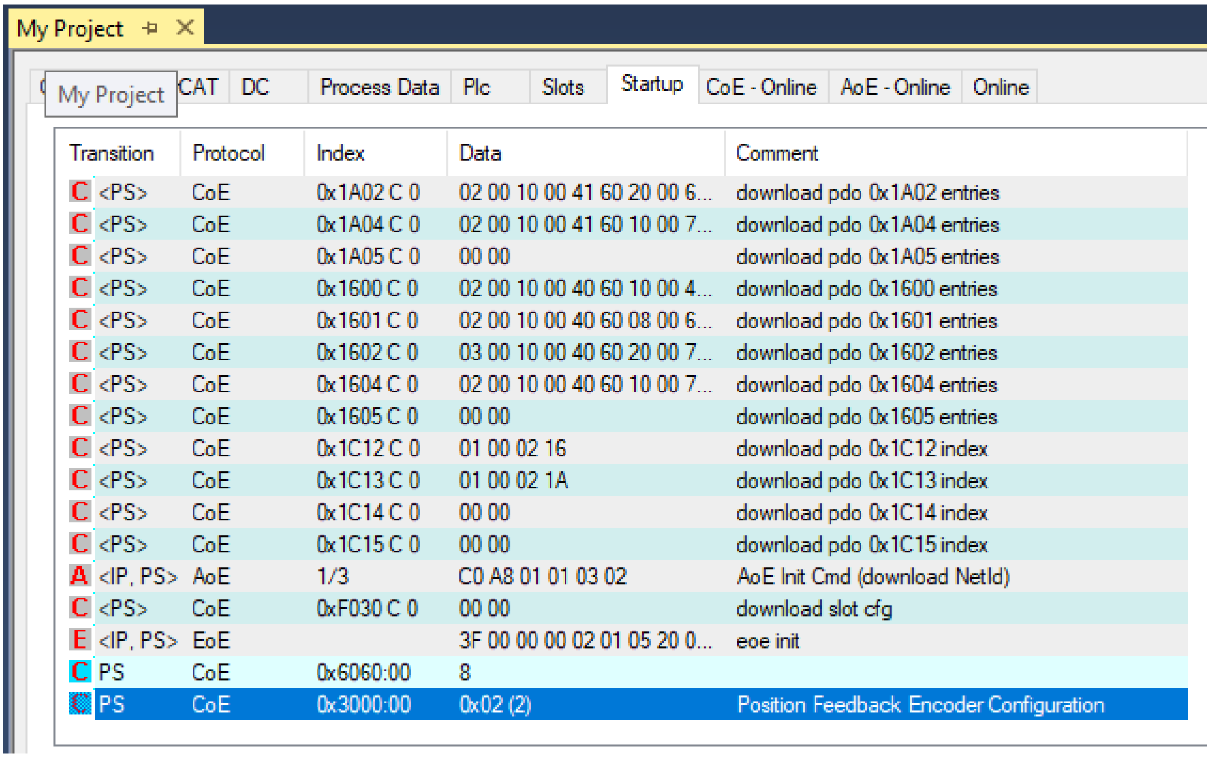
1 – Drive feedback source, P1 interface.
2 – Drive feedback source, P2 interface.
3 – Slot 1 position feedback module, P1 interface.
4 – Slot 1 position feedback module, P2 interface.
5 – Slot 2 position feedback module, P1 interface.
6 – Slot 2 position feedback module, P2 interface.
7 – Slot 3 position 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.

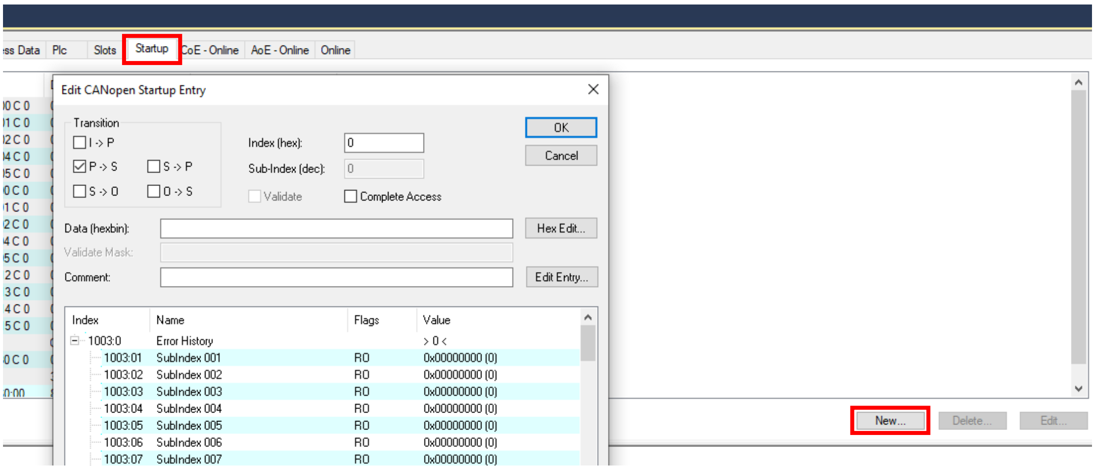
This list is an extract from Section 6.3.5 Feedback encoder source taken from the SI-EtherCAT User Guide.

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



Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1A02 C 0	02 00 10 00 41 60 20 00 6...	download pdo 0x1A02 entries
C <PS>	CoE	0x1A04 C 0	02 00 10 00 41 60 10 00 7...	download pdo 0x1A04 entries
C <PS>	CoE	0x1A05 C 0	00 00	download pdo 0x1A05 entries
C <PS>	CoE	0x1600 C 0	02 00 10 00 40 60 10 00 4...	download pdo 0x1600 entries
C <PS>	CoE	0x1601 C 0	02 00 10 00 40 60 08 00 6...	download pdo 0x1601 entries
C <PS>	CoE	0x1602 C 0	03 00 10 00 40 60 20 00 7...	download pdo 0x1602 entries
C <PS>	CoE	0x1604 C 0	02 00 10 00 40 60 10 00 7...	download pdo 0x1604 entries
C <PS>	CoE	0x1605 C 0	00 00	download pdo 0x1605 entries
C <PS>	CoE	0x1C12 C 0	01 00 02 16	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	01 00 02 1A	download pdo 0x1C13 index
C <PS>	CoE	0x1C14 C 0	00 00	download pdo 0x1C14 index
C <PS>	CoE	0x1C15 C 0	00 00	download pdo 0x1C15 index
A <IP, PS>	AoE	1/3	C0 A8 01 01 03 02	AoE Init Cmd (download NetId)
C <PS>	CoE	0xF030 C 0	00 00	download slot cfg
E <IP, PS>	EoE		3F 00 00 00 02 01 05 20 0...	eoe init
C PS	CoE	0x6060:00	8	
C PS	CoE	0x3000:00	0x02 (2)	Position Feedback Encoder Configuration

6. From the “Startup” view, select the “New…” button.



Index (hex): 0

Sub-Index (dec): 0

Data (hexbin):

Validate Mask:

Comment:

Index	Name	Flags	Value
1003:0	Error History		> 0 <
1003:01	SubIndex 001	RO	0x00000000 (0)
1003:02	SubIndex 002	RO	0x00000000 (0)
1003:03	SubIndex 003	RO	0x00000000 (0)
1003:04	SubIndex 004	RO	0x00000000 (0)
1003:05	SubIndex 005	RO	0x00000000 (0)
1003:06	SubIndex 006	RO	0x00000000 (0)
1003:07	SubIndex 007	RO	0x00000000 (0)

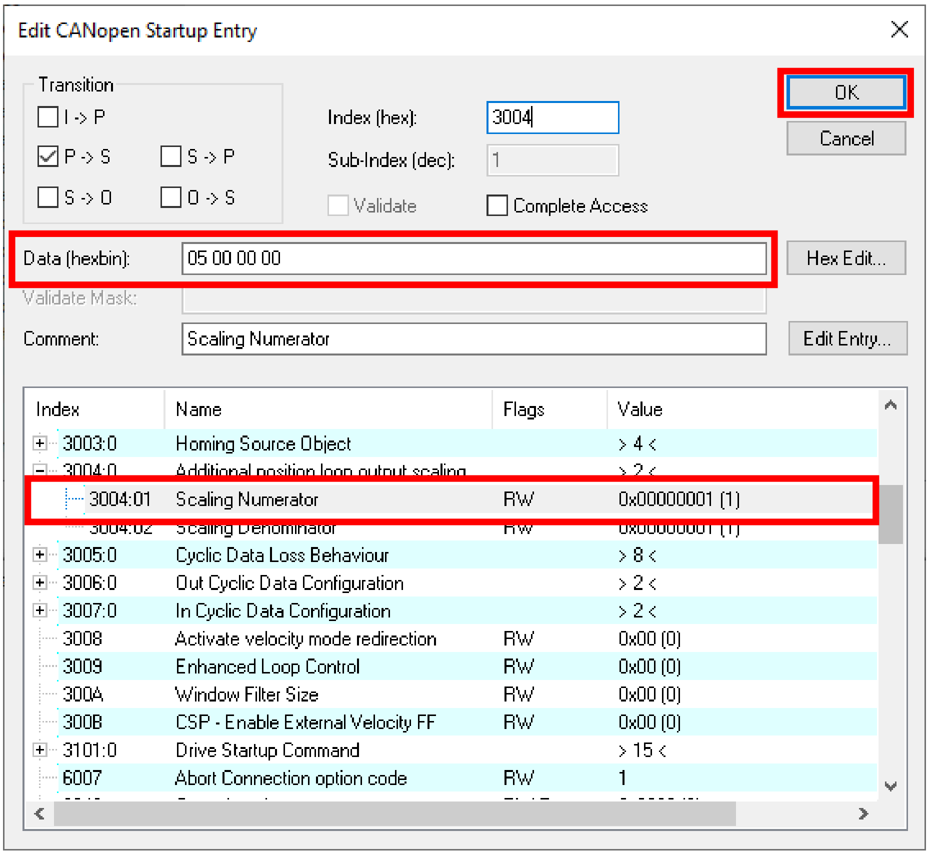
7. The “Edit CANopen Startup Entry” dialog will be displayed.

From the list of available mappings, select “3004 Scaling Numerator”.

The value of the numerator will represent the number of motor revolutions required to achieve the distance of the denominator.

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”.



Index (hex): 3004

Sub-Index (dec): 1

Data (hexbin): 05 00 00 00

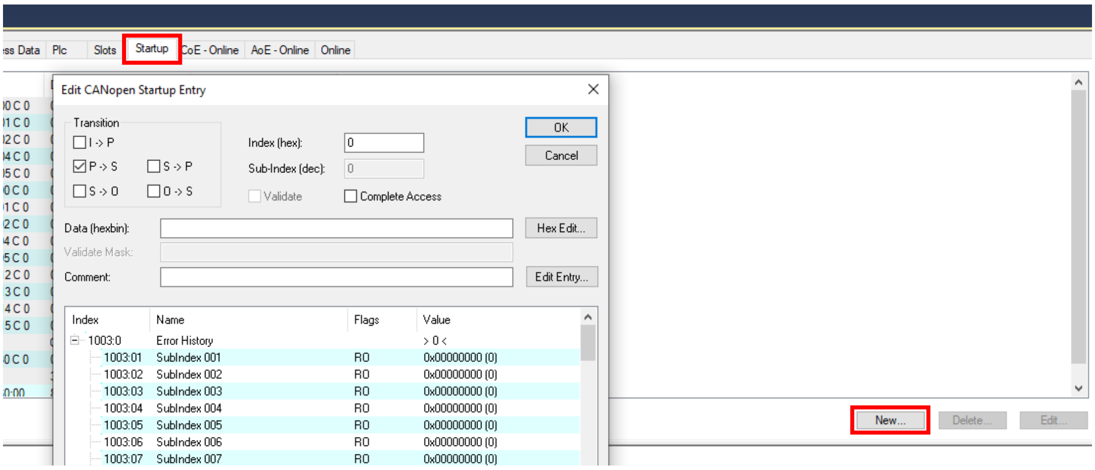
Validate Mask:

Comment: Scaling Numerator

Index	Name	Flags	Value
3003:0	Homing Source Object		> 4 <
3004:0	Additional position loop output scaling		> 2 <
3004:01	Scaling Numerator	RW	0x00000001 (1)
3004:02	Scaling Denominator	RW	0x00000001 (1)
3005:0	Cyclic Data Loss Behaviour		> 8 <
3006:0	Out Cyclic Data Configuration		> 2 <
3007:0	In Cyclic Data Configuration		> 2 <
3008	Activate velocity mode redirection	RW	0x00 (0)
3009	Enhanced Loop Control	RW	0x00 (0)
300A	Window Filter Size	RW	0x00 (0)
300B	CSP - Enable External Velocity FF	RW	0x00 (0)
3101:0	Drive Startup Command		> 15 <
6007	Abort Connection option code	RW	1

HINT: The values specified in the “Data (hexbin)” are in the hex format, e.g. hex 14 = decimal 20 etc.
“Data” is entered in reverse endian format where the least significant bytes are entered first followed by next most significant and so on. This is a quirk of the TwinCAT Start up list editor.
E.g. conventional format hex value is 0x 12 34 56 78, however in reverse endian format it is entered as 0x 78 56 34 12.

8. From the “Startup” object view, select the “New” button.



Edit CANopen Startup Entry

Transition: ☐ I -> P ☑ P -> S ☐ S -> P ☐ S -> O ☐ O -> S

Index (hex): 0

Sub-Index (dec): 0

☐ Validate ☐ Complete Access

Data (hexbin):

Comment:

Index	Name	Flags	Value
1003:0	Error History		> 0 <
1003:01	SubIndex 001	RO	0x00000000 (0)
1003:02	SubIndex 002	RO	0x00000000 (0)
1003:03	SubIndex 003	RO	0x00000000 (0)
1003:04	SubIndex 004	RO	0x00000000 (0)
1003:05	SubIndex 005	RO	0x00000000 (0)
1003:06	SubIndex 006	RO	0x00000000 (0)
1003:07	SubIndex 007	RO	0x00000000 (0)

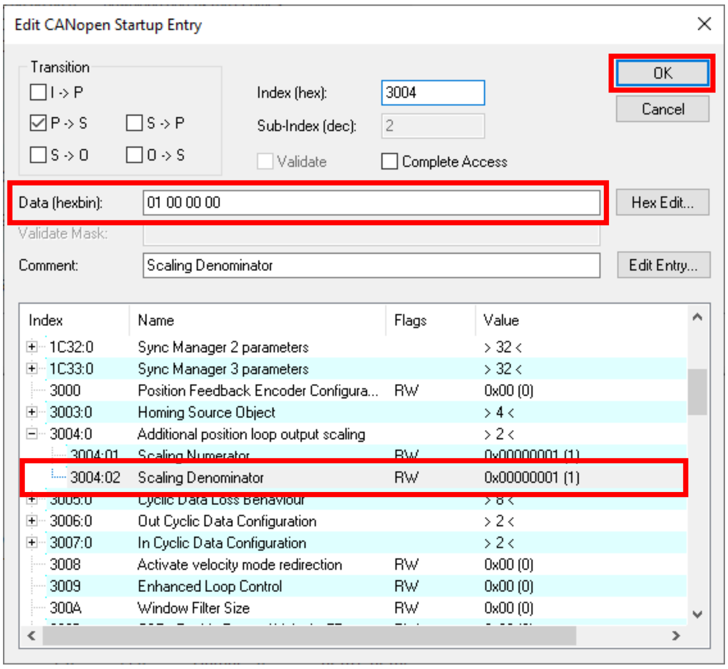
9. The “Edit CANopen Startup Entry” dialog will be displayed.

   From the list of available mappings, select “3004 Scaling Denominator”.

   The value of the denominator will represent the revolutions achieved per number of motor revolutions as specified by the numerator.

   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 1 should be entered into the denominator “Value” field.

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



Edit CANopen Startup Entry

Transition: ☐ I -> P ☑ P -> S ☐ S -> P ☐ S -> O ☐ O -> S

Index (hex): 3004

Sub-Index (dec): 2

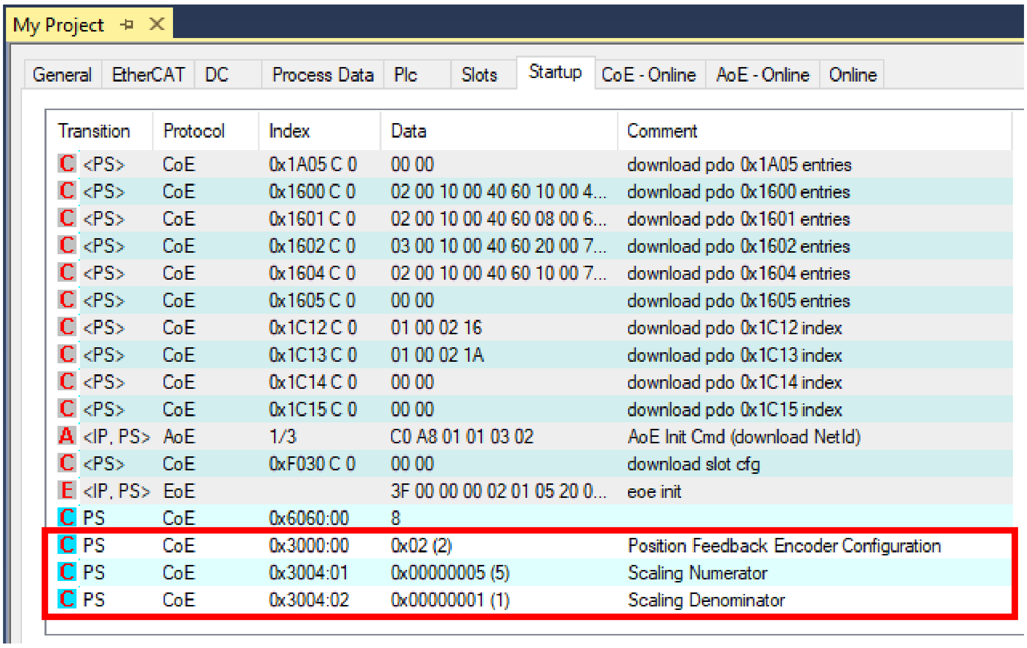
☐ Validate ☐ Complete Access

Data (hexbin): 01 00 00 00

Comment: Scaling Denominator

Index	Name	Flags	Value
1C32:0	Sync Manager 2 parameters		> 32 <
1C33:0	Sync Manager 3 parameters		> 32 <
3000	Position Feedback Encoder Configura...	RW	0x00 (0)
3003:0	Homing Source Object		> 4 <
3004:0	Additional position loop output scaling		> 2 <
3004:01	Scaling Numerator	RW	0x00000001 (1)
3004:02	Scaling Denominator	RW	0x00000001 (1)
3005:0	Cyclic Data Loss Behaviour		> 8 <
3006:0	Out Cyclic Data Configuration		> 2 <
3007:0	In Cyclic Data Configuration		> 2 <
3008	Activate velocity mode redirection	RW	0x00 (0)
3009	Enhanced Loop Control	RW	0x00 (0)
300A	Window Filter Size	RW	0x00 (0)

10. The Scaling Numerator, Scaling Denominator and Position Feedback Encoder Configuration mappings will now be present in the “Startup” 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 NC axis associated with this drive.

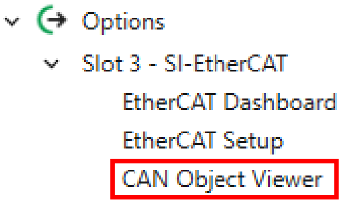


Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1A05 C 0	00 00	download pdo 0x1A05 entries
C <PS>	CoE	0x1600 C 0	02 00 10 00 40 60 10 00 4...	download pdo 0x1600 entries
C <PS>	CoE	0x1601 C 0	02 00 10 00 40 60 08 00 6...	download pdo 0x1601 entries
C <PS>	CoE	0x1602 C 0	03 00 10 00 40 60 20 00 7...	download pdo 0x1602 entries
C <PS>	CoE	0x1604 C 0	02 00 10 00 40 60 10 00 7...	download pdo 0x1604 entries
C <PS>	CoE	0x1605 C 0	00 00	download pdo 0x1605 entries
C <PS>	CoE	0x1C12 C 0	01 00 02 16	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	01 00 02 1A	download pdo 0x1C13 index
C <PS>	CoE	0x1C14 C 0	00 00	download pdo 0x1C14 index
C <PS>	CoE	0x1C15 C 0	00 00	download pdo 0x1C15 index
A <IP, PS>	AoE	1/3	C0 A8 01 01 03 02	AoE Init Cmd (download NetId)
C <PS>	CoE	0xF030 C 0	00 00	download slot cfg
E <IP, PS>	EoE		3F 00 00 00 02 01 05 20 0...	eoe init
C PS	CoE	0x6060:00	8	
C PS	CoE	0x3000:00	0x02 (2)	Position Feedback Encoder Configuration
C PS	CoE	0x3004:01	0x00000005 (5)	Scaling Numerator
C PS	CoE	0x3004:02	0x00000001 (1)	Scaling Denominator

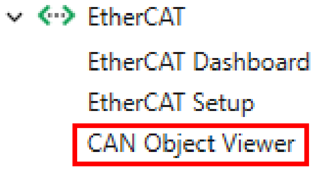
3.13 How to view CAN objects

Connect has a powerful diagnostic aid to help the user view CANopen object live called the CAN Object Viewer. This helps the user to diagnose setup and operational issues with the CiA402 system layer upon which EtherCAT and NC axes run. If the TwinCAT 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.

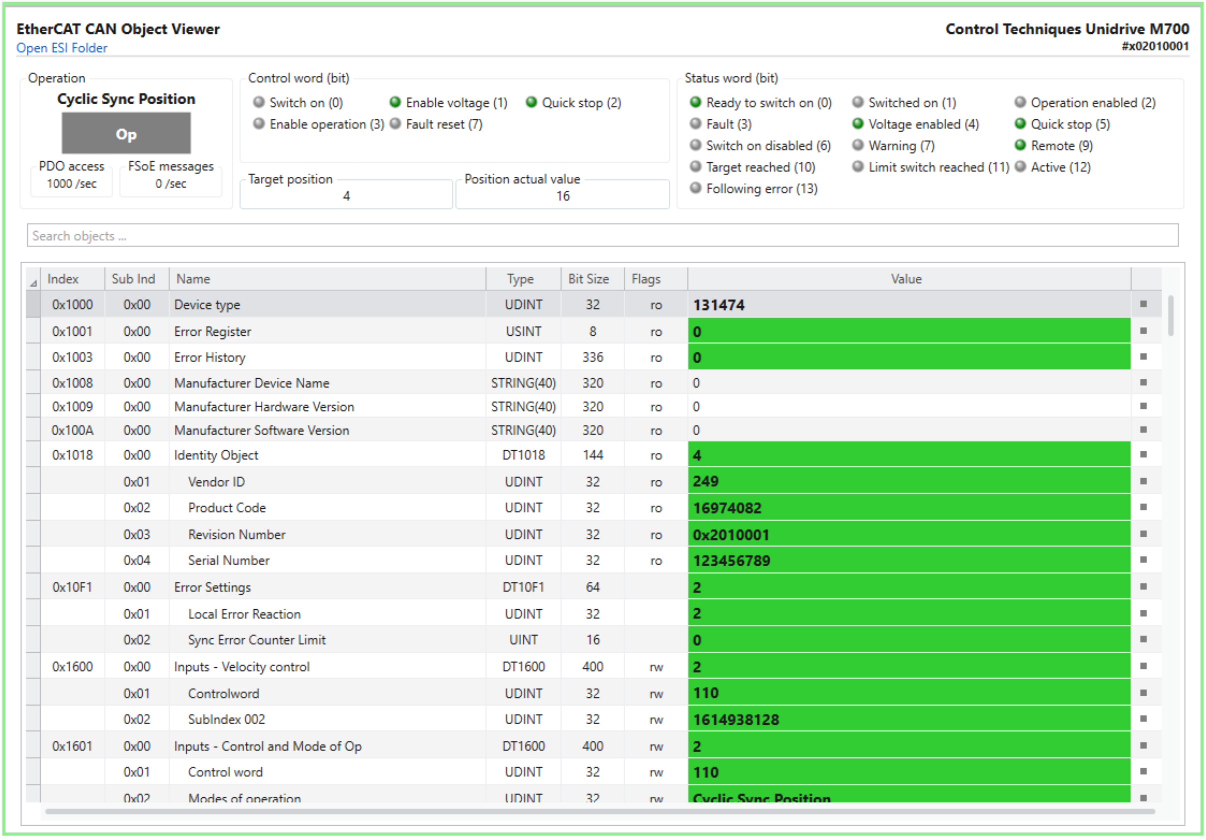
The tool is found in the Drive explorer tree under "Options" > "Slot[n]" > "CAN Object Viewer" where [n] is the drive slot in which the SI-EtherCAT interface is located.



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:



EtherCAT CAN Object Viewer
Open ESI Folder

Control Techniques Unidrive M700
#x02010001

Operation
Cyclic Sync Position
Op
PDO access: 1000 /sec
FSoE messages: 0 /sec

Control word (bit)
○ Switch on (0) ● Enable voltage (1) ● Quick stop (2)
○ Enable operation (3) ○ Fault reset (7)

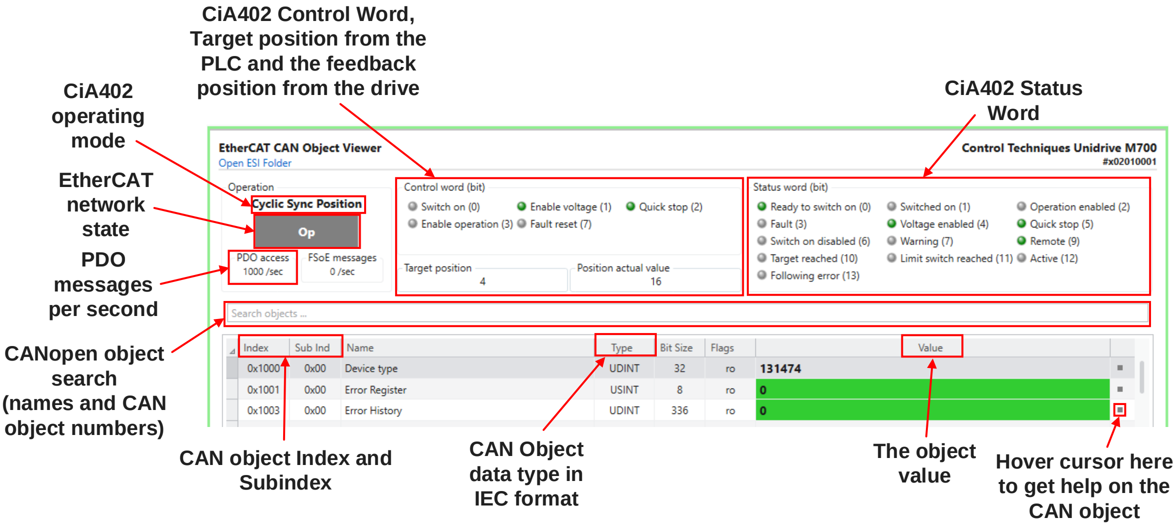
Target position: 4
Position actual value: 16

Status word (bit)
● Ready to switch on (0) ○ Switched on (1) ○ Operation enabled (2)
○ Fault (3) ● Voltage enabled (4) ● Quick stop (5)
○ Switch on disabled (6) ○ Warning (7) ● Remote (9)
○ Target reached (10) ○ Limit switch reached (11) ○ Active (12)
○ Following error (13)

Search objects ...

Index	Sub Ind	Name	Type	Bit Size	Flags	Value
0x1000	0x00	Device type	UDINT	32	ro	**131474**
0x1001	0x00	Error Register	USINT	8	ro	**0**
0x1003	0x00	Error History	UDINT	336	ro	**0**
0x1008	0x00	Manufacturer Device Name	STRING(40)	320	ro	0
0x1009	0x00	Manufacturer Hardware Version	STRING(40)	320	ro	0
0x100A	0x00	Manufacturer Software Version	STRING(40)	320	ro	0
0x1018	0x00	Identity Object	DT1018	144	ro	**4**
	0x01	Vendor ID	UDINT	32	ro	**249**
	0x02	Product Code	UDINT	32	ro	**16974082**
	0x03	Revision Number	UDINT	32	ro	**0x2010001**
	0x04	Serial Number	UDINT	32	ro	**123456789**
0x10F1	0x00	Error Settings	DT10F1	64		**2**
	0x01	Local Error Reaction	UDINT	32		**2**
	0x02	Sync Error Counter Limit	UINT	16		**0**
0x1600	0x00	Inputs - Velocity control	DT1600	400	rw	**2**
	0x01	Controlword	UDINT	32	rw	**110**
	0x02	SubIndex 002	UDINT	32	rw	**1614938128**
0x1601	0x00	Inputs - Control and Mode of Op	DT1600	400	rw	**2**
	0x01	Control word	UDINT	32	rw	**110**
	0x02	Modes of operation	UDINT	32	rw	**Cyclic Sync Position**

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



CiA402 operating mode

EtherCAT network state

PDO messages per second

CANopen object search (names and CAN object numbers)

CiA402 Control Word, Target position from the PLC and the feedback position from the drive

CiA402 Status Word

CAN object Index and Subindex

CAN Object data type in IEC format

The object value

Hover cursor here to get help on the CAN object

PDO access 1000 /sec

FSoE messages 0 /sec

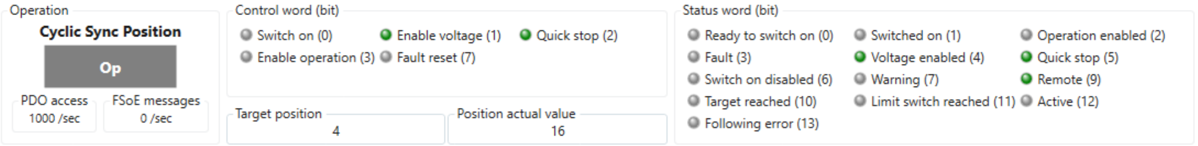
Target position 4

Position actual value 16

Index	Sub Ind	Name	Type	Bit Size	Flags	Value
0x1000	0x00	Device type	UDINT	32	ro	131474
0x1001	0x00	Error Register	USINT	8	ro	0
0x1003	0x00	Error History	UDINT	336	ro	0

3.13.1 MC_Power Disabled and Drive Disabled

This screenshot acts a guide to understand the expected CiA402 Status and Control word for a TwinCAT NC axis with a disabled drive (STO = 0V) and MC_Power.Enable = FALSE.



PDO access 1000 /sec

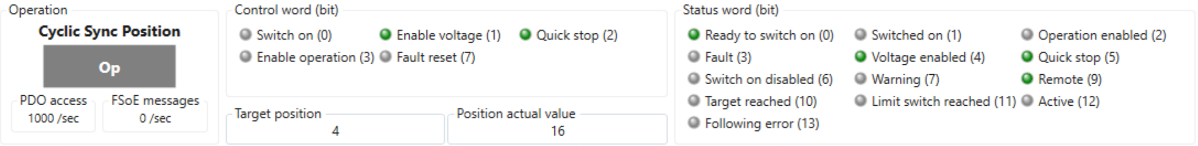
FSoE messages 0 /sec

Target position 4

Position actual value 16

3.13.2 MC_Power Disabled and Drive Enabled

This screenshot acts a guide to understand the expected CiA402 Status and Control word for a TwinCAT NC axis with a disabled drive (STO = 24V) and MC_Power.Enable = FALSE.



PDO access 1000 /sec

FSoE messages 0 /sec

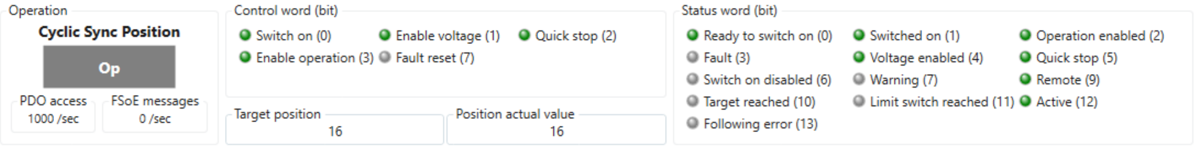
Target position 4

Position actual value 16

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

3.13.3 MC_Power Enabled and Drive Enabled

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



PDO access 1000 /sec

FSoE messages 0 /sec

Target position 16

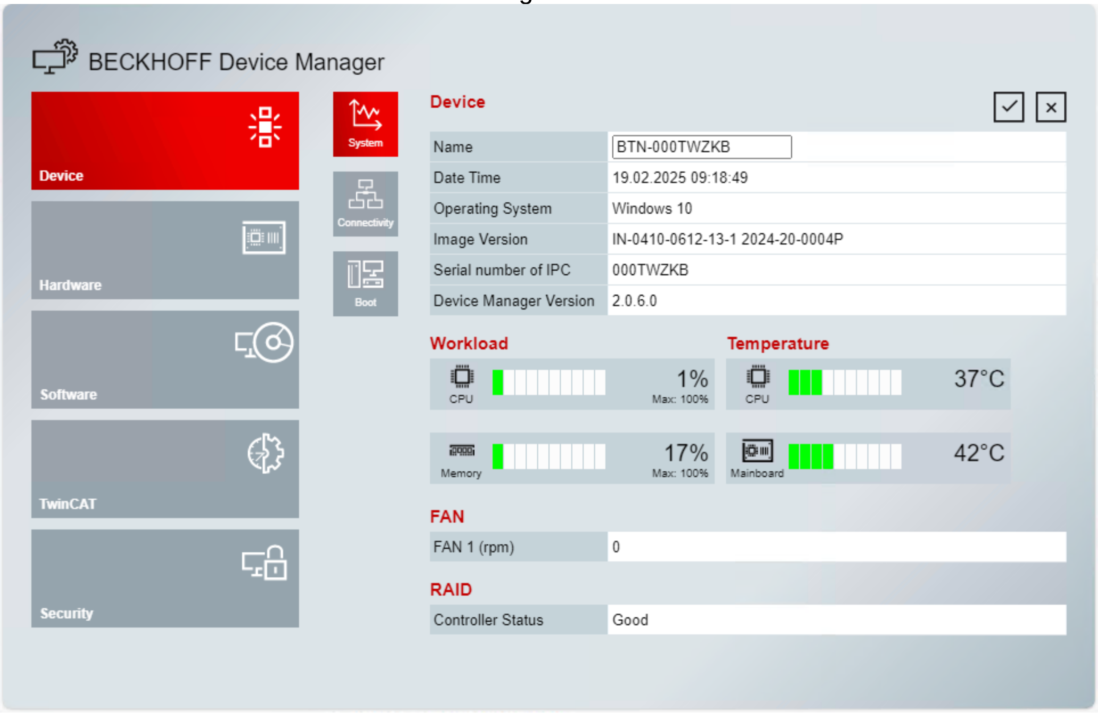
Position actual value 16

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

3.14 How to install TwinCAT Runtime and configure Ethernet ports for EtherCAT use on a BECKHOFF IPC

IPCs from Beckhoff such as C6025 are not supplied with the TwinCAT runtime installed or any EtherCAT port setup. The following steps can be used to turn a Beckhoff IPC into a TwinCAT controller.

1. Apply 24V to the IPC. Follow the user guide for the IPC for the details on the power connector wiring.
2. Plug a monitor, keyboard and mouse into the IPC.
3. After a short wait the Beckhoff Device manager will be shown:



Device

Name	BTN-000TWZKB
Date Time	19.02.2025 09:18:49
Operating System	Windows 10
Image Version	IN-0410-0612-13-1 2024-20-0004P
Serial number of IPC	000TWZKB
Device Manager Version	2.0.6.0

Workload

CPU 1% Max: 100%

Memory 17% Max: 100%

Temperature

CPU 37°C

Mainboard 42°C

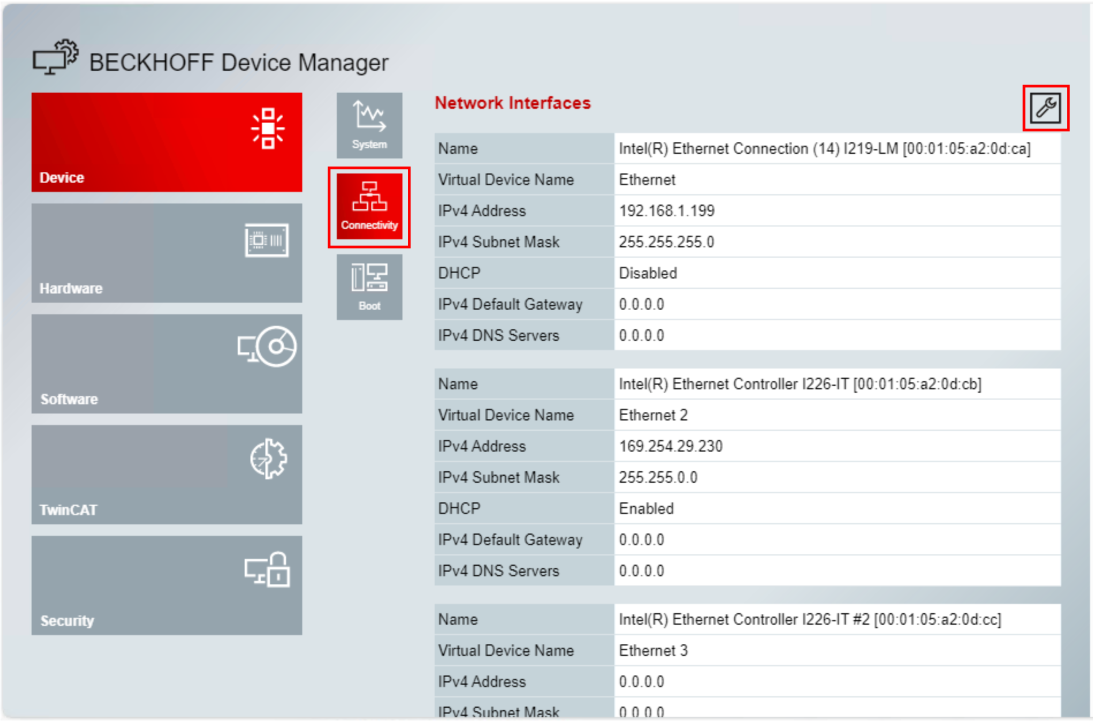
FAN

FAN 1 (rpm)	0

RAID

Controller Status	Good

4. Select Connectivity and then click the spanner icon on the top right of the Device Manager to configure a static IP address:



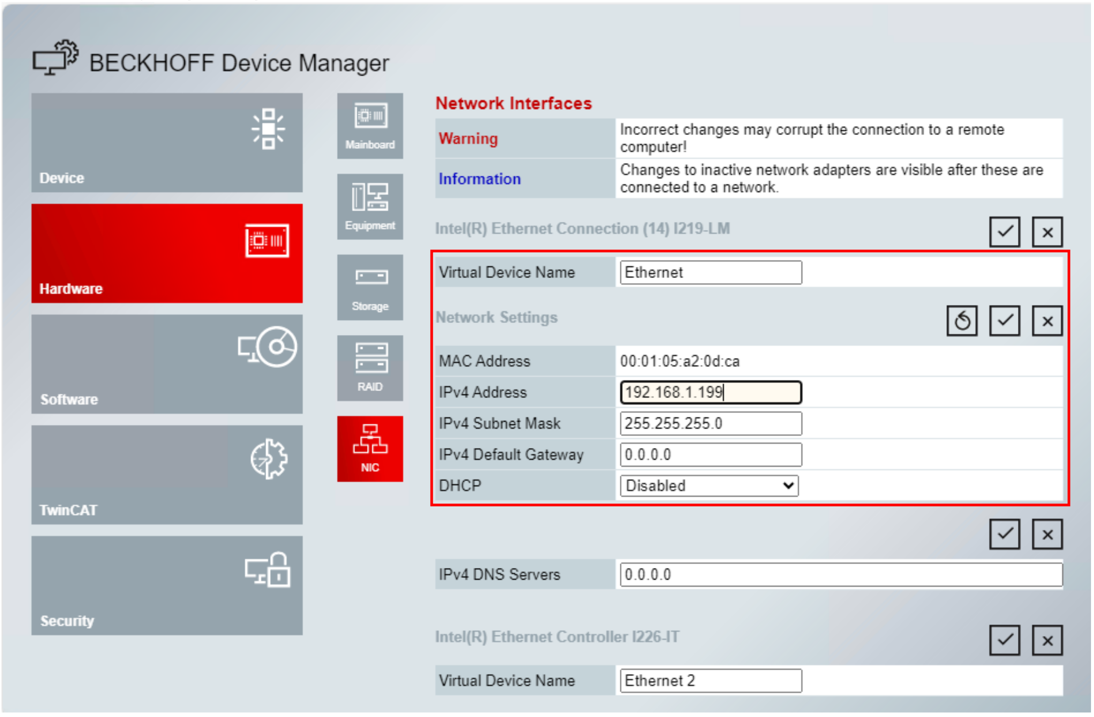
Network Interfaces

Name	Intel(R) Ethernet Connection (14) I219-LM [00:01:05:a2:0d:ca]
Virtual Device Name	Ethernet
IPv4 Address	192.168.1.199
IPv4 Subnet Mask	255.255.255.0
DHCP	Disabled
IPv4 Default Gateway	0.0.0.0
IPv4 DNS Servers	0.0.0.0

Name	Intel(R) Ethernet Controller I226-IT [00:01:05:a2:0d:cb]
Virtual Device Name	Ethernet 2
IPv4 Address	169.254.29.230
IPv4 Subnet Mask	255.255.0.0
DHCP	Enabled
IPv4 Default Gateway	0.0.0.0
IPv4 DNS Servers	0.0.0.0

Name	Intel(R) Ethernet Controller I226-IT #2 [00:01:05:a2:0d:cc]
Virtual Device Name	Ethernet 3
IPv4 Address	0.0.0.0
IPv4 Subnet Mask	0.0.0.0

5. Choose an ethernet port that will become the Ethernet connection to your PC. There is normally 1 port that is slower than the others, this is the best port to use for the PC connection since the high-speed ports should be used for EtherCAT real time communication.



Network Interfaces

Warning	Incorrect changes may corrupt the connection to a remote computer!
Information	Changes to inactive network adapters are visible after these are connected to a network.

Intel(R) Ethernet Connection (14) I219-LM

Virtual Device Name	Ethernet

Network Settings

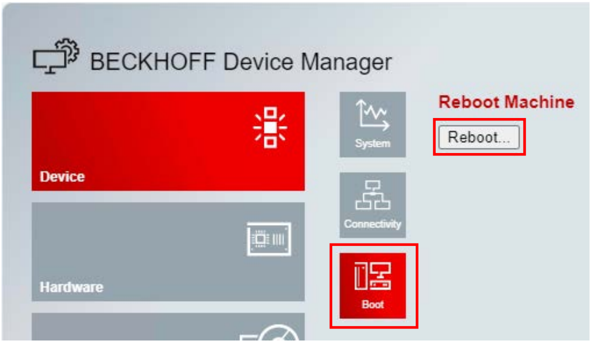
MAC Address	00:01:05:a2:0d:ca
IPv4 Address	192.168.1.199
IPv4 Subnet Mask	255.255.255.0
IPv4 Default Gateway	0.0.0.0
DHCP	Disabled

IPv4 DNS Servers	0.0.0.0

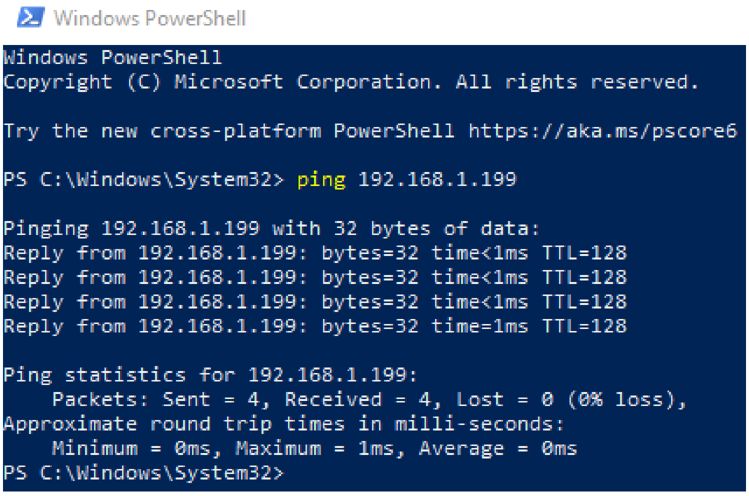
Intel(R) Ethernet Controller I226-IT

Virtual Device Name	Ethernet 2

6. Click the tick ☑ to apply the settings and then reboot.



7. Using the command prompt or PowerShell, ping the new IP address to make sure it is operational. When the command prompt is shown type “ping ” followed by the IP address assigned e.g. “ping 192.168.1.199”. The screen shot below shows a successful ping:

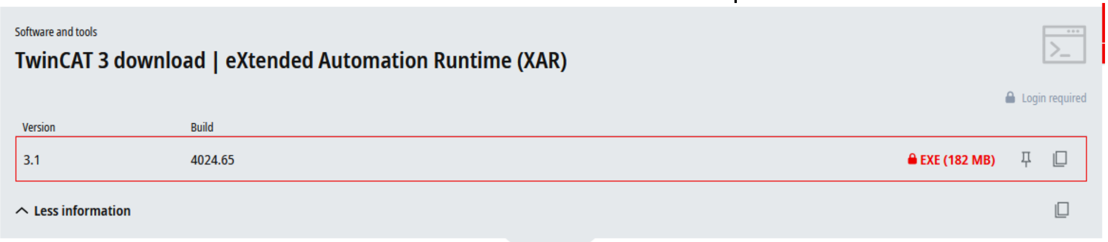


Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\System32> ping 192.168.1.199

Pinging 192.168.1.199 with 32 bytes of data:
Reply from 192.168.1.199: bytes=32 time<1ms TTL=128
Reply from 192.168.1.199: bytes=32 time<1ms TTL=128
Reply from 192.168.1.199: bytes=32 time<1ms TTL=128
Reply from 192.168.1.199: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.199:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Windows\System32>
```

8. From this point onwards, the Beckhoff Device Manager can be accessed using a web browser e.g. type in “https://192.168.1.199/config/”.

9. Get the TwinCAT XAR runtime form the Beckhoff website and put it onto a USB stick:



Software and tools

TwinCAT 3 download | eXtended Automation Runtime (XAR)

Login required

Version	Build	
3.1	4024.65	EXE (182 MB)

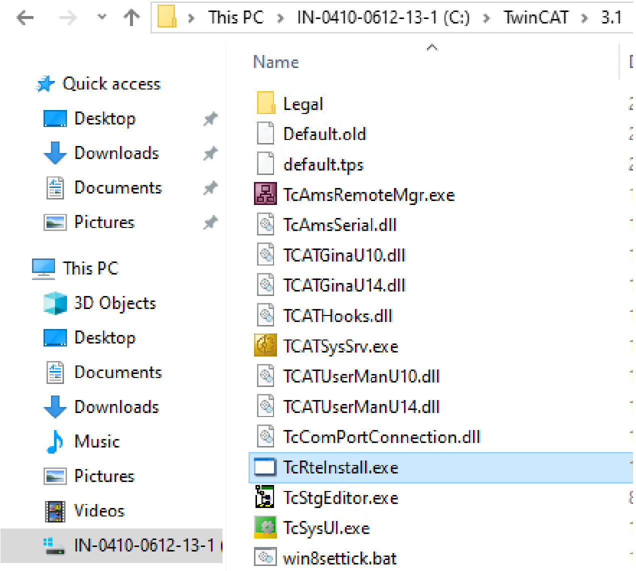
Less information

Put the USB stick with the runtime on it into one of the IPCs USB ports.

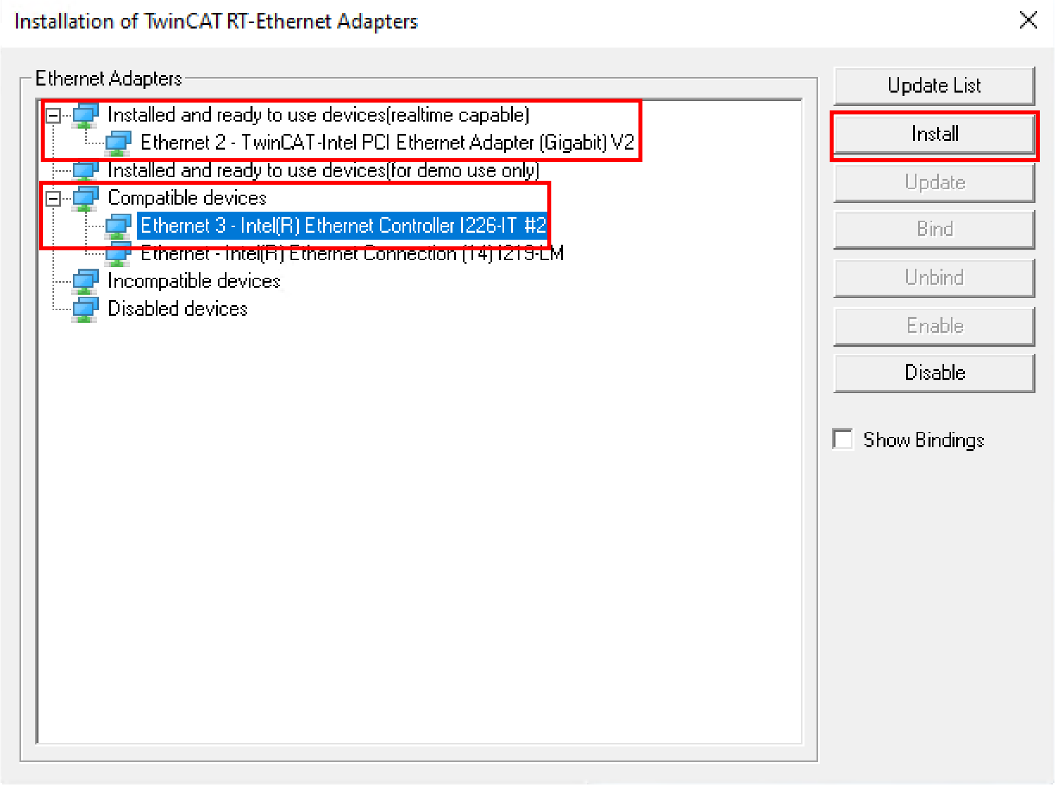
10. Using file explorer on the IPC locate the USB stick, copy the files from the USB to the IPCs documents folder, and then double click on the installer file to start the installation. When the installation completes the IPC must be rebooted.

11. When the IPC boots up the “TC” LED will now be lit up blue.

12. Access the IPC file system using File Explorer and navigate to C:/TwinCAT/3.1/System. In this folder there is a file called “TcRteInstall.exe”. Double click on this to assign the EtherCAT network port on the IPC.

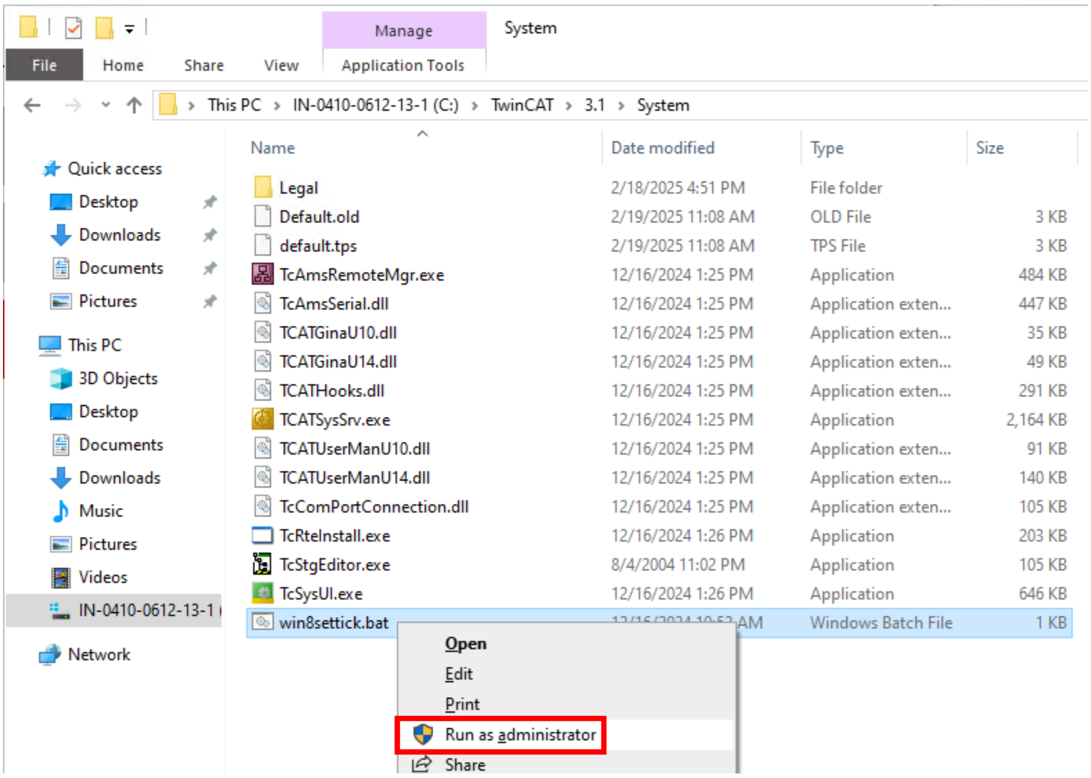


13. When the installer opens, click on the port to become the EtherCAT network port in the “Compatible Devices” list and then click “Install”. When the process is completed the newly installed port will appear in the “Installed and ready to use devices (realtime capable)” list.



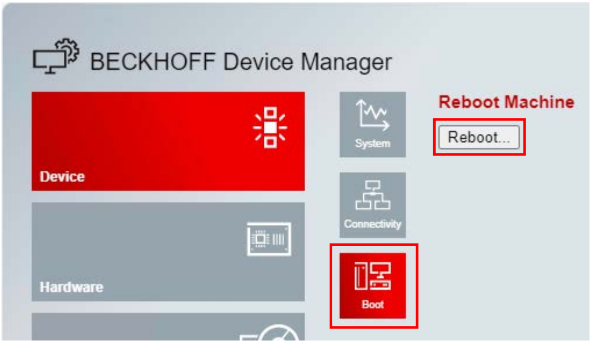
Close the installer when finished.

14. In the same folder location, C:/TwinCAT/3.1/System, run win8settick.bat as an Administrator, (right click on the file select "Run as Administrator…").



Name	Date modified	Type	Size
Legal	2/18/2025 4:51 PM	File folder	
Default.old	2/19/2025 11:08 AM	OLD File	3 KB
default.tps	2/19/2025 11:08 AM	TPS File	3 KB
TcAmsRemoteMgr.exe	12/16/2024 1:25 PM	Application	484 KB
TcAmsSerial.dll	12/16/2024 1:25 PM	Application exten...	447 KB
TCATGinaU10.dll	12/16/2024 1:25 PM	Application exten...	35 KB
TCATGinaU14.dll	12/16/2024 1:25 PM	Application exten...	49 KB
TCATHooks.dll	12/16/2024 1:25 PM	Application exten...	291 KB
TCATSysSrv.exe	12/16/2024 1:25 PM	Application	2,164 KB
TCATUserManU10.dll	12/16/2024 1:25 PM	Application exten...	91 KB
TCATUserManU14.dll	12/16/2024 1:25 PM	Application exten...	140 KB
TcComPortConnection.dll	12/16/2024 1:25 PM	Application exten...	105 KB
TcRteInstall.exe	12/16/2024 1:26 PM	Application	203 KB
TcStgEditor.exe	8/4/2004 11:02 PM	Application	105 KB
TcSysUI.exe	12/16/2024 1:26 PM	Application	646 KB
win8settick.bat	12/16/2024 10:5[illegible] AM	Windows Batch File	1 KB

15. When the batch file completes, reboot the IPC.

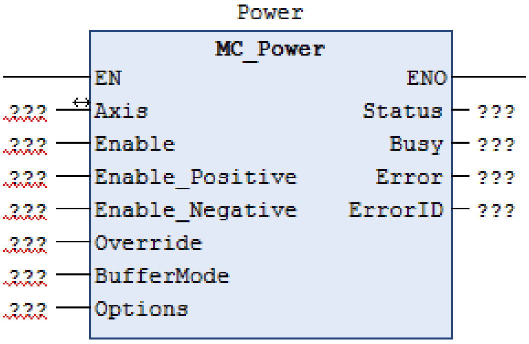


16. The IPC can now be used with TwinCAT.

3.15 How to Remove “???” from LD diagram POU inputs and outputs

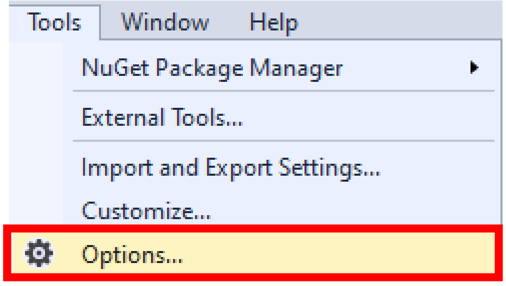
LD diagram language can be a nice way to explore the functionality of the NC axis PLCopen function blocks. Each input and output can be seen side by side, any enumerated values selected, and all output values may be seen.

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.

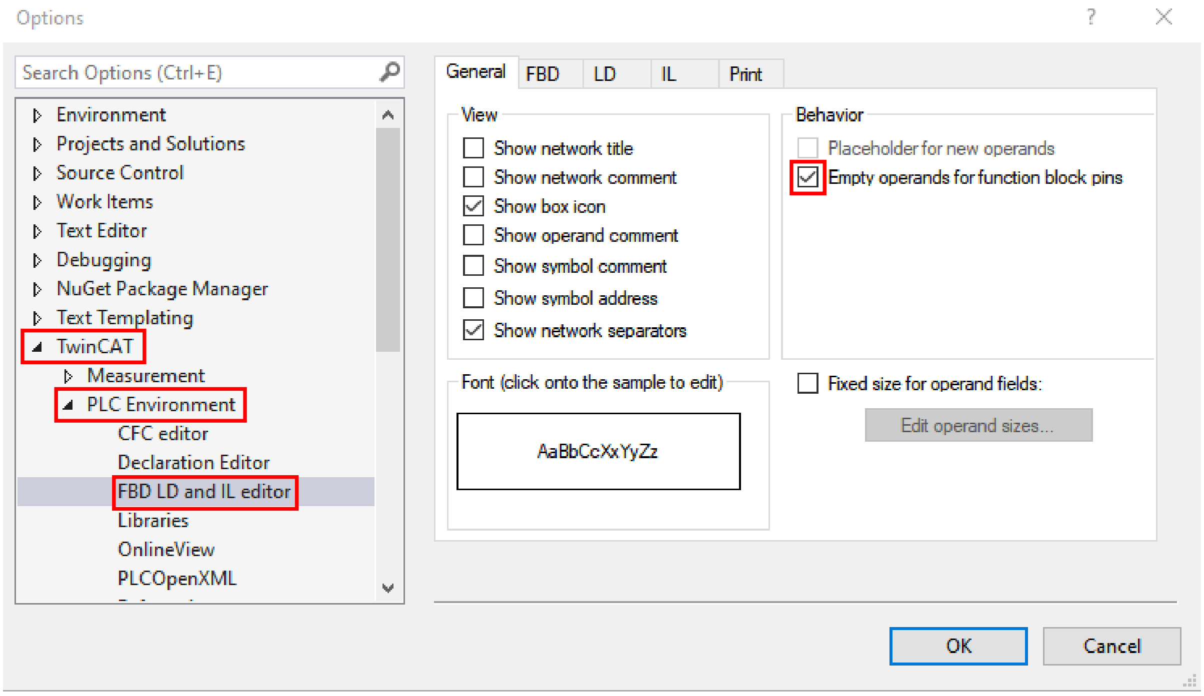


The “???” at the POU’s I/O is optional. The following steps show how to turn this off:

- Go to “Tools” >” Options”
- Automatic generation of these question marks is a feature that can be disabled in the TwinCAT environment itself, to do this, select “Tools” > “Options…” from the main toolbar.



- Go to “TwinCAT” > “PLC Environment” > “FB LD and IL editor”.
- Check the “Empty operands for function block pins”

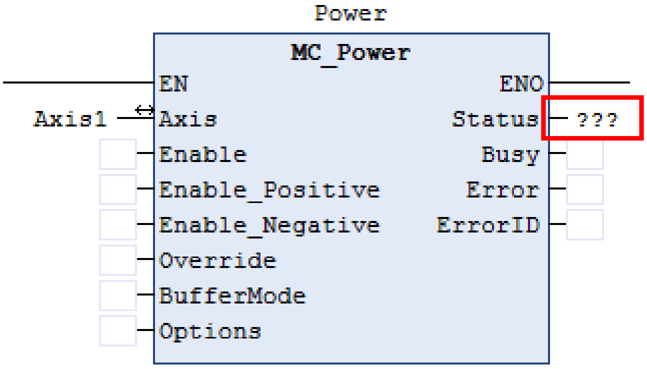


View
☐ Show network title
☐ Show network comment
☑ Show box icon
☐ Show operand comment
☐ Show symbol comment
☐ Show symbol address
☑ Show network separators

Behavior
☐ Placeholder for new operands
☑ Empty operands for function block pins

☐ Fixed size for operand fields:

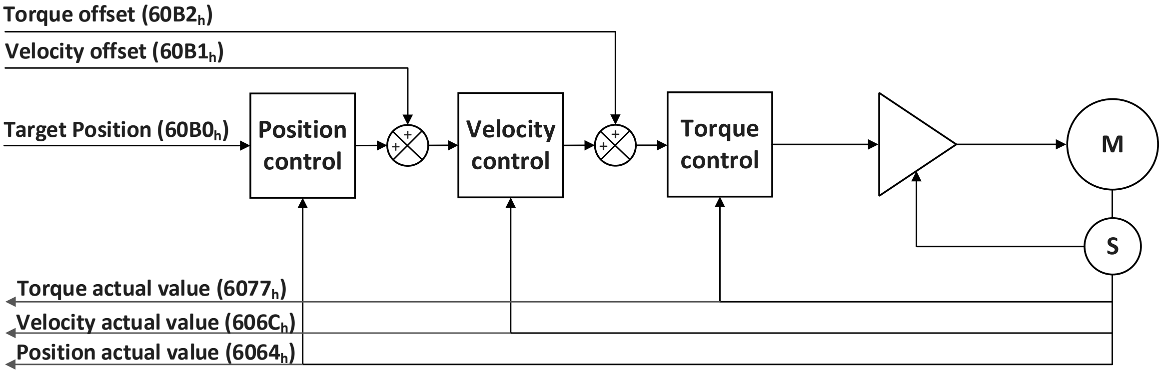
After making this change, when new POUs are inserted to a rung, only the first output has “???” which is much easier to delete:



3.16 How to apply velocity feedforward to an NC axis

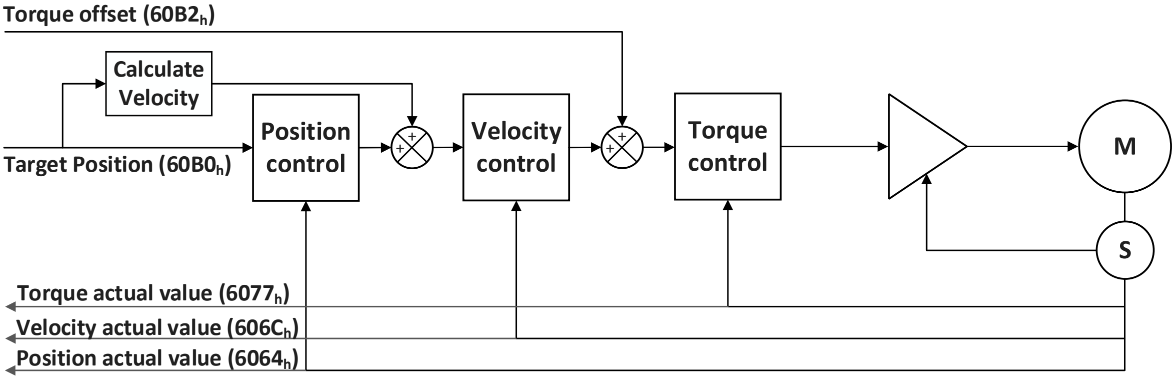
This section relates to velocity feedforward with a NC axis being controlled using EtherCAT and DS402 in CSP (Cyclic Synchronous Position) mode, where the velocity calculated by the NC axis motion profiler is fed forwards directly to the target axis which improves the granularity of the velocity profiling, 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.NcToPlc.SetVelo. 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)

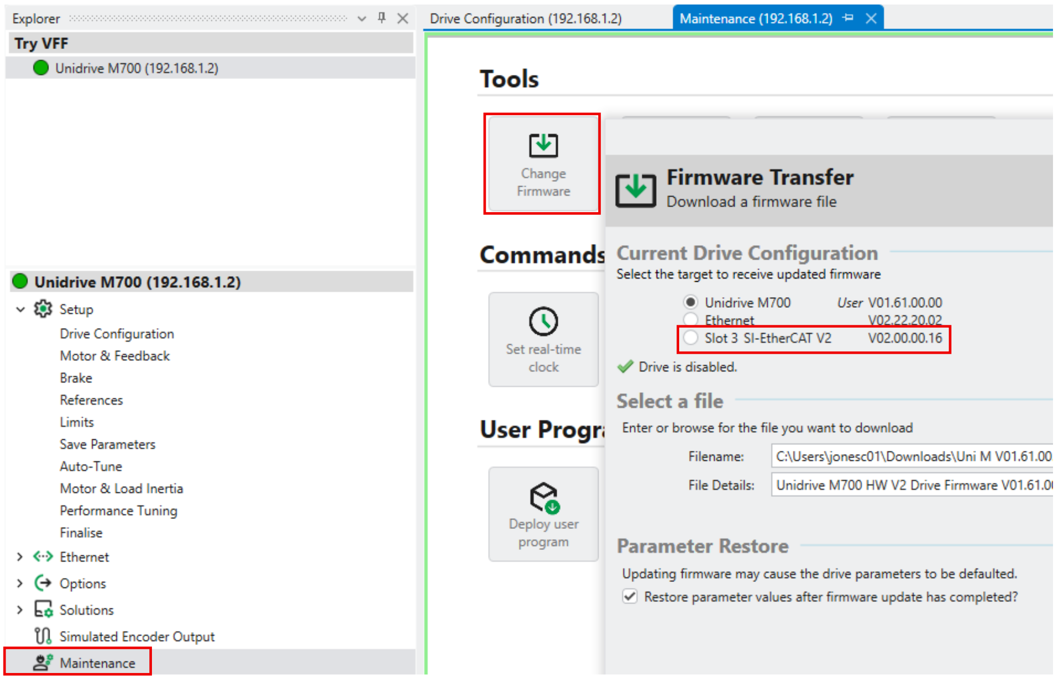
For a normal axis using the default setup i.e. no velocity feedforward implemented, the drive calculates the velocity feedforward term by differentiating the position reference passed into 0x60B0. This tends to be noisier since the velocity granularity is based on whole position units.



(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.



Firmware Transfer
Download a firmware file

Current Drive Configuration
Select the target to receive updated firmware

● Unidrive M700	*User*	V01.61.00.00
○ Ethernet		V02.22.20.02
○ Slot 3 SI-EtherCAT V2		V02.00.00.16

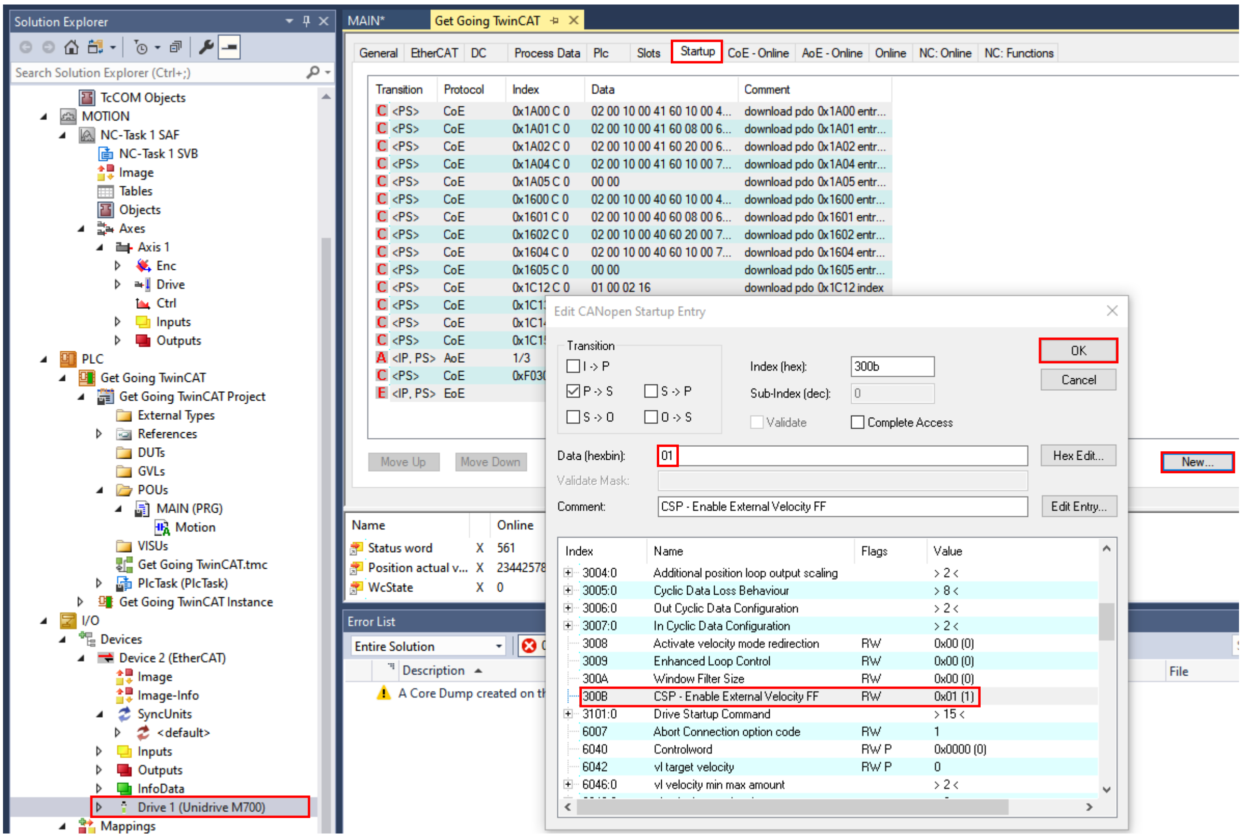
✓ Drive is disabled.

Select a file
Enter or browse for the file you want to download
Filename: C:\Users\jonesc01\Downloads\Uni M V01.61.00
File Details: Unidrive M700 HW V2 Drive Firmware V01.61.0

Parameter Restore
Updating firmware may cause the drive parameters to be defaulted.
☑ Restore parameter values after firmware update has completed?

2. Put TwinCAT in Config mode by clicking on the button.

3. Enable the velocity feedforward term by setting 0x300B =1 in the startup list. Double click on the EtherCAT slave in the “Solution Explorer” tree. Select the “Startup” tab. From the list select the entry that starts with “300B”. When the PDO link boxes update, set “Data” to “01”. Click “OK” when finished.



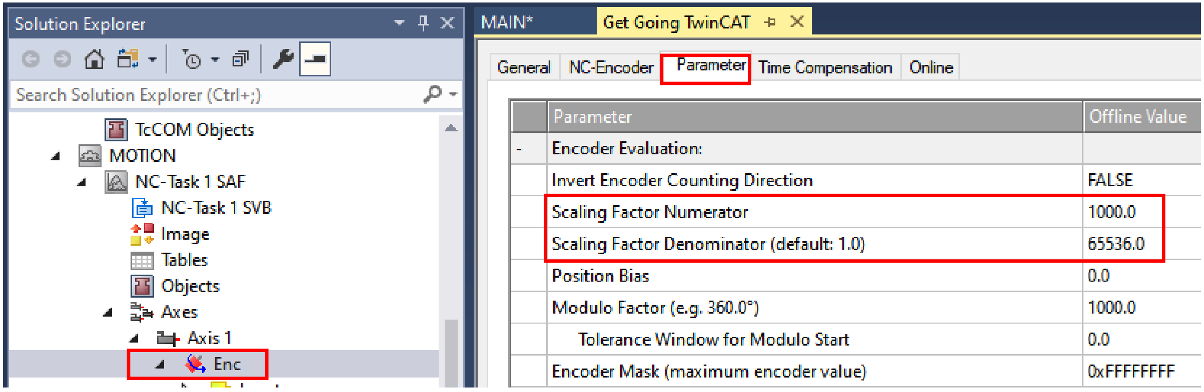
Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1A00 C 0	02 00 10 00 41 60 10 00 4...	download pdo 0x1A00 entr...
C <PS>	CoE	0x1A01 C 0	02 00 10 00 41 60 08 00 6...	download pdo 0x1A01 entr...
C <PS>	CoE	0x1A02 C 0	02 00 10 00 41 60 20 00 6...	download pdo 0x1A02 entr...
C <PS>	CoE	0x1A04 C 0	02 00 10 00 41 60 10 00 7...	download pdo 0x1A04 entr...
C <PS>	CoE	0x1A05 C 0	00 00	download pdo 0x1A05 entr...
C <PS>	CoE	0x1600 C 0	02 00 10 00 40 60 10 00 4...	download pdo 0x1600 entr...
C <PS>	CoE	0x1601 C 0	02 00 10 00 40 60 08 00 6...	download pdo 0x1601 entr...
C <PS>	CoE	0x1602 C 0	02 00 10 00 40 60 20 00 7...	download pdo 0x1602 entr...
C <PS>	CoE	0x1604 C 0	02 00 10 00 40 60 10 00 7...	download pdo 0x1604 entr...
C <PS>	CoE	0x1605 C 0	00 00	download pdo 0x1605 entr...
C <PS>	CoE	0x1C12 C 0	01 00 02 16	download pdo 0x1C12 index
C <PS>	CoE	0x1C1		
C <PS>	CoE	0x1C1		
C <PS>	CoE	0x1C1		
A <IP, PS>	AoE	1/3		
C <PS>	CoE	0xF03		
E <IP, PS>	EoE			

Edit CANopen Startup Entry

Transition: ☐ I -> P ☑ P -> S ☐ S -> P ☐ S -> O ☐ O -> S
Index (hex): 300b
Sub-Index (dec): 0
☐ Validate ☐ Complete Access
Data (hexbin): 01
Validate Mask:
Comment: CSP - Enable External Velocity FF

Index	Name	Flags	Value
3004:0	Additional position loop output scaling		> 2 <
3005:0	Cyclic Data Loss Behaviour		> 8 <
3006:0	Out Cyclic Data Configuration		> 2 <
3007:0	In Cyclic Data Configuration		> 2 <
3008	Activate velocity mode redirection	RW	0x00 (0)
3009	Enhanced Loop Control	RW	0x00 (0)
300A	Window Filter Size	RW	0x00 (0)
300B	CSP - Enable External Velocity FF	RW	0x01 (1)
3101:0	Drive Startup Command		> 15 <
6007	Abort Connection option code	RW	1
6040	Controlword	RW P	0x0000 (0)
6042	vl target velocity	RW P	0
6046:0	vl velocity min max amount		> 2 <

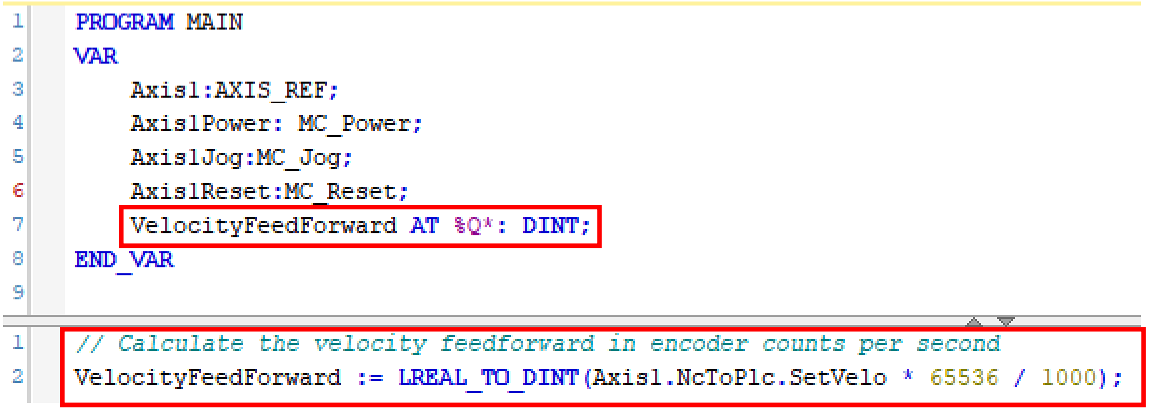
4. For a normal NC Axis i.e. not a CNC axis, the user must calculate the velocity feedforward reference manually. To do this, open the axis in the solution explorer and check the encoder scaling:



	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	1000.0
	Scaling Factor Denominator (default: 1.0)	65536.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	1000.0
	Tolerance Window for Modulo Start	0.0
	Encoder Mask (maximum encoder value)	0xFFFFFFFF

For the purposes of this description A numerator of 1000 units per revolution, and a scaling denominator of 65536 encoder counts per revolution is used.

Add the highlighted code in the following code snippet to the PLC programming area e.g. “Main”. This reads the velocity from the axis and then converts it to into a velocity feedforward reference in Encoder Counts per Second suitable for use with the Velocity Offset 0x60B1. “VelocityFeedForward” is mapped to PLC Q registers using “AT %Q*” so that the PDO can be linked with the feedforward value.

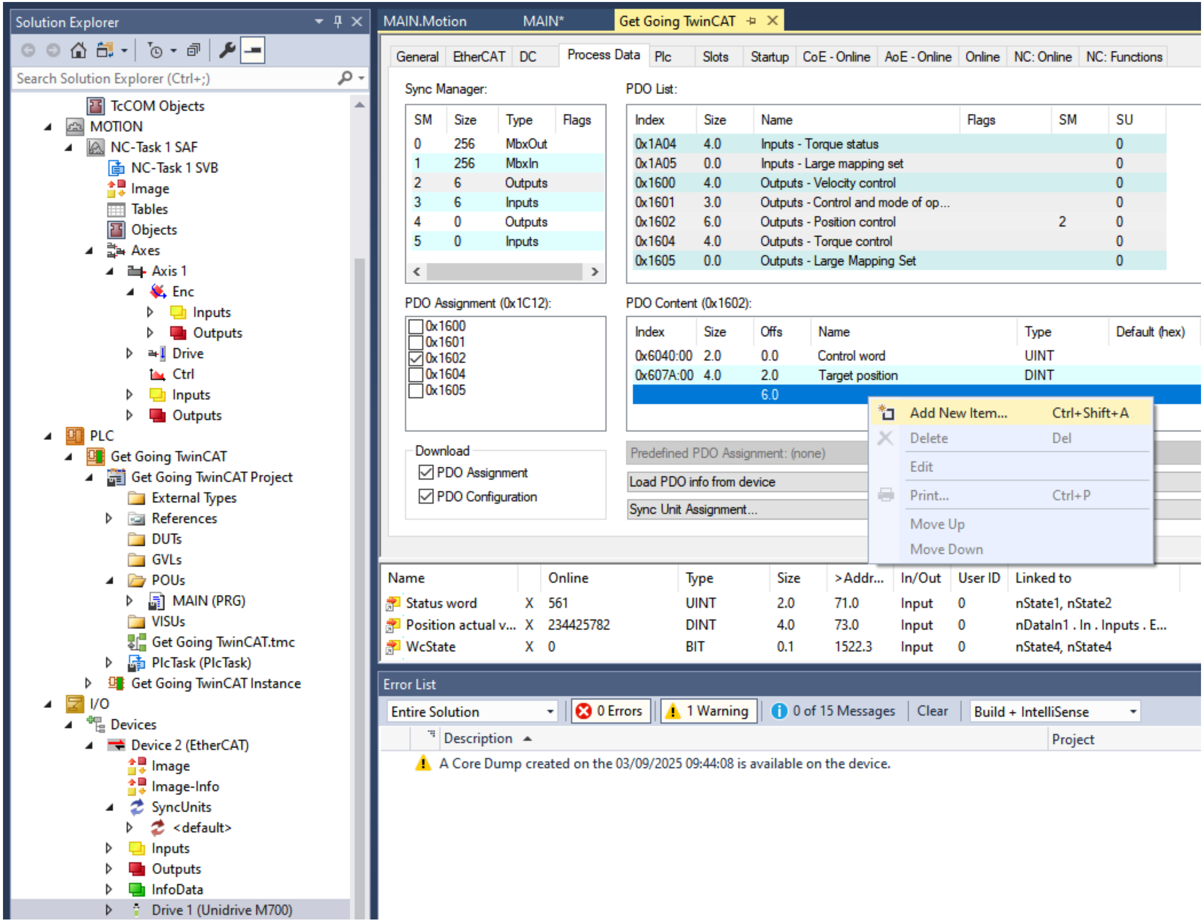


```
PROGRAM MAIN
VAR
    Axis1:AXIS_REF;
    Axis1Power: MC_Power;
    Axis1Jog:MC_Jog;
    Axis1Reset:MC_Reset;
    VelocityFeedForward AT %Q*: DINT;
END_VAR

```

```
// Calculate the velocity feedforward in encoder counts per second
VelocityFeedForward := LREAL_TO_DINT(Axis1.NcToPlc.SetVelo * 65536 / 1000);
```

Build the code (Ctrl+Shit+B) after making this change so that “VelocityFeedForward” is selectable in the PDO linking editor.

5. Add a PDO mapping to 0x60B1. Double click on the EtherCAT slave drive, then select the “Process Data” tab. In the “PDO List” select the row that starts “0x1602”. In the “PDO Content” window select an empty row and the right click and select “Add New Item…”



Sync Manager:

SM	Size	Type	Flags
0	256	MbxOut	
1	256	MbxIn	
2	6	Outputs	
3	6	Inputs	
4	0	Outputs	
5	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A04	4.0	Inputs - Torque status			0
0x1A05	0.0	Inputs - Large mapping set			0
0x1600	4.0	Outputs - Velocity control			0
0x1601	3.0	Outputs - Control and mode of op...			0
0x1602	6.0	Outputs - Position control		2	0
0x1604	4.0	Outputs - Torque control			0
0x1605	0.0	Outputs - Large Mapping Set			0

PDO Assignment (0x1C12):

☐ 0x1600
☐ 0x1601
☑ 0x1602
☐ 0x1604
☐ 0x1605

PDO Content (0x1602):

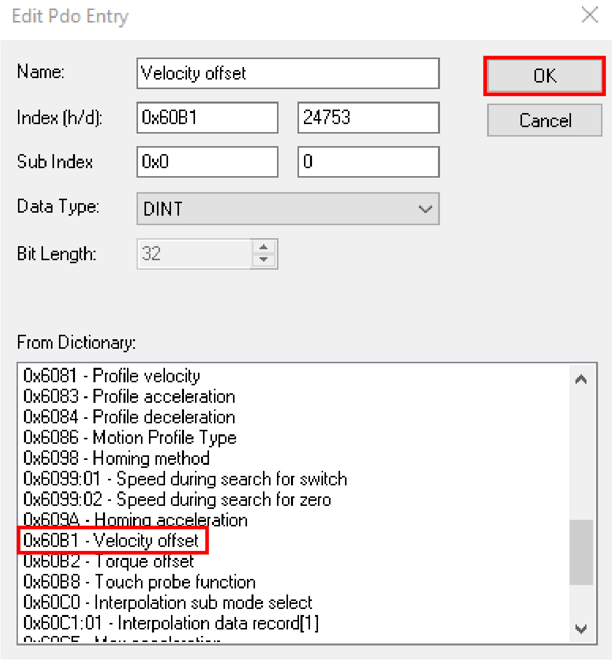
Index	Size	Offs	Name	Type	Default (hex)
0x6040:00	2.0	0.0	Control word	UINT	
0x607A:00	4.0	2.0	Target position	DINT	
		6.0			

Download
☑ PDO Assignment
☑ PDO Configuration

Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Status word	X	561	UINT	2.0	71.0	Input	0	nState1, nState2
Position actual v...	X	234425782	DINT	4.0	73.0	Input	0	nDataIn1 . In . Inputs . E...
WcState	X	0	BIT	0.1	1522.3	Input	0	nState4, nState4

Description	Project
A Core Dump created on the 03/09/2025 09:44:08 is available on the device.	

When the “Edit PDO Entry” dialog starts, select “0x60B1 – Velocity Offset” from the dictionary and then click “OK”.



Name: Velocity offset

Index (h/d): 0x60B1 24753

Sub Index: 0x0 0

Data Type: DINT

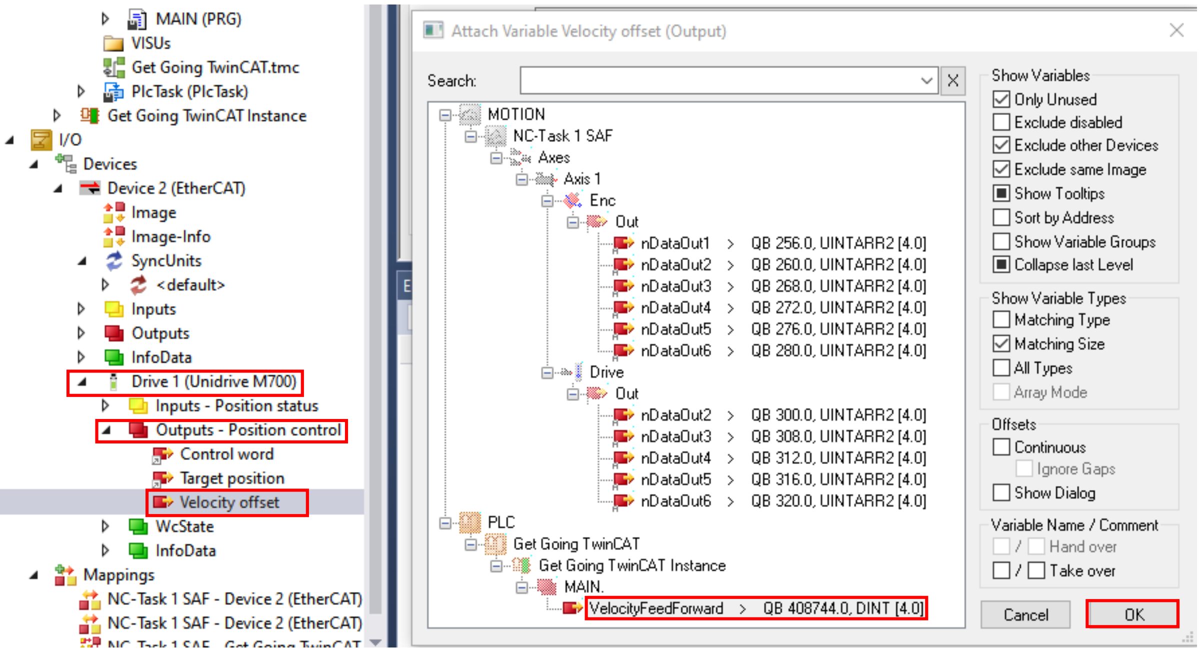
Bit Length: 32

From Dictionary:

0x6081 - Profile velocity
0x6083 - Profile acceleration
0x6084 - Profile deceleration
0x6086 - Motion Profile Type
0x6098 - Homing method
0x6099:01 - Speed during search for switch
0x6099:02 - Speed during search for zero
0x609A - Homing acceleration
0x60B1 - Velocity offset
0x60B2 - Torque offset
0x60B8 - Touch probe function
0x60C0 - Interpolation sub mode select
0x60C1:01 - Interpolation data record[1]

The PDO is now mapped.

6. Link the PDO mapping to the calculated velocity feedforward value. Expand the EtherCAT slave drive, then expand “Outputs – Position control”. Double click on “Velocity Offset” to open the “Attach Variable Velocity offset (Output)” dialog. Select “VelocityFeedForward” from the list and click “OK”.



Show Variables
☑ Only Unused
☐ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
■ Show Tooltips
☐ Sort by Address
☐ Show Variable Groups
■ Collapse last Level

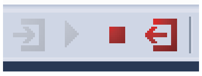
Show Variable Types
☐ Matching Type
☑ Matching Size
☐ All Types
☐ Array Mode

Offsets
☐ Continuous
☐ Ignore Gaps
☐ Show Dialog

Variable Name / Comment
☐ / ☐ Hand over
☐ / ☐ Take over

The setup is complete. Save the changes by pressing and activate the configuration by pressing . Click the login icon to load / view the code while it runs. Run the application (if it is not already running) by pressing the ▶ button.

When the system is running the online tool bar looks like this:



4 CT NC Axis Interface Library Functionality

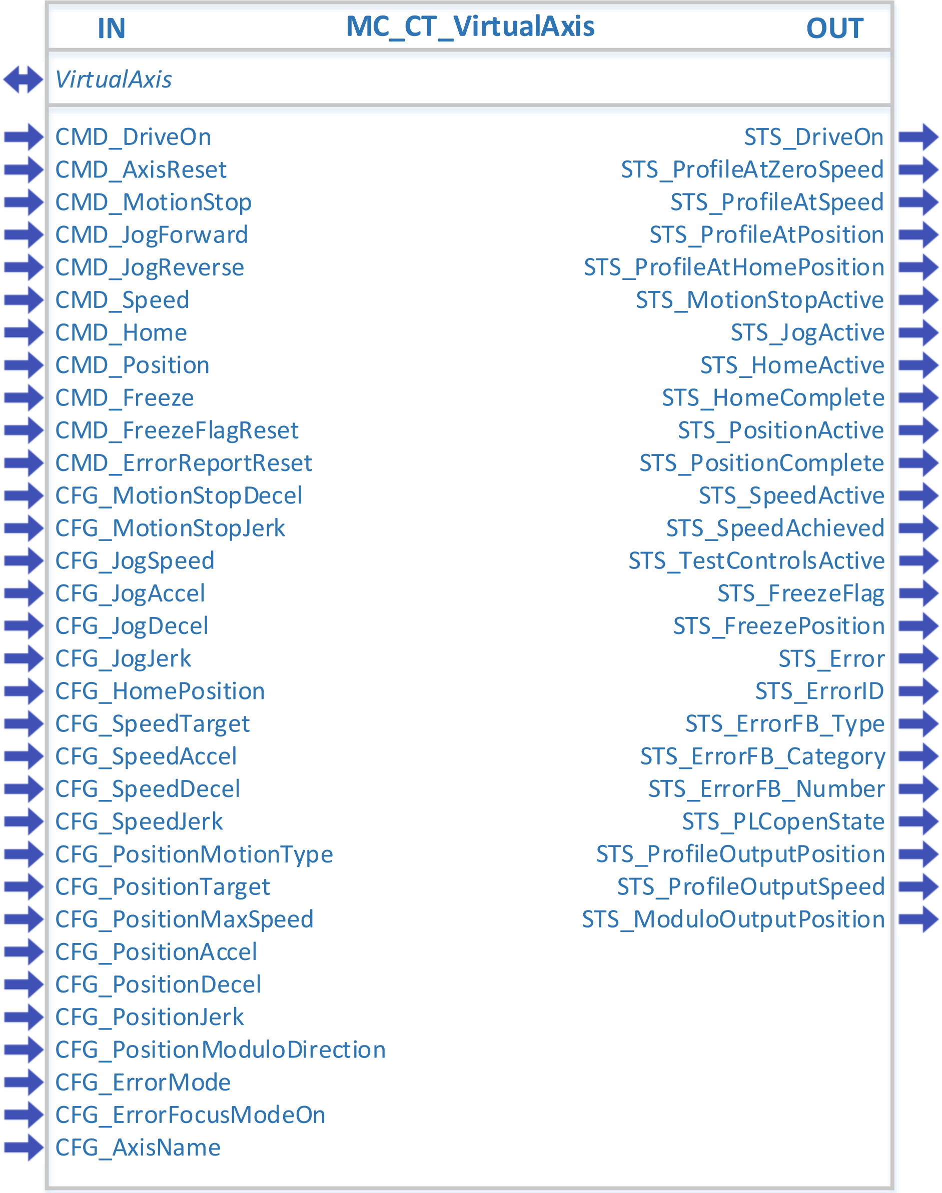
4.1 Motion Interface

4.1.1 FB Diagrams

These are the 4 main motion control interfaces that are included with the CT NC Axis Interface Library. This section of the user guide contains detailed information on the functionality of the FB I/O for each of these motion control interfaces:

IN	MC_CT_PositionAxis	OUT
Axis		
CMD_DriveOn		STS_CyclicCommsOK
CMD_AxisReset		STS_AxisCommsReady
CMD_MotionStop		STS_DriveOn
CMD_JogForward		STS_DriveOK
CMD_JogReverse		STS_DriveEnabled
CMD_Home		STS_BrakeReleased
CMD_HomeSwitch		STS_CurrentLimitActive
CMD_Position		STS_DriveAtZeroSpeed
CMD_SoftwareTravelLimitEnable		STS_DriveAtSpeed
CMD_FreezeFlagReset		STS_DriveAtPosition
CMD_ErrorReportReset		STS_ProfileAtZeroSpeed
CFG_MotionStopDecel		STS_ProfileAtSpeed
CFG_MotionStopJerk		STS_ProfileAtPosition
CFG_JogSpeed		STS_ProfileAtHomePosition
CFG_JogAccel		STS_MotionStopActive
CFG_JogDecel		STS_JogActive
CFG_JogJerk		STS_HomeActive
CFG_HomeMode		STS_HomeComplete
CFG_HomeDirection		STS_PositionActive
CFG_HomeSwitchLogic		STS_PositionComplete
CFG_HomeTorqueThreshold		STS_TestControlsActive
CFG_HomeSearchPosition		STS_SoftwareTravelLimitNegativeOK
CFG_HomeSearchSpeed		STS_SoftwareTravelLimitPositiveOK
CFG_HomeSearchBackoffSpeed		STS_HardwareTravelLimitNegativeOK
CFG_HomeOffsetPosition		STS_HardwareTravelLimitPositiveOK
CFG_HomeOffsetMaxSpeed		STS_PositionFollowingErrorLimitOK
CFG_HomeAccel		STS_SpeedFollowingErrorLimitOK
CFG_HomeDecel		STS_FreezeFlag
CFG_HomeJerk		STS_FreezePosition
CFG_HomeDriveDigitalInput		STS_StopOnFollowingErrorActive
CFG_PositionMotionType		STS_StopOnCommsFailedActive
CFG_PositionTarget		STS_Error
CFG_PositionMaxSpeed		STS_ErrorID
CFG_PositionAccel		STS_ErrorFB_Type
CFG_PositionDecel		STS_ErrorFB_Category
CFG_PositionJerk		STS_ErrorFB_Number
CFG_PositionModuloDirection		STS_PLCopenState
CFG_SoftwareTravelLimitNegative		STS_DriveActiveAlarm
CFG_SoftwareTravelLimitPositive		STS_DriveLastTrip
CFG_SoftwareTravelLimitDecel		STS_DriveLastSubTrip
CFG_CurrentLimit		STS_DriveLoadPercent
CFG_AtSpeedWindow		STS_DriveActualPosition
CFG_AtPositionWindow		STS_DriveActualSpeed
CFG_PositionFollowingErrorLimit		STS_ProfileOutputPosition
CFG_SpeedFollowingErrorLimit		STS_ProfileOutputSpeed
CFG_StopOnFollowingErrorMode		STS_PositionFollowingError
CFG_StopOnFollowingErrorDecel		STS_SpeedFollowingError
CFG_StopOnFollowingErrorTime		
CFG_ErrorMode		
CFG_ErrorFocusModeOn		
CFG_FreezeSource		
CFG_AxisName		


IN
MC_CT_EGB_Axis
OUT
SlaveAxis
MasterAxis
CMD_DriveOn
CMD_AxisReset
CMD_MotionStop
CMD_JogForward
CMD_JogReverse
CMD_Home
CMD_HomeSwitch
CMD_Position
CMD_EGB
CMD_SoftwareTravelLimitEnable
CMD_FreezeFlagReset
CMD_ErrorReportReset
CFG_MotionStopDecel
CFG_MotionStopJerk
CFG_JogSpeed
CFG_JogAccel
CFG_JogDecel
CFG_JogJerk
CFG_HomeMode
CFG_HomeDirection
CFG_HomeSwitchLogic
CFG_HomeTorqueThreshold
CFG_HomeSearchPosition
CFG_HomeSearchSpeed
CFG_HomeSearchBackoffSpeed
CFG_HomeOffsetPosition
CFG_HomeOffsetMaxSpeed
CFG_HomeAccel
CFG_HomeDecel
CFG_HomeJerk
CFG_HomeDriveDigitalInput
CFG_PositionMotionType
CFG_PositionTarget
CFG_PositionMaxSpeed
CFG_PositionAccel
CFG_PositionDecel
CFG_PositionJerk
CFG_PositionModuloDirection
CFG_EGB_Mode
CFG_EGB_PhaseOffset
CFG_EGB_RatioNumerator
CFG_EGB_RatioDenominator
CFG_EGB_MaxSpeed
CFG_EGB_Accel
CFG_EGB_Decel
CFG_EGB_Jerk
CFG_EGB_MasterRegistrationSensorDelay
CFG_EGB_MasterFreezeFlag
CFG_EGB_MasterFreezePosition
CFG_SoftwareTravelLimitNegative
CFG_SoftwareTravelLimitPositive
CFG_SoftwareTravelLimitDecel
CFG_CurrentLimit
CFG_AtSpeedWindow
CFG_AtPositionWindow
CFG_PositionFollowingErrorLimit
CFG_SpeedFollowingErrorLimit
CFG_StopOnFollowingErrorMode
CFG_StopOnFollowingErrorDecel
CFG_StopOnFollowingErrorTime
CFG_ErrorMode
CFG_ErrorFocusOn
CFG_FreezeSource
CFG_AxisName
STS_CyclicCommsOK
STS_AxisCommsReady
STS_MasterDriveOn
STS_DriveOn
STS_DriveOK
STS_DriveEnabled
STS_BrakeReleased
STS_CurrentLimitActive
STS_DriveAtZeroSpeed
STS_DriveAtSpeed
STS_DriveAtPosition
STS_ProfileAtZeroSpeed
STS_ProfileAtSpeed
STS_ProfileAtPosition
STS_ProfileAtHomePosition
STS_MotionStopActive
STS_JogActive
STS_HomeActive
STS_HomeComplete
STS_PositionActive
STS_PositionComplete
STS_EGB_Active
STS_EGB_Locked
STS_TestControlsActive
STS_SoftwareTravelLimitNegativeOK
STS_SoftwareTravelLimitPositiveOK
STS_HardwareTravelLimitNegativeOK
STS_HardwareTravelLimitPositiveOK
STS_PositionFollowingErrorLimitOK
STS_SpeedFollowingErrorLimitOK
STS_FreezeFlag
STS_FreezePosition
STS_StopOnFollowingErrorActive
STS_StopOnCommsFailedActive
STS_Error
STS_ErrorID
STS_ErrorFB_Type
STS_ErrorFB_Category
STS_ErrorFB_Number
STS_PLCopenState
STS_DriveActiveAlarm
STS_DriveLastTrip
STS_DriveLastSubTrip
STS_DriveLoadPercent
STS_DriveActualPosition
STS_DriveActualSpeed
STS_ProfileOutputPosition
STS_ProfileOutputSpeed
STS_PositionFollowingError
STS_SpeedFollowingError

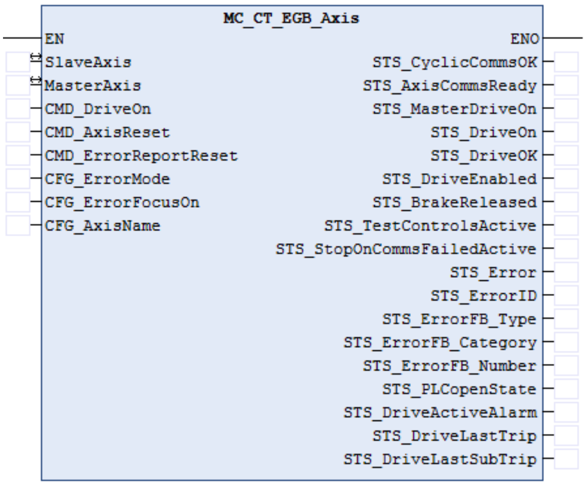

IN
MC_CT_VirtualAxis
OUT
VirtualAxis
CMD_DriveOn
CMD_AxisReset
CMD_MotionStop
CMD_JogForward
CMD_JogReverse
CMD_Speed
CMD_Home
CMD_Position
CMD_Freeze
CMD_FreezeFlagReset
CMD_ErrorReportReset
CFG_MotionStopDecel
CFG_MotionStopJerk
CFG_JogSpeed
CFG_JogAccel
CFG_JogDecel
CFG_JogJerk
CFG_HomePosition
CFG_SpeedTarget
CFG_SpeedAccel
CFG_SpeedDecel
CFG_SpeedJerk
CFG_PositionMotionType
CFG_PositionTarget
CFG_PositionMaxSpeed
CFG_PositionAccel
CFG_PositionDecel
CFG_PositionJerk
CFG_PositionModuloDirection
CFG_ErrorMode
CFG_ErrorFocusModeOn
CFG_AxisName
STS_DriveOn
STS_ProfileAtZeroSpeed
STS_ProfileAtSpeed
STS_ProfileAtPosition
STS_ProfileAtHomePosition
STS_MotionStopActive
STS_JogActive
STS_HomeActive
STS_HomeComplete
STS_PositionActive
STS_PositionComplete
STS_SpeedActive
STS_SpeedAchieved
STS_TestControlsActive
STS_FreezeFlag
STS_FreezePosition
STS_Error
STS_ErrorID
STS_ErrorFB_Type
STS_ErrorFB_Category
STS_ErrorFB_Number
STS_PLCopenState
STS_ProfileOutputPosition
STS_ProfileOutputSpeed
STS_ModuloOutputPosition

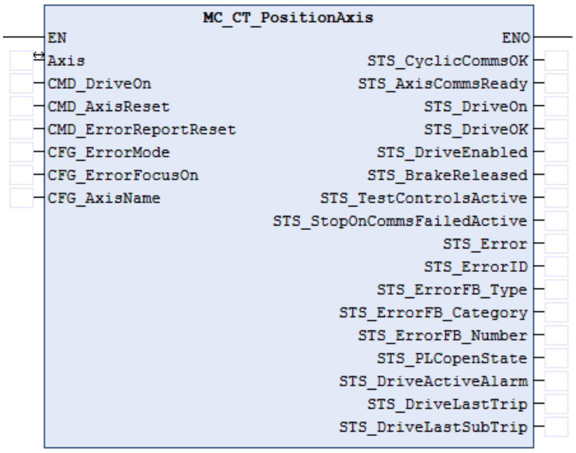
4.1.2 Common FB Functionality

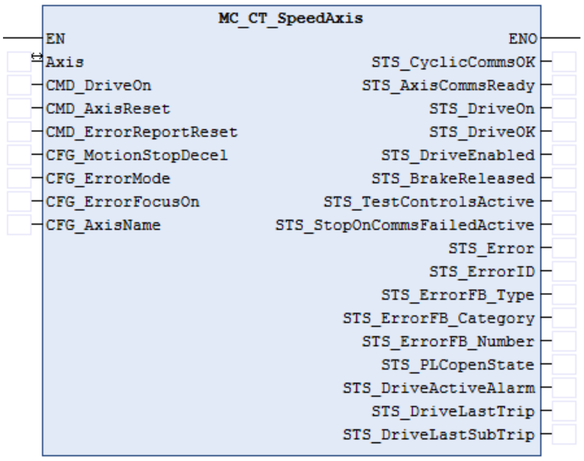
This section shows the groups of functionality that are shared between the 4 main motion interface function blocks. An overview of the features is provided that helps the user to identify which type of functionality a given input or output is related to.

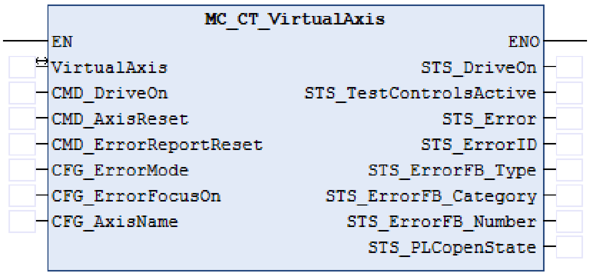
4.1.2.1 General

Function block I/O that is common between the motion interfaces and categorised as “General” functionality is shown in the diagrams below. The interface has been simplified so only the relevant ones are shown.







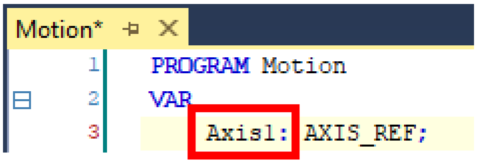


The general functionality offers the user:

- The ability to specify the target axis for motion to be invoked on.
- The ability to enable / disable the target axis via CMD_DriveOn, with status information via STS_DriveOn and STS_Drive_Enabled.
- The ability to reset the target axis in the event of a trip or NC Axis error via CMD_AxisReset.
- Status information on the cyclic communications to a drive, (STS_CyclicCommsOK), and the EtherCAT network status of a drive (STS_AxisCommsReady).
- Target axis alarm and trip information via STS_DriveOK, STS_DriveActiveAlarm, STS_DriveLastTrip and STS_DriveLastSubTrip.

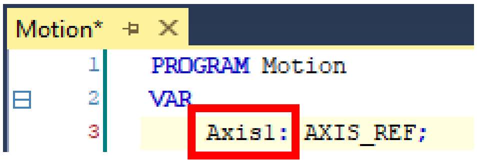
- The ability to specify the error mode (continuous update or one-shot trigger) via CFG_ErrorMode, choose which type of internal NC function block the user would like to monitor for errors via CFG_ErrorFocusOn and monitor the result via STS_Error, STS_ErrorID, STS_ErrorFB_Type and STS_ErrorFB_Category.
- Monitoring of the current PLCopen state machine state via STS_PLCopenState.
- The ability to monitor the state of the mechanical brake controller via STS_BrakeReleased, when it has been configured via drive Menu 12.
- Web visualisation control monitoring of the interface via STS_TestControlsActive.

FB Parameter	**Axis**		
Applies To	**MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis.**		

Description	The FB interface to the NC axis. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

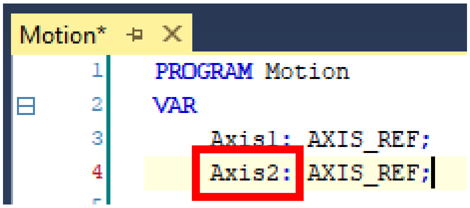
Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

FB Parameter	**SlaveAxis**		
Applies To	**MC_CT_EGB_Axis**		

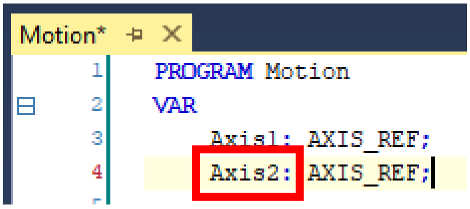
Description	Specific to the Electronic Gearbox Axis FB, this is the FB interface to the NC axis for Slave motion. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

<table>
<tr><td>FB Parameter</td><td colspan="3">MasterAxis</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_EGB_Axis</td></tr>
<tr><td>Description</td><td colspan="3">Specific to the Electronic Gearbox Axis FB, this is the FB interface to the NC axis that will be used as a master position reference, send and receive data, and to monitor the PLCopen state machine for the specified axis.

This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled, which could take the form of a virtual drive axis, an encoder axis or a physical drive axis.

 <pre>PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    Axis2: AXIS_REF;</pre></td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

<table>
<tr><td>FB Parameter</td><td colspan="3">VirtualAxis</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_VirtualAxis</td></tr>
<tr><td>Description</td><td colspan="3">Specific to the Virtual Axis FB, this is the FB interface to the NC axis. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.

This input/output must be assigned to the name given to the NC axis.

 <pre>PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    Axis2: AXIS_REF;</pre></td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

FB Parameter	**CMD_DriveOn**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	When set to FALSE, the drive will become inhibited and the NC axis will transition into the “Power off” state. When set to TRUE, the drive will be software enabled, in the case of a physical drive a run signal will be applied. Provided the STO is healthy and no other motion command is simultaneously being executed, the NC axis will transition to the “Standstill” state. The drive STO is enabled by applying 24V to T31 on M700 or T11 and T13 on M702 or T2 and T6 on Digitax HD. If the axis is disabled via CMD_DriveOn = FALSE, the position count in STS_DriveActualPosition will be retained. If CMD_MotionStop = TRUE at the same time as CMD_DriveOn changes from TRUE to FALSE (motion was in progress), CMD_MotionStop has priority and will bring the axis to a controlled stop before disabling the target drive. If the drive is still powered and EtherCAT communications to the drive are lost, the motor behaviour will of that specified in CoE object “Cyclic data loss behaviour” (Index: 0x3005, Sub-index 0x2), see section **3.2.2 How to add a single CAN object to the TwinCAT Startup List & Cyclic comms loss action** for more information. If CMD_DriveOn = FALSE and a motion CMD_ bit = TRUE and STS_CyclicCommsOK = FALSE (network loss), when the drive recovers it will remain in Inhibit where the user will need to set all motion CMD_ bits to FALSE before the axis can be powered via CMD_DriveOn. If CMD_DriveOn = TRUE and a motion CMD_ bit = TRUE and STS_CyclicCommsOK = FALSE (network loss while motion was running), when the drive recovers it will remain in Inhibit where the user will need to set CMD_DriveOn to FALSE **in addition to** setting all motion CMD_ bits to FALSE before the axis can be powered via CMD_DriveOn and any motion reinstated by setting the relevant motion CMD_ bits to TRUE. If CMD_DriveOn = TRUE, a motion CMD_ bit = TRUE and Drive Under voltage = TRUE, STS_DriveOn and STS_DriveOK will = FALSE. When the drive recovers, the STS_DriveOn and STS_DriveOK will = TRUE, where any motion CMD_ previously set TRUE will need to be toggled FALSE then back to TRUE to reinstate motion. If CMD_DriveOn = TRUE, a motion CMD_ bit = TRUE and a drive trip condition occurs: - If the trip is reset via CMD_AxisReset, the drive will enter the Run state and any motion CMD_ bit set TRUE will instantly restart motion on the axis. - If the trip is reset via the drive (keypad or otherwise), the drive will remain in an Inhibit state until the axis has been reset via the CMD_AxisReset input.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	CMD_AxisReset		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	On a rising edge transition, from FALSE to TRUE, both the NC axis and the drive will be reset. This is the equivalent of pressing the red reset button on the keypad of the target drive axis followed by executing a MC_Reset.		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	0	Maximum	1

FB Parameter	CMD_ErrorReportReset		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	On a rising edge transition, from FALSE to TRUE, the latch that allows NC axis errors to be captured is reset. This applies when CFG_ErrorMode = MC_CT_ErrorModes.OneShotTrigger (1).		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	0	Maximum	1

FB Parameter	CFG_ErrorMode		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Specifies how NC axis errors will be captured, where: **MC_CT_ErrorModes.ContinuousUpdate (0)** NC errors will be captured and displayed continuously. **MC_CT_ErrorModes.OneShotTrigger (1)** NC errors will be captured and displayed on a one-shot basis. Once an error has been captured, the CMD_ErrorReportReset input will need to be set TRUE to reset the error outputs. This input accepts the MC_CT_ErrorModes (ENUM) data type, for more information see section **5.9 MC_CT_ErrorModes**.		
Usage	Input	Data Type	MC_CT_ErrorModes
Unit	N/A	Initial Value	0 (Continuous Update)
Minimum	0 (Continuous Update)	Maximum	1 (One Shot Trigger)

FB Parameter	**CFG_ErrorFocusOn**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Selects which internal function block to focus on. When set to 0, all internal function blocks are checked where the error status is presented at the “STS_Error”, “STS_ErrorID”, “STS_ErrorFB_Type” and “STS_ErrorFB_Category” outputs. If a non-zero value is set, the “STS_Error”, “STS_ErrorID”, “STS_ErrorFB_Type” and “STS_ErrorFB_Category” outputs will only display error information for the specified function block number. “STS_ErrorFB_Number” can be used to identify the internal function block number to be chosen for isolated monitoring.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	20

FB Parameter	**STS_CyclicCommsOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	When set to FALSE, indicates that the SDO communications used to monitor the target drive are not working. When set to TRUE, indicates that the SDO communications used to monitor the target drive are working correctly.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_AxisCommsReady**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	When set to FALSE, indicates that EtherCAT communications which are used to pass references between target CT drives are either not configured or not running. When set to TRUE, indicates that EtherCAT communications have been configured correctly and are running on the target drive axis. This is provided so that if the target drive axis is also part of a synchronous real time position reference configuration, that the status of the EtherCAT communications used to transmit / receive reference is known.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_MasterDriveOn**		
Applies To	**MC_CT_EGB_Axis**		
Description	When set to FALSE, indicates that the drive is not on i.e. inhibited, and the FB motion commands can't be used. When set to TRUE, indicates that the drive is on and ready for motion i.e. enabled, a run signal is applied, there are no trips, where the FB motion command inputs may be used to move the target axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_DriveOn**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	When set to FALSE, indicates that the drive is not on i.e. inhibited, and the FB motion commands can't be used. When set to TRUE, indicates that the drive is on and ready for motion i.e. enabled, a run signal is applied, there are no trips, where the FB motion command inputs may be used to move the target axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_DriveOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	When set to FALSE, indicates that the drive is not on i.e. inhibited, and the FB motion commands can't be used. When set to TRUE, indicates that the drive is on and ready for motion i.e. enabled, a run signal is applied, there are no trips, where the FB motion command inputs may be used to move the target axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_DriveEnabled**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	When FALSE, indicates that the target drive axis safe torque off feature is active i.e. 0V is applied to T31 on M700 or T11 and T13 on M702 or T2 and T6 on Digitax HD. When set to TRUE, indicates that the drive is hardware enabled i.e. 24V is applied to T31 on M700 or T11 and T13 on M702 or T2 and T6 on Digitax HD. When CMD_DriveOn = FALSE, STS_DriveEnabled = FALSE.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_BrakeReleased**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the state of the drives internal brake controller, where: FALSE = the brake has not been released by the drives internal brake controller. TRUE = the motor has been energised, and the drives internal brake controller has set the brake release signal high, and the “BrakeReleaseDelay” time has elapsed. **HINT:** If the brake controller has not been set up in drive menu 12 or using the guided setup in Connect, this output will always remain FALSE.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_TestControlsActive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the CT NC Axis Visualisation has control of the axis, where: FALSE = User controls at the function block interface are active TRUE = Test controls are active, and the function block interface controls have been superseded. The values present at the input of the function block are ignored when this output is TRUE. The control can only be changed over when CMD_DriveOn = FALSE.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_StopOnCommsFailedActive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the axis power has been removed due to a detect loss of cyclic communications to the drive, where: FALSE = The comms have not failed. TRUE = The comms have failed; the “bRegulatorOn” and “bDriveStart” inputs of the internal instance of MC_Power are forced FALSE.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_Error**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that a NC axis or function block error is present where: FALSE = No error. TRUE = There is an error. More information on the error and its cause can be found at the STS_ErrorID, STS_ErrorFB_Type and STS_ErrorFB_Category outputs.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_ErrorID**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the NC axis error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

FB Parameter	**STS_ErrorFB_Type**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the function block type that had the error. The datatype is MC_CT_FB_Types (ENUM). This ENUM originates from the CT NC Axis Interface Library and contains strings for each NC function block used within the CT NC Axis Interface Library. For more information see section **5.10 MC_CT_FB_Types**.		
Usage	Output	**Data Type**	MC_CT_FB_Types
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	17

FB Parameter	**STS_ErrorFB_Category**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the reference category of function block that had the NC axis error. The datatype is MC_CT_ErrorCategories (ENUM). This ENUM originates from the CT NC Axis Interface Library and contains strings for each NC function block category used within the CT NC Axis Interface Library. For more information see section **5.7 MC_CT_ErrorCategories**.		
Usage	Output	**Data Type**	MC_CT_ErrorCategories
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	13

FB Parameter	**STS_ErrorFB_Number**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the internal function block number that is currently in error. This output is designed to be used with “CFG_ErrorFocusOn” when the user would like to focus on a specific function block for error checking when troubleshooting a system.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	20

FB Parameter	**STS_PLCOpenState**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the current PLCopen axis state of the NC axis as an enumerated text string. The datatype is MC_CT_AxisStates (ENUM). This ENUM based upon MC_AxisStates from the NC axis library and contains strings for each possible PLCopen state. For more information see section **5.3 MC_CT_AxisStates**.		
Usage	Output	**Data Type**	MC_CT_AxisStates
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	8

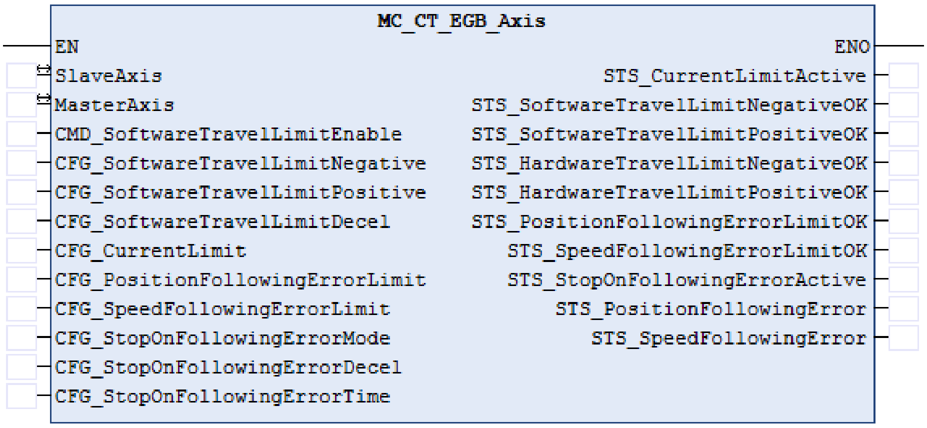
FB Parameter	**STS_DriveActiveAlarm**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the active drive alarm number as an enumerated text string. The value originates from the alarm list present in drive Pr10.104 (Active Alarm). The datatype is MC_CT_ActiveAlarms (ENUM) which contains strings for each possible drive alarm number. For more information see section **5.1 MC_CT_ActiveAlarms**.		
Usage	Output	**Data Type**	MC_CT_ActiveAlarms
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	12

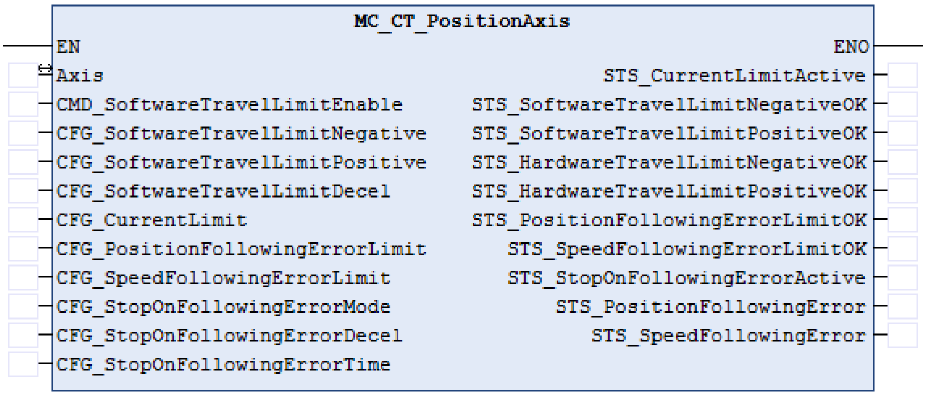
FB Parameter	**STS_DriveLastTrip**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the last drive trip number that occurred as an enumerated text string. The value originates from drive Pr10.020 (Trip 0) The datatype is MC_CT_TripStrings (ENUM) which contains strings for each possible drive trip number. For more information see section **5.20 MC_CT_TripStrings**.		
Usage	Output	**Data Type**	MC_CT_TripStrings
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

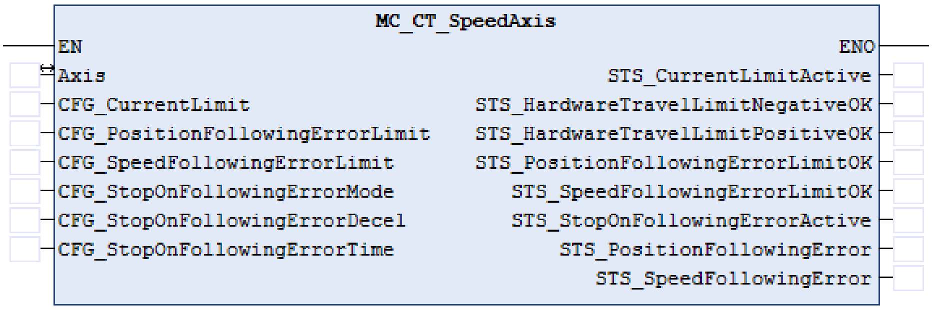
FB Parameter	**STS_DriveLastSubTrip**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the last drive sub trip number that occurred, originating from drive Pr10.070 (Trip 0 Sub-trip Number).		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	65535

4.1.2.2 Limits

Function block I/O that is common between the motion interfaces and categorised as “Limit” functionality is shown in the diagrams below. The interface has been simplified so only the relevant inputs and outputs are shown.







HINT: The “Axis” inputs are shown in the above diagrams because items in the scope “VAR_IN_OUT” can’t be hidden in an LD diagram.

The operating limits available for the target axis are:

- Software axis travel limits enabled by CMD_SoftwareTravelLimitEnable, configured by CFG_SoftwareTravelLimitNegative, CFG_SoftwareTravelLimitPositive and CFG_SoftwareTravelLimitDecel and monitored by STS_SoftwareTravelLimitNegativeOK and STS_SoftwareTravelLimitPositiveOK.
- Hardware axis travel limits monitored by STS_HardwareTravelLimitNegativeOK and STS_HardwareTravelLimitPositiveOK. The terminal used for the hardware limits are configured using the Connect guided setup.
- Symmetrical current limit control via CFG_CurrentLimit. This may be used to limit the maximum percentage of motor rated current and torque applied to the motor. The STS_CurrentLimitActive output indicates when the current limit has been reached, and the percentage of motor rated current, and torque applied to the motor is being limited.
- Speed and position following error limits and monitoring, configured by, CFG_PositionFollowingErrorLimit, CFG_SpeedFollowingErrorLimit, CFG_StopOnFollowingErrorMode and CFG_StopOnFollowingErrorDecel. Monitoring is carried out via the STS_PositionFollowingErrorLimitOK, STS_SpeedFollowingErrorLimitOK, STS_PositionFollowingError and STS_SpeedFollowingError outputs. In the event of breaching a limit the axis will automatically stop depending on how CFG_StopOnFollowingErrorMode is configured.

FB Parameter	**CMD_SoftwareTravelLimitEnable**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Enables software travel limits, where: FALSE = Software travel limits are disabled. TRUE = Software travel limits are enabled.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CFG_SoftwareTravelLimitNegative**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Defines the negative software travel limit for the axis.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**CFG_SoftwareTravelLimitPositive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Defines the positive software travel limit for the axis.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**CFG_SoftwareTravelLimitDecel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Defines the deceleration rate to be used to bring the axis to a stop when a software travel limit has been triggered. A linear deceleration profile is used to stop the axis.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	10000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_CurrentLimit**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Defines the symmetrical current limit to be used for the drive, fixed precision of 1 decimal place. When the drive is in a current limit condition, the STS_CurrentLimitActive output is TRUE.		
Usage	Input	**Data Type**	LREAL
Unit	%	**Initial Value**	150.0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_PositionFollowingErrorLimit**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Sets the symmetrical positional following error limit (in Application units). When the axis is outside of this window, the STS_PositionFollowingErrorOK output will be FALSE.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	100.0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_SpeedFollowingErrorLimit**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Sets the symmetrical speed following error limit (in Application units). When the axis is outside of this window, the STS_SpeedFollowingError output will be TRUE.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	100.0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_StopOnFollowingErrorMode**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Defines the detection mechanism for automatic stopping when an excessive following error is detected, where: **MC_CT_StopOnFollowingErrorModes.Disabled (0)** The stop on following error feature is disabled. **MC_CT_StopOnFollowingErrorModes.PositionFollowingError (1)** Stop if the position following error is greater than CFG_PositionFollowingErrorLimit for the duration specified at CFG_StopOnFollowingErrorTime. **MC_CT_StopOnFollowingErrorModes.SpeedFollowingError (2)** Stop if the speed following error is greater than CFG_SpeedFollowingErrorLimit for the duration specified at CFG_StopOnFollowingErrorTime. **MC_CT_StopOnFollowingErrorModes.PositionAndSpeedFollowingError (3)** Stop if the position following error is greater than CFG_PositionFollowingErrorLimit or if the speed following error is greater than CFG_SpeedFollowingErrorLimit for the duration specified at CFG_StopOnFollowingErrorTime. This input accepts the MC_CT_StopOnFollowingErrorModes (ENUM) data type, for more information see section **5.17 MC_CT_StopOnFollowingErrorModes**.		
Usage	Input	**Data Type**	MC_CT_StopOnFollowingErrorModes
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	3

FB Parameter	**CFG_StopOnFollowingErrorDecel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Defines the deceleration rate to be used to bring the axis to a stop when an excessive following error has been detected.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	1.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_StopOnFollowingErrorTime**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Defines the amount of time that the position or speed following error must be outside of their respective limits before the axis is stopped due to following error.		
Usage	Input	**Data Type**	TIME
Unit	TIME	**Initial Value**	100ms
Minimum	0ms	**Maximum**	49d17h2m47s295ms

FB Parameter	**STS_CurrentLimitActive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the current limit has been reached or exceeded where: FALSE = The current limit has not been reached. TRUE = The current limit has been reached or exceeded.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_SoftwareTravelLimitNegativeOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Indicates that the negative software travel limit has been reached, where: FALSE = The negative software travel limit is activated i.e. the axis has run into the limit negative software limit and must stop. TRUE = The negative software travel limit is not active i.e. the axis has not run into the limit negative software limit.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_SoftwareTravelLimitPositiveOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Indicates that the positive software travel limit has been reached, where: FALSE = The positive software travel limit is activated i.e. the axis has run into the limit positive software limit and must stop. TRUE = The positive software travel limit is not active i.e. the axis has not run into the limit positive software limit.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_HardwareTravelLimitNegativeOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the negative hardware travel limit has been reached, where: FALSE = The negative hardware travel limit is activated i.e. the axis has run into the limit negative hardware limit and must stop. TRUE = The negative hardware travel limit is not active i.e. the axis has not run into the limit negative hardware limit.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_HardwareTravelLimitPositiveOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the positive hardware travel limit has been reached, where: FALSE = The positive hardware travel limit is activated i.e. the axis has run into the limit positive hardware limit and must stop. TRUE = The positive software travel limit is not active i.e. the axis has not run into the limit positive hardware limit.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_PositionFollowingErrorLimitOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the position following error is within the window defined by CFG_PositionFollowingErrorLimit, where: FALSE = The following error is outside of the limit set by CFG_PositionFollowingErrorLimit. TRUE = The position following error is within limits.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_SpeedFollowingErrorLimitOK**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the speed following error is within the window defined by CFG_SpeedFollowingErrorLimit, where: FALSE = The following error is outside of the limit set by CFG_SpeedFollowingErrorLimit. TRUE = The speed following error is within limits.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

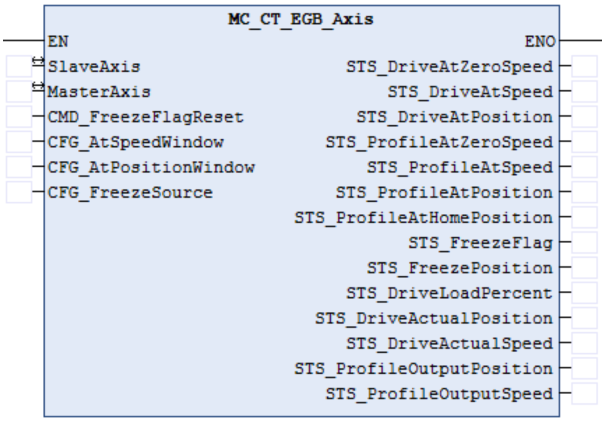
FB Parameter	**STS_StopOnFollowingErrorActive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the axis has been stopped due to an excessive following error, where: FALSE = The axis hasn’t stopped due to excessive following error. TRUE = The axis has stopped due to excessive following error.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

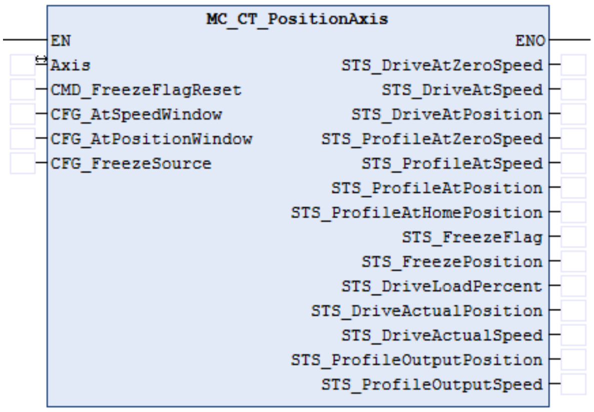
FB Parameter	**STS_PositionFollowingError**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the delta between the profile position and the actual position.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

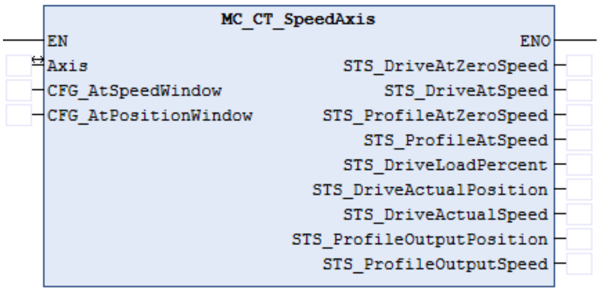
FB Parameter	**STS_SpeedFollowingError**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the delta between the profile speed and the actual speed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

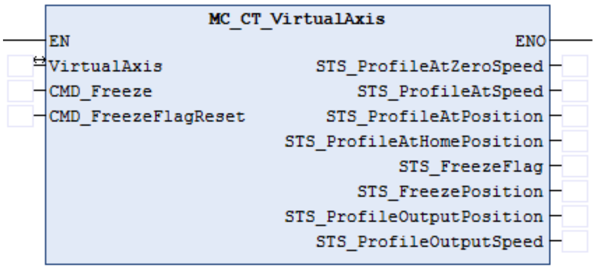
4.1.2.3 Common Motion

Function block inputs and outputs that are common between the motion interfaces and categorised as “Common motion” functionality, is shown in the diagrams below. The interface has been simplified so only the relevant inputs and outputs are shown.









HINT: The “Axis” inputs are shown in the above diagrams because items in the scope “VAR_IN_OUT” can’t be hidden in an LD diagram.

The common motion functions include:

- The ability to capture the slave position feedback as a high-speed edge triggered capture, (<1us) via the drives internal freeze functionality via the touch probe mechanism.

  HINT: For a virtual axis there is no freeze hardware present, therefore the freeze position is only accurate to the nearest EtherCAT task sample.

  CMD_FreezeFlagReset, CFG_FreezeSource, STS_FreezeFlag and STS_FreezePosition are used with this functionality.

- Drive load monitoring via STS_DriveLoadPercent.
- Profile speed, profile position, drive speed and drive position monitoring.

  “Profile” monitoring is an indication of the motion profile generator output (NC axis), and “Drive” monitoring is an indication of the speed and position feedback originating from the drive encoder.

  CFG_ProfileAtSpeedWindow, CFG_ProfileAtPositionWindow, STS_DriveAtZeroSpeed, STS_DriveAtSpeed, STS_DriveAtPosition, STS_ProfileAtZeroSpeed, STS_ProfileAtSpeed, STS_ProfileAtPosition, STS_ProfileAtHomePosition, STS_DriveActualPosition, STS_DriveActualSpeed, STS_ProfileOutputPosition and STS_ProfileOutputSpeed are used with this functionality.

FB Parameter	**CMD_Freeze**		
Applies To	**MC_CT_VirtualAxis**		
Description	On a transition of FALSE to TRUE the position of the virtual axis is captured and presented at the STS_FreezePosition output. Once a valid value is present at the STS_FreezePosition output, the STS_FreezeFlag output is set to TRUE. **HINT:** This is a software-based capture, and therefore the freeze position is captured at the EtherCAT task sample rate.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CFG_FreezeFlagReset**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	On a transition from FALSE to TRUE the STS_FreezeFlag output is reset to FALSE along with the freeze / touch probe capture mechanism. To allow further freeze / touch probe capture this flag must be set to FALSE.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CFG_AtSpeedWindow**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Sets the speed window size (in Application units) that determines when the drive is at speed. When the axis is within this speed window the STS_DriveAtSpeed output will be TRUE. The default window may require increasing in size if the feedback encoder is of low resolution or the units per revolution are low, to avoid STS_DriveAtSpeed flickering (TRUE/FALSE) when the axis is stopped.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	10.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_AtPositionWindow**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Sets the position window size (in Application units) that determines when the drive is at position. When the axis is within this position window the STS_DriveAtPosition output will be TRUE.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	256.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_FreezeSource**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Selects which drive signal is used for the freeze position capture, (touch probe capture). Data type is of the MC_CT_FreezeCaptureSource (ENUM) where: **MC_CT_FreezeCaptureSource.DigitalInput4 (0)** Digital Input 4 is selected as the freeze source. **MC_CT_FreezeCaptureSource.DigitalInput5 (1)** Digital Input 5 is selected as the freeze source. **MC_CT_FreezeCaptureSource.DriveP1MarkerPulse (2)** Drive P1 Marker Pulse is selected as the freeze source. **MC_CT_FreezeCaptureSource.DriveP2MarkerPulse (3)** Drive P2 Marker Pulse is selected as the freeze source. **MC_CT_FreezeCaptureSource.CommonFreezeBus (4)** Common Freeze Bus is selected as the freeze source. **MC_CT_FreezeCaptureSource.DriveP1ZeroPosition (5)** Drive P1 Zero Position is selected as the freeze source. **MC_CT_FreezeCaptureSource.DriveP2ZeroPosition (6)** Drive P2 Zero Position is selected as the freeze source. For more information see section **5.12 MC_CT_FreezeCaptureSource**.		
Usage	Input	**Data Type**	MC_CT_FreezeCaptureSource
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	6

FB Parameter	**STS_DriveAtZeroSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the physical axis (drive encoder feedback) is at zero speed where: FALSE = The drive feedback encoder speed is not equal to 0. TRUE = The drive feedback encoder speed is equal to 0. The window for detecting this condition is defined at the CFG_AtSpeedWindow input.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_DriveAtSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the physical axis (drive encoder feedback) is at the commanded speed where: FALSE = The drive feedback encoder speed is not at the commanded speed. TRUE = The drive feedback encoder speed is at the commanded speed. The window for detecting this condition is defined at the CFG_AtSpeedWindow input.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_DriveAtPosition**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates that the physical axis (drive encoder feedback) is at the commanded position where: FALSE = The drive feedback encoder speed is not at the commanded position. TRUE = The drive feedback encoder speed is at the commanded position. The window for detecting this condition is defined at the CFG_AtPositionWindow input.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_ProfileAtZeroSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that the profile generator output speed from the NC axis is at zero speed where: FALSE = The profile generator output speed is not equal to 0. TRUE = The profile generator output speed is equal to 0.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_ProfileAtSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that the profile generator output speed from the NC axis is at the commanded speed where: FALSE = The profile generator output is not at the commanded speed. TRUE = The profile generator output is at the commanded speed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STS_ProfileAtPosition**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that the profile generator output position from the NC axis is at the commanded position where: FALSE = The profile generator output is not at the commanded position. TRUE = The profile generator output is at the commanded position.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	STS_ProfileAtHomePosition		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that the profile generator output position from the NC axis is at the final home position where: FALSE = The profile generator output is not at the final homing position. TRUE = The profile generator output is at the final homing position.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	STS_FreezeFlag		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that a new freeze position, (touch probe), has been captured and may be read in STS_FreezePosition / STS_SlaveFreezePosition. FALSE = The freeze flag is reset. TRUE = A new freeze position has been captured and may be read in STS_FreezePosition / STS_SlaveFreezePosition.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	STS_FreezePosition		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the current feedback freeze position, (touch probe).		
Usage	Output	Data Type	LREAL
Unit	Application units	Initial Value	0.0
Minimum	-LREAL MAX	Maximum	LREAL MAX

FB Parameter	STS_DriveLoadPercent		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the percentage of load current supplied to the motor by the drive, with a fixed precision of one decimal place.		
Usage	Output	Data Type	LREAL
Unit	%	Initial Value	0.0
Minimum	-LREAL MAX	Maximum	LREAL MAX

FB Parameter	**STS_DriveActualPosition**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the drives actual feedback position.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

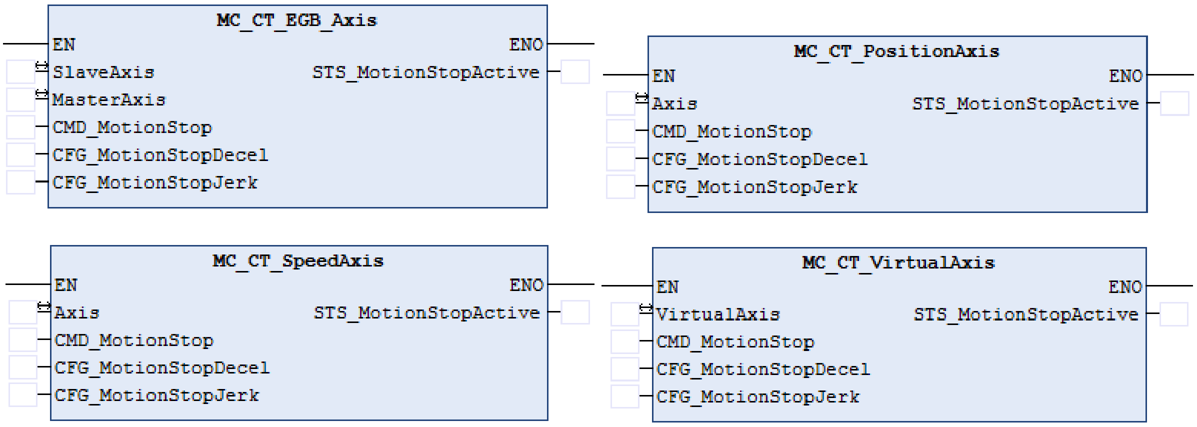
FB Parameter	**STS_DriveActualSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis**		
Description	Indicates the drives actual feedback speed.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**STS_ProfileOutputPosition**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the profile output position for the NC axis.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**STS_ProfileOutputSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates the profile output speed for the NC axis.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

4.1.2.4 Motion Stop

Function block I/O that is common between the motion interfaces and categorised as “Motion stop” functionality is shown in the diagrams below. The interface has been simplified so only the relevant inputs and outputs are shown.



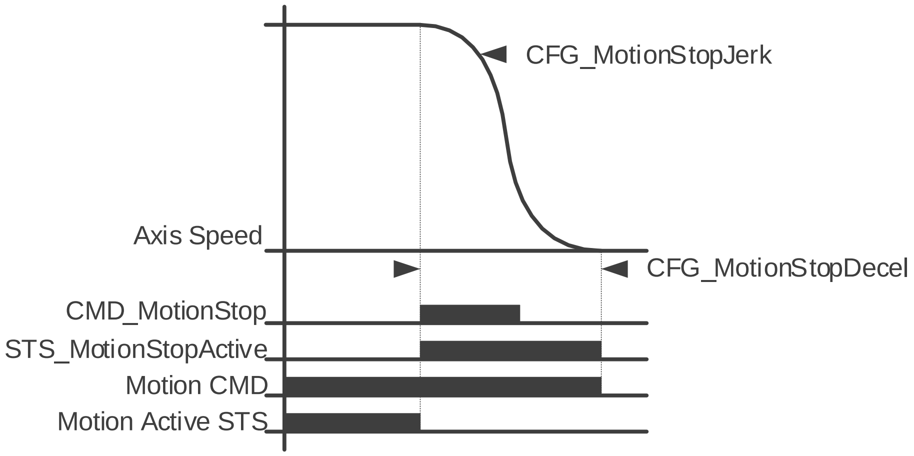
HINT: The “Axis” inputs are shown in the above diagrams because items in the scope “VAR_IN_OUT” can’t be hidden in an LD diagram.

Motion stop gives the user the ability to ramp the NC axis motion profile to a stop. This reference has individual deceleration rate and jerk values so the deceleration may be customised to the machine requirements.

Motion Stop is intended to be used to rapidly stop the axis where the user would normally set the jerk value to 0.0, (giving linear speed ramps), and a large deceleration rate.

When CMD_MotionStop is activated, it has priority over all other motion references, e.g. if the system is jogging using CMD_JogForward and later motion stop is activated by setting the CMD_MotionStop input to TRUE the motion stop profile will be activated, overriding the jog command.

Monitoring of an active motion stop command is provided via the STS_MotionStopActive output.



FB Parameter	**CMD_MotionStop**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Triggers a motion stop which has priority over all other motion commands, where: FALSE = No action TRUE = Motion stop is triggered, with priority over all other motion commands. To allow new motion after commanding a motion stop, CMD_MotionStop must be set to FALSE and any motion CMD_ bits that were TRUE when CMD_MotionStop was activated must be reset to FALSE.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

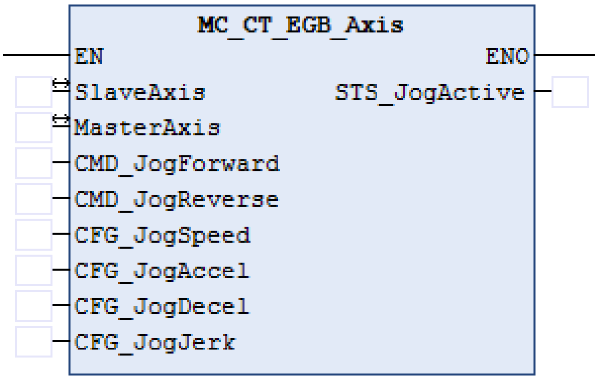
FB Parameter	**CFG_MotionStopDecel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Specifies the maximum deceleration rate to be used when applying the motion stop, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	10000.0
Minimum	>0.0	**Maximum**	LREAL MAX

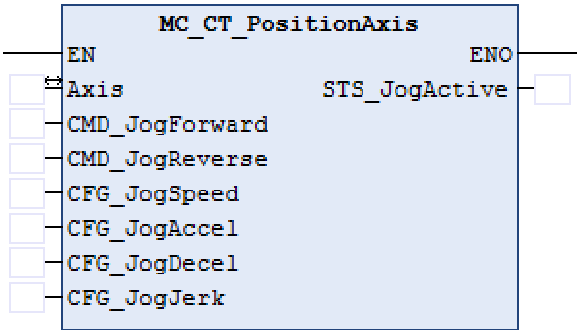
FB Parameter	**CFG_MotionStopJerk**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Specifies the jerk rate to be used when applying the motion stop, in Application units/s^3. A value of 0 results in linear a ramp.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

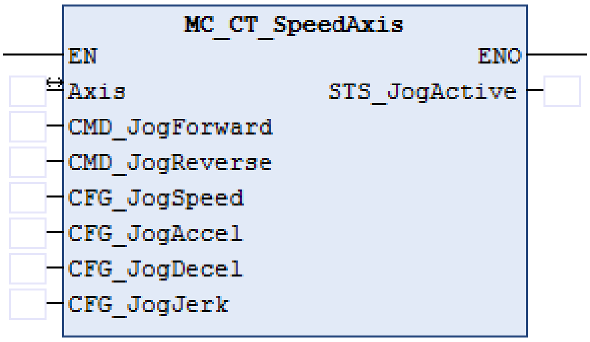
FB Parameter	**STS_MotionStopActive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that a motion stop is the active reference and that all other motion references are suspended, where: FALSE = Motion stop is not active. TRUE = Motion stop is the active reference and all other motion references are suspended. To allow new motion after commanding a motion stop, CMD_MotionStop must be set to FALSE and any motion CMD_ bits that were TRUE when CMD_MotionStop was activated must be reset to FALSE.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

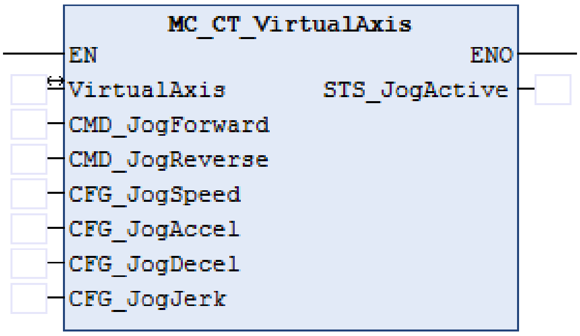
4.1.2.5 Jog

Function block I/O that is common between the motion interfaces and categorised as “Jog” functionality is shown in the diagrams below. The interface has been simplified so only the relevant inputs and outputs are shown.







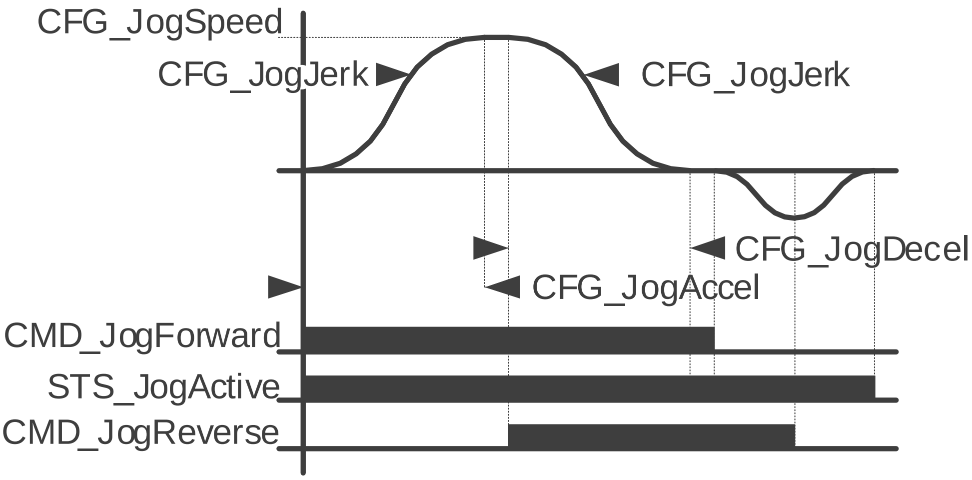


HINT: The “Axis” inputs are shown in the above diagrams because items in the scope “VAR_IN_OUT” can’t be hidden in an LD diagram.

The jog function gives the user a standard interface to implement bi-directional jogging via the CMD_JogForward and CMD_JogReverse inputs.

The user can define the motion profile used for jogging such as the speed, accel rate, decel rate and jerk rate via the CFG_JogSpeed, CFG_JogAccel, CFG_JogDecel and CFG_JogJerk inputs.

Monitoring of an active jog command is provided via the STS_JogActive output.



FB Parameter	CMD_JogForward		
Applies To	MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis		
Description	Causes a jog movement to occur on the axis in the forward direction, where: FALSE = No action TRUE = The jog forward command is selected If CMD_JogForward and CMD_JogReverse are both set to TRUE at the same time, jogging is cancelled.		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	CMD_JogReverse		
Applies To	MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis		
Description	Causes a jog movement to occur on the axis in the reverse direction, where: FALSE = No action TRUE = The jog reverse command is selected If CMD_JogForward and CMD_JogReverse are both set to TRUE at the same time, jogging is cancelled.		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	CFG_JogSpeed		
Applies To	MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis		
Description	Specifies the target speed to be used when jogging, in Application units/s.		
Usage	Input	Data Type	LREAL
Unit	Application units/s	Initial Value	1000.0
Minimum	>0.0	Maximum	LREAL MAX

FB Parameter	**CFG_JogAccel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Specifies the maximum acceleration rate to be used when jogging, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

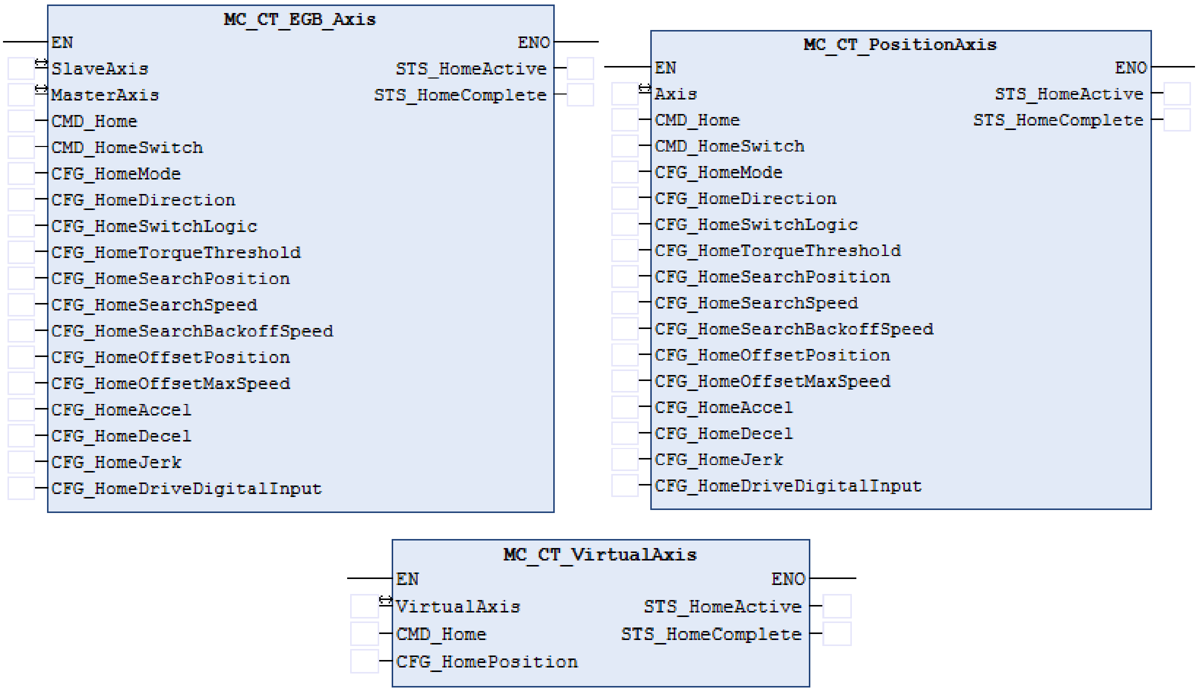
FB Parameter	**CFG_JogDecel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Specifies the maximum deceleration rate to be used when jogging, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_JogJerk**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Specifies the jerk rate to be used when jogging, in Application units/s^3. A value of 0 will result in linear ramps.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**STS_JogActive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Indicates that jog is the active motion reference, where: FALSE = Jog is not the active motion reference. TRUE = Jog is the active motion reference.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

4.1.2.6 Home

Function block I/O that is common between the motion interfaces and categorised as “Home” functionality is shown in the diagrams below. The interface has been simplified so only the relevant inputs and outputs are shown.



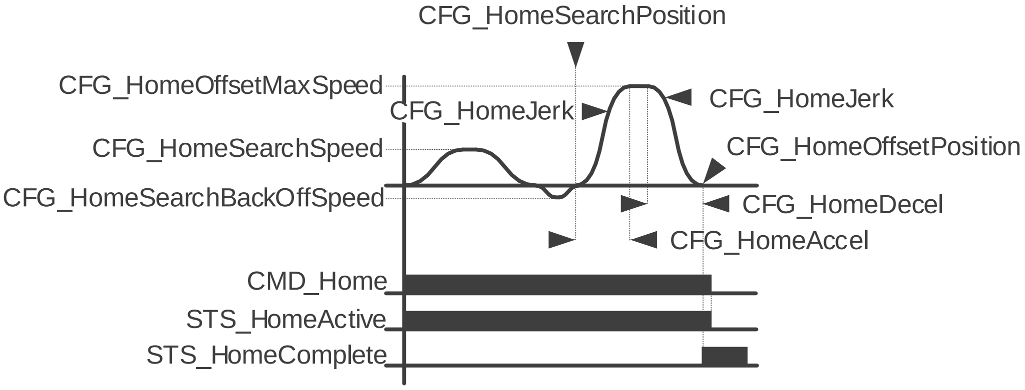
HINT: The “Axis” inputs are shown in the above diagrams because items in the scope “VAR_IN_OUT” can’t be hidden in an LD diagram.

The homing functionality comes with 10 different homing methods that allow it to be applied to many different mechanical arrangements. The homing routines consist of:

- Home at current position.
- Home to function block homing switch (function block input).
- Home to drive homing switch (selectable via CFG_HomeDriveDigitalInput).
- Home to function block homing switch then index marker pulse.
- Home to function block homing switch then reverse to index marker pulse.
- Home to drive homing switch (selectable via CFG_HomeDriveDigitalInput) then index marker pulse.
- Home to drive homing switch (selectable via CFG_HomeDriveDigitalInput) then reverse to index marker pulse.
- Home to index freeze (selectable via CFG_FreezeSource).
- Home to torque threshold (set via CFG_HomeTorqueThreshold).
- Home to torque threshold (set via CFG_HomeTorqueThreshold) then reverse to index marker pulse.

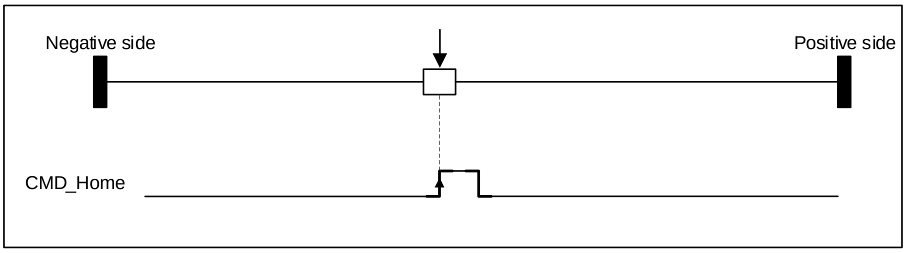
The full details of these homing modes can be found on the CFG_HomeMode tables below.

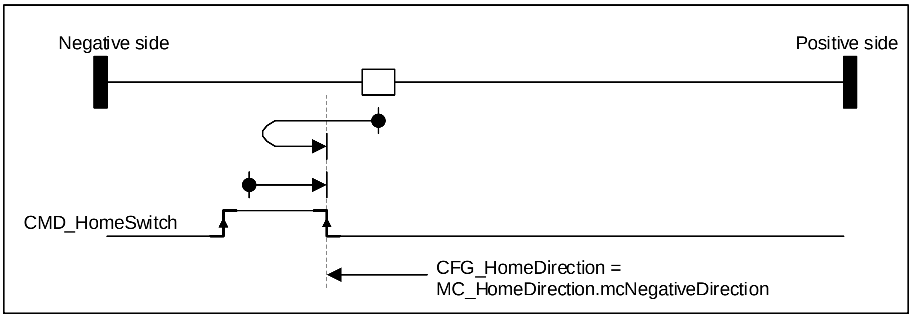
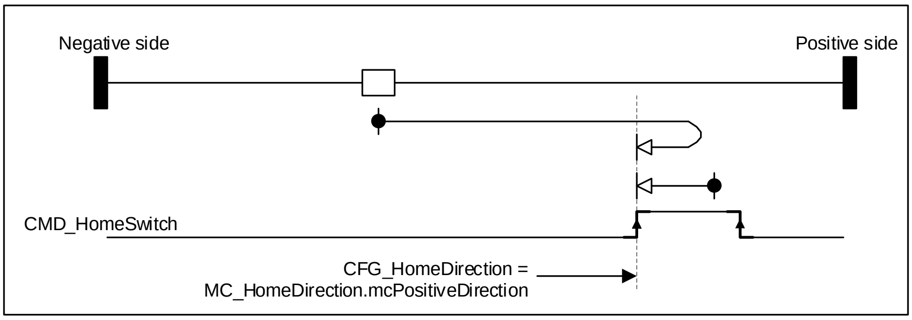
Homing is started using the CMD_Home input, where the motion profile used to find the homing sensor (home search phase) and to reach the final resting position (home offset phase) is fully configurable. Status information is given to indicate when the homing reference has been activated and when the homing routine has been completed.

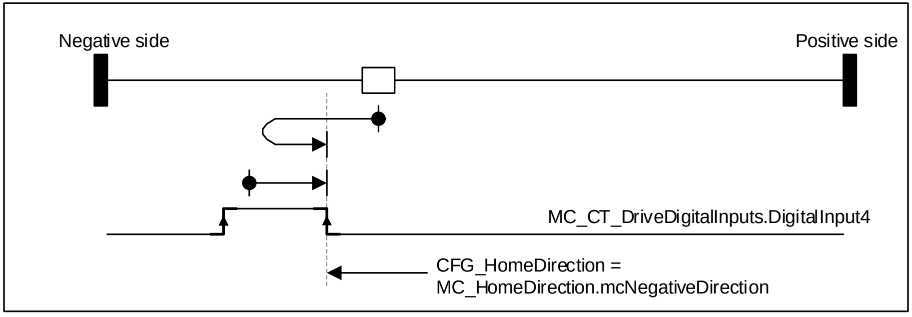
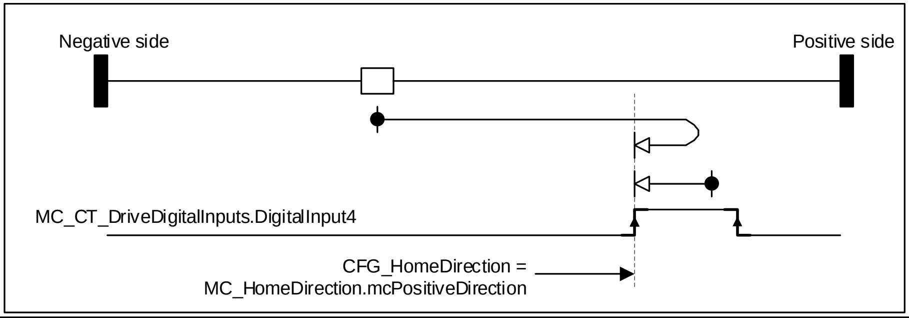


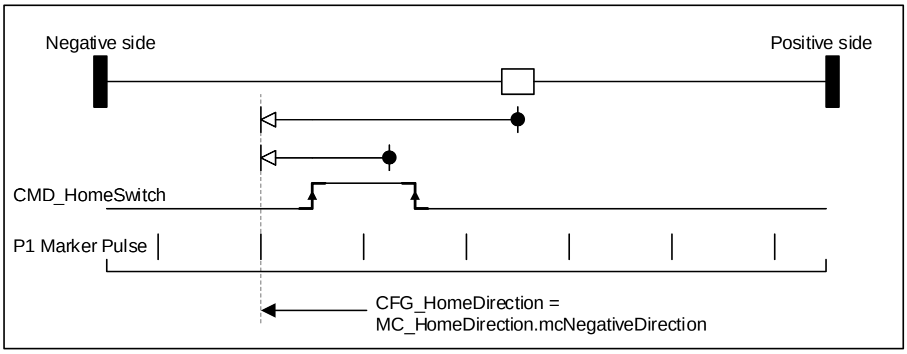
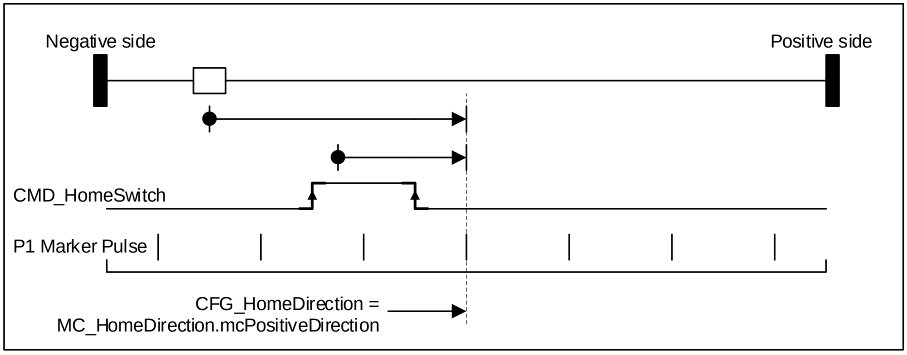
FB Parameter	**CMD_Home**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis, MC_CT_VirtualAxis**		
Description	Causes a homing routine to be executed, where: FALSE = No action TRUE = Begin homing **HINT:** When MC_CT_VirtualAxis is in use, the only homing method available is “HomeAtCurrentPosition”.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

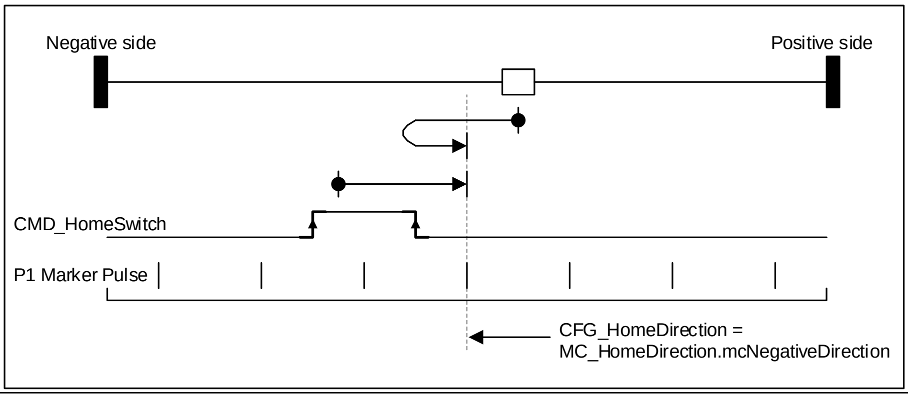
FB Parameter	**CMD_HomeSwitch**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Homing switch input that can be used in conjunction with the following homing modes as defined by CFG_HomeMode: **CFG_HomeMode.HomeToFbHomingSwitch (1)** Home to FB homing switch. **CFG_HomeMode.HomeToFbHomingSwitchThenIndexMarkerPulse (3)** Home to FB homing switch then index marker pulse. **CFG_HomeMode.HomeToFbHomingSwitchThenReverseIndexMarkerPulse (4)** Home to FB homing switch then reverse to index marker pulse. FALSE = Homing switch is not activated TRUE = Homing switch is activated		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CFG_HomeMode.HomeAtCurrentPosition (0)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home at current position** The axis remains stationary, and the position counter is reset to the value present at the CFG_HomeSearchPosition input. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input. 

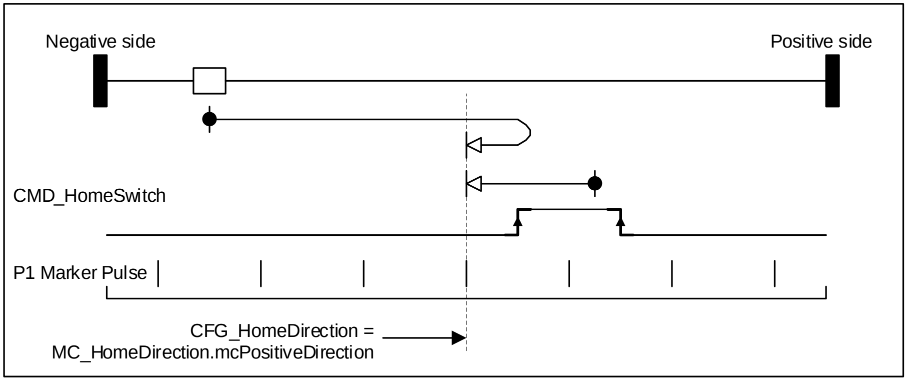
FB Parameter	**CFG_HomeMode.HomeToFbHomingSwitch (1)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to FB homing swich** The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the CMD_HomeSwitch signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. The home position is set once the CMD_HomeSwitch signal is no longer detected where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input. The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcNegativeDirection:  The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection: 

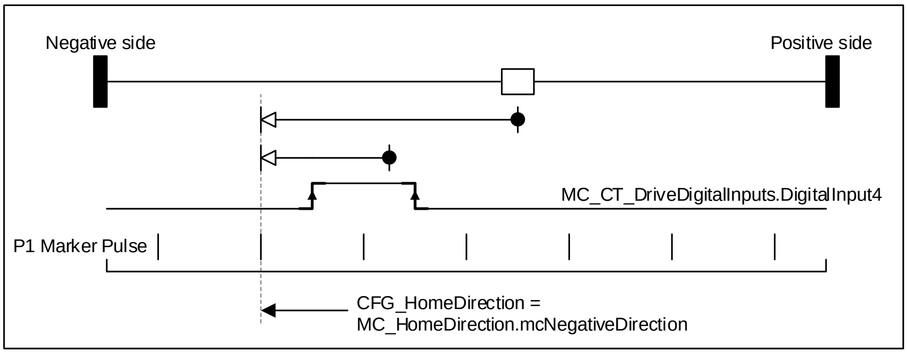
FB Parameter	**CFG_HomeMode.HomeToDriveHomingSwitch (2)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to drive homing switch** The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the drive digital input signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. The home position is set once the drive digital input signal is no longer detected where the axis is brought to a complete stop. The drive digital input source can be selected via the CFG_HomeDriveDigitalInput input where the data type is of MC_CT_DriveDigitalInputs (ENUM). For more information see section **5.5 MC_CT_DriveDigitalInputs**. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input. The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcNegativeDirection:  The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection: 

FB Parameter	**CFG_HomeMode. HomeToFbHomingSwitchThenIndexMarkerPulse (3)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to FB homing switch then index marker pulse** The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the CMD_HomeSwitch signal has been detected, the axis continues to move in the same direction, but the velocity will change to that specified at the CFG_HomeSearchBackoffSpeed input. Once the CMD_HomeSwitch signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcNegativeDirection:  The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection: 

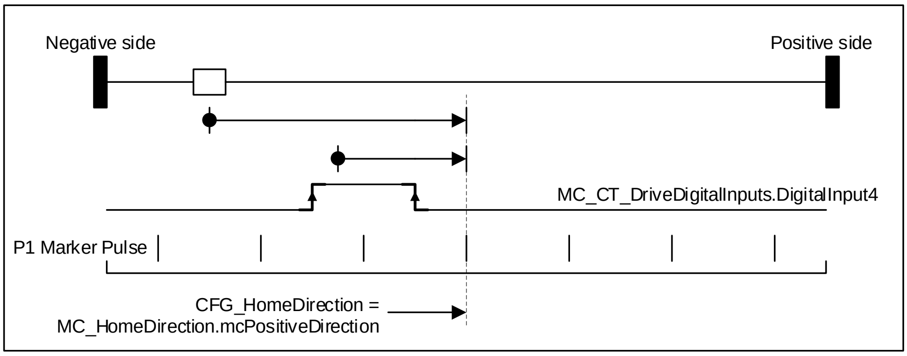
FB Parameter	**CFG_HomeMode. HomeToFbHomingSwitchThenReverseToIndexMarkerPulse (4)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to FB homing switch then reverse to index marker pulse** The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the CMD_HomeSwitch signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. Once the CMD_HomeSwitch signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input. The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcNegativeDirection: 

The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection:



FB Parameter	**CFG_HomeMode. HomeToDriveHomingSwitchThenIndexMarkerPulse (5)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to drive homing switch then index marker pulse** The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the drive digital input signal has been detected, the axis continues to move in the same direction, but the velocity will change to that specified at the CFG_HomeSearchBackoffSpeed input. Once the drive digital input signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The drive digital input source can be selected via the CFG_HomeDriveDigitalInput input where the data type is of MC_CT_DriveDigitalInputs (ENUM). For more information see section **5.5 MC_CT_DriveDigitalInputs**. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input. The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcNegativeDirection: 

The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection:



FB Parameter

CFG_HomeMode.
HomeToDriveHomingSwitchThenReverseIndexMarkerPulse (6)

Applies To

MC_CT_EGB_Axis, MC_CT_PositionAxis

Description

Specifies homing mode to be used when CMD_Home is set to TRUE.

Data type is of the MC_CT_HomingModes ENUM.

Home to drive homing switch then reverse to index marker pulse

The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed.

Once the drive digital input signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. Once the drive digital input signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop.

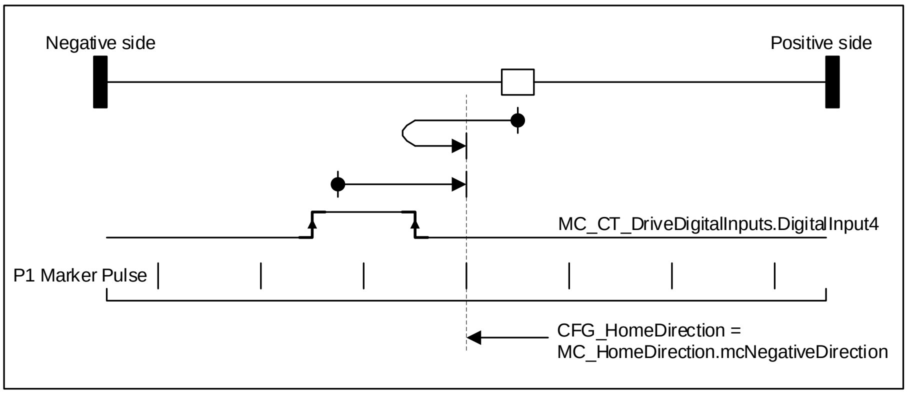
The drive digital input source can be selected via the CFG_HomeDriveDigitalInput input where the data type is of MC_CT_DriveDigitalInputs (ENUM). For more information see section **5.5 MC_CT_DriveDigitalInputs**.

The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs.

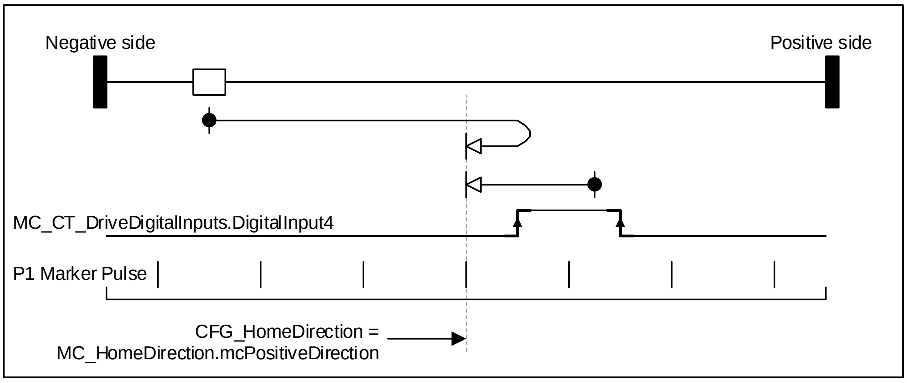
The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.

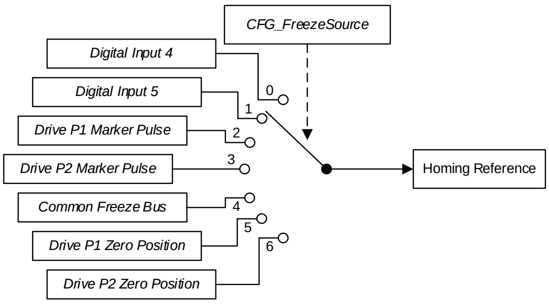
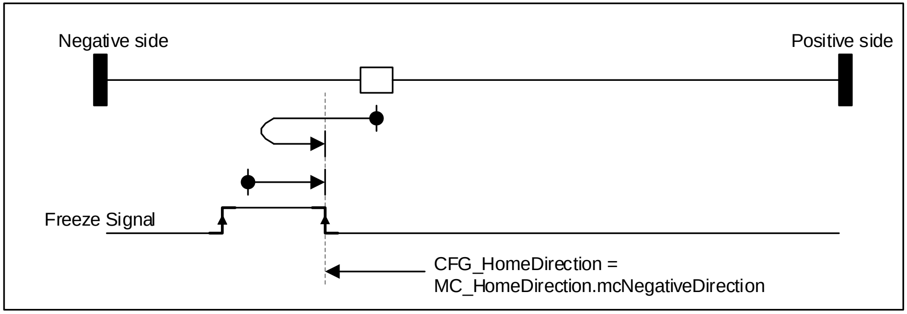
Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input.

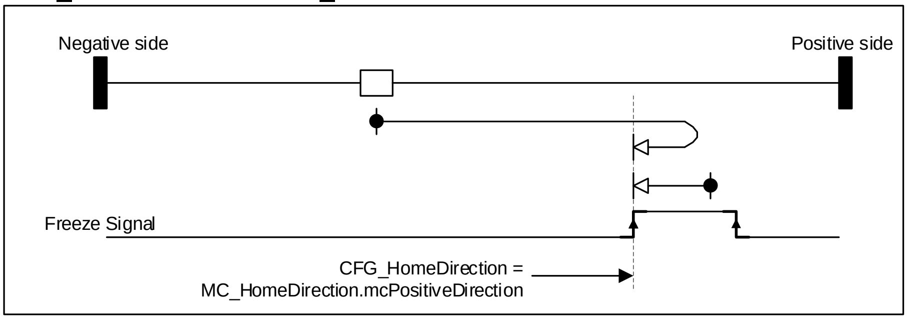
The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcNegativeDirection:



The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection



FB Parameter	**CFG_HomeMode.HomeToIndexFreeze (7)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to index freeze** The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. The axis uses the Drive F1 freeze input (Digital input 4 by default) freeze mechanism as the home switch input, to complete a 1 stage (search only) homing routine. Once the freeze signal has been detected a direction reversal is performed where the axis moves to the absolute position specified at the CFG_HomeOffsetPosition input (0 by default), where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. The freeze source can be modified via the CFG_FreezeSource function block input where the datatype accepted at this input is of type MC_CT_FreezeCaptureSource. For more information see section **5.12 MC_CT_FreezeCaptureSource**.  The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input. The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcNegativeDirection: 

	The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection: 

FB Parameter	**CFG_HomeMode.HomeToTorqueThreshold (8)**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**
Description	Specifies homing mode to be used when CMD_Home is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to torque threshold** The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. The axis uses a Percentage Load >= CFG_HomeTorqueThreshold condition to satisfy the homing switch input. Once this homing switch condition has been detected, the axis will perform a reversal where the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed to perform a 2 stage (Search and Backoff) homing routine. Once the Percentage Load >= CFG_HomeTorqueThreshold condition is no longer detected, the axis is brought to a complete stop and the home position is set. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input.

<table>
<tr><td>FB Parameter</td><td colspan="3">CFG_HomeMode.
HomeToTorqueThresholdThenReverseIndexMarkerPulse (9)</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_EGB_Axis, MC_CT_PositionAxis</td></tr>
<tr><td>Description</td><td colspan="3">Home to torque threshold then index marker pulse
The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed.
The axis uses a Percentage Load >= CFG_HomeTorqueThreshold condition to satisfy the homing switch input. Once this homing switch condition has been detected, the axis will perform a reversal where the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed to perform a 2 stage (Search and Backoff) homing routine.
Once the Percentage Load >= CFG_HomeTorqueThreshold condition is no longer detected, the home position is set at the rising edge of the next index marker pulse where the axis is then is brought to a complete stop.
are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs
The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.
Once homing has completed, the axis position is set to the value present at the CFG_HomeSearchPosition input.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>MC_CT_HomingModes</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>0</td><td>Maximum</td><td>9</td></tr>
</table>

FB Parameter	**CFG_HomeDirection**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Specifies the axis direction when homing begins. For example, to start the homing routine in the forward direction, CFG_HomeDirection = MC_Home_Direction.mcPositiveDirection or to start the homing routine in the reverse direction, CFG_HomeDirection = MC_Home_Direction.mcNegativeDirection. **HINT:** Although this input accepts the type MC_Home_Direction (ENUM), the only values accepted at this input are “**MC_HomeDirection.mcNegativeDirection**” and “**MC_HomeDirection.mcPositiveDirection**”.		
Usage	Input	**Data Type**	MC_Home_Direction
Unit	N/A	**Initial Value**	N/A
Minimum	mcPositiveDirection	**Maximum**	mcNegativeDirection

FB Parameter	**CFG_HomeSwitchLogic**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Specifies the polarity of the home switch logic using the MC_CT_SwitchLogic (ENUM) where: **MC_CT_SwitchLogic.PositiveLogic** The home switch input must be set TRUE to active the home switch. **MC_CT_SwitchLogic.NegativeLogic** The home switch input must be set FALSE to activate the home switch. For more information see section **5.18 MC_CT_SwitchLogic**.		
Usage	Input	**Data Type**	MC_CT_SwitchLogic
Unit	N/A	**Initial Value**	N/A
Minimum	PositiveLogic	**Maximum**	NegativeLogic

FB Parameter	**CFG_HomeTorqueThreshold**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Specifies the percentage load threshold for the “**MC_CT_HomingModes.HomeToTorqueThreshold**” and “**HomeToTorqueThresholdThenReverseIndexMarkerPulse**” homing modes. A drive percentage load value greater than or equal to the CFG_HomeTorqueThreshold value results in a home switch TRUE condition for the home to torque threshold homing modes. During homing the current limit for the drive is set to this value so that excessive torque is not applied to the motor when bumping into the end stops.		
Usage	Input	**Data Type**	LREAL
Unit	%	**Initial Value**	20.0
Minimum	>0.0	**Maximum**	100.0

FB Parameter	**CFG_HomeSearchPosition**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis,**		
Description	Specifies the position that the axis will be reset to once homing completes.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_HomePosition**		
Applies To	**MC_CT_VirtualAxis**		
Description	Specifies the position that the axis will be reset to once homing completes.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**CFG_HomeSearchSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis,**		
Description	Specifies the speed used for the home switch search phase of a homing routine.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_HomeSearchBackoffSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis,**		
Description	Specifies the speed used for backing from the home switch, this should typically be 10% of the velocity specified for CFG_HomeSearchSpeed.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	100.0
Minimum	>0.0	**Maximum**	LREAL MAX

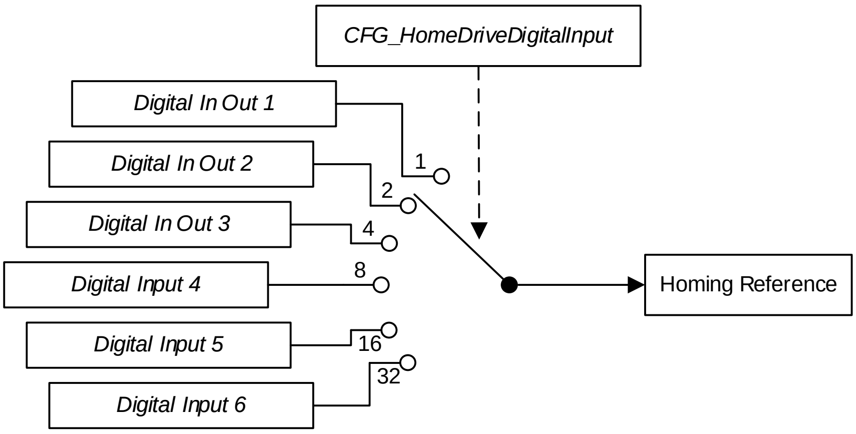
FB Parameter	**CFG_HomeOffsetPosition**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis,**		
Description	Specifies the final absolute resting position of the axis used during the offset phase of a homing routine.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_HomeOffsetMaxSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis,**		
Description	Specifies the speed used for moving to the offset position specified at the CFG_HomeOffsetPosition input used during the offset phase of a homing routine.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_HomeAccel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Specifies the maximum acceleration rate to be used when homing, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_HomeDecel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Specifies the maximum deceleration rate to be used when homing, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_HomeJerk**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis**		
Description	Specifies the jerk rate to be used when homing, in Application units/s^3. A value of 0 will result in linear ramps.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

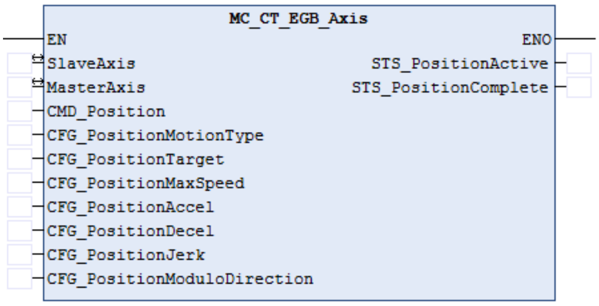
<table>
<tr><th>FB Parameter</th><th colspan="3">CFG_HomeDriveDigitalInput</th></tr>
<tr><td>**Applies To**</td><td colspan="3">**MC_CT_EGB_Axis, MC_CT_PositionAxis**</td></tr>
<tr><td>**Description**</td><td colspan="3">Selects which drive signal is used as the drive homing switch used in homing modes “**HomeToDriveHomingSwitch**”, “**HomeToDriveHomingSwitchThenIndexMarkerPulse**” and “**HomeToDriveHomingSwitchThenReverseIndexMarkerPulse**”.
Data type is of the MC_CT_DriveDigitalInputs (ENUM) where:
MC_CT_DriveDigitalInputs.DigitalInOut1
Digital In Out 1 is selected.
MC_CT_DriveDigitalInputs.DigitalInOut2
Digital In Out 2 is selected.
MC_CT_DriveDigitalInputs.DigitalInOut3
Digital In Out 3 is selected.
MC_CT_DriveDigitalInputs.DigitalInput4
Digital Input 4 is selected.
MC_CT_DriveDigitalInputs.DigitalInput5
Digital Input 5 is selected.
MC_CT_DriveDigitalInputs.DigitalInput6
Digital Input 6 is selected.

For more information see section **5.5 MC_CT_DriveDigitalInputs**.</td></tr>
<tr><td>**Usage**</td><td>Input</td><td>**Data Type**</td><td>MC_CT_DriveDigitalInputs</td></tr>
<tr><td>**Unit**</td><td>N/A</td><td>**Initial Value**</td><td>N/A</td></tr>
<tr><td>**Minimum**</td><td>DigitalInOut1</td><td>**Maximum**</td><td>DigitalInput6</td></tr>
</table>

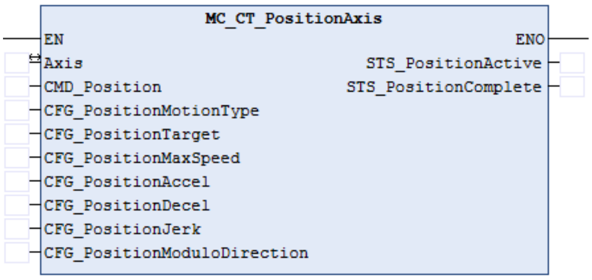
<table>
<tr><td>FB Parameter</td><td colspan="3">STS_HomeActive</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis</td></tr>
<tr><td>Description</td><td colspan="3">Indicates that home is the active motion reference, where:
FALSE = Home is not the active motion reference.
TRUE = Home is the active motion reference.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

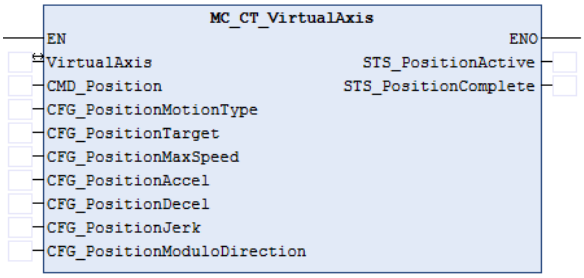
<table>
<tr><td>FB Parameter</td><td colspan="3">STS_HomeComplete</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis</td></tr>
<tr><td>Description</td><td colspan="3">Indicates that homing has completed, where:
FALSE = The homing routine has not completed or is in progress.
TRUE = The homing routine has completed successfully.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

4.1.2.7 Position

Function block I/O that is common between the motion interfaces and categorised as “Position” functionality is shown in the diagrams below. The interface has been simplified so only the relevant inputs and outputs are shown.







HINT: The “Axis” inputs are shown in the above diagrams because items in the scope “VAR_IN_OUT” can’t be hidden in an LD diagram.

There are 5 positioning modes available which are selectable via the CFG_PositionMotionType input with the enumeration MC_CT_PositionMotionTypes:

MC_CT_PositionModeTypes.Absolute allows the user to move the axis to a specific position defined by CFG_PositionTarget. The movement is started when the CMD_Position input is set TRUE.

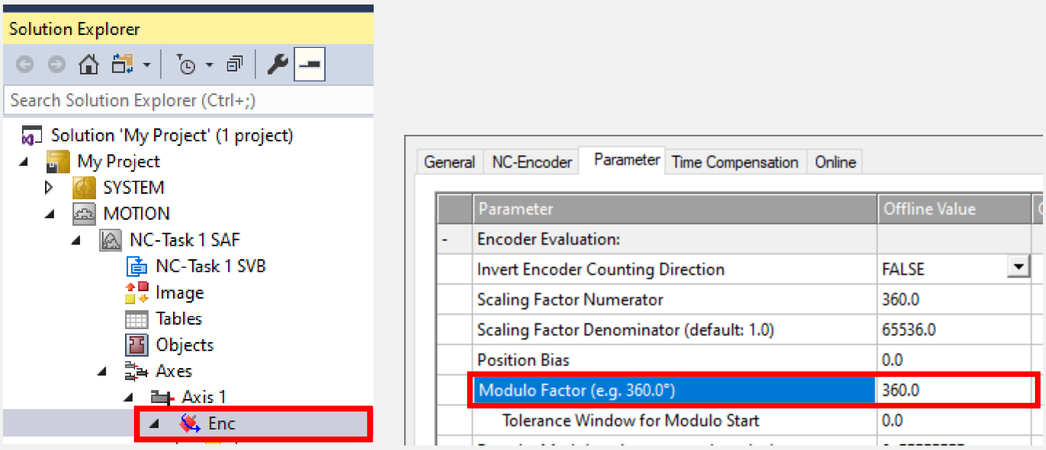
MC_CT_PositionModeTypes.Relative allows the user to move the axis by a set distance from the current position. The movement is relative to the axis’s position at the time the relative reference was applied, where the distance is defined by CFG_PositionTarget, where the direction is determined by the sign of this value, e.g. positive = forward movement, negative = backward movement. Motion is started when CMD_Position input is set TRUE. Once the movement has completed, the distance can be repeated by toggling the CMD_Position input to FALSE and then TRUE again, or by setting the distance to 0 and then to value <>0. The user is recommended to use this mode to achieve a relative indexing system.

MC_CT_PositionModeTypes.AutoIndex continuously indexes the axis in one direction whilst CMD_Position is TRUE. The index distance is defined at the CFG_PositionTarget input. The user recommended to use this mode for motor performance tuning or initial machine setup.

MC_CT_PositionModeTypes.AutoReciprocate continuously reciprocates the axis alternating between two directions whilst the CMD_Position input is TRUE. The distance to move back and forth by is defined by CFG_PositionTarget. It is recommended to use this mode for motor performance tuning. The user recommended to use this mode for motor performance tuning or initial machine setup.

MC_CT_PositionModeTypes.Modulo modulo mode allows the user to make movements within a rotary range e.g. the range of positions available could be within 1 revolution of a rotary actuator. The profile handling in rotary mode is selected using CFG_PositionModuloDirection e.g. shortest distance, negative only, positive only, current direction and fastest.

HINT: The modulo rollover factor is defined in the encoder parameter area which can be reached via the project tree under "**Project Name**" > "**Motion**" > "**NC-Task SAF**" > "**Axes**" > "**Axis**" > "**Enc**" and then selecting the "**Parameter**" tab:



Solution Explorer

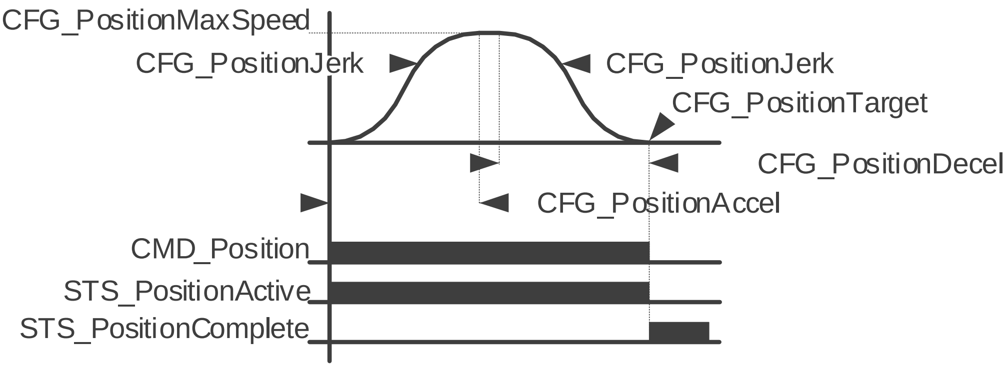
Search Solution Explorer (Ctrl+;)

- Solution 'My Project' (1 project)
  - My Project
    - SYSTEM
    - MOTION
      - NC-Task 1 SAF
        - NC-Task 1 SVB
        - Image
        - Tables
        - Objects
        - Axes
          - Axis 1
            - Enc

General | NC-Encoder | Parameter | Time Compensation | Online

	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	360.0
	Scaling Factor Denominator (default: 1.0)	65536.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	360.0
	Tolerance Window for Modulo Start	0.0

Status information is available to allow the used to detect when the position reference is active, when the target has been reached, and if the position has not been reached e.g. CMD_Position = TRUE, STS_PositionActive = TRUE and STS_PositionComplete = FALSE.



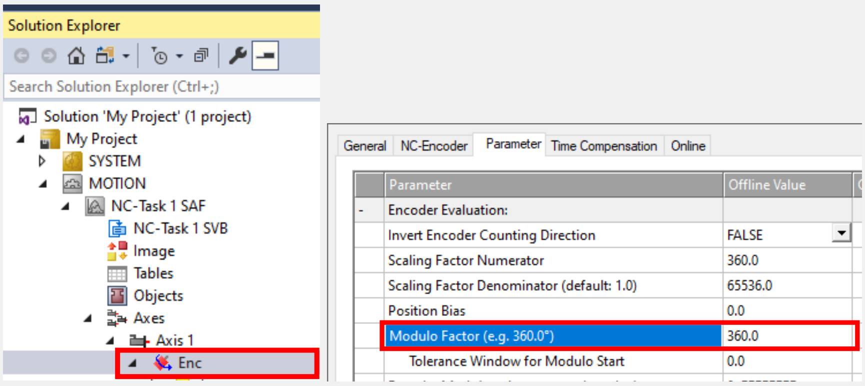
FB Parameter	**CMD_Position**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Causes the axis to move to the position specified at the CFG_PositionTarget input, where: FALSE = No action TRUE = Run selected position reference.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CFG_PositionMotionType**
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**
Description	Defines the type of positional movement. Data type is of the MC_CT_PositionMotionTypes ENUM where: **MC_CT_PositionModeTypes.Absolute** When the CMD_Position input is set TRUE the axis will move to the absolute position specified at the CFG_Position input. **MC_CT_PositionModeTypes.Relative** On a rising edge of the CMD_Position input, the axis will move incrementally from its current position by the distance specified at the CFG_Position input, where a positive value will result in a forward movement and a negative value in a backward movement. **MC_CT_PositionModeTypes.AutoIndex** When the CMD_Position input is set TRUE the axis will continually index by the distance specified at the CFG_Position input, where a positive value will result in a forward movement and a negative value in a backward movement. **HINT:** The axis will be brought to a stop as soon as the CMD_Position input is set FALSE meaning the axis may stop before the current movement has completed. This reference is intended for use as a tuning aid. **MC_CT_PositionModeTypes.AutoReciprocate** When the CMD_Position input is set TRUE the axis will continually reciprocate by the distance specified at the CFG_Position input, where a positive value will result in an initial forward movement and a negative value in an initial backward movement. **HINT:** The axis will be brought to a stop as soon as the CMD_Position input is set FALSE meaning the axis may stop before the current movement has completed.

MC_CT_PositionModeTypes.Modulo

On a rising edge of the CMD_Position input, the axis will move to the modulo position specified at the CFG_Position input using the modulo direction specified at the CFG_PositionModuloDirection input.

HINT: The modulo rollover factor is defined in the encoder parameter area which can be reached via the project tree under “**Project Name**” > “**Motion**” > “**NC-Task SAF**” > “**Axes**” > “**Axis**” > “**Enc**” and then selecting the “**Parameter**” tab:



	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	360.0
	Scaling Factor Denominator (default: 1.0)	65536.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	360.0
	Tolerance Window for Modulo Start	0.0

Usage	Input	**Data Type**	MC_CT_PositionMotionTypes
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4

FB Parameter	**CFG_PositionTarget**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Specifies the position target / distance, in Application units.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**CFG_PositionMaxSpeed**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Specifies the target speed to be used when positioning, in Application units/s.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_PositionAccel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Specifies the maximum acceleration rate to be used when positioning, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_PositionDecel**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Specifies the maximum deceleration rate to be used when positioning, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_PositionJerk**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Specifies the jerk rate to be used when positioning, in Application units/s^3. A value of 0 will result in linear ramps.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

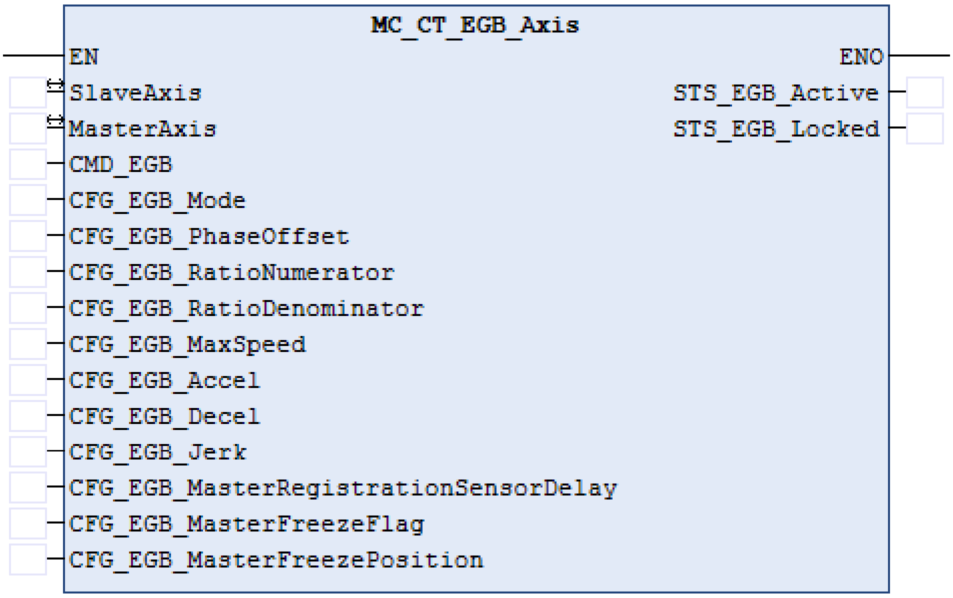
FB Parameter	**CFG_PositionModuloDirection**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Sets the direction of travel when the axis has been configured to be a modulo axis and CFG_PositionMotionType = 0 (Absolute), where: **CFG_PositionMotionType.MC_Positive_Direction** Moves in positive direction. **CFG_PositionMotionType.MC_Shortest_Way** Selects the direction according to the shortest distance. **CFG_PositionMotionType.MC_Negative_Direction** Moves in negative direction. **CFG_PositionMotionType.MC_Current_Direction** Keeps the current direction to reach the target. **CFG_PositionMotionType.MC_Undefined_Direction** Undefined direction of movement		
Usage	Input	**Data Type**	MC_Direction
Unit	N/A	**Initial Value**	MC_Shortest_Way
Minimum	0	**Maximum**	3

FB Parameter	**STS_PositionActive**		
Applies To	**MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis**		
Description	Indicates that position is the active motion reference, where: FALSE = Position is not the active motion reference. TRUE = Position is the active motion reference.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><td>FB Parameter</td><td colspan="3">STS_PositionComplete</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_VirtualAxis</td></tr>
<tr><td>Description</td><td colspan="3">Indicates that the position target has been reached, where:
FALSE = The position target has not been reached.
TRUE = The position target has been reached.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

4.1.2.8 EGB

The diagrams below show the function block I/O that is categorised as “EGB” functionality is shown in the diagrams below. The interface has been simplified so only the relevant inputs and outputs are shown.



HINT: The “Axis” inputs are shown in the above diagrams because items in the scope “VAR_IN_OUT” can’t be hidden in an LD diagram.

The EGB reference allows the user to make an Electronic Gearbox or Digital lock or Gear In application between a Slave Axis and Master Axis.

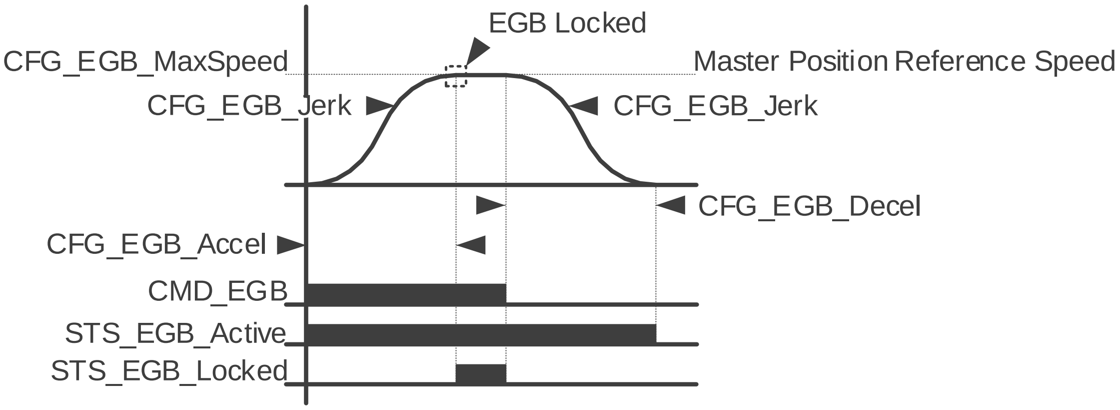
The EGB reference can run in the following modes:

- Non-rigid lock, (CFG_EGB_Mode = MC_CT_EGB_Modes.NonRigidLock), where the EGB slave axis will catch up with the speed of the master axis position reference, and then become locked.
- Rigid lock, (CFG_EGB_Mode = MC_CT_EGB_Modes.RigidLock), where the EGB slave axis will catch up with the speed and relative position of the master axis position reference, and then become locked.
- Registration, (CFG_EGB_Mode = MC_CT_EGB_Modes.Registration), where a EGB master axis freeze event will cause the EGB slave axis to catch up with the speed and relative position of the master axis position reference that was captured when the freeze event took place, and then become locked.

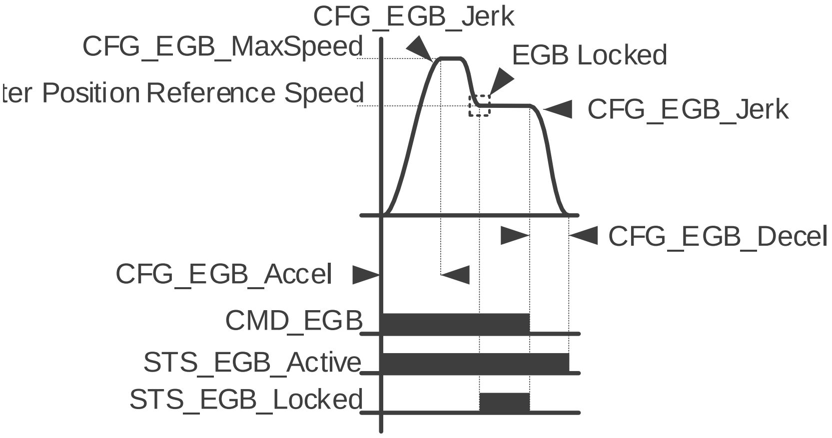
The EGB functionality has its own motion profile limits e.g. maximum speed, maximum acceleration rate, maximum deceleration rate, jerk rate and EGB locking ratio numerator / denominator.

Status information is available to allow the used to detect when the EGB reference is active and when the EGB has become locked to the master encoder position reference.

CMD_EGB_LockMode = 0 (Non-rigid lock)



CMD_EGB_LockMode = 1 (Rigid lock)



FB Parameter	**CMD_EGB**		
Applies To	**MC_CT_EGB_Axis**		
Description	Causes the SlaveAxis to form an EGB lock with the MasterAxis position, where: FALSE = No action TRUE = Form an EGB lock with the MasterAxis position.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CFG_EGB_Mode**		
Applies To	**MC_CT_EGB_Axis**		
Description	Defines the type of electronic gearbox locking Data type is of the MC_CT_EGB_Modes ENUM where: **MC_CT_EGB_Modes.NonRigidLock** The EGB slave axis will catch up with the speed of the master axis position reference and then become locked. **MC_CT_EGB_Modes.RigidLock** The EGB slave axis will catch up with the speed and relative position of the master axis position reference and then become locked. **MC_CT_EGB_Modes.NonRigidLock.Registration** A EGB master axis freeze event will cause the EGB slave axis to catch up with the speed and relative position of the master axis position reference that was captured when the freeze event took place and then become locked.		
Usage	Input	**Data Type**	MC_CT_EGB_Modes
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	2

FB Parameter	**CFG_EGB_PhaseOffset**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies a phase offset distance that is added to the slave axis position. The position is absolute to 0.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_EGB_RatioNumerator**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies the EGB ratio numerator used for locking the slave axis to master axis position reference. Slave speed = master speed x CFG_EGB_RatioNumerator / CFG_EGB_RatioDenominator		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	1000
Minimum	1	**Maximum**	2147483647

FB Parameter	**CFG_EGB_RatioDenominator**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies the EGB ratio denominator used for locking the slave axis to master axis position reference. Slave speed = master speed x CFG_EGB_RatioNumerator / CFG_EGB_RatioDenominator		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	1000
Minimum	1	**Maximum**	2147483647

FB Parameter	**CFG_EGB_MaxSpeed**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies the maximum slave axis speed permitted when synchronising to the master axis in Application units/s. For NonRigidLock speed must be a minimum of the master speed x CFG_EGB_RatioNumerator / CFG_EGB_RatioDenominator. For RigidLock and Registration mode it should be set to a minimum of 110% of master speed x CFG_EGB_RatioNumerator / CFG_EGB_RatioDenominator.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_EGB_Accel**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies the maximum slave axis acceleration rate permitted when synchronising to the master axis in Application units/s². For RigidLock or Registration mode, this acceleration limit applies when the slave axis is attempting to catch up to the relative position of the master axis for locking.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_EGB_Decel**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies the maximum slave axis deceleration rate permitted when synchronising to the master axis in Application units/s². For RigidLock or Registration mode, this deceleration limit applies when the slave axis is attempting to catch up to the relative position of the master axis for locking.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_EGB_Jerk**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies the maximum slave axis jerk permitted when synchronising to the master axis in Application units/s³. For RigidLock or Registration mode, this jerk limit applies when the slave axis is attempting to catch up to the relative position of the master axis for locking. A value of 0 will result in linear ramps.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s³	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**CFG_EGB_MasterRegistrationSensorDelay**		
Applies To	**MC_CT_EGB_Axis**		
Description	Specifies the time in μs for the registration sensor to process the registration mark, (throughput delay), and provide a change in electrical output e.g. 0V to 24V transition.		
Usage	Input	**Data Type**	DINT
Unit	μs	**Initial Value**	0
Minimum	0	**Maximum**	MAX

FB Parameter	**CFG_EGB_MasterFreezeFlag**		
Applies To	**MC_CT_EGB_Axis**		
Description	An input to capture STS_FreezeFlag from of the master axis. This input is only used when CFG_EGB_Mode = Registration.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	0	**Maximum**	TRUE

FB Parameter	**CFG_EGB_MasterFreezePosition**		
Applies To	**MC_CT_EGB_Axis**		
Description	An input to capture STS_FreezePosition from of the master axis. This input is only used when CFG_EGB_Mode = Registration.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**STS_EGB_Active**		
Applies To	**MC_CT_EGB_Axis**		
Description	Indicates that EGB is the active motion reference, where: FALSE = EGB is not the active motion reference. TRUE = EGB is the active motion reference.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><td>FB Parameter</td><td colspan="3">STS_EGB_Locked</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_EGB_Axis</td></tr>
<tr><td>Description</td><td colspan="3">Indicates that the EGB target has become locked to the master axis reference, where:
FALSE = The EGB reference has not locked to the master axis reference.
TRUE = The EGB reference has locked to the master axis reference.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

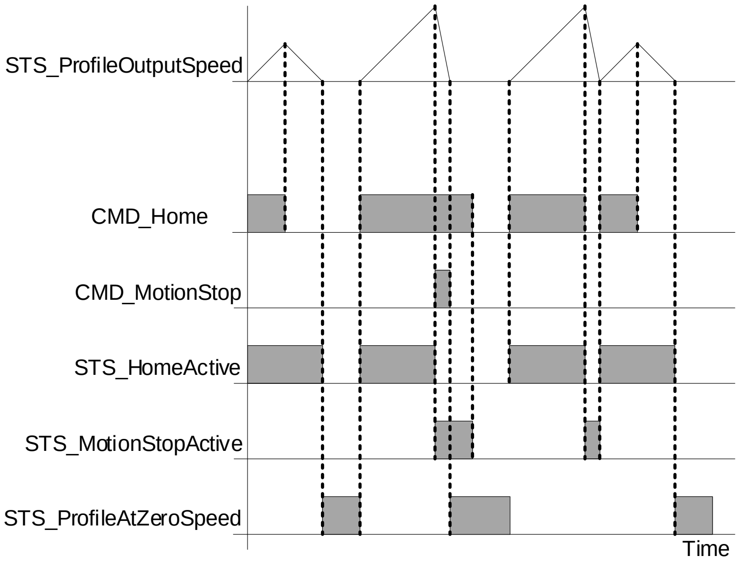
4.1.3 Timing Diagrams

The following section shows timing diagrams for each of the CMD inputs that are common between the 4 main motion interface function blocks (MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis and MC_CT_VirtualAxis).

4.1.3.1 CMD_Home

CMD_Home applies to the MC_CT_EGB_Axis, MC_CT_PositionAxis and MC_CT_VirtualAxis motion interfaces.

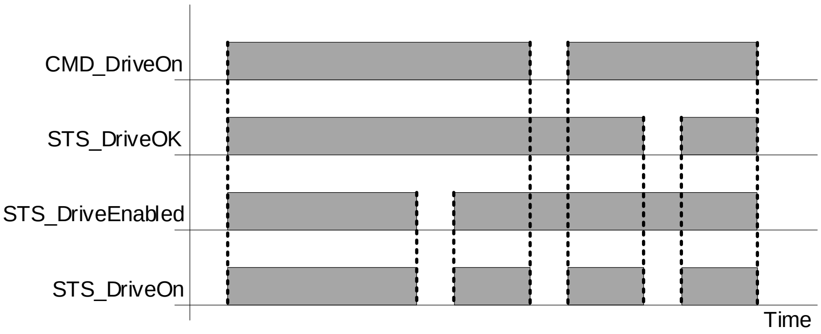
The illustration below shows the interaction between CMD_Home, CMD_MotionStop and the resultant motion profile (STS_ProfileOutputSpeed) and output conditions that appear at the "STS_" outputs relating to home functionality.



4.1.3.2 CMD_DriveOn

CMD_Home applies to the MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis and MC_CT_VirtualAxis motion interfaces.

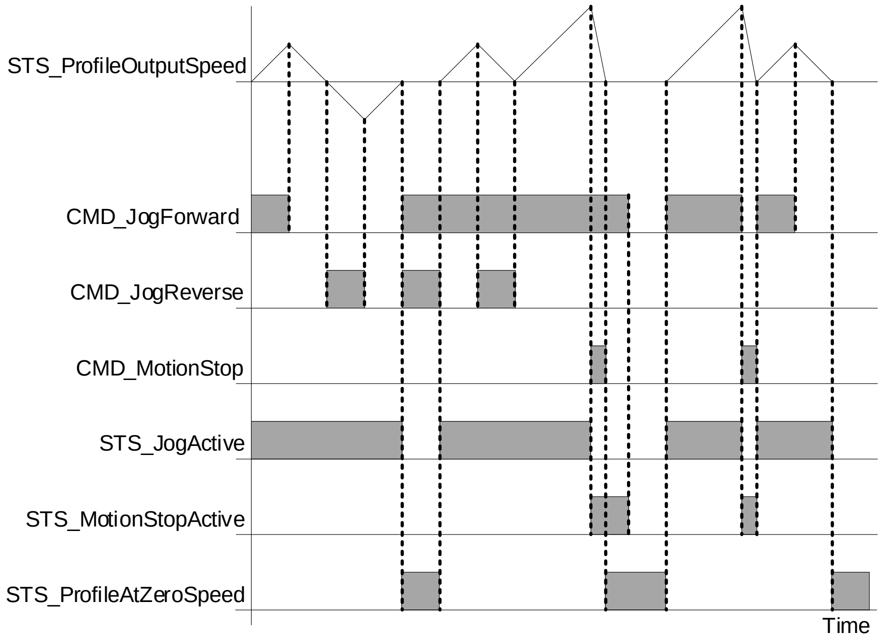
The illustration below shows the interaction between CMD_DriveOn and the resultant output conditions that appear at the "STS_" outputs.



4.1.3.3 CMD_JogForward and CMD_JogReverse

CMD_JogForward and CMD_JogReverse applies to the MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis and MC_CT_VirtualAxis motion interfaces.

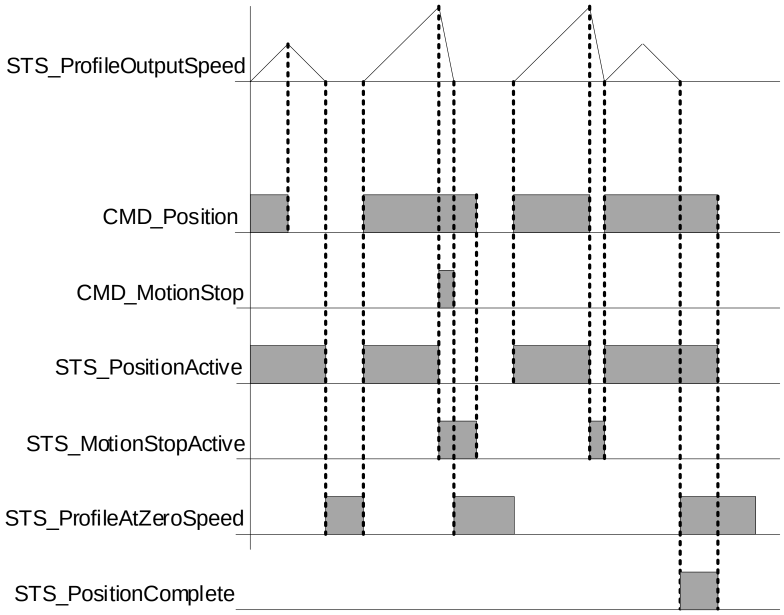
The illustration below shows the interaction between CMD_JogForward, CMD_JogReverse, CMD_MotionStop and the resultant motion profile (STS_ProfileOutputSpeed) and output conditions that appear at the “STS_” outputs relating to jogging functionality.



4.1.3.4 CMD_Position

CMD_Position applies to the MC_CT_EGB_Axis, MC_CT_PositionAxis and MC_CT_VirtualAxis motion interfaces.

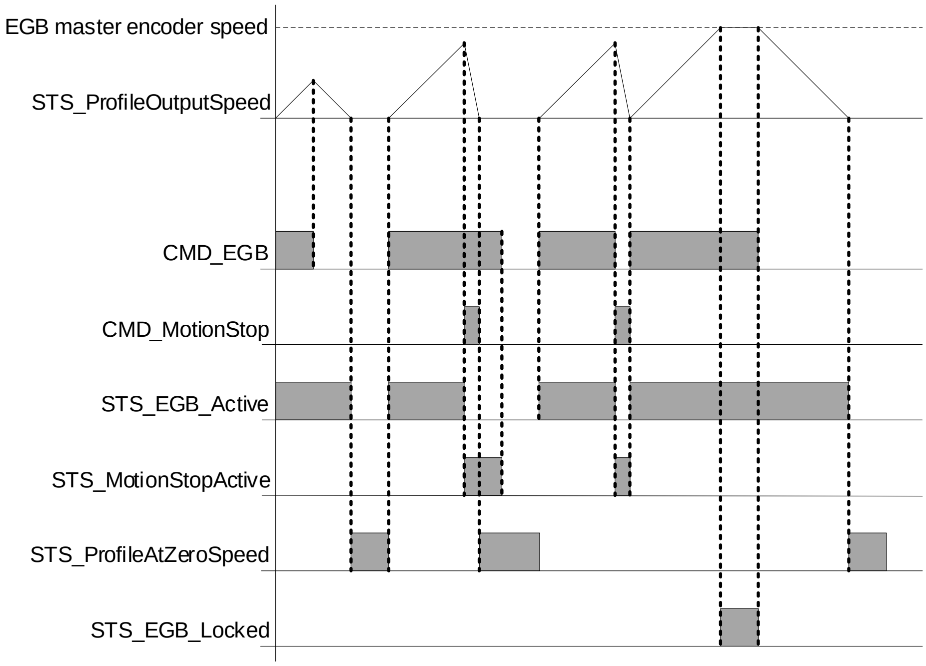
The illustration below shows the interaction between CMD_Position, CMD_MotionStop and the resultant motion profile (STS_ProfileOutputSpeed) and output conditions that appear at the “STS_” outputs relating to position functionality.



4.1.3.5 CMD_EGB

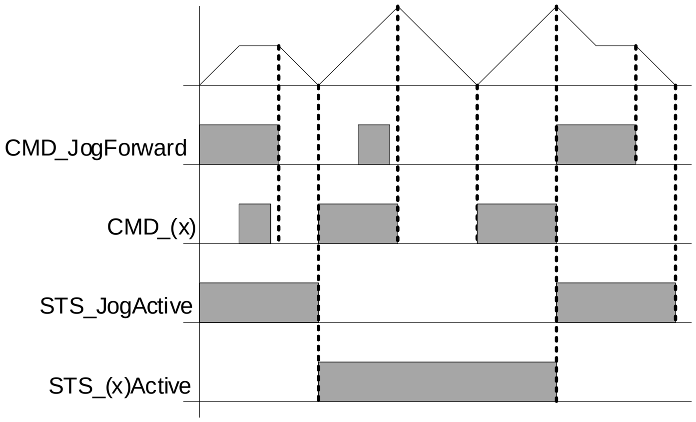
CMD_EGB applies to the MC_CT_EGB_Axis motion interface.

The illustration below shows the interaction between CMD_EGB, CMD_MotionStop and the resultant motion profile (STS_ProfileOutputSpeed) and output conditions that appear at the “STS_” outputs relating to EGB functionality.



4.1.3.6 Behaviour when more than one reference is selected

The illustration below shows the interaction between motion commands, when mor than one motion command is selected.



(x) = CMD_Home, CMD_Position, CMD_EGB
CMD_Speed as appropriate.

The active CMD reference will remain active regardless of other CMD references that are subsequently selected until the active CMD reference has been released, at which point next CMD reference will take effect.

4.2 Master Reference Functionality

This group of function blocks are suitable for use when one or more axis is being used to generate a master position reference that is then consumed by a slave axis in a master follower application like EGB, or Digital Lock, or MC_GearIn.

Filtering capability is built in to both MC_CT_AxisFilter and MC_CT_AxisSelector offering the user the choice between a windowing average filter, which gives the smoothest response at constant speed but is less accurate when the speed is changing or a weighted average filter, which gives a less smooth response at constant speed but is more accurate when the speed is changing. Phase lag compensation is offered for both filter types.

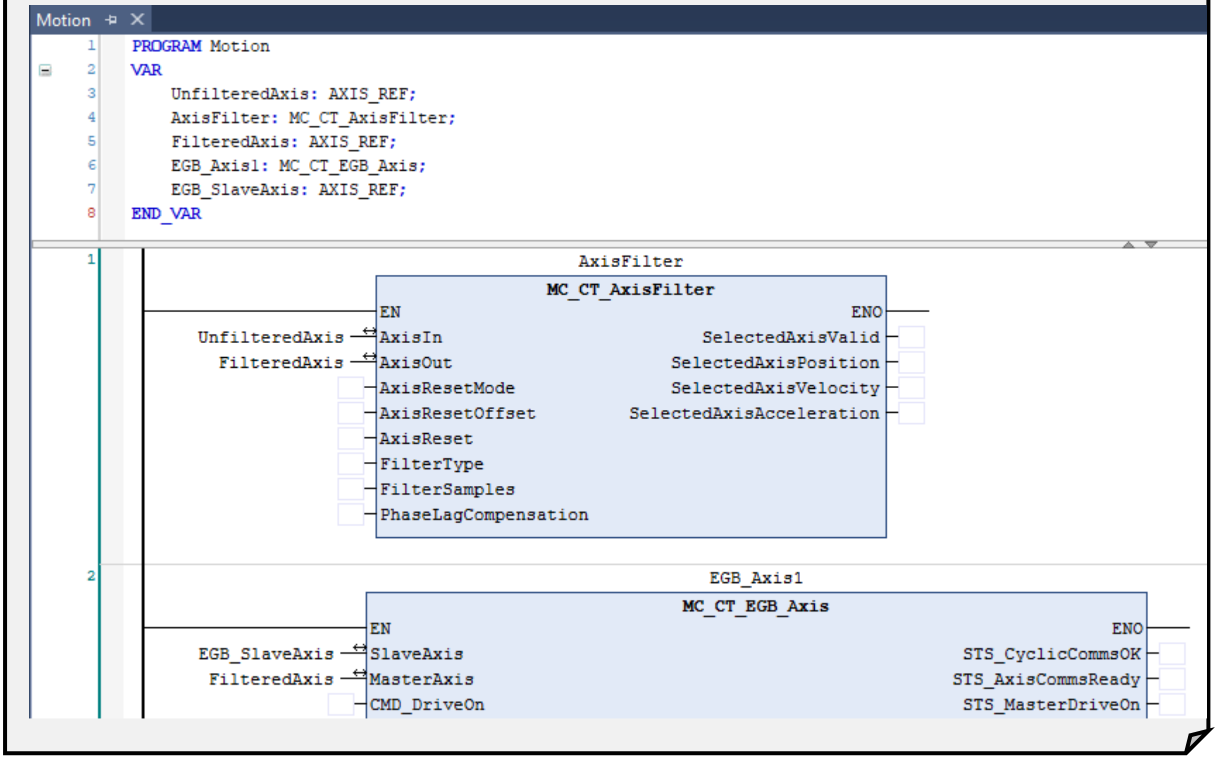
4.2.1 MC_CT_AxisFilter

This function block is used to filter an NC axis position, where the resulting AxisOut may be used as the master reference for an MC_CT_EGB_Axis.

The master axis to filter is connected to the AxisIn input and the AxisOut output is connected to an instance of AXIS_REF / NC axis that will later be connected to the MasterAxis input of an EGB or MC_GearIn interface such as MC_CT_EGB_Axis.

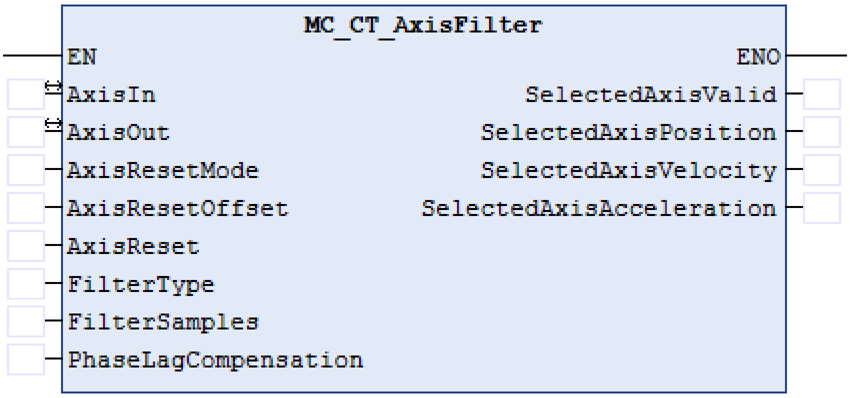
The position value should only be reset (AxisReset) when the slave axis is not actively following the master reference since large position changes of the AxisOut value are possible as a result of the values at the AxisResetMode and AxisResetOffset inputs.

HINT: The example below illustrates how MC_CT_AxisFilter can be used with MC_CT_EGB_Axis.



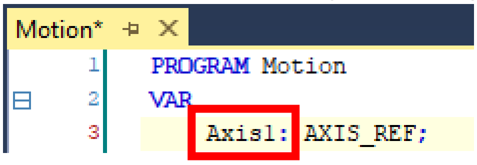
```
PROGRAM Motion
VAR
    UnfilteredAxis: AXIS_REF;
    AxisFilter: MC_CT_AxisFilter;
    FilteredAxis: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    EGB_SlaveAxis: AXIS_REF;
END_VAR
```

4.2.1.1 FB Diagram



4.2.1.2 Inputs / Outputs

FB Parameter	**AxisIn**		
Applies To	**MC_CT_AxisFilter**		

Description	The FB interface to the AXIS_REF and NC axis to be filtered, e.g. an unfiltered master encoder reference.
	This function block can be used with a physical drive axis, virtual drive axis, or an encoder axis.
	This input/output must be assigned to the instance name of "AXIS_REF" for the axis to be controlled. Naming the slave axis something more appropriate like "Axis1" as shown in the image below is good practice.
	
	
	`PROGRAM Motion`
	`VAR`
	`    Axis1: AXIS_REF;`		

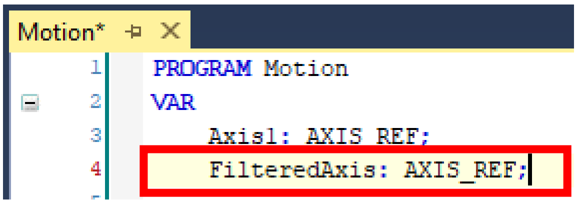
Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

FB Parameter	**AxisOut**
Applies To	**MC_CT_AxisFilter**

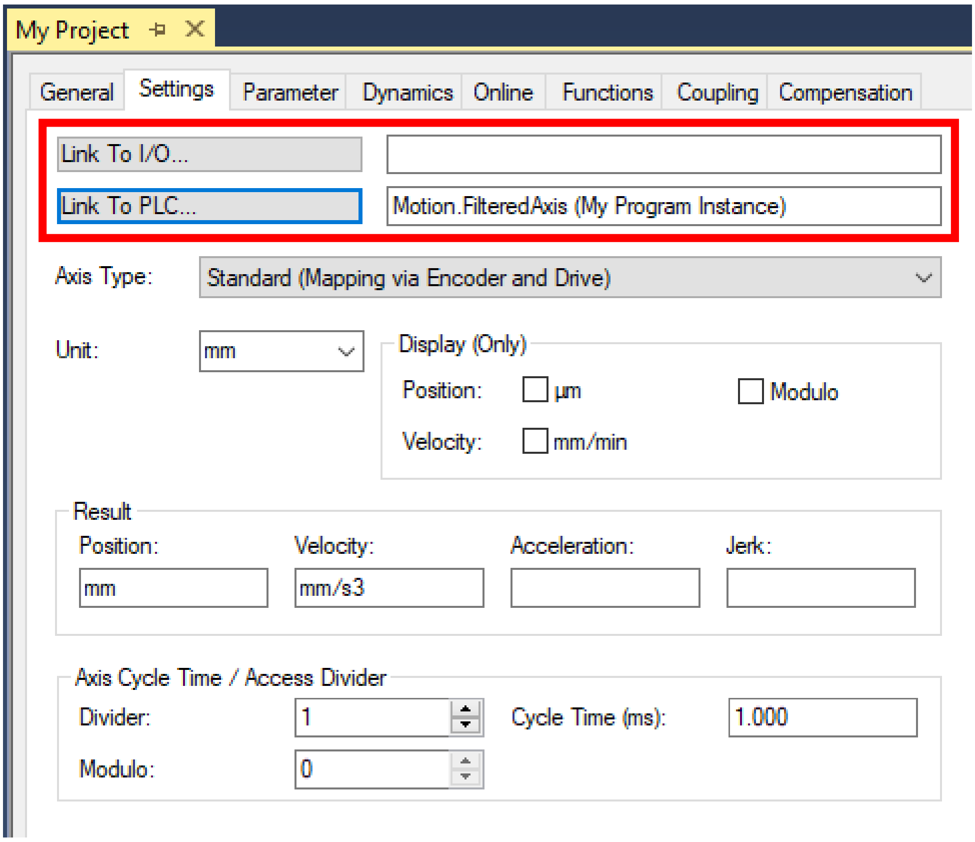
Description

The filtered output of AxisIn e.g. e.g. the final filtered encoder reference to be consumed at the MasterAxis input of an MC_CT_EGB_Axis or an MC_GearIn function block.

This must be connected to an isolated instance of AXIS_REF that is not linked to drive communications (I/O):



```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    FilteredAxis: AXIS_REF;
```



Link To PLC...: Motion.FilteredAxis (My Program Instance)

Axis Type: Standard (Mapping via Encoder and Drive)

Unit: mm

Display (Only)

Position: ☐ µm ☐ Modulo

Velocity: ☐ mm/min

Result

Position:	Velocity:	Acceleration:	Jerk:
mm	mm/s3		

Axis Cycle Time / Access Divider

Divider: 1

Cycle Time (ms): 1.000

Modulo: 0

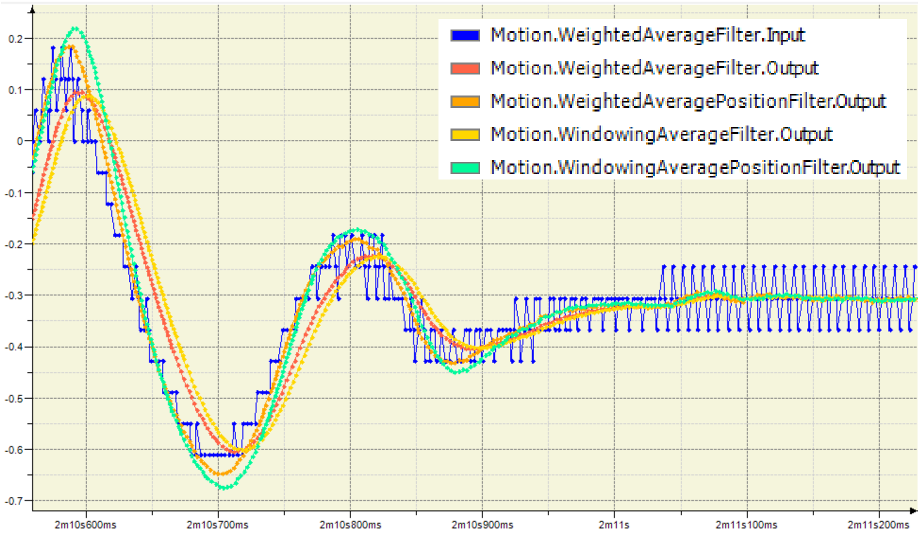
Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.2.1.3 Inputs

FB Parameter	**AxisResetMode**		
Applies To	**MC_CT_AxisFilter**		
Description	Defines the way that the position that the position accumulator is reset to when AxisReset = TRUE. Data type is of the MC_CT_AxisMode ENUM where: **MC_CT_AxisMode.Absolute (0)** Relative means that the output position integrator will be reset to 0 + ResetOffset when AxisReset = TRUE (Rising Edge). **MC_CT_AxisMode.Relative (1)** Absolute means that the output position integrator will be reset to the selected master absolute position + ResetOffset when AxisReset = TRUE (Rising Edge).		
Usage	Input	**Data Type**	MC_CT_AxisMode
Unit	N/A	**Initial Value**	FALSE
Minimum	0	**Maximum**	1

FB Parameter	**AxisResetOffset**		
Applies To	**MC_CT_AxisFilter**		
Description	The offset distance to be added to the pre-filter position during an axis reset.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	FALSE
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**AxisReset**		
Applies To	**MC_CT_AxisFilter**		
Description	On a FALSE to TRUE transition, the position will be reset according to the setting of AxisResetMode and AxisResetOffset. **HINT:** It is recommended to disengage the slave axis from the master reference during an axis reset since the position output can change by a large amount.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**FilterType**
Applies To	**MC_CT_AxisFilter**
Description	Defines the type of positional movement. Data type is of the MC_CT_FilterTypes ENUM where: **MC_CT_FilterTypes.No Filter** No Filter is selected. **MC_CT_FilterTypes.Windowing Average** The windowing average filter gives the smoothest response at constant speed but is less accurate when the speed is changing. **MC_CT_FilterTypes Weighted Average** The weighted average filter gives a less smooth response at constant speed but is more accurate when the speed is changing. The data trace below shows the effect each filter mode has on the raw encoder feedback (Motion.WeightedAverageFilter.Input). **HINT:** “Motion.WeightedAveragePositionFilter.Output” and “Motion.WindowingAveragePositionFilter.Output” have Phase Lag Compensation Enabled. 

Usage	Input	**Data Type**	MC_CT_FilterTypes
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><td>FB Parameter</td><td colspan="3">FilterSamples</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_AxisFilter</td></tr>
<tr><td>Description</td><td colspan="3">Specifies the number of samples that the filter prediction is based upon.
This value is read on the first scan only, or when the filter is disabled.
HINT: A value of 32 samples normally delivers the best results.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>DINT</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>2</td><td>Maximum</td><td>1024</td></tr>
</table>

<table>
<tr><td>FB Parameter</td><td colspan="3">PhaseLagCompensation</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_AxisFilter</td></tr>
<tr><td>Description</td><td colspan="3">Enables phase lag compensation.
FALSE = the position is filtered but there is a phase lag between the input and output at constant speed, but the filtered output is the smoothest.
TRUE = The output is filtered and the phase lag is compensated, but there is extra noise in the output as a result.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

4.2.1.4 Outputs

FB Parameter	**SelectedAxisValid**		
Applies To	**MC_CT_AxisFilter**		
Description	When set to FALSE the axis is resetting. When this happens any axes which are slaved to the AxisOut position source must be disconnected since the position output may change by a large amount when AxisReset = TRUE. For example, for MC_CT_EGB_Axis, set CMD_EGB = FALSE by using SelectedAxisValid as a part of an AND logic condition for setting CMD_EGB. When set to TRUE the axis is not resetting and the AxisOut value may be consumed by a slave axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**SelectedAxisPosition**		
Applies To	**MC_CT_AxisFilter**		
Description	Displays the selected axis position from the feedback device.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SelectedAxisVelocity**		
Applies To	**MC_CT_AxisFilter**		
Description	Displays the selected axis velocity from the feedback device.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SelectedAxisAcceleration**		
Applies To	**MC_CT_AxisFilter**		
Description	Displays the selected axis acceleration from the feedback device.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

4.2.2 MC_CT_AxisSelector

This function block is used to allow the master input of a NC Axis EGB or CAM to be switched between 2 different sources.

The Axis1 and Axis2 inputs are intended to be connected to the two separate encoder reference sources.

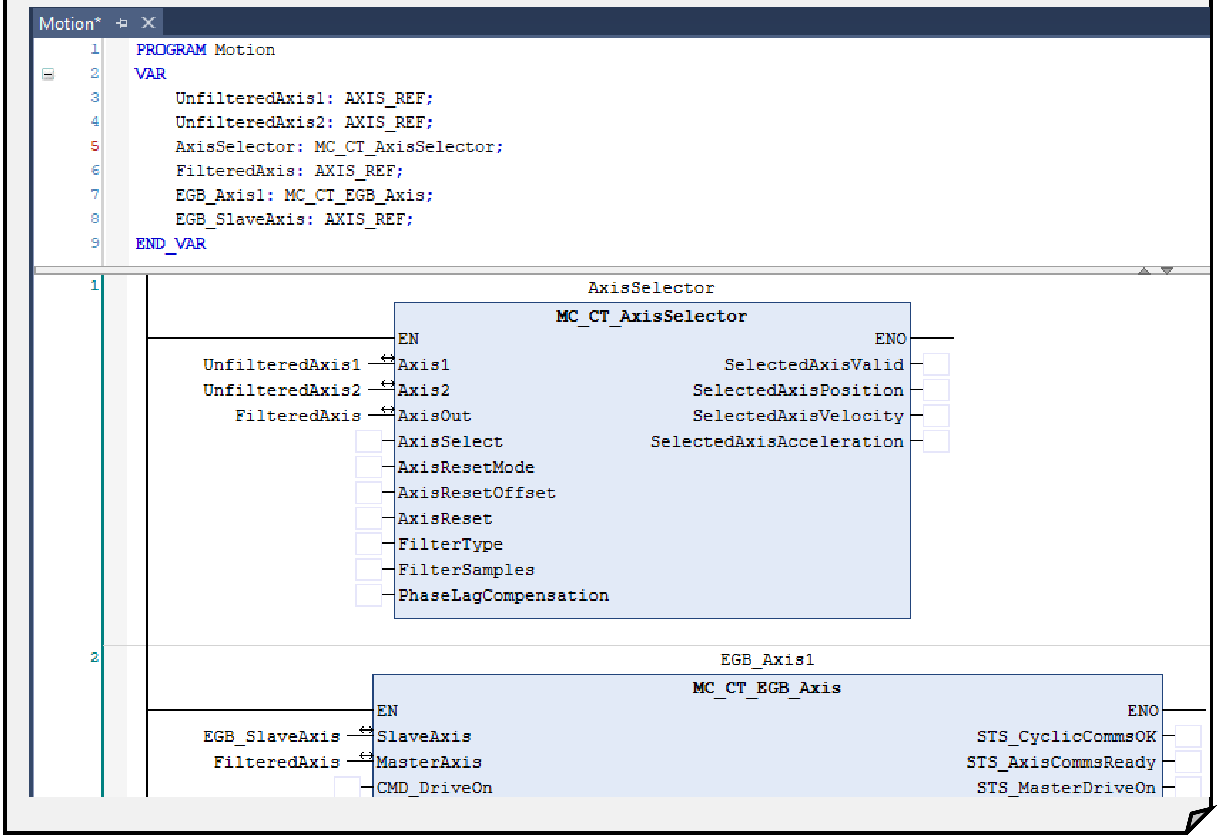
If AxisSelect = 0 then Axis1 is routed to AxisOut, if AxisSelect = 1 then Axis2 is routed to AxisOut.

The AxisOut output can then be used as the MasterAxis input to the MC_CT_EGB_Axis function block.

Relative distance logic is used when switching between Axis1 and Axis2 to ensure a bumpless switch over, provided that the speeds match, e.g. both Axis1 and Axis2 must be at zero speed or moving at near identical speeds to avoid any sudden changes in AxisOut velocity when the switch over occurs.

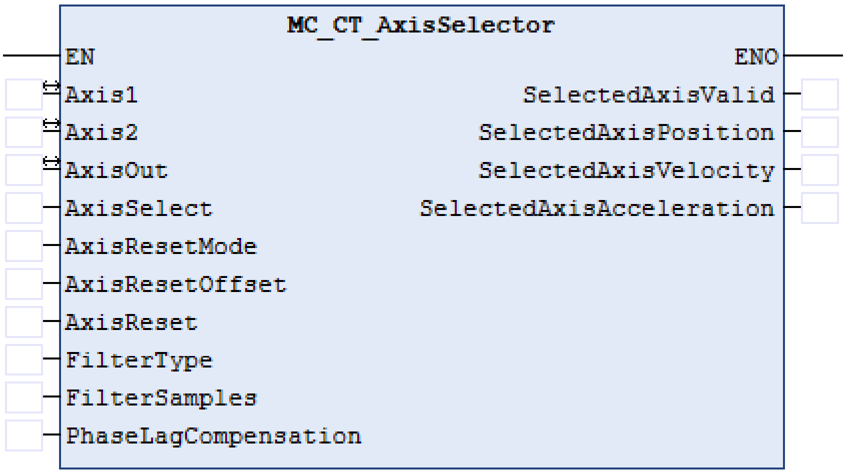
The position value should only be reset (AxisReset) when the slave axis is not actively following the master reference since large position changes of the AxisOut value are possible as a result of the values at the AxisResetMode and AxisResetOffset inputs.

HINT: The example below illustrates how MC_CT_AxisSelector can be used with MC_CT_EGB_Axis.



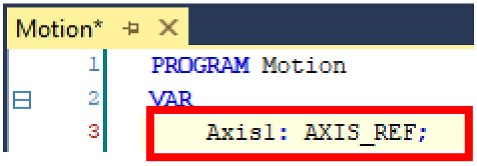
```
PROGRAM Motion
VAR
    UnfilteredAxis1: AXIS_REF;
    UnfilteredAxis2: AXIS_REF;
    AxisSelector: MC_CT_AxisSelector;
    FilteredAxis: AXIS_REF;
    EGB_Axis1: MC_CT_EGB_Axis;
    EGB_SlaveAxis: AXIS_REF;
END_VAR
```

4.2.2.1 FB Diagram

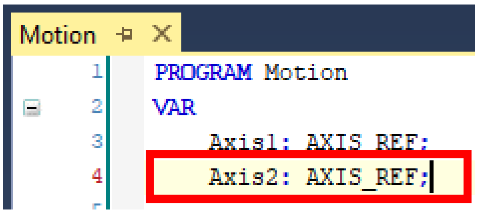


4.2.2.2 Inputs / Outputs

FB Parameter	**Axis1**		
Applies To	**MC_CT_AxisSelector**		

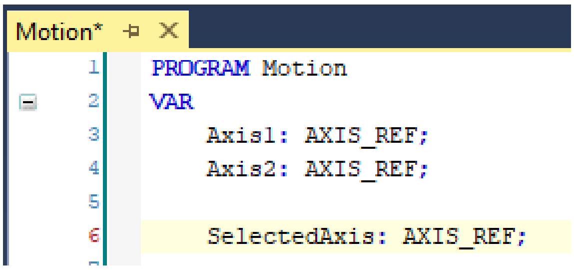
Description	The function block interface to the first selectable NC axis / AXIS_REF encoder reference.
	This function block can be used with a physical drive axis, virtual drive axis, or an encoder axis.
	This input/output must be assigned to the instance name of "AXIS_REF" for the axis to be linked. Naming the axis something more appropriate like "Axis1" as shown in the image below is good practice.
	
	
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

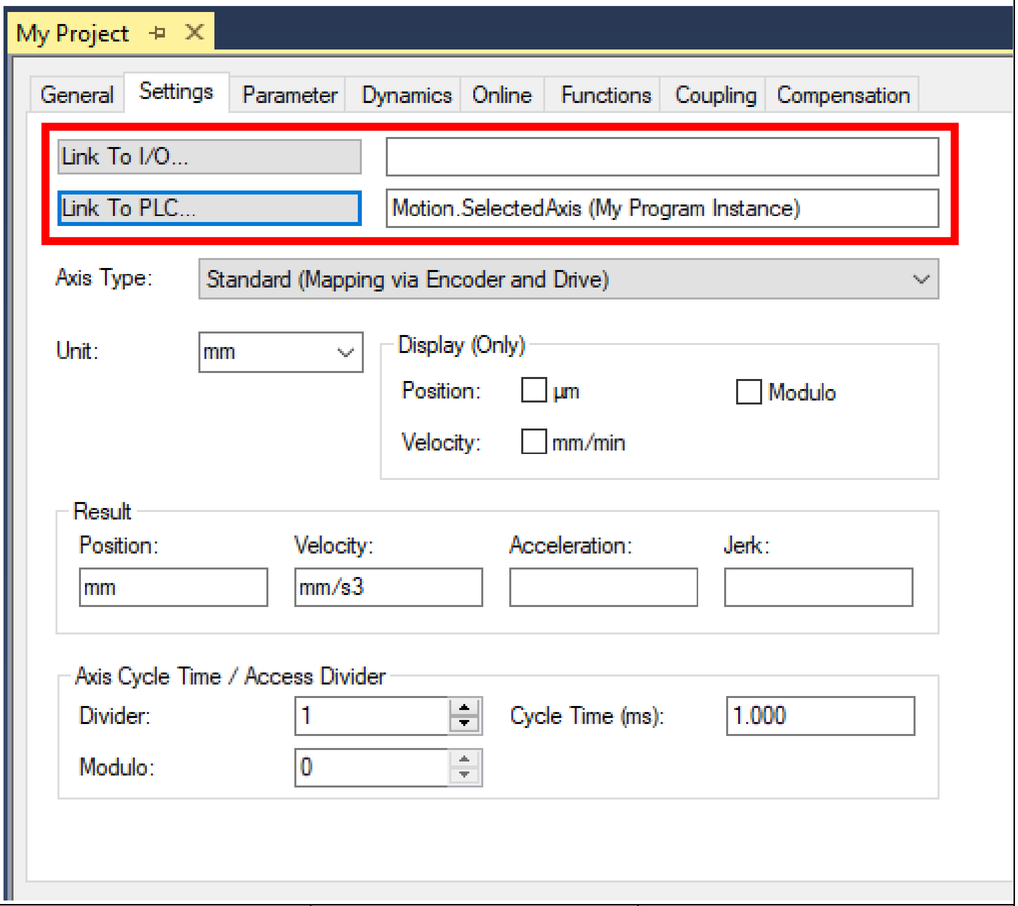
FB Parameter	**Axis2**		
Applies To	**MC_CT_AxisSelector**		
Description	The function block interface to the second selectable NC axis / AXIS_REF encoder reference. This function block can be used with a physical drive axis, virtual drive axis, or an encoder axis. This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be linked. Naming the slave axis something more appropriate like “Axis2” as shown in the image below is good practice. 		
Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    Axis2: AXIS_REF;
```

FB Parameter	**AxisOut**
Applies To	**MC_CT_AxisSelector**
Description	Specifies the instance of NC axis / AXIS_REF to output the data of the selected axis (as per the value present at the “AxisSelect” input). The final filtered encoder reference is consumed at the MasterAxis input of an MC_CT_EGB_Axis or an MC_GearIn function block. This must be connected to an isolated instance of AXIS_REF that is not linked to drive communications (I/O):



```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    Axis2: AXIS_REF;

    SelectedAxis: AXIS_REF;
```



Link To PLC...: Motion.SelectedAxis (My Program Instance)

Axis Type: Standard (Mapping via Encoder and Drive)

Unit: mm

Display (Only)

Position: ☐ µm ☐ Modulo

Velocity: ☐ mm/min

Result

Position: mm

Velocity: mm/s3

Acceleration:

Jerk:

Axis Cycle Time / Access Divider

Divider: 1

Cycle Time (ms): 1.000

Modulo: 0

Usage	InOut	Data Type	AXIS_REF
Unit	N/A	Initial Value	N/A
Minimum	N/A	Maximum	N/A

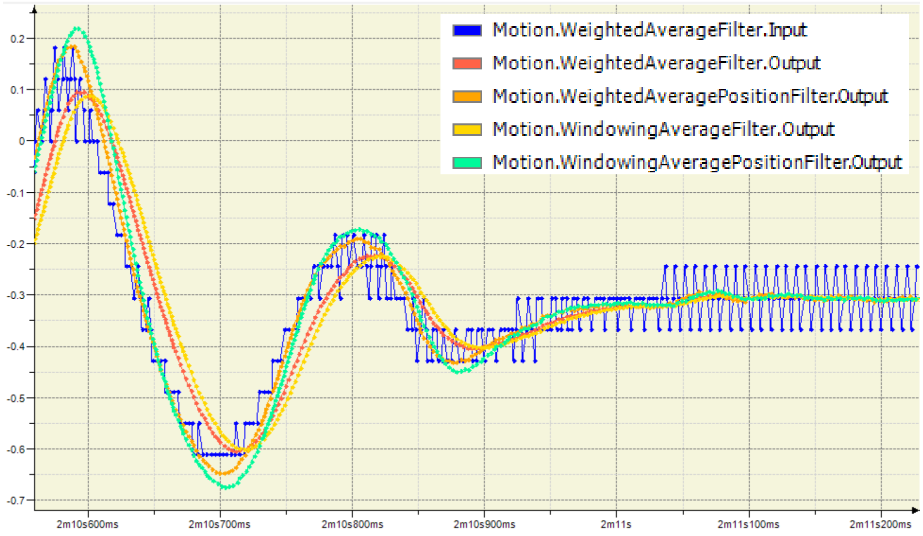
4.2.2.3 Inputs

FB Parameter	**AxisSelect**		
Applies To	**MC_CT_AxisSelector**		
Description	Selects which master axis is passed to the AxisOut output, where: FALSE = Axis1 is passed to AxisOut. TRUE = Axis2 is passed to AxisOut. Relative distance logic is used when switching between Axis1 and Axis2 to ensure a bumpless switch over, provided that the speeds match. This means the position at the AxisOut output will remain the same and the user will not see large positional changes at the AxisOut output, provided the speed of each axis is equal when the switch over occurs, e.g. both Axis1 and Axis2 must be at zero speed or moving at near identical speeds to avoid any sudden changes in AxisOut velocity when the switch over occurs.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**AxisResetMode**		
Applies To	**MC_CT_AxisSelector**		
Description	Defines the way that the position that the position accumulator is reset to when AxisReset = TRUE. Data type is of the MC_CT_AxisMode ENUM where: **MC_CT_AxisMode.Absolute** Relative means that the output position integrator will be reset to 0 + ResetOffset when AxisReset = TRUE (Rising Edge). **MC_CT_AxisMode.Relative** Absolute means that the output position integrator will be reset to the selected master absolute position + ResetOffset when AxisReset = TRUE (Rising Edge).		
Usage	Input	**Data Type**	MC_CT_AxisMode
Unit	N/A	**Initial Value**	FALSE
Minimum	0	**Maximum**	1

FB Parameter	**AxisResetOffset**		
Applies To	**MC_CT_AxisSelector**		
Description	The offset distance to be added to the pre-filter position during an axis reset.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	FALSE
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**AxisReset**		
Applies To	**MC_CT_AxisSelector**		
Description	On a FALSE to TRUE transition, the position will be reset according to the setting of AxisResetMode and AxisResetOffset. **HINT:** It is recommended to disengage the slave axis from the master reference during an axis reset since the position output can change by a large amount.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><th>FB Parameter</th><th colspan="3">FilterType</th></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_AxisFilter</td></tr>
<tr><td>Description</td><td colspan="3">
<p>Defines the filtering type applied to Axis1 or Axis2 before being passed to AxisOut.</p>
<p>Data type is of the MC_CT_FilterTypes ENUM where:</p>
<p>MC_CT_FilterTypes.No Filter</p>
<p>No Filter is selected.</p>
<p>MC_CT_FilterTypes.WindowingAverage</p>
<p>The windowing average filter gives the smoothest response at constant speed but is less accurate when the speed is changing.</p>
<p>MC_CT_FilterTypes.WeightedAverage</p>
<p>The weighted average filter gives a less smooth response at constant speed but is more accurate when the speed is changing.</p>
<p>The data trace below shows the effect each filter mode has on the raw encoder feedback (Motion.WeightedAverageFilter.Input).</p>
<blockquote>HINT: “Motion.WeightedAveragePositionFilter.Output” and “Motion.WindowingAveragePositionFilter.Output” have Phase Lag Compensation Enabled.</blockquote>


</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>MC_CT_FilterTypes</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_AxisSelector**		
Description	Specifies the number of samples that the filter prediction is based upon. This value is read on the first scan only, or when the filter is disabled. **HINT:** A value of 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	FALSE
Minimum	2	**Maximum**	1024

FB Parameter	**PhaseLagCompensation**		
Applies To	**MC_CT_AxisSelector**		
Description	Enables phase lag compensation. FALSE = the position is filtered but there is a phase lag between the input and output at constant speed, but the filtered output is the smoothest. TRUE = The output is filtered and the phase lag is compensated, but there will be extra noise in the output as a result.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

4.2.2.4 Outputs

FB Parameter	**SelectedAxisValid**		
Applies To	**MC_CT_AxisSelector**		
Description	When set to FALSE the axis is resetting. When this happens any axes which are slaved to the AxisOut position source must be disconnected since the position output may change by a large amount when AxisReset = TRUE. For example, for MC_CT_EGB_Axis, set CMD_EGB = FALSE by using SelectedAxisValid as a part of an AND logic condition for setting CMD_EGB. When set to TRUE the axis is not resetting and the AxisOut value may be consumed by a slave axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**SelectedAxisPosition**		
Applies To	**MC_CT_AxisSelector**		
Description	Displays the selected axis position from the feedback device.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SelectedAxisVelocity**		
Applies To	**MC_CT_AxisSelector**		
Description	Displays the selected axis velocity from the feedback device.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SelectedAxisAcceleration**		
Applies To	**MC_CT_AxisSelector**		
Description	Displays the selected axis acceleration from the feedback device.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

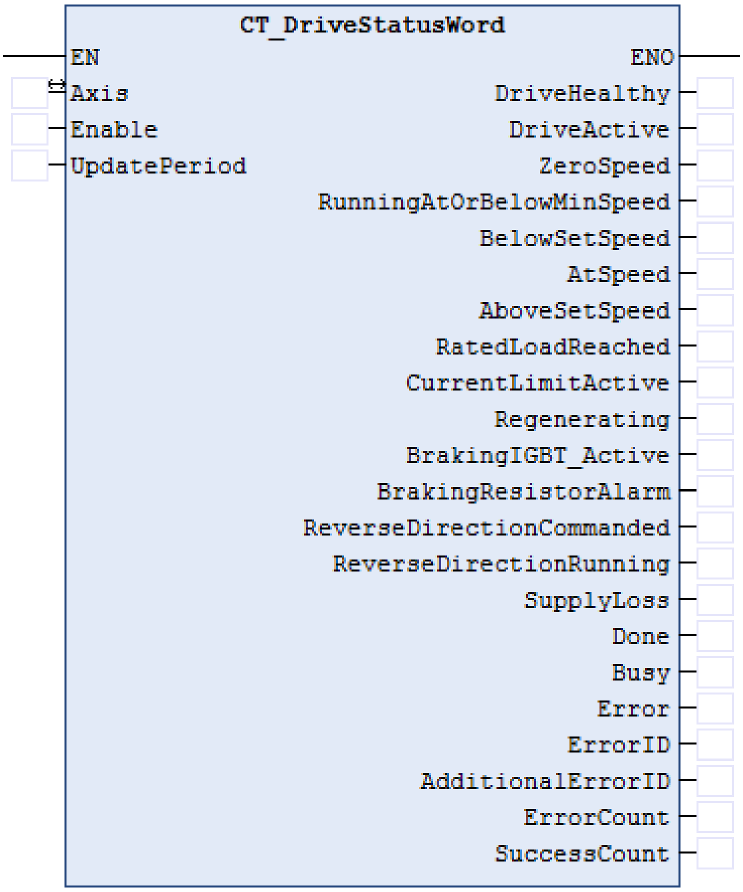
4.3 Drive Functionality

This group of function blocks provide the user with an interface which allows for the programmatic monitoring of the drive status word, additional status word and the drive IO words.

4.3.1 CT_DriveStatusWord

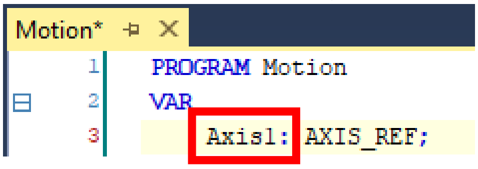
CT_DriveStatusWord indicates the status bits from Pr10.040. An SDO read is used to sample the data with an update period specified by the user.

4.3.1.1 FB Diagram



4.3.1.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**CT_DriveStatusWord**		

Description	The FB interface to the NC axis to read the data from.
	This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.3.1.3 Inputs

FB Parameter	**Enable**		
Applies To	**CT_DriveStatusWord**		
Description	Enables reading of the drive’s digital input word, where: FALSE = The digital input word is not read. TRUE = The digital input word is read at the rate defined by UpdatePeriod.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**UpdatePeriod**		
Applies To	**CT_DriveStatusWord**		
Description	Specifies the time between successive reads of the digital input word.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

4.3.1.4 Outputs

FB Parameter	**DriveHealthy**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Drive Healthy Pr10.001.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DriveActive**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Drive Active Pr10.002.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ZeroSpeed**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Zero Speed Pr10.003.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**RunningAtOrBelowMinSpeed**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Running At Or Below Minimum Speed Pr10.004.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**BelowSetSpeed**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Below Set Speed Pr10.005.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**AtSpeed**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of At Speed Pr10.006.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**AboveSetSpeed**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Above Set Speed Pr10.007.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**RatedLoadReached**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Rated Load Reached Pr10.008.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CurrentLimitActive**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Current Limit Active Pr10.009.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Regenerating**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Regenerating Pr10.010.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**BrakingIGBT_Active**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Braking IGBT Active Pr10.011.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**BrakingResistorAlarm**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Braking Resistor Alarm Pr10.012.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ReverseDirectionCommanded**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Reverse Direction Commanded Pr10.013.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ReverseDirectionRunning**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Reverse Direction Running Pr10.014.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**SupplyLoss**		
Applies To	**CT_DriveStatusWord**		
Description	Displays the state of Supply Loss Pr10.015.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Done**		
Applies To	**CT_DriveStatusWord**		
Description	Indicates that the function block has successfully completed, where: FALSE = The function block has not successfully completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_DriveStatusWord**		
Description	Indicates that the function block is running and must continue to be called, where: FALSE = The function block is not running. TRUE = The function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_DriveStatusWord**		
Description	Indicates that a NC axis or function block error is present where: FALSE = No error. TRUE = Error present. More information on the error and its cause can be found at the ErrorID, and AdditionalErrorID outputs.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_DriveStatusWord**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_DriveStatusWord**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	Data Type	MC_CT_ErrorList
Unit	N/A	Initial Value	0
Minimum	0	Maximum	5

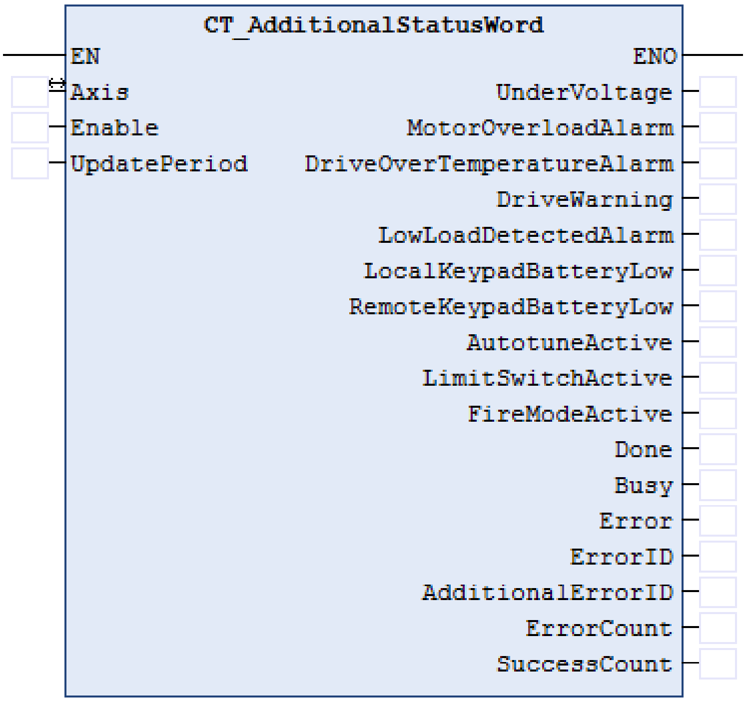
FB Parameter	**ErrorCount**		
Applies To	**CT_DriveStatusWord**		
Description	This is incremented every time there is an error.		
Usage	Output	Data Type	USINT
Unit	N/A	Initial Value	0
Minimum	0	Maximum	255

FB Parameter	**SuccessCount**		
Applies To	**CT_DriveStatusWord**		
Description	This is incremented every time there is a successful read.		
Usage	Output	Data Type	USINT
Unit	N/A	Initial Value	0
Minimum	0	Maximum	255

4.3.2 CT_AdditionalStatusWord

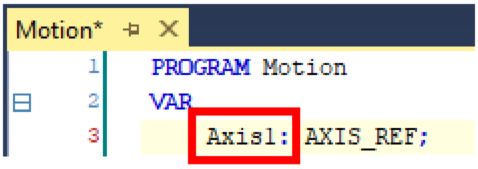
CT_AdditionalStatusWord indicates the additional status word from Pr10.069. An SDO read is used to sample the data with an update period specified by the user.

4.3.2.1 FB Diagram



4.3.2.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**CT_AdditionalStatusWord**		

Description	The FB interface to the NC to read the data from.
	This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	Motion*
	`PROGRAM Motion`
	`VAR`
	`    Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.3.2.3 Inputs

FB Parameter	**Enable**		
Applies To	**CT_AdditionalStautsWord**		
Description	Enables reading of the drive’s additional status word, where: FALSE = The additional status word is not read. TRUE = The additional status word is read at the rate defined by the UpdatePeriod.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**UpdatePeriod**		
Applies To	**CT_AdditionalStautsWord**		
Description	Specifies the time between successive reads of the additional status word.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

4.3.2.4 Outputs

FB Parameter	**UnderVoltage**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Under Voltage Active Pr10.016.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**MotorOverloadAlarm**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Motor Overload Alarm Pr10.017.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DriveOverTemperatureAlarm**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Drive Over-temperature Alarm Pr10.018.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DriveWarning**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Drive Warning Pr10.019.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**LowLoadDetectedAlarm**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Low Load Detected Alarm Pr10.062.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**LocalKeypadBatteryLow**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Local Keypad Battery Low Pr10.063.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**RemoteKeypadBatteryLow**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Remote Keypad Battery Low Pr10.064.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**AutotuneActive**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Auto-tune Active Pr10.065.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**LimitSwitchActive**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Limit Switch Active Pr10.066.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**FireModeActive**		
Applies To	**CT_AdditionalStautsWord**		
Description	Displays the state of Fire Mode Active Pr10.067.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Done**		
Applies To	**CT_AdditionalStautsWord**		
Description	Indicates that the function block has successfully completed, where: FALSE = The function block has not successfully completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_AdditionalStautsWord**		
Description	Indicates that the function block is running and must continue to be called, where: FALSE = The function block is not running. TRUE = The function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_AdditionalStautsWord**		
Description	Indicates that a NC axis or function block error is present where: FALSE = No error. TRUE = Error present. More information on the error and its cause can be found at the ErrorID, and AdditionalErrorID outputs.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_AdditionalStautsWord**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	Data Type	MC_CT_SDO_ErrorList
Unit	N/A	Initial Value	0
Minimum	0	Maximum	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_AdditionalStautsWord**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	Data Type	MC_CT_ErrorList
Unit	N/A	Initial Value	0
Minimum	0	Maximum	5

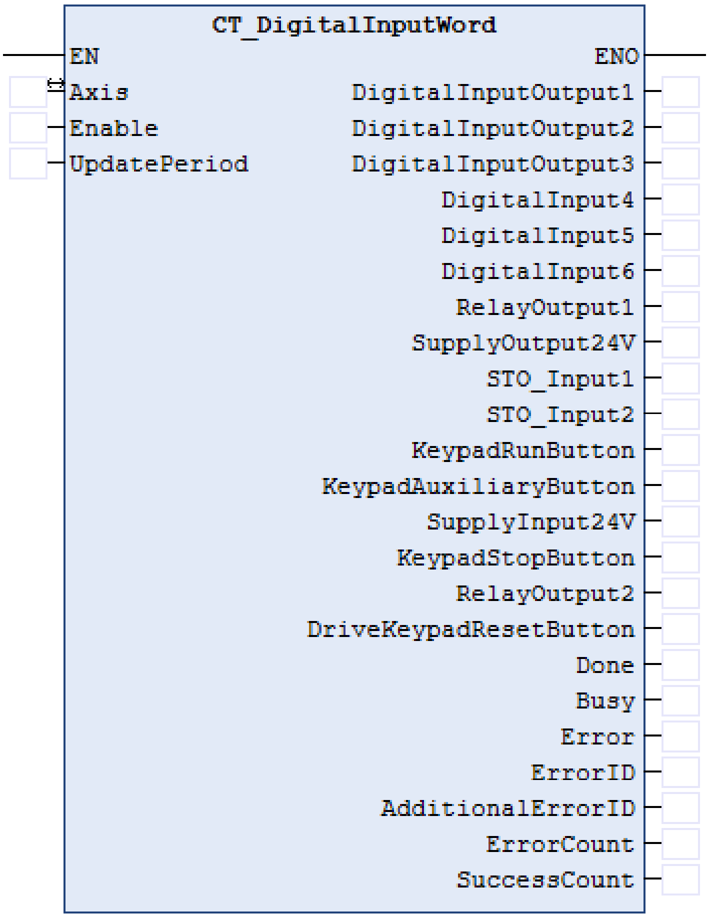
FB Parameter	**ErrorCount**		
Applies To	**CT_AdditionalStautsWord**		
Description	This is incremented every time there is an error.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**SuccessCount**		
Applies To	**CT_AdditionalStautsWord**		
Description	This is incremented every time there is a successful read.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.3.3 CT_DigitalInputWord

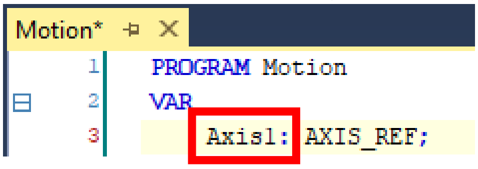
CT_DigitalInputWord indicates the Digital Input word from Pr08.020 and Pr08.072. SDO reads are used to sample the data with an update period specified by the user.

4.3.3.1 FB Diagram



4.3.3.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**CT_DigitalInputWord**		

Description	The FB interface to the NC to read the data from.
	This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of "AXIS_REF" for the axis to be controlled. Naming the slave axis something more appropriate like "Axis1" as shown in the image below is good practice.
	
	
	`PROGRAM Motion`
	`VAR`
	`    Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.3.3.3 Inputs

FB Parameter	**Enable**		
Applies To	**CT_DigitalInputWord**		
Description	Enables reading of the drive's digital input word, where: FALSE = The digital input word is not read. TRUE = The digital input word is read at the rate defined by the UpdatePeriod.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**UpdatePeriod**		
Applies To	**CT_DigitalInputWord**		
Description	Specifies the time between successive reads of the digital input word.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

4.3.3.4 Outputs

FB Parameter	**DigitalInputOutput1**		
Applies To	**CT_DigitalInputWord**		
Description	Digital Input/Output 1 for M7xx M6xx Drives Pr08.001.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DigitalInputOutput2**		
Applies To	**CT_DigitalInputWord**		
Description	Digital Input/Output 2 for M7xx M6xx Drives Pr08.002.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DigitalInputOutput3**		
Applies To	**CT_DigitalInputWord**		
Description	Digital Input/Output 3 for M7xx M6xx Drives Pr08.003.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DigitalInput4**		
Applies To	**CT_DigitalInputWord**		
Description	Digital Input 4 for M7xx M6xx Drives Pr08.004.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DigitalInput5**		
Applies To	**CT_DigitalInputWord**		
Description	Digital Input 5 for M7xx M6xx Drives Pr08.005.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**DigitalInput6**		
Applies To	**CT_DigitalInputWord**		
Description	Digital Input 6 for M7xx M6xx Drives Pr08.006.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**RelayOutput1**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of Relay 1 for M7xx M6xx Drives Pr08.007.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**SupplyOutput24V**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of 24Vdc Output for M7xx M6xx Drives Pr08.008.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**STO_Input1**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of safe torque off input 1 for M7xx M6xx Pr08.009.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**STO_Input2**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of safe torque off input 2 for M7xx M6xx Drives Pr08.040.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**KeypadRunButton**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of keypad run button for M7xx M6xx drives Pr08.041.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**KeypadAuxiliaryButton**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of the keypad auxiliary button for M7xx M6xx drives Pr08.042.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**SupplyInput24V**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of the 24Vdc input for M7xx M6xx drives Pr08.043.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**KeypadStopButton**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of the keypad stop button for M7xx M6xx drives Pr08.044.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**RelayOutput2**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of Relay 2 state for M7xx M6xx drives Pr08.045.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DriveKeypadResetButton**		
Applies To	**CT_DigitalInputWord**		
Description	Displays the state of the Drive reset button for M75x drives Pr08.046.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Done**		
Applies To	**CT_DigitalInputWord**		
Description	Indicates that the function block has successfully completed, where: FALSE = The function block has not successfully completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_DigitalInputWord**		
Description	Indicates that the function block is running and must continue to be called, where: FALSE = The function block is not running. TRUE = The function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_DigitalInputWord**		
Description	Indicates that a NC axis or function block error is present where: FALSE = No error. TRUE = Error present. More information on the error and its cause can be found at the ErrorID, and AdditionalErrorID outputs.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_DigitalInputWord**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_DigitalInputWord**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

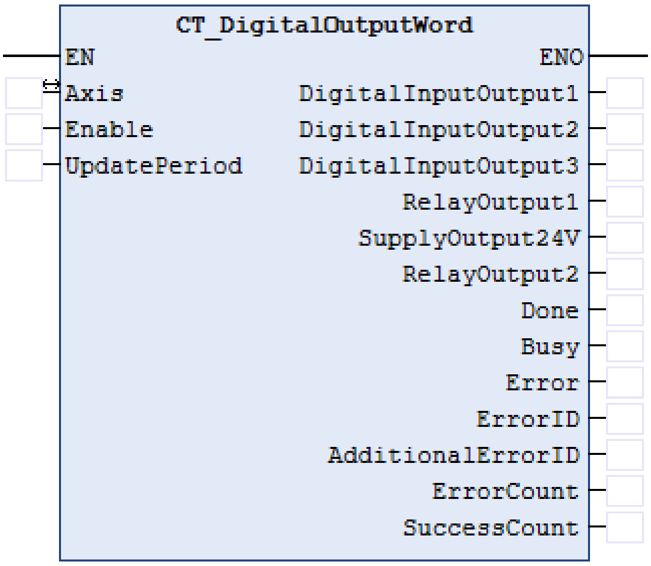
FB Parameter	**ErrorCount**		
Applies To	**CT_DigitalInputWord**		
Description	This is incremented every time there is an error.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**SuccessCount**		
Applies To	**CT_DigitalInputWord**		
Description	This is incremented every time there is a successful read.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.3.4 CT_DigitalOutputWord

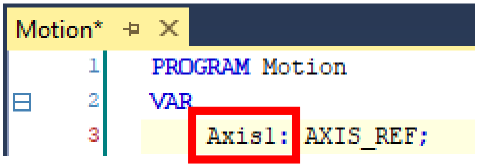
CT_DigitalInputWord indicates the Digital Output word from Pr8.073. An SDO read is used to sample the data with an update period specified by the user.

4.3.4.1 FB Diagram



4.3.4.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**CT_DigitalOutputWord**		

Description	The FB interface to the NC to read the data from.
	This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	Motion*
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.3.4.3 Inputs

FB Parameter	**Enable**		
Applies To	**CT_DigitalOutputWord**		
Description	Enables reading of the drive's digital output word, where: FALSE = The digital output word is not read. TRUE = The digital output word is read at the rate defined by the UpdatePeriod.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**UpdatePeriod**		
Applies To	**CT_DigitalOutputWord**		
Description	Specifies the time between successive reads of the digital output word.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

4.3.4.4 Outputs

FB Parameter	**DigitalInputOutput1**		
Applies To	**CT_DigitalOutputWord**		
Description	Digital Input/Output 1 for M7xx M6xx Drives Pr08.001.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DigitalInputOutput2**		
Applies To	**CT_DigitalOutputWord**		
Description	Digital Input/Output 2 for M7xx M6xx Drives Pr08.002.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**DigitalInputOutput3**		
Applies To	**CT_DigitalOutputWord**		
Description	Digital Input/Output 3 for M7xx M6xx Drives Pr08.003.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**RelayOutput1**		
Applies To	**CT_DigitalOutputWord**		
Description	Displays the state of Relay 1 state for M7xx M6xx drives Pr08.007.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**SupplyOutput24V**		
Applies To	**CT_DigitalOutputWord**		
Description	Displays the state of 24Vdc Output for M7xx M6xx Drives Pr08.008.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**RelayOutput2**		
Applies To	**CT_DigitalOutputWord**		
Description	Displays the state of Relay 2 state for M7xx M6xx drives Pr08.045.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Done**		
Applies To	**CT_DigitalOutputWord**		
Description	Indicates that the function block has successfully completed, where: FALSE = The function block has not successfully completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_DigitalOutputWord**		
Description	Indicates that the function block is running and must continue to be called, where: FALSE = The function block is not running. TRUE = The function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_DigitalOutputWord**		
Description	Indicates that a NC axis or function block error is present where: FALSE = No error. TRUE = Error present. More information on the error and its cause can be found at the ErrorID, and AdditionalErrorID outputs.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_DigitalOutputWord**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	Data Type	MC_CT_SDO_ErrorList
Unit	N/A	Initial Value	0
Minimum	0	Maximum	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_DigitalOutputWord**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	Data Type	MC_CT_ErrorList
Unit	N/A	Initial Value	0
Minimum	0	Maximum	5

FB Parameter	**ErrorCount**		
Applies To	**CT_DigitalOutputWord**		
Description	This is incremented every time there is an error.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

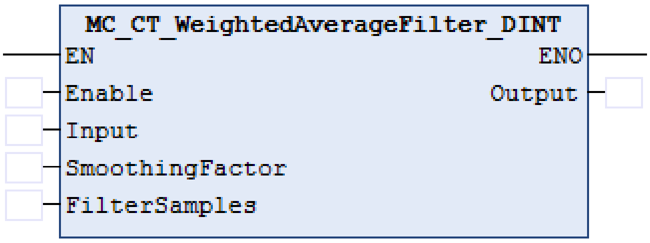
FB Parameter	**SuccessCount**		
Applies To	**CT_DigitalOutputWord**		
Description	This is incremented every time there is a successful read.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.4 Filter Functionality

4.4.1 MC_CT_WeightedAverageFilter_DINT

MC_CT_WeightedAverageFilter_DINT is a filter used to filter out position feedback noise, such as low-speed encoder quantisation noise, in the encoder feedback value when read as a DINT type. At constant speed the input and output speed will match, but there is a phase lag between the input and output position. This filter offers the best dynamic performance when filtering compared to MC_CT_WeightedAveragePositionFilter_DINT.

4.4.1.1 FB Diagram



4.4.1.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WeightedAverageFilter_DINT**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WeightedAverageFilter_DINT**		
Description	The DINT position input to filter, taken from an NC axis for example.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WeightedAverageFilter_DINT**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WeightedAverageFilter_DINT**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

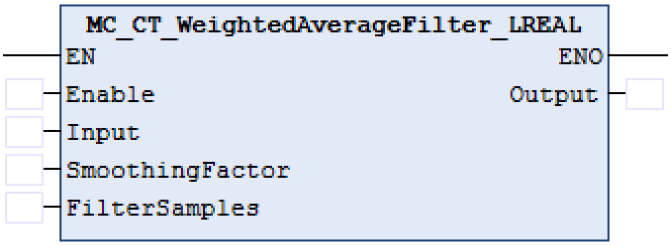
4.4.1.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WeightedAverageFilter_DINT**		
Description	Output value after filtering. This may be written to a free encoder axis to be used as a master e.g. write to axis.diEncoderPosition.		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

4.4.2 MC_CT_WeightedAverageFilter_LREAL

MC_CT_WeightedAverageFilter_LREAL is a filter used to filter out position feedback noise, such as low-speed encoder quantisation noise, in the encoder feedback value when read as a LREAL type. At constant speed the input and output speed will match, but there is a phase lag between the input and output position.

4.4.2.1 FB Diagram



4.4.2.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WeightedAverageFilter_LREAL**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to the Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WeightedAverageFilter_LREAL**		
Description	The LREAL position input to filter. This may be read from an NC axis using MC_ReadParameter or by reading the outputs from and MC_CT_ axis function block.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WeightedAverageFilter_LREAL**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WeightedAverageFilter_LREAL**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

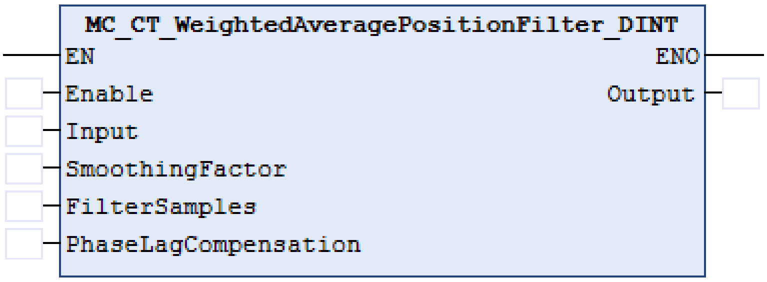
4.4.2.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WeightedAverageFilter_LREAL**		
Description	Output value after filtering. This may be written to a free encoder axis to be used as a master e.g. write to axis.fEncoderPosition.		
Usage	Output	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

4.4.3 MC_CT_WeightedAveragePositionFilter_DINT

MC_CT_WeightedAveragePositionFilter_DINT is a filter used to smooth out position feedback noise such as encoder quantisation noise in encoder counts / DINT. At constant speed there is no position or speed phase lag. This filter offers good dynamic performance when filtering. When phase lag compensation is enabled, the filter has slightly increased noise output.

4.4.3.1 FB Diagram



4.4.3.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WeightedAveragePositionFilter_DINT**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to the Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WeightedAveragePositionFilter_DINT**		
Description	The DINT position input to filter, taken from an NC axis for example.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WeightedAveragePositionFilter_DINT**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WeightedAveragePositionFilter_DINT**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

FB Parameter	**PhaseLagCompensation**		
Applies To	**MC_CT_WeightedAveragePositionFilter_DINT**		
Description	Enables phase lag compensation, where: FALSE = The output is filtered, but there is a phase lag between the input and output at constant speed, but the filtered output is the smoothest. TRUE = The output is filtered, and the phase lag is compensated, but there will be extra noise in the output as a result. **HINT:** This parameter should only be changed when the axis is stopped.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

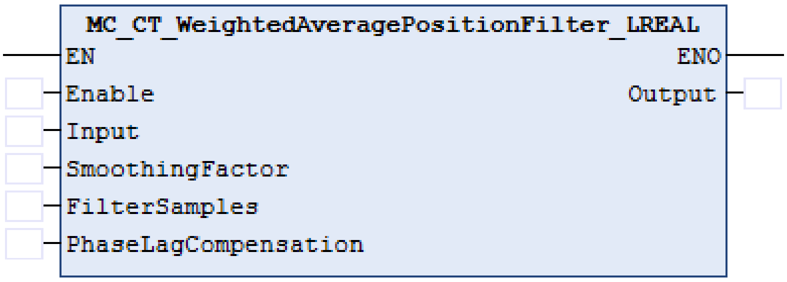
4.4.3.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WeightedAveragePositionFilter_DINT**		
Description	Output value after filtering.		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

4.4.4 MC_CT_WeightedAveragePositionFilter_LREAL

MC_CT_WeightedAveragePositionFilter_ LREAL is a filter used to smooth out position feedback noise such as encoder quantisation noise in encoder counts / DINT. At constant speed there is no position or speed phase lag. This filter offers good dynamic performance when filtering but compared to MC_CT_WeightedAverageFilter_ LREAL. When phase lag compensation is enabled, the filter has slightly increased noise output.

4.4.4.1 FB Diagram



4.4.4.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WeightedAveragePositionFilter_LREAL**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to the Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WeightedAveragePositionFilter_LREAL**		
Description	The LREAL position input to filter. This may be read from an NC axis using MC_ReadParameter or by reading the outputs from and MC_CT_ axis function block.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WeightedAveragePositionFilter_LREAL**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WeightedAveragePositionFilter_LREAL**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

FB Parameter	**PhaseLagCompensation**		
Applies To	**MC_CT_WeightedAveragePositionFilter_DINT**		
Description	Enables phase lag compensation, where: FALSE = The output is filtered, but there is a phase lag between the input and output at constant speed, but the filtered output is the smoothest. TRUE = The output is filtered, and the phase lag is compensated, but there will be extra noise in the output as a result. **HINT:** This parameter should only be changed when the axis is stopped.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

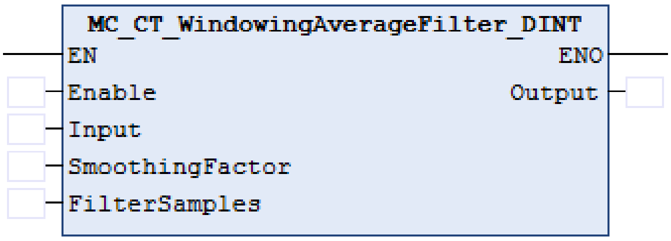
4.4.4.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WeightedAveragePositionFilter_LREAL**		
Description	The filtered output value.		
Usage	Output	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

4.4.5 MC_CT_WindowingAverageFilterDINT

MC_CT_WindowingAverageFilter_DINT is a windowing average filter used to smooth out position feedback noise such as encoder quantisation noise. At constant speed the input and output speed will match, but there is a phase lag between the input and output position. This filter offers the least output noise performance.

4.4.5.1 FB Diagram



4.4.5.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WindowingAverageFilter_DINT**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to the Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WindowingAverageFilter_DINT**		
Description	The DINT position input to filter, taken from an NC axis for example.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WindowingAverageFilter_DINT**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WindowingAverageFilter_DINT**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

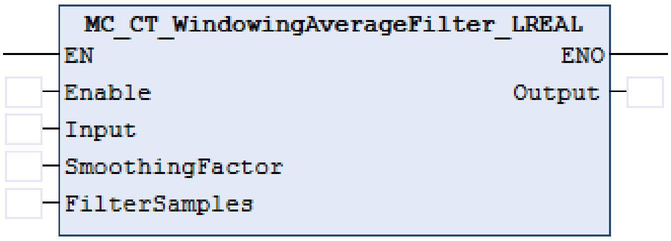
4.4.5.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WindowingAverageFilter_DINT**		
Description	The filtered output value.		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

4.4.6 MC_CT_WindowingAverageFilter_LREAL

MC_CT_WindowingAverageFilter_LREAL is a windowing average filter used to smooth out position feedback noise such as encoder quantisation noise. At constant speed the input and output speed will match, but there is a phase lag between the input and output position.

4.4.6.1 FB Diagram



4.4.6.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to the Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	The LREAL position input to filter. This may be read from an NC axis using MC_ReadParameter or by reading the outputs from and MC_CT_ axis function block.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

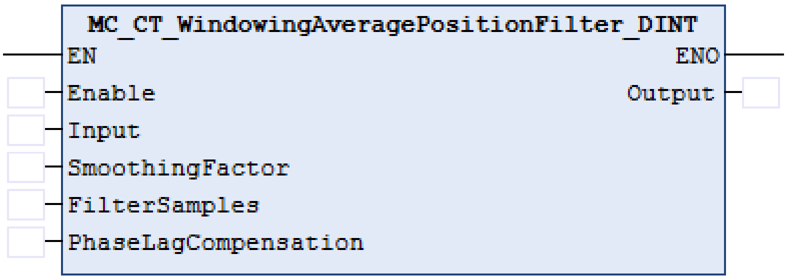
4.4.6.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	Output value after filtering.		
Usage	Output	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

4.4.7 MC_CT_WindowingAveragePositionFilter_DINT

MC_CT_WindowingAveragePositionFilter_DINT is a filter used to smooth out position feedback noise such as encoder quantisation noise in encoder counts / DINT. At constant speed there is no position or speed phase lag when phase lag compensation is enabled. This filter offers good noise performance when filtering. When phase lag compensation is enabled, the filter has a slightly increased noise output.

4.4.7.1 FB Diagram



4.4.7.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WindowingAveragePositionFilter_DINT**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to the Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WindowingAveragePositionFilter_DINT**		
Description	The DINT position input to filter, taken from an NC axis for example.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WindowingAveragePositionFilter_DINT**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WindowingAveragePositionFilter_DINT**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

FB Parameter	**PhaseLagCompensation**		
Applies To	**MC_CT_WindowingAveragePositionFilter_DINT**		
Description	Enables phase lag compensation, where: FALSE = The output is filtered, but there is a phase lag between the input and output at constant speed, but the filtered output is the smoothest. TRUE = The output is filtered, and the phase lag is compensated, but there will be extra noise in the output as a result. **HINT:** This parameter should only be changed when the axis is stopped.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

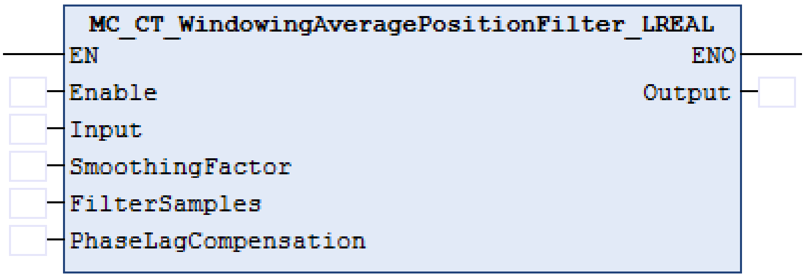
4.4.7.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WindowingAveragePositionFilter_DINT**		
Description	The filtered output value.		
Usage	Output	Data Type	DINT
Unit	N/A	Initial Value	0
Minimum	-2147483648	Maximum	2147483647

4.4.8 MC_CT_WindowingAveragePositionFilter_LREAL

MC_CT_WindowingAveragePositionFilter_ LREAL is a filter used to smooth out position feedback noise such as encoder quantisation noise. At constant speed there is no position or speed phase lag when phase lag compensation is enabled. This filter offers good noise performance when filtering. When phase lag compensation is enabled, the filter has a slightly increased noise output.

4.4.8.1 FB Diagram



4.4.8.2 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	Enables the filtering functionality when set to TRUE, where: FALSE = pass the input to the output i.e. no filter. TRUE = filter Input and pass the result to the Output. **HINT:** The filter must only be enabled when the axis position is available e.g. when the communications network is up and running in an EtherCAT based system.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Input**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	The LREAL position input to filter. This may be read from an NC axis using MC_ReadParameter or by reading the outputs from and MC_CT_ axis function block.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SmoothingFactor**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	Control parameter for smoothing where: 0 no smoothing, 1 = smooth, 10 = rough. A smoothing value of 2 normally delivers the best results.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	2.0
Minimum	0.0	**Maximum**	10.0

FB Parameter	**FilterSamples**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	Selects the number of samples that the filter prediction is based upon. 2 to 256. Only read on the first scan or when the filter is disabled. 32 samples normally delivers the best results.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	32
Minimum	2	**Maximum**	256

FB Parameter	**PhaseLagCompensation**		
Applies To	**MC_CT_WindowingAveragePositionFilter_DINT**		
Description	Enables phase lag compensation, where: FALSE = The output is filtered, but there is a phase lag between the input and output at constant speed, but the filtered output is the smoothest. TRUE = The output is filtered, and the phase lag is compensated, but there will be extra noise in the output as a result. **HINT:** This parameter should only be changed when the axis is stopped.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

4.4.8.3 Outputs

FB Parameter	**Output**		
Applies To	**MC_CT_WindowingAverageFilter_LREAL**		
Description	The filtered output value.		
Usage	Output	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

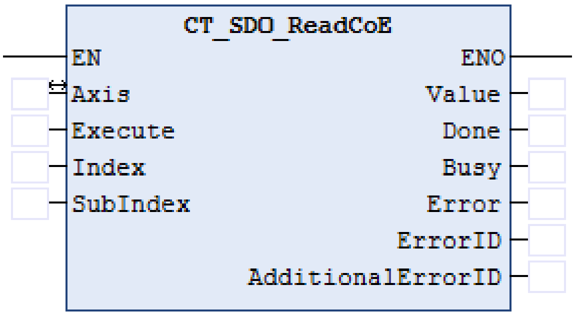
4.5 SDO Read / Write

4.5.1 CT_SDO_ReadCoE

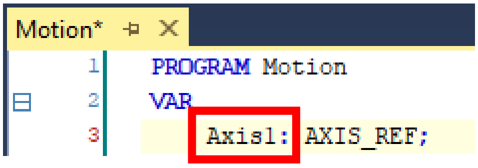
CT_SDO_ReadCoE is an SDO read function block that performs a one shot read of the specified CoE (CAN over EtherCAT) object.

A CoE object interface is provided i.e. "Index" and "SubIndex" for specifying the CoE CAN object to be read. The timeout for the read is fixed at 1s.

4.5.1.1 FB Diagram



4.5.1.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadCoE</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis to read data from.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.

Motion*
PROGRAM Motion
VAR
    Axis1: AXIS_REF;</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.5.1.3 Inputs

FB Parameter	**Execute**		
Applies To	**CT_SDO_ReadCoE**		
Description	Enables reading, where: FALSE = Stop reading or reset after completing a read. TRUE = Start reading the parameter.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Index**		
Applies To	**CT_SDO_ReadCoE**		
Description	Specifies the CoE object index to read from. For example, object 0x6060 is entered as 16#6060.		
Usage	Input	**Data Type**	WORD
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	65535

FB Parameter	**SubIndex**		
Applies To	**CT_SDO_ReadCoE**		
Description	Specifies the CoE object sub-index to read from.		
Usage	Input	**Data Type**	BYTE
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.5.1.4 Outputs

FB Parameter	**Value**		
Applies To	**CT_SDO_ReadCoE**		
Description	Displays the value that has been read from the CoE object once the Done output is set to TRUE.		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**Done**		
Applies To	**CT_SDO_ReadCoE**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_ReadCoE**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_SDO_ReadCoE**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_SDO_ReadCoE**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_SDO_ReadCoE**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

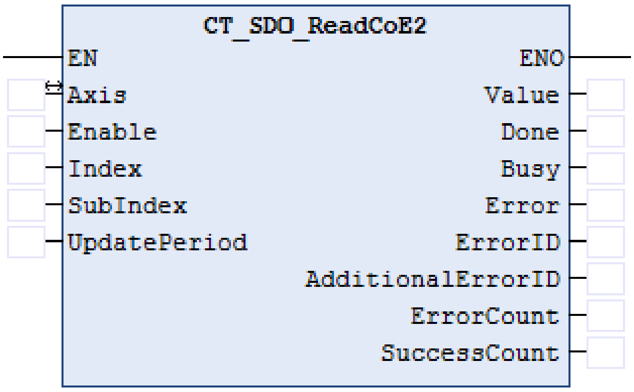
4.5.2 CT_SDO_ReadCoE2

CT_SDO_ReadCoE2 is an SDO read function block that performs continuous reading of the specified CoE (CAN over EtherCAT) object.

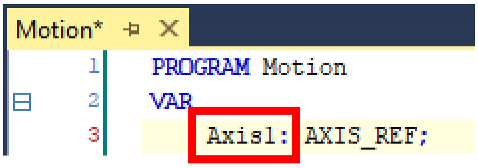
The read repeat rate can be specified by "UpdatePeriod".

A CoE object interface is provided i.e. "Index" and "SubIndex for specifying the CoE object to be read. The timeout for the read is fixed at 1s.

4.5.2.1 FB Diagram



4.5.2.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadCoE2</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis to read data from.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.

`PROGRAM Motion`
`VAR`
`Axis1: AXIS_REF;`</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.5.2.3 Inputs

FB Parameter	**Enable**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Enables reading, where: FALSE = Stop reading or reset after completing a read. TRUE = Start reading the parameter.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Index**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Specifies the CoE object index to read from. For example, object 0x6060 is entered as 16#6060.		
Usage	Input	**Data Type**	WORD
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	65535

FB Parameter	**SubIndex**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Specifies the CoE object sub-index to read from.		
Usage	Input	**Data Type**	BYTE
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**UpdatePeriod**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Specifies the time between successive reads. T#0MS = update as fast as possible.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

4.5.2.4 Outputs

FB Parameter	**Value**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Displays the value that has been read from the CoE object once the Done output is set TRUE.		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**Done**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><td>FB Parameter</td><td colspan="3">Error</td></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadCoE2</td></tr>
<tr><td>Description</td><td colspan="3">Indicates that the function block has entered the error state, where:
FALSE = No error.
TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

<table>
<tr><td>FB Parameter</td><td colspan="3">ErrorID</td></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadCoE2</td></tr>
<tr><td>Description</td><td colspan="3">Indicates the error ID when the Error output is set to TRUE.
The datatype is MC_CT_SDO_ErrorList (ENUM).
For more information see section 5.16 MC_CT_SDO_ErrorList.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>MC_CT_SDO_ErrorList</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>0</td><td>Maximum</td><td>4294967295</td></tr>
</table>

<table>
<tr><td>FB Parameter</td><td colspan="3">AdditionalErrorID</td></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadCoE2</td></tr>
<tr><td>Description</td><td colspan="3">Indicates the additional error ID when the Error output is set to TRUE.
The datatype is MC_CT_ErrorList (ENUM).
For more information see section 5.8 MC_CT_ErrorList.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>MC_CT_ErrorList</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>0</td><td>Maximum</td><td>5</td></tr>
</table>

FB Parameter	**ErrorCount**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Displays the number of errors encountered since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

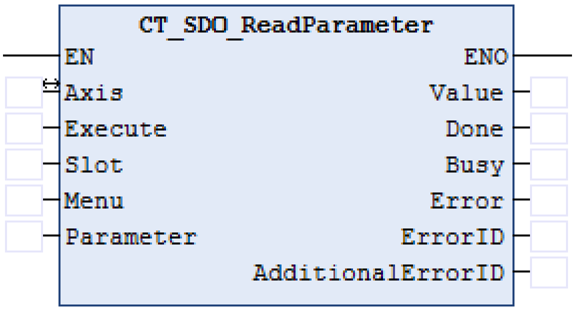
FB Parameter	**SuccessCount**		
Applies To	**CT_SDO_ReadCoE2**		
Description	Displays the number of times the function block has successfully completed since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.5.3 CT_SDO_ReadParameter

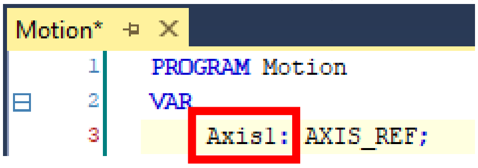
CT_SDO_ReadParameter is an SDO read function block that performs a one shot read of the specified drive parameter.

A drive parameter interface is provided i.e. "Slot", "Menu" and "Parameter" for specifying the parameter to be read. The timeout for the read is fixed at 1s.

4.5.3.1 FB Diagram



4.5.3.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadParameter</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis to read data from.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.


`PROGRAM Motion`
`VAR`
`    Axis1: AXIS_REF;`</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.5.3.3 Inputs

FB Parameter	**Execute**		
Applies To	**CT_SDO_ReadParameter**		
Description	Enables reading, where: FALSE = stop reading or reset after completing a read. TRUE = start reading the parameter.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><th>FB Parameter</th><th colspan="3">Slot</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadParameter</td></tr>
<tr><td>Description</td><td colspan="3">Specifies the drive slot to read from.

HINT: Slot 0 = the drive itself. This can be used for reading from drive menus.

The table below shows the number of slots each product has, and which slot is consumed by the factory fit option module.

<table>
<tr><th>Product</th><th>Available Slot Numbers</th></tr>
<tr><td>Unidrive M</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Option slot 3 = 3
Factory fitted option = 4</td></tr>
<tr><td>Digitax HD</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Factory fitted option = 3</td></tr>
</table>
</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>USINT</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>0</td><td>Maximum</td><td>255</td></tr>
</table>

FB Parameter	**Menu**		
Applies To	**CT_SDO_ReadParameter**		
Description	Specifies the drive menu to read from.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**Parameter**		
Applies To	**CT_SDO_ReadParameter**		
Description	Specifies the drive parameter to read from.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.5.3.4 Outputs

FB Parameter	**Value**		
Applies To	**CT_SDO_ReadParameter**		
Description	Displays the value that has been read from the drive parameter once the Done output is set TRUE.		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**Done**		
Applies To	**CT_SDO_ReadParameter**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_ReadParameter**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_SDO_ReadParameter**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_SDO_ReadParameter**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

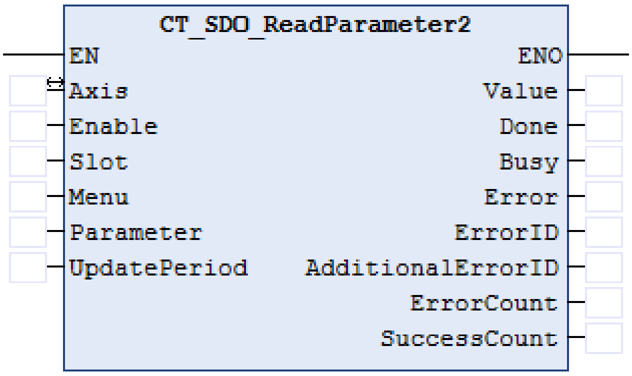
FB Parameter	**AdditionalErrorID**		
Applies To	**CT_SDO_ReadParameter**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

4.5.4 CT_SDO_ReadParameter2

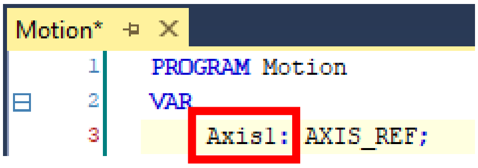
CT_SDO_ReadParameter2 is an SDO read function block that performs continuous reading of the specified drive parameter. The read repeat rate can be specified by “UpdatePeriod”.

A drive parameter interface is provided i.e. “Slot”, “Menu” and “Parameter” for specifying the parameter to be read. The timeout for the read is fixed at 1s.

4.5.4.1 FB Diagram



4.5.4.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_ReadParameter2</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis to read data from.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.

Motion*
PROGRAM Motion
VAR
Axis1: AXIS_REF;</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.5.4.3 Inputs

FB Parameter	**Enable**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Enables reading, where: FALSE = stop reading or reset after completing a read. TRUE = start reading the parameter.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><th>FB Parameter</th><td colspan="3">Slot</td></tr>
<tr><th>Applies To</th><td colspan="3">CT_SDO_ReadParameter2</td></tr>
<tr><th>Description</th><td colspan="3">
Specifies the drive slot to read from.
<p>HINT: Slot 0 = the drive itself. This can be used for reading from drive menus.</p>
<p>The table below shows the number of slots each product has, and which slot is consumed by the factory fit option module.</p>
<table>
<tr><th>Product</th><th>Available Slot Numbers</th></tr>
<tr><td>Unidrive M</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Option slot 3 = 3
Factory fitted option = 4</td></tr>
<tr><td>Digitax HD</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Factory fitted option = 3</td></tr>
</table>
</td></tr>
<tr><th>Usage</th><td>Input</td><th>Data Type</th><td>USINT</td></tr>
<tr><th>Unit</th><td>N/A</td><th>Initial Value</th><td>0</td></tr>
<tr><th>Minimum</th><td>0</td><th>Maximum</th><td>255</td></tr>
</table>

FB Parameter	**Menu**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Specifies the drive menu to read from.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**Parameter**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Specifies the drive parameter to read from.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**UpdatePeriod**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Specifies the time between successive reads. T#0MS = update as fast as possible.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

4.5.4.4 Outputs

FB Parameter	**Value**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Displays the value that has been read from the drive parameter once the Done output is set TRUE.		
Usage	Output	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**Done**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the "ErrorID" output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_SDO_ReadParameter2**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

FB Parameter	**ErrorCount**		
Applies To	**CT_SDO_ReadParameter**		
Description	Displays the number of errors encountered since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

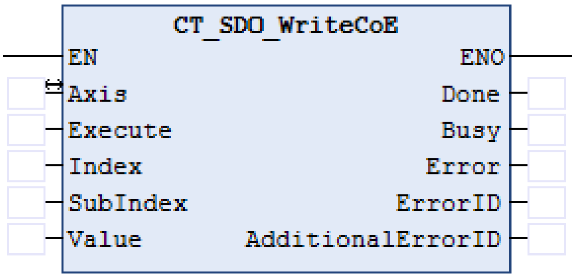
FB Parameter	**SuccessCount**		
Applies To	**CT_SDO_ReadParameter**		
Description	Displays the number of times the function block has successfully completed since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.5.5 CT_SDO_WriteCoE

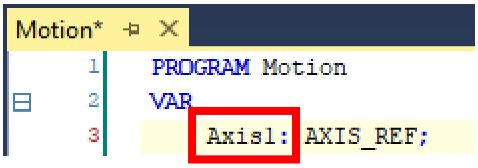
CT_SDO_WriteCoE is an SDO write function block that performs a one shot write to the specified CoE (CAN over EtherCAT) object.

A CoE object interface is provided i.e. "Index" and "SubIndex for specifying the CoE object to be written. The timeout for the write is fixed at 1s.

4.5.5.1 FB Diagram



4.5.5.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_WriteCoE</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis to write data to.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.


PROGRAM Motion
VAR
    Axis1: AXIS_REF;</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.5.5.3 Inputs

FB Parameter	**Execute**		
Applies To	**CT_SDO_WriteCoE**		
Description	Enables writing, where: FALSE = Stop writing or reset after completing a write. TRUE = Start writing the parameter.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Index**		
Applies To	**CT_SDO_WriteCoE**		
Description	Specifies the CoE object index to be written to. For example, object 0x6060 is entered as 16#6060.		
Usage	Input	**Data Type**	WORD
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	65535

FB Parameter	**SubIndex**		
Applies To	**CT_SDO_WriteCoE**		
Description	Specifies the CoE object sub-index to write to.		
Usage	Input	**Data Type**	BYTE
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**Value**		
Applies To	**CT_SDO_WriteCoE**		
Description	Specifies the value to be written to the CoE object when the Execute input is set TRUE.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

4.5.5.4 Outputs

FB Parameter	**Done**		
Applies To	**CT_SDO_WriteCoE**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_WriteCoE**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_SDO_WriteCoE**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_SDO_WriteCoE**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

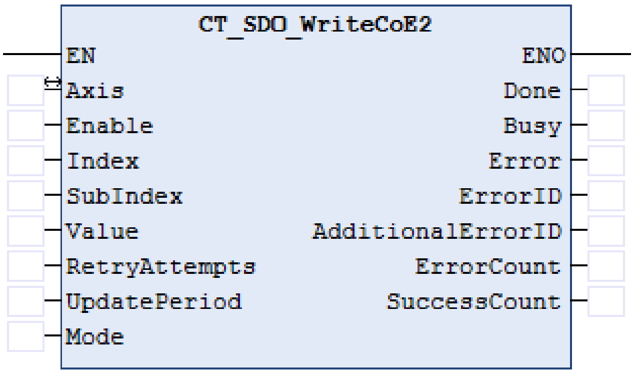
FB Parameter	**AdditionalErrorID**		
Applies To	**CT_SDO_WriteCoE**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

4.5.6 CT_SDO_WriteCoE2

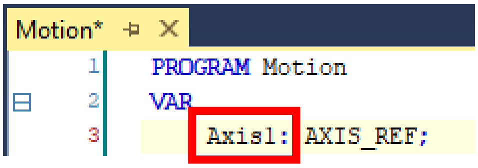
CT_SDO_WriteCoE2 is an SDO write function block that performs continuous writing to the specified CoE (CAN over EtherCAT) object. The write repeat rate can be specified by "UpdatePeriod".

A CoE object interface is provided i.e. "Index" and "SubIndex" for specifying the CoE object to be written. The timeout for the write is fixed at 1s.

4.5.6.1 FB Diagram



4.5.6.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_WriteCoE2</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis to write data to.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.


<pre>PROGRAM Motion
VAR
    Axis1: AXIS_REF;</pre></td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.5.6.3 Inputs

FB Parameter	**Enable**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Enables writing, where: FALSE = Stop writing or reset after completing a write. TRUE = Start writing the parameter.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Index**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Specifies the CoE object index to be written to. For example, object 0x6060 is entered as 16#6060.		
Usage	Input	**Data Type**	WORD
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	65535

FB Parameter	**SubIndex**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Specifies the CoE object sub-index to write to.		
Usage	Input	**Data Type**	BYTE
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**Value**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Specifies the value to be written to the CoE object when the Enable input is set TRUE.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**RetryAttempts**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Specifies the number of retry attempts when write has failed.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**UpdatePeriod**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Specifies the time between successive writes. T#0MS = update as fast as possible.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

FB Parameter	**Mode**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Selects when to write the value. The datatype is MC_CT_WriteModes (ENUM) where: **MC_CT_WriteModes.WriteWhenInputChanges** Write when a change has been detected at the “Value” input. **MC_CT_WriteModes.WriteContinuous** Continuously write at the repeat rate specified at the “UpdatePeriod” input.		
Usage	Input	**Data Type**	MC_CT_WriteModes
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	1

4.5.6.4 Outputs

FB Parameter	**Done**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

FB Parameter	**ErrorCount**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Displays the number of errors encountered since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

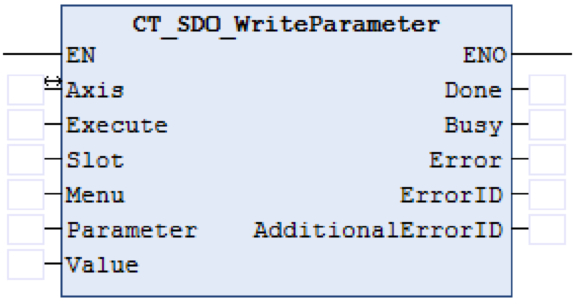
FB Parameter	**SuccessCount**		
Applies To	**CT_SDO_WriteCoE2**		
Description	Displays the number of times the function block has successfully completed since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.5.7 CT_SDO_WriteParameter

CT_SDO_WriteParameter is an SDO write function block that performs a one shot write to the specified drive parameter.

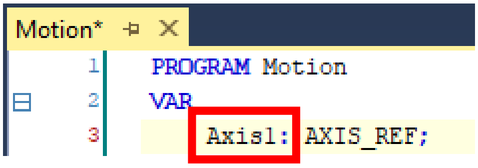
A drive parameter interface is provided e.g. "Slot", "Menu" and "Parameter" for specifying the parameter to be written. The timeout for the write is fixed at 1s.

4.5.7.1 FB Diagram



4.5.7.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**CT_SDO_WriteParameter**		

Description	The FB interface to the NC axis to write data to.
	This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	Motion*
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.5.7.3 Inputs

<table>
<tr><td>FB Parameter</td><td colspan="3">Execute</td></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_WriteParameter</td></tr>
<tr><td>Description</td><td colspan="3">Enables writing, where:
FALSE = Stop writing or reset after completing a write.
TRUE = Start writing the parameter.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

<table>
<tr><td>FB Parameter</td><td colspan="3">Slot</td></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_WriteParameter</td></tr>
<tr><td>Description</td><td colspan="3">Specifies the drive slot to write to.

HINT: Slot 0 = the drive itself. This can be used for writing to drive menus.

The table below shows the number of slots each product has, and which slot is consumed by the factory fit option module.
<table>
<tr><th>Product</th><th>Available Slot Numbers</th></tr>
<tr><td>Unidrive M</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Option slot 3 = 3
Factory fitted option = 4</td></tr>
<tr><td>Digitax HD</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Factory fitted option = 3</td></tr>
</table></td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>USINT</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>0</td><td>Maximum</td><td>255</td></tr>
</table>

FB Parameter	**Menu**		
Applies To	**CT_SDO_WriteParameter**		
Description	Specifies the drive menu to write to.		
Usage	Input	Data Type	USINT
Unit	N/A	Initial Value	0
Minimum	0	Maximum	255

FB Parameter	**Parameter**		
Applies To	**CT_SDO_WriteParameter**		
Description	Specifies the drive parameter to write to.		
Usage	Input	Data Type	USINT
Unit	N/A	Initial Value	0
Minimum	0	Maximum	255

FB Parameter	**Value**		
Applies To	**CT_SDO_WriteParameter**		
Description	Specifies the value to be written to the drive parameter object when the Execute output is set TRUE.		
Usage	Input	Data Type	DINT
Unit	N/A	Initial Value	0
Minimum	-2147483648	Maximum	2147483647

4.5.7.4 Outputs

FB Parameter	**Done**		
Applies To	**CT_SDO_WriteParameter**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_WriteParameter**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_SDO_WriteParameter**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_SDO_WriteParameter**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

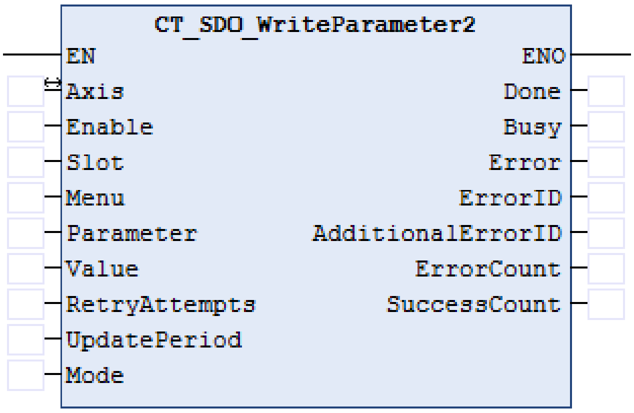
FB Parameter	**AdditionalErrorID**		
Applies To	**CT_SDO_WriteParameter**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

4.5.8 CT_SDO_WriteParameter2

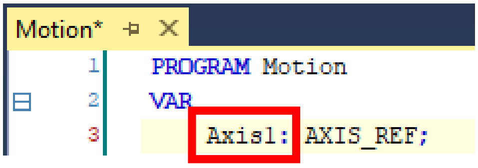
CT_SDO_WriteParameter2 is an SDO write function block that performs continuous writing to the specified drive parameter. The write repeat rate can be specified using "UpdatePeriod".

A drive parameter interface is provided e.g. "Slot", "Menu" and "Parameter" for specifying the parameter to be written. The timeout for the write is fixed at 1s.

4.5.8.1 FB Diagram



4.5.8.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_WriteParameter2</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis to write data to.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.


PROGRAM Motion
VAR
Axis1: AXIS_REF;</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.5.8.3 Inputs

<table>
<tr><td>FB Parameter</td><td colspan="3">Enable</td></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_WriteParameter2</td></tr>
<tr><td>Description</td><td colspan="3">Enables writing, where:
FALSE = Stop writing or reset after completing a write.
TRUE = Start writing the parameter.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>BOOL</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>FALSE</td></tr>
<tr><td>Minimum</td><td>FALSE</td><td>Maximum</td><td>TRUE</td></tr>
</table>

<table>
<tr><td>FB Parameter</td><td colspan="3">Slot</td></tr>
<tr><td>Applies To</td><td colspan="3">CT_SDO_WriteParameter</td></tr>
<tr><td>Description</td><td colspan="3">Specifies the drive slot to write to.

HINT: Slot 0 = the drive itself. This can be used for writing to drive menus.

The table below shows the number of slots each product has, and which slot is consumed by the factory fit option module.

<table>
<tr><th>Product</th><th>Available Slot Numbers</th></tr>
<tr><td>Unidrive M</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Option slot 3 = 3
Factory fitted option = 4</td></tr>
<tr><td>Digitax HD</td><td>Drive = 0
Option slot 1 = 1
Option slot 2 = 2
Factory fitted option = 3</td></tr>
</table>
</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>USINT</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>0</td><td>Maximum</td><td>255</td></tr>
</table>

FB Parameter	**Menu**		
Applies To	**CT_SDO_WriteParameter**		
Description	Specifies the drive menu to write to.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**Parameter**		
Applies To	**CT_SDO_WriteParameter**		
Description	Specifies the drive parameter to write to.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**Value**		
Applies To	**CT_SDO_WriteParameter**		
Description	Specifies the value to be written to the drive parameter object when the Execute output is set TRUE.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	0
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**RetryAttempts**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Specifies the number of retry attempts when write has failed.		
Usage	Input	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**UpdatePeriod**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Specifies the time between successive writes. T#0MS = update as fast as possible.		
Usage	Input	**Data Type**	TIME
Unit	Time	**Initial Value**	FALSE
Minimum	T#0MS	**Maximum**	T#49D 17H 2M 47S 295MS

FB Parameter	**Mode**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Selects when to write the value. The datatype is MC_CT_WriteModes (ENUM) where: **MC_CT_WriteModes.WriteWhenInputChanges** Write when a change has been detected at the “Value” input. **MC_CT_WriteModes.WriteContinuous** Continuously write at the repeat rate specified at the “UpdatePeriod” input.		
Usage	Input	**Data Type**	MC_CT_WriteModes
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	1

4.5.8.4 Outputs

FB Parameter	**Done**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Displays when the function block has completed, where: FALSE = The function block has not completed. TRUE = The function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Indicates the error ID when the Error output is set to TRUE. The datatype is MC_CT_SDO_ErrorList (ENUM). For more information see section **5.16 MC_CT_SDO_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_SDO_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4294967295

FB Parameter	**AdditionalErrorID**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Indicates the additional error ID when the Error output is set to TRUE. The datatype is MC_CT_ErrorList (ENUM). For more information see section **5.8 MC_CT_ErrorList**.		
Usage	Output	**Data Type**	MC_CT_ErrorList
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	5

FB Parameter	**ErrorCount**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Displays the number of errors encountered since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

FB Parameter	**SuccessCount**		
Applies To	**CT_SDO_WriteParameter2**		
Description	Displays the number of times the function block has successfully completed since the Enable input was set TRUE.		
Usage	Output	**Data Type**	USINT
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	255

4.6 Basic NC Axis function blocks

The Control Techniques NC Axis Interface Library contains a set of continuous update motion control function blocks which replicate the standard NC motion control function blocks, but include a "Continuous Update" input e.g. MC_CT_MoveVelocity is similar to MC_MoveVelocity except the velocity, acceleration rate, deceleration rate and jerk rate may all be updated during execution of the block provided the "Continuous" update input = TRUE. Any new values relating to target or rates will take immediate effect without the need to re-execute the function block or to use multiple instances of the same function block to switch between different targets or rates.

The following section describes each of the continuous update function blocks included with the Control Techniques NC Axis Interface Library.

4.6.1 MC_CT_GearIn

This function block is a continuous update version of MC_GearIn

If the "ContinuousUpdate" input is set TRUE and a value is changed at the "RatioNumerator", "RatioDenominator", "Acceleration", "Deceleration" or "Jerk" inputs, the new value takes immediate effect.

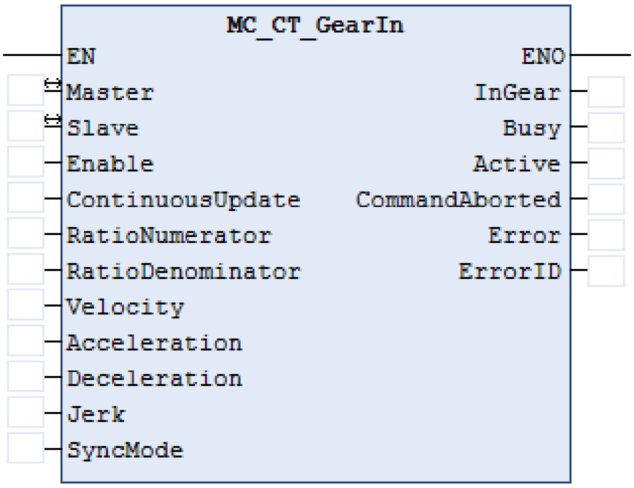
The function block couples the slave axis to the master axis, and applies a user specified velocity ratio between master and slave axis.

When MC_CT_GearIn is started, the slave axis is ramped up to the given ratio of the master velocity. Coupling takes place once this is achieved. The distance travelled prior to coupling, e.g. while accelerating, is only compensated for in Rigid Lock and Registration Sync Modes.

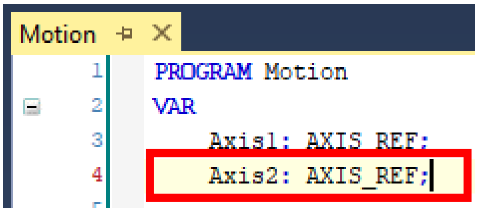
HINT: MC_CT_GearIn should be called after the motion FB controlling the Master axis has been called. Otherwise, the Slave axis will lag behind the Master axis by 1 sample.

InGear is set as soon as the axis is coupled. The final coupling speed of the slave axis depends on the "RatioNumerator" and "RatioDenominator" that is applied to the master reference for the slave to follow. For example, with a ratio of Numerator =1, Denominator = 2, the slave is moved by exactly half the distance of the master at half the velocity.

4.6.1.1 FB Diagram

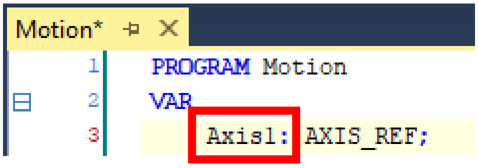


4.6.1.2 Inputs / Outputs

FB Parameter	**Master**
Applies To	**MC_CT_GearIn**
Description	This is the FB interface to the NC master axis that will be used as the position reference master. This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice. 

```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
    Axis2: AXIS_REF;
```

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

FB Parameter	**Slave**
Applies To	**MC_CT_GearIn**
Description	This is the FB interface to the NC slave axis that will be geared to the master axis. This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice. 

```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
```

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.1.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_GearIn**		
Description	Executes the function block, where: FALSE = No action. TRUE = Rising edge starts execution of the function block.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_GearIn**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - RatioNumerator - RatioDenominator - Acceleration - Deceleration - Jerk		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**RatioNumerator**		
Applies To	**MC_CT_GearIn**		
Description	Defines numerator of coupling ratio between the master and slave axis.		
Usage	Input	**Data Type**	DINT
Unit	N/A	**Initial Value**	FALSE
Minimum	-2147483648	**Maximum**	2147483647

FB Parameter	**RatioDenominator**		
Applies To	**MC_CT_GearIn**		
Description	Defines denominator of coupling ratio between the master and slave axis.		
Usage	Input	**Data Type**	UDINT
Unit	N/A	**Initial Value**	FALSE
Minimum	1	**Maximum**	4294967295

FB Parameter	**Acceleration**		
Applies To	**MC_CT_GearIn**		
Description	Specifies the maximum acceleration rate to be used when ramping up, in Application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_GearIn**		
Description	Specifies the maximum deceleration rate to be used when ramping up, in Application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_GearIn**		
Description	Specifies the maximum jerk rate to be used when ramping up, in Application units/s³.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s³	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**SyncMode**		
Applies To	**MC_CT_GearIn**		
Description	Specifies how the slave axis is synchronised to the master axis. Data type is of the MC_CT_SyncMode (ENUM) where: **MC_CT_SyncMode.InstantLock** Instant lock is selected. **MC_CT_SyncMode.MatchSpeed** Match speed is selected. **MC_CT_SyncMode.MatchSpeedProfile** Match speed profile is selected. **MC_CT_SyncMode.MatchPosition** Match position is selected. **MC_CT_SyncMode.MatchPositionProfile** Match position profile is selected. For more information see section **5.19 MC_CT_SyncMode**.		
Usage	Input	**Data Type**	MC_CT_SyncMode
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	4

4.6.1.4 Outputs

FB Parameter	**InGear**		
Applies To	**MC_CT_GearIn**		
Description	Indicates that coupling has taken place and that the slave axis will now move at the specified gear ratio with master axis movement.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_GearIn**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Active**		
Applies To	**MC_CT_GearIn**		
Description	Indicates that the FB has control of the axis, where: FALSE = The FB does not have control of the axis. TRUE = The FB has control of the axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_GearIn**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_GearIn**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_GearIn**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

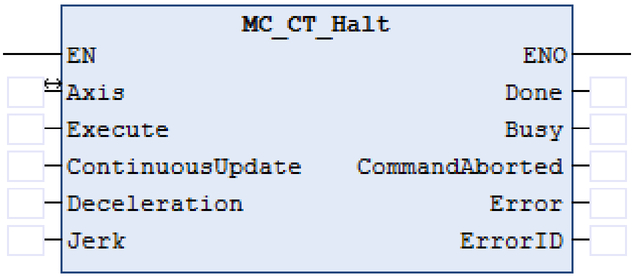
4.6.2 MC_CT_Halt

This function block is a continuous update version of MC_Halt.

If the “ContinuousUpdate” input is set TRUE and a value is changed at the “Deceleration” or “Jerk” inputs, the new values to take immediate effect.

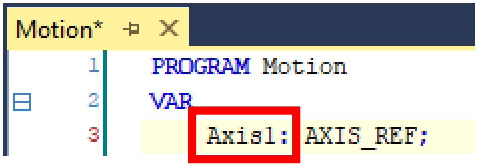
MC_CT_Halt stops the referenced axis in a controlled manner. If actions of other function blocks are running at this time, the actions are aborted. The axis enters the “discrete motion” state until the velocity 0 is reached. If the Done output of MC_CT_Halt is set, the state of the axis changes to standstill. As long as MC_CT_Halt is active, a new motion command can be issued that interrupts the execution of MC_CT_Halt. Unlike MC_Stop, MC_CT_Halt can be aborted.

4.6.2.1 FB Diagram



4.6.2.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**MC_CT_Halt**		

Description	The FB interface to the NC axis. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	Data Type	AXIS_REF
Unit	N/A	Initial Value	N/A
Minimum	N/A	Maximum	N/A

4.6.2.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_Halt**		
Description	Executes the function block, where: FALSE = No action. TRUE = Rising edge starts execution of the function block.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_Halt**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Deceleration - Jerk		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Deceleration**		
Applies To	**MC_CT_Halt**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in Application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_Halt**		
Description	Specifies the maximum jerk rate to be used when decelerating, in Application units/s³.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s³	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

4.6.2.4 Outputs

FB Parameter	**Done**		
Applies To	**MC_CT_Halt**		
Description	Indicates that 0 velocity has been reached.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_Halt**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_Halt**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_Halt**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_Halt**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

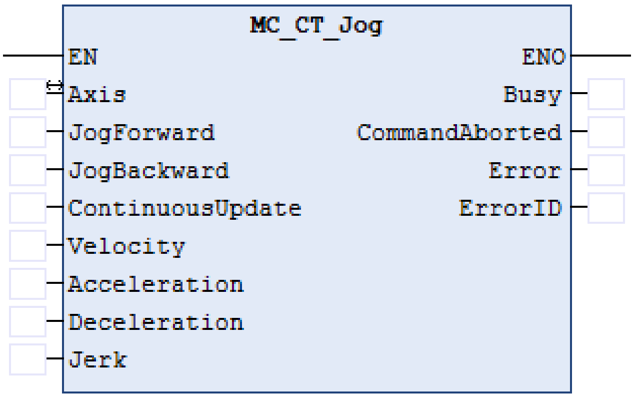
4.6.3 MC_CT_Jog

This function block is a continuous update version of MC_Jog.

If the “ContinuousUpdate” input is set TRUE and a value is changed at the “Velocity”, “Acceleration”, “Deceleration” or “Jerk” inputs, the new values take immediate effect.

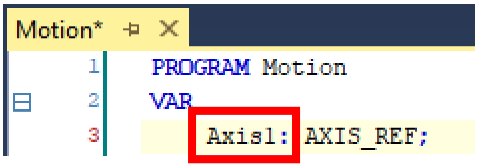
MC_CT_Jog causes the axis to move continuously while “JogForward” or “JogBackward” inputs are set TRUE using the velocity and rates defined at the “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs. Motion is stopped when the “JogForward” or “JogBackward” input is reset to FALSE.

4.6.3.1 FB Diagram



4.6.3.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**MC_CT_Jog**		

Description	The FB interface to the NC axis.
	This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	Motion*
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.3.3 Inputs

FB Parameter	**JogForward**		
Applies To	**MC_CT_Jog**		
Description	Moves the axis in the positive direction with the velocity and rates specified at the “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs: FALSE = Motion is stopped. TRUE = Motion is started in the positive direction.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**JogBackward**		
Applies To	**MC_CT_Jog**		
Description	Moves the axis in the negative direction with the velocity and rates specified at the “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs: FALSE = Motion is stopped. TRUE = Motion is started in the negative direction.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_Jog**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Velocity - Acceleration - Deceleration - Jerk		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Velocity**		
Applies To	**MC_CT_Jog**		
Description	Specifies the maximum velocity to be used during the movement, in Application units/s.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Acceleration**		
Applies To	**MC_CT_Jog**		
Description	Specifies the maximum acceleration rate to be used when accelerating, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_Jog**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_Jog**		
Description	Specifies the maximum jerk rate to be used when accelerating and decelerating, in Application units/s^3.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

4.6.3.4 Outputs

FB Parameter	**Busy**		
Applies To	**MC_CT_Jog**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_Jog**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_Jog**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><th>FB Parameter</th><th colspan="3">ErrorID</th></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_Jog</td></tr>
<tr><td>Description</td><td colspan="3">Indicates the NC error when Error = TRUE.
The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks.
For more information see section 5.14 MC_CT_NC_Error.</td></tr>
<tr><td>Usage</td><td>Output</td><td>Data Type</td><td>MC_CT_NC_Error</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>0</td><td>Maximum</td><td>19536</td></tr>
</table>

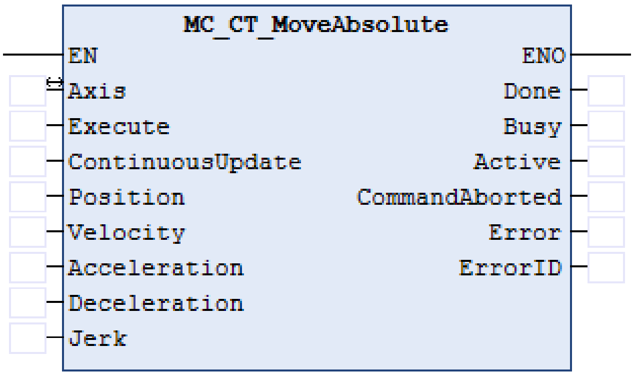
4.6.4 MC_CT_MoveAbsolute

This function block is a continuous update version of MC_MoveAbsolute.

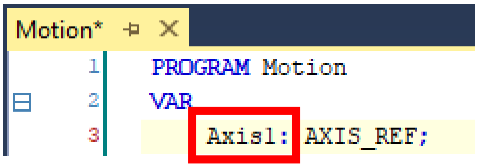
If the “ContinuousUpdate” input is set TRUE and a value is changed at the “Position”, “Velocity”, “Acceleration”, “Deceleration” or “Jerk” inputs, the new values take immediate effect.

MC_CT_MoveAbsolute causes the axis to move to an absolute position using the velocity and rates specified at the values for “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs. If no other motion control function blocks are executed, the movement ends with a velocity of 0.

4.6.4.1 FB Diagram



4.6.4.2 Inputs / Outputs

FB Parameter	**Axis**
Applies To	**MC_CT_MoveAbsolute**
Description	The FB interface to the NC axis. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis. This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice. 

```
PROGRAM Motion
VAR
    Axis1: AXIS_REF;
```

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.4.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Executes a movement to an absolute position specified at the “Position” input, using the velocity and rates specified at the “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs: FALSE = No action. TRUE = Execution of the function block is started.		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_MoveAbsolute**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Position - Velocity - Acceleration - Deceleration - Jerk		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**Position**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Specifies the absolute target position to be achieved on rising edge of the “Execute” input in Application units. Can be a positive or negative value.		
Usage	Input	Data Type	LREAL
Unit	Application units	Initial Value	0
Minimum	-LREAL MAX	Maximum	LREAL MAX

FB Parameter	**Velocity**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Specifies the maximum velocity to be used during the movement, in Application units/s.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Acceleration**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Specifies the maximum acceleration rate to be used when accelerating, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in Application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Specifies the maximum jerk rate to be used when accelerating and decelerating, in application units/s^3.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

4.6.4.4 Outputs

FB Parameter	**Done**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Indicates that the target position has been reached.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Active**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Indicates that the function block has control of the axis, where: FALSE = The function block is not running. TRUE = Indicates the function block has control of the axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_MoveAbsolute**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

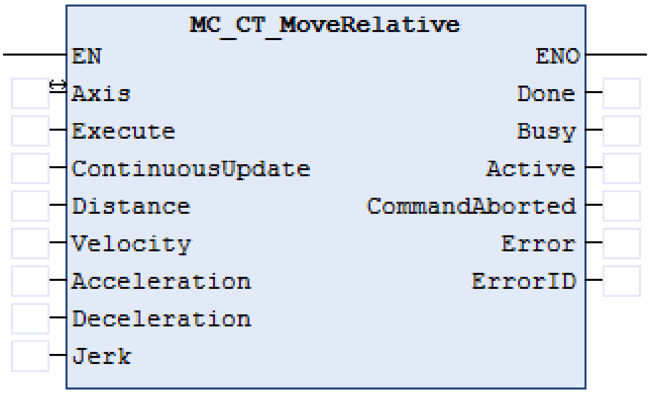
4.6.5 MC_CT_MoveRelative

This function block is a continuous update version of MC_MoveRelative.

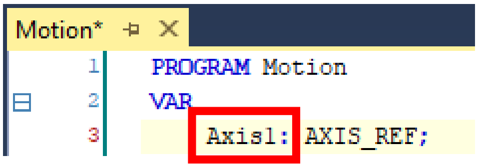
If the “ContinuousUpdate” input is set TRUE and a value is changed at the “Distance”, “Velocity”, “Acceleration”, “Deceleration” or “Jerk” inputs, the new values take immediate effect.

MC_CT_MoveRelative commands a controlled motion of a specified distance. The motion ends in standstill (unless blended to a follow-up movement).

4.6.5.1 FB Diagram



4.6.5.2 Inputs / Outputs

<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_MoveRelative</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.

Motion*
PROGRAM Motion
VAR
Axis1: AXIS_REF;</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.6.5.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_MoveRelative**		
Description	Executes a movement from the current axis position to a relative position specified at the "Distance" input, using the velocity and rates specified at the "Velocity", "Acceleration", "Deceleration" and "Jerk" inputs: FALSE = No action. TRUE = Execution of the function block is started on rising edge.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_MoveRelative**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Distance - Velocity - Acceleration - Deceleration - Jerk		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Distance**		
Applies To	**MC_CT_MoveRelative**		
Description	Specifies the relative target position to be achieved on rising edge of the "Execute" input in application units. Can be a positive or negative value.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**Velocity**		
Applies To	**MC_CT_MoveRelative**		
Description	Specifies the maximum velocity to be used during the movement, in application units/s.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Acceleration**		
Applies To	**MC_CT_MoveRelative**		
Description	Specifies the maximum acceleration rate to be used when accelerating, in application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_MoveRelative**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_MoveRelative**		
Description	Specifies the maximum jerk rate to be used when accelerating and decelerating, in application units/s^3.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

4.6.5.4 Outputs

FB Parameter	**Done**		
Applies To	**MC_CT_MoveRelative**		
Description	Indicates that the target position has been reached.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_MoveRelative**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Active**		
Applies To	**MC_CT_MoveRelative**		
Description	Indicates that the function block has control of the axis, where: FALSE = The function block is not running. TRUE = Indicates the function block has control of the axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_MoveRelative**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_MoveRelative**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the "ErrorID" output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_MoveRelative**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

4.6.6 MC_CT_MoveSuperimposed

The function block causes an additional movement, on top of an ongoing (original) movement of the axis. The original movement is not aborted, and instead the movement by the given distance is superimposed, or added to, the original motion.

If MC_CT_MoveSuperImposed is active, a further aborting command on the same axis interrupts both commands, i.e. MC_CT_MoveSuperImposed and the original motion command. A further buffered or blending command does not interrupt the superimposed movement but rather the superimposed movement is continued in parallel.

The function block MC_CT_MoveSuperImposed must be called after the function block of the original movement. If a movement command is called after MC_CT_MoveSuperImposed, the error SMC_MSI_INVALID_EXECUTION_ORDER is returned.

If an instance of MC_CT_MoveSuperImposed is active and the “ContinuousUpdate” input = TRUE, a value change at the “Distance” input will cause the active superimposed movement to be aborted and superseded with the new distance value, resulting in a new superimposed movement.

The new distance is relative to the start of the first superimposed movement. This means that a new distance of 0 will result in a slowdown that cancels the effect of the first superimposed movement.

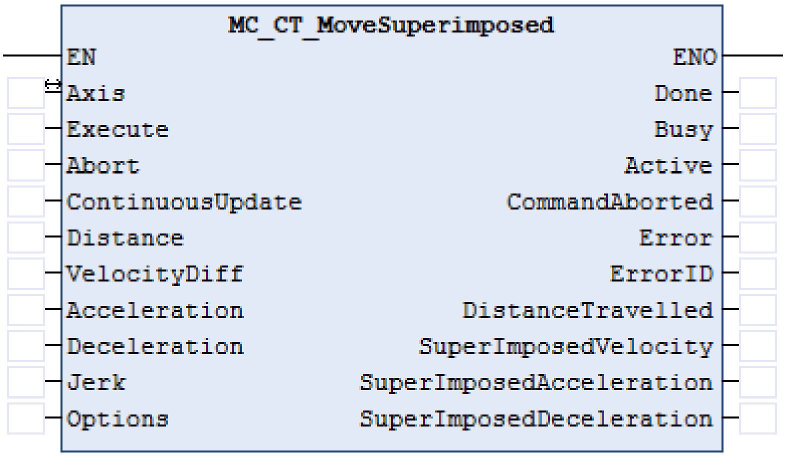
To halt a superimposed movement, use the function block MC_HaltSuperImposed. While MC_HaltSuperImposed is active, no further superimposed movement can be started.

If the axis is in the standstill state, MC_CT_MoveSuperImposed behaves like MC_MoveRelative.

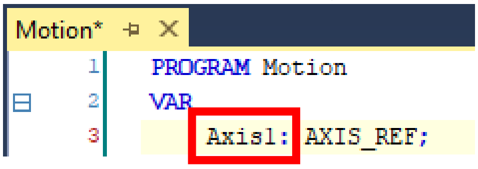
The Acceleration, Deceleration and Jerk inputs are additional values (not absolute values), which are added to the original movement. The original function block will always complete its movement within the same duration and with no consideration for any instance of MC_MoveSuperImposed acting at the same time.

MC_CT_MoveSuperImposed acts on the slave axis while MC_Phasing operates on the master axis.

4.6.6.1 FB Diagram



4.6.6.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	The FB interface to the NC axis. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis. This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.  Motion* PROGRAM Motion VAR Axis1: AXIS_REF;		
Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.6.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	A rising edge starts the superimposed movement using the distance and rates specified at the “Distance”, “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs: FALSE = No action. TRUE = Execution of the function block is started on a rising edge.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Abort**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Aborts an ongoing movement and resets all outputs. If set during an active movement, then the axis position will jump back to the position it would have had without the superimposed movement.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Distance		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Distance**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Specifies the additional distance in application units. If this value is changed during execution or a superimposed movement is aborted by a new superimposed movement, then the distance is relative to the start position of the interrupted superimposed movement.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**VelocityDiff**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Specifies the maximum velocity to be used during the movement, in application units/s.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Acceleration**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Specifies the maximum acceleration rate to be used when accelerating, in application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Specifies the maximum jerk rate to be used when accelerating and decelerating, in application units/s³.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s³	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Options**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Data structure containing additional, rarely used parameters. This input can normally remain open. AbortOption: defines the behaviour when the subordinate movement stops. The superimposed movement can be aborted or continued later.		
Usage	Input	**Data Type**	ST_SuperpositionOptions (STRUCT)
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.6.4 Outputs

FB Parameter	**Done**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Indicates that the target distance has been reached.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Active**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Displays when the command is being executed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_MoveSuperimposed**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

FB Parameter	**DistanceTravelled**		
Applies To	**MC_CT_ MoveSuperimposed**		
Description	Indicates the superimposed distance travelled so far. This output is only updated /valid when either the Busy or Done outputs are TRUE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SuperimposedVelocity**		
Applies To	**MC_CT_ MoveSuperimposed**		
Description	Indicates the current superimposed velocity. This output is only updated /valid when either the Busy or Done outputs are TRUE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SuperimposedAcceleration**		
Applies To	**MC_CT_ MoveSuperimposed**		
Description	Indicates the current superimposed acceleration. This output is only updated / valid when either the Busy or Done outputs are TRUE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**SuperimposedDeceleration**		
Applies To	**MC_CT_ MoveSuperimposed**		
Description	Indicates the current superimposed deceleration. This output is only updated / valid when either the Busy or Done outputs are TRUE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	0.0	**Maximum**	LREAL MAX

4.6.7 MC_CT_MoveSuperimposedAbsolute

The function block causes an additional movement, on top of an ongoing (original) movement of the axis. The original movement is not aborted, instead the movement by the given distance is superimposed.

This deffers from MC_CT_MoveSuperImposed in that although the starting point is relative to the original motion being super imposed, the Position input value is entered absolute to this value making it suitable for slave axis phasing applications e.g. if Position input is changed from 10 to 11 the slave axis superimposed position will only change by 1.

If MC_CT_MoveSuperImposedAbsolute is active, a further aborting command on the same axis interrupts both commands, i.e. MC_CT_MoveSuperImposedAbsolute and the original command. A further buffered or blending command does not interrupt the superimposed movement but rather the superimposed movement is continued in parallel.

The function block MC_CT_MoveSuperImposedAbsolute must be called after the function block of the original movement. If a movement command is called after MC_CT_MoveSuperImposedAbsolute, the error SMC_MSI_INVALID_EXECUTION_ORDER is returned.

If an instance of MC_CT_MoveSuperImposedAbsolute is active and the “ContinuousUpdate” input = TRUE, a value change at the “Position”, “VelocityDiff”, “Acceleration”, “Deceleration” or “Jerk” inputs will cause the active superimposed movement to be aborted and superseded with the new position value, resulting in a new superimposed movement.

The new distance is relative to the start of the first superimposed movement. This means that a new distance of 0 will result in a slowdown that cancels the effect of the first superimposed movement.

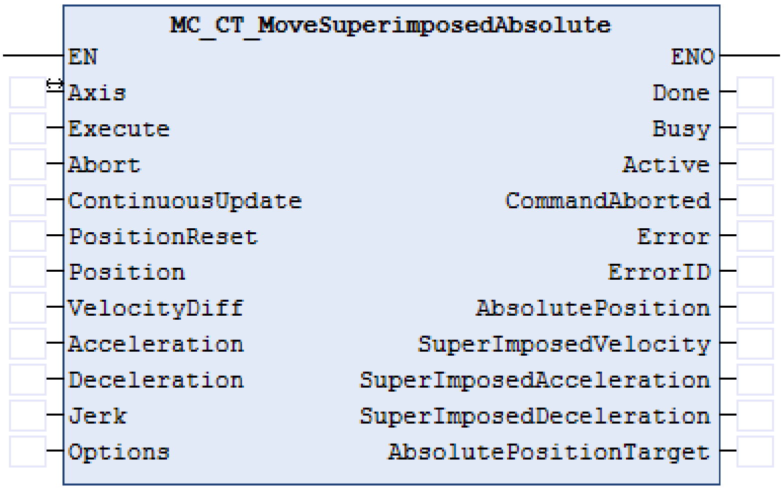
To halt a superimposed movement, use the function block MC_HaltSuperImposed. While MC_HaltSuperImposed is active, no further superimposed movement can be started.

If the axis is in the standstill state, MC_CT_MoveSuperImposedAbsolute behaves like MC_MoveAbsolute.

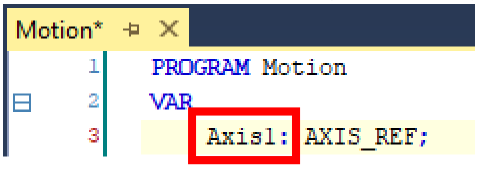
The Acceleration, Deceleration and Jerk inputs are additional values (not absolute values), which are added to the original movement. The original function block will always complete its movement within the same duration and with no consideration for any instance of MC_MoveSuperImposed acting at the same time.

MC_CT_MoveSuperImposedAbsolute acts on the slave axis while MC_Phasing operates on the master axis.

4.6.7.1 FB Diagram



4.6.7.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	The FB interface to the NC axis. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis. This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.  PROGRAM Motion VAR Axis1: AXIS_REF;		
Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.7.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	A rising edge starts the superimposed movement using the distance and rates specified at the “Distance”, “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs: FALSE = No action. TRUE = Execution of the function block is started on a rising edge.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Abort**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Aborts an ongoing movement and resets all outputs. If set during an active movement, then the axis position will jump back to the position it would have had without the superimposed movement.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Position - VelocityDiff - Acceleration - Deceleration - Jerk		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**PositionReset**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Resets the absolute position accumulator to 0 when set to TRUE and the Execute input is set to FALSE.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Position**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Specifies the additional position (absolute to 0) in application units. If this value is changed during execution or a superimposed movement is aborted by a new superimposed movement, then the distance is relative to the start position of the interrupted superimposed movement.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**VelocityDiff**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Specifies the maximum velocity to be used during the movement, in application units/s.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Acceleration**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Specifies the maximum acceleration rate to be used when accelerating, in application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Specifies the maximum jerk rate to be used when accelerating and decelerating, in application units/s^3.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Options**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Data structure containing additional, rarely used parameters. This input can normally remain open. AbortOption: defines the behaviour when the subordinate movement stops. The superimposed movement can be aborted or continued later.		
Usage	Input	**Data Type**	ST_SuperpositionOptions (STRUCT)
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.7.4 Outputs

FB Parameter	**Done**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates that the target distance has been reached.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Active**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Displays when the command is being executed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	Data Type	MC_CT_NC_Error
Unit	N/A	Initial Value	0
Minimum	0	Maximum	19536

FB Parameter	**AbsolutePosition**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates the absolute offset position travelled so far, absolute from 0. This output is only updated / valid when either the Busy or Done outputs are TRUE.		
Usage	Output	Data Type	LREAL
Unit	Application units	Initial Value	0
Minimum	-LREAL MAX	Maximum	LREAL MAX

FB Parameter	**SuperimposedVelocity**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates the current superimposed velocity. This output is only updated /valid when either the Busy or Done outputs are TRUE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SuperimposedAcceleration**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates the current superimposed Acceleration. This output is only updated /valid when either the Busy or Done outputs are TRUE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SuperimposedDeceleration**		
Applies To	**MC_CT_ MoveSuperimposedAbsolute**		
Description	Indicates the current superimposed deceleration. This output is only updated / valid when either the Busy or Done outputs are TRUE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**AbsolutePositionTarget**		
Applies To	**MC_CT_MoveSuperimposedAbsolute**		
Description	Indicates the current absolute position target for the function block. This value is retained when the “Execute” input is set to FALSE.		
Usage	Output	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

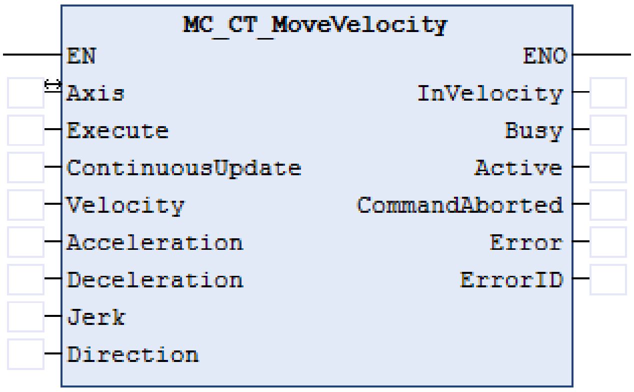
4.6.8 MC_CT_MoveVelocity

This function block is a continuous update version of MC_MoveVelocity.

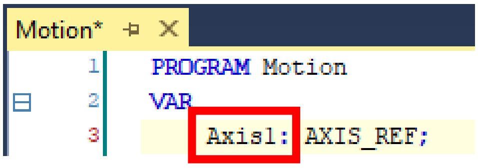
If the “ContinuousUpdate” input is set to TRUE and a value is changed at the “Velocity”, “Acceleration”, “Deceleration”, “Jerk” or “Direction” inputs, the new values take immediate effect.

MC_CT_MoveVelocity causes an endless motion at a specified velocity.

4.6.8.1 FB Diagram



4.6.8.2 Inputs / Outputs

<table>
<tr><td>FB Parameter</td><td colspan="3">Axis</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_MoveVelocity</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.


PROGRAM Motion
VAR
Axis1: AXIS_REF;</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

4.6.8.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_MoveVelocity**		
Description	Executes an endless movement in the direction specified at the “Direction” input using the velocity and rates specified at the “Velocity”, “Acceleration”, “Deceleration” and “Jerk” inputs: FALSE = No action. TRUE = Execution of the function block is started on a rising edge.		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_MoveVelocity**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Velocity - Acceleration - Deceleration - Jerk - Direction		
Usage	Input	Data Type	BOOL
Unit	N/A	Initial Value	FALSE
Minimum	FALSE	Maximum	TRUE

FB Parameter	**Velocity**		
Applies To	**MC_CT_MoveVelocity**		
Description	Specifies the maximum velocity to be used during the movement, in application units/s.		
Usage	Input	Data Type	LREAL
Unit	Application units/s	Initial Value	0
Minimum	>0.0	Maximum	LREAL MAX

FB Parameter	**Acceleration**		
Applies To	**MC_CT_MoveVelocity**		
Description	Specifies the maximum acceleration rate to be used when accelerating, in application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_MoveVelocity**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_MoveVelocity**		
Description	Specifies the maximum jerk rate to be used when accelerating and decelerating, in application units/s^3.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

<table>
<tr><td>FB Parameter</td><td colspan="3">Direction</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_MoveVelocity</td></tr>
<tr><td>Description</td><td colspan="3">Used to specify the direction of movement, e.g. “Current”, “Positive” or “Negative”.
Input is of type MC_Direction, for more information see section 5.4 MC_Direction.
HINT: Only “Current”, “Positive” and “Negative” values are allowed at this input.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>MC_Direction</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>0</td></tr>
<tr><td>Minimum</td><td>-1</td><td>Maximum</td><td>3</td></tr>
</table>

4.6.8.4 Outputs

FB Parameter	**InVelocity**		
Applies To	**MC_CT_MoveVelocity**		
Description	Indicates that the target velocity has been reached for the first time since execution of the function block.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_MoveVelocity**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Active**		
Applies To	**MC_CT_MoveVelocity**		
Description	Indicates that the function block has control of the axis, where: FALSE = The function block is not running. TRUE = Indicates the function block has control of the axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_MoveVelocity**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_MoveVelocity**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_MoveVelocity**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

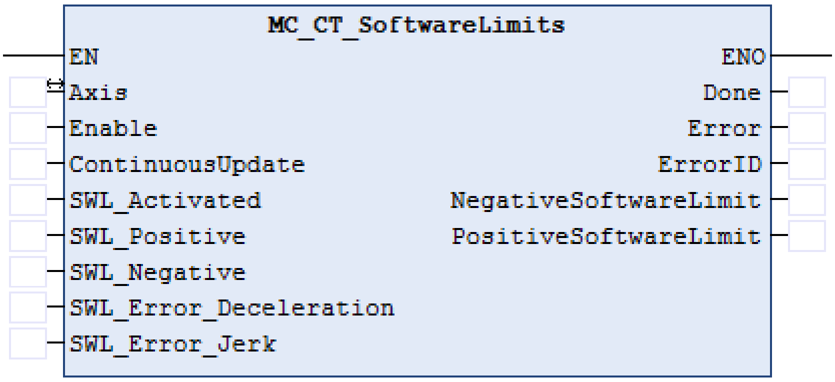
4.6.9 MC_CT_SoftwareLimits

This function block allows the software limits of a NC motion axis to be set during runtime.

If the “ContinuousUpdate” input is set TRUE and a value is changed at the “SWL_Activated”, “SWL_Positive”, “SWL_Negative”, “SWL_Error_Deceleration” or “SWL_Error_Jerk” inputs, the new values take immediate effect.

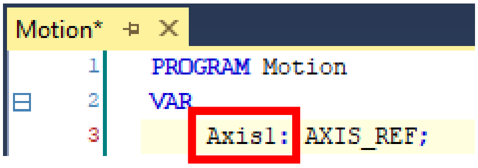
MC_CT_SoftwareLimits sets the position limits and the corresponding error reaction of an axis.

4.6.9.1 FB Diagram



4.6.9.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**MC_CT_SoftwareLimits**		

Description	The FB interface to the NC axis.
	This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
	This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.
	
	
	`PROGRAM Motion`
	`VAR`
	`Axis1: AXIS_REF;`		

Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.9.3 Inputs

FB Parameter	**Enable**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Enables or disables function block execution where: FALSE = The function block is no longer enabled and execution is stopped. TRUE = The function block is enabled and execution is started.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_SoftwareLimits**		
Description	When set to FALSE, any changes made to the software limit inputs after Enable = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - SWL_Activated - SWL_Positive - SWL_Negative - SWL_Error_Deceleration - SWL_Error_Jerk		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**SWL_Activated**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Enables or disables the position limits depending on the state when the “Execute” input is set TRUE, where: FALSE = Disables the position limits. TRUE = Activates the position limits.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**SWL_Positive**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Defines the absolute position of the positive software limit switch in application units.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SWL_Negative**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Defines the absolute position of the negative software limit switch in application units.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**SWL_Error_Deceleration**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Specifies the deceleration rate to be used when a software limit is activated.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**SWL_Error_Jerk**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Specifies the maximum jerk to be used when a software limit is activated. **HINT:** This only takes effect when set to a non-zero value. When set to a zero value the deceleration ramp is linear.		
Usage	Input	**Data Type**	LREAL
Unit	N/A	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

4.6.9.4 Outputs

FB Parameter	**Done**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Indicates that the function block has successfully completed.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

FB Parameter	**NegativeSoftwareLimit**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Indicates that the axis has entered a software limit state, where: FALSE = No error. TRUE = Indicates when the negative software limit has been reached.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**PositiveSoftwareLimit**		
Applies To	**MC_CT_SoftwareLimits**		
Description	Indicates that axis has entered a software limit state, where: FALSE = No error. TRUE = Indicates when the positive software limit has been reached.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

4.6.10 MC_CT_MoveModulo

This function block is a continuous update version of the MC_MoveModulo function block found in the TwinCAT NC Axis library.

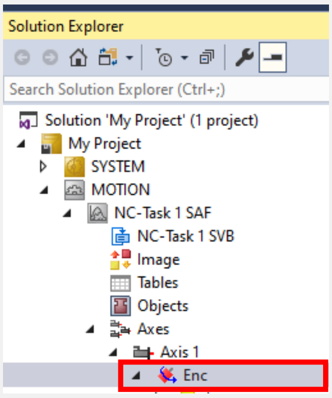
If the “ContinuousUpdate” input is set TRUE and a value is changed at the “Position”, “Velocity”, “Acceleration”, “Deceleration”, “Jerk” or “Direction” inputs, the new values take immediate effect.

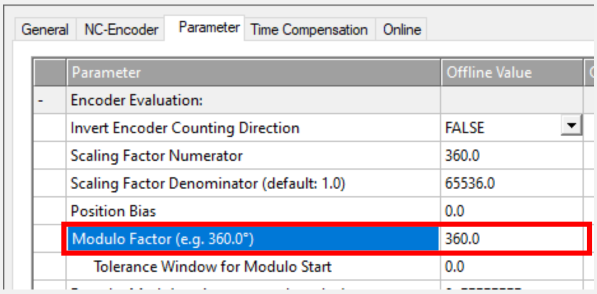
MC_CT_MoveModulo allows for the execution of modulo movements on a specified motion axis. A modulo rotation is based on the adjustable axis parameter modulo factor (e.g. 360°).

There are three possible movement types, specified by the value present at the "Direction" input:

- Positioning in positive direction
- Positioning in negative direction
- Positioning along shortest path

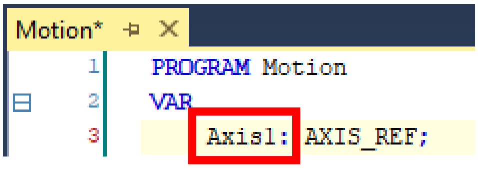
HINT: The modulo rollover factor is defined in the encoder parameter area which can be reached via the project tree under “**Project Name**” > “**Motion**” > “**NC-Task SAF**” > “**Axes**” > “**Axis**” > “**Enc**” and then selecting the “**Parameter**” tab:





	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	360.0
	Scaling Factor Denominator (default: 1.0)	65536.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	360.0
	Tolerance Window for Modulo Start	0.0

4.6.10.2 Inputs / Outputs

FB Parameter	**Axis**		
Applies To	**MC_CT_MoveModulo**		
Description	The FB interface to the NC axis. This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis. This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.  PROGRAM Motion VAR Axis1: AXIS_REF;		
Usage	InOut	**Data Type**	AXIS_REF
Unit	N/A	**Initial Value**	N/A
Minimum	N/A	**Maximum**	N/A

4.6.10.3 Inputs

FB Parameter	**Execute**		
Applies To	**MC_CT_MoveModulo**		
Description	A rising edge starts the modulo movement using the position, velocity, rates and movement type specified at the “Position”, “Velocity”, “Acceleration”, “Deceleration”, “Jerk” and “Direction” inputs: FALSE = No action. TRUE = Execution of the function block is started on a rising edge.		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ContinuousUpdate**		
Applies To	**MC_CT_MoveModulo**		
Description	When set to FALSE, any changes made to the target or rates after Execute = TRUE are ignored. When set to TRUE, any changes to values at the inputs listed below will cause the values present at the function block inputs to take immediate effect: - Position - Velocity - Acceleration - Deceleration - Jerk - Direction		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Position**		
Applies To	**MC_CT_MoveModulo**		
Description	Specifies the modulo target position to be used for positioning. If the axis is started from standstill, Target positions greater than 360° result in additional turns. Negative positions are not permitted.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0
Minimum	-LREAL MAX	**Maximum**	LREAL MAX

FB Parameter	**Velocity**		
Applies To	**MC_CT_MoveModulo**		
Description	Specifies the maximum velocity to be used during the movement, in application units/s.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Acceleration**		
Applies To	**MC_CT_MoveModulo**		
Description	Specifies the maximum acceleration rate to be used when accelerating, in application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Deceleration**		
Applies To	**MC_CT_MoveModulo**		
Description	Specifies the maximum deceleration rate to be used when decelerating, in application units/s^2.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^2	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Jerk**		
Applies To	**MC_CT_MoveModulo**		
Description	Specifies the maximum jerk rate to be used when accelerating and decelerating, in application units/s^3.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s^3	**Initial Value**	0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**Direction**		
Applies To	**MC_CT_MoveModulo**		
Description	Used to specify the direction of movement, e.g. “Current”, “Positive” or “Negative”. Input is of type MC_Direction (ENUM) where: **MC_Direction.MC_Positive_Direction** Moves in positive direction. **MC_Direction.MC_Shortest_Way** Selects the direction according to the shortest distance (only modulo axes). **MC_Direction.MC_Negative_Direction** Moves in negative direction. **MC_Direction.MC_Current_Direction** Keeps the current direction to reach the target (only modulo axes). For more information see section **5.4 MC_Direction.**		
Usage	Input	Data Type	MC_Direction
Unit	N/A	Initial Value	0
Minimum	0	Maximum	3

4.6.10.4 Outputs

FB Parameter	**Busy**		
Applies To	**MC_CT_MoveModulo**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Active**		
Applies To	**MC_CT_MoveModulo**		
Description	Indicates that the function block has control of the axis, where: FALSE = The function block is not running. TRUE = Indicates the function block has control of the axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_MoveModulo**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_MoveModulo**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_MoveModulo**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

4.6.11 MC_CT_Homing

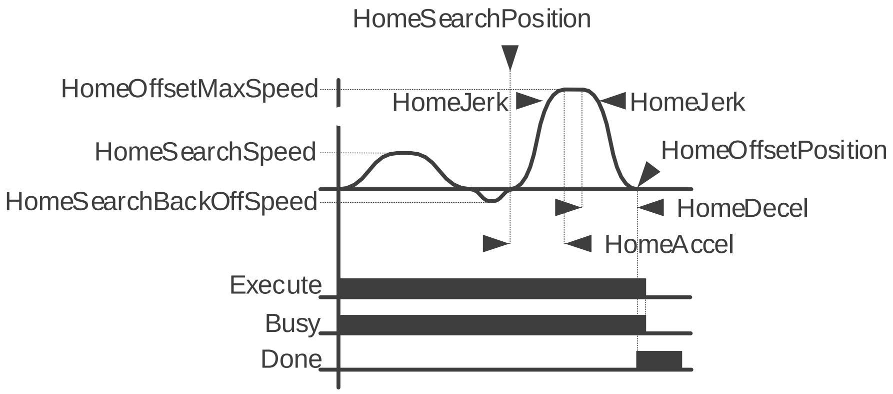
MC_CT_Homing provides the user with a more complete version of MC_Home, where the CoE writes that would normally be required to configure the CoE homing objects and any further sequencing of motion control function blocks that would typically be required to set the home position and carry out an offset movement have been encapsulated into a single function block, where the desired functionality can be selected via the function block inputs.

MC_CT_Homing comes with 10 different homing methods that allow it to be applied to many different mechanical arrangements. The homing routines consist of:

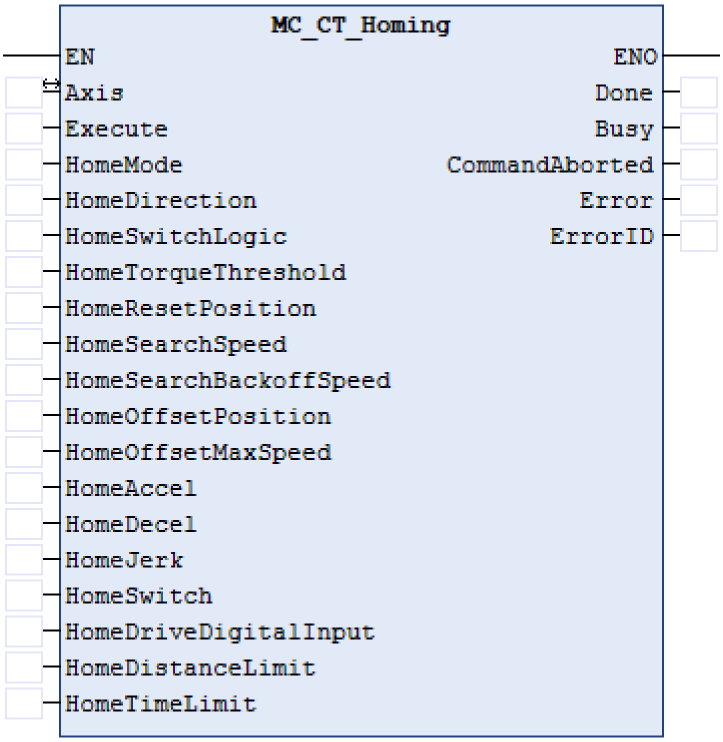
- Home at current position.
- Home to function block homing switch (function block input).
- Home to drive homing switch (selectable via HomeDriveDigitalInput).
- Home to function block homing switch then index marker pulse.
- Home to function block homing switch then reverse to index marker pulse.
- Home to drive homing switch (selectable via HomeDriveDigitalInput) then index marker pulse.
- Home to drive homing switch (selectable via HomeDriveDigitalInput) then reverse to index marker pulse.
- Home to index freeze (freeze source is selectable via drive Pr03.100).
- Home to torque threshold (set via HomeTorqueThreshold).
- Home to torque threshold (set via HomeTorqueThreshold) then reverse to index marker pulse.

The full details of these homing modes can be found on the HomeMode tables found in section **4.6.11.3 Inputs** below.

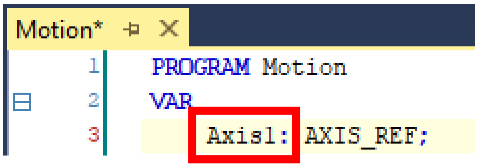
Homing is started using the Execute input, where the motion profile used to find the homing sensor (home search phase) and to reach the final resting position (home offset phase) is fully configurable. Status information is given to indicate when the homing reference has been activated and when the homing routine has been completed.



4.6.11.1 FB Diagram

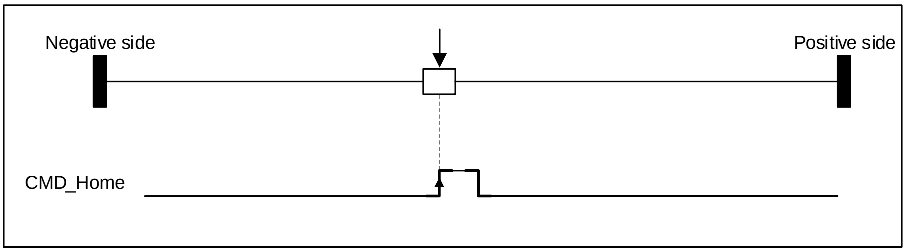


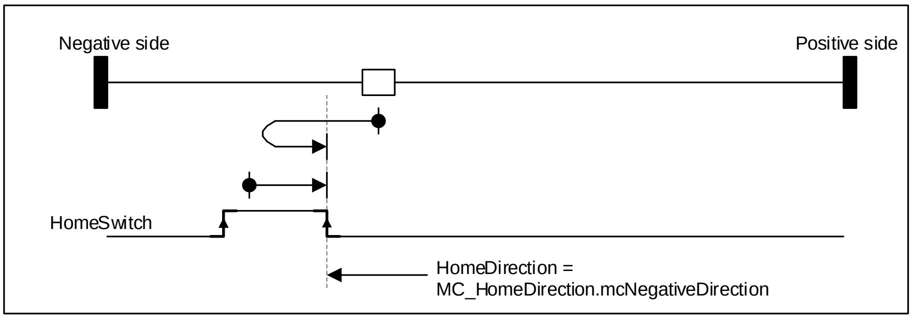
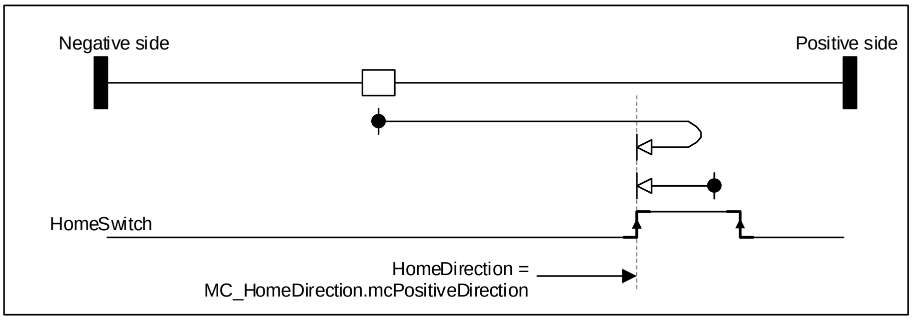
4.6.11.2 Inputs / Outputs

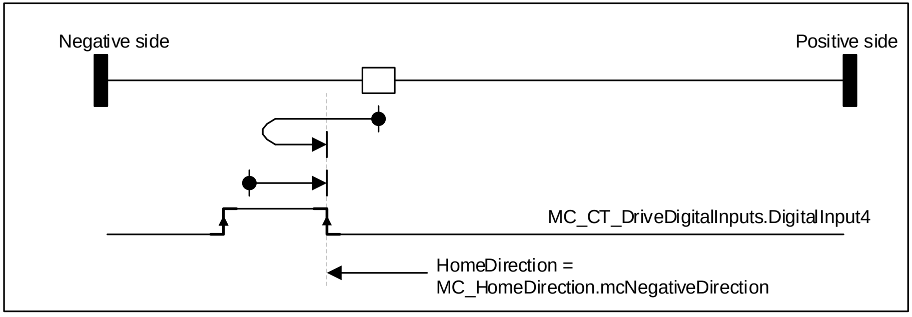
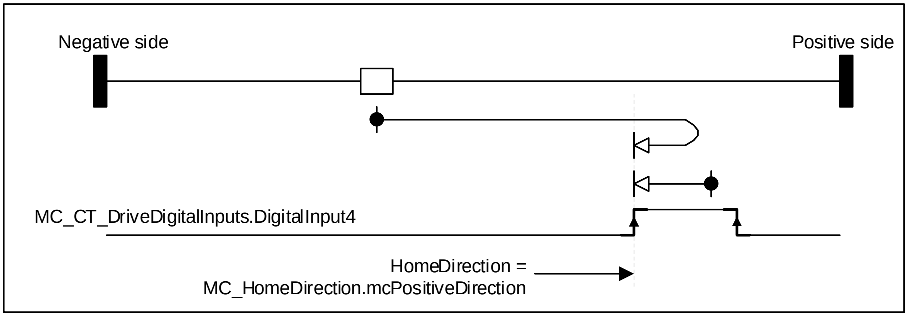
<table>
<tr><th>FB Parameter</th><th colspan="3">Axis</th></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_Homing</td></tr>
<tr><td>Description</td><td colspan="3">The FB interface to the NC axis.
This is used to invoke motion in the axis, send and receive data, and to monitor the PLCopen state machine for the specified axis.
This input/output must be assigned to the instance name of “AXIS_REF” for the axis to be controlled. Naming the slave axis something more appropriate like “Axis1” as shown in the image below is good practice.

Motion*
PROGRAM Motion
VAR
Axis1: AXIS_REF;</td></tr>
<tr><td>Usage</td><td>InOut</td><td>Data Type</td><td>AXIS_REF</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>N/A</td><td>Maximum</td><td>N/A</td></tr>
</table>

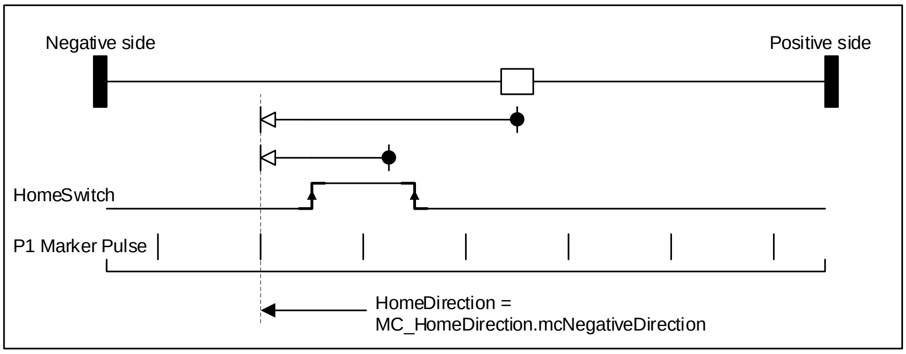
4.6.11.3 Inputs

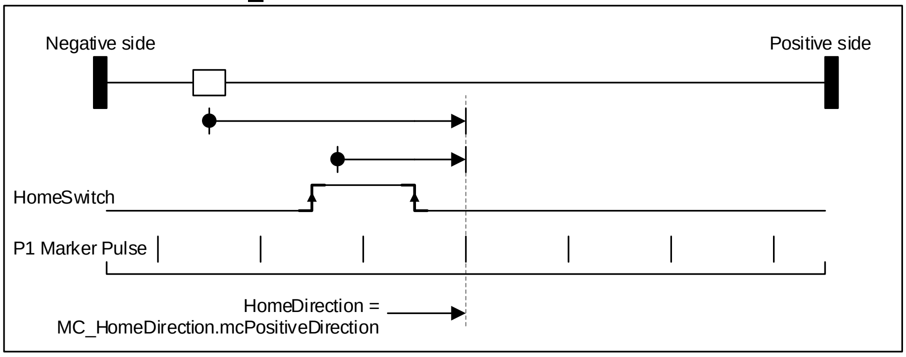
FB Parameter	**Execute**		
Applies To	**MC_CT_Homing**		
Description	Causes a homing routine to be executed, where: FALSE = No action TRUE = Begin homing **HINT:** When a virtual axis is in use, the only homing method available is "HomeAtCurrentPosition".		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

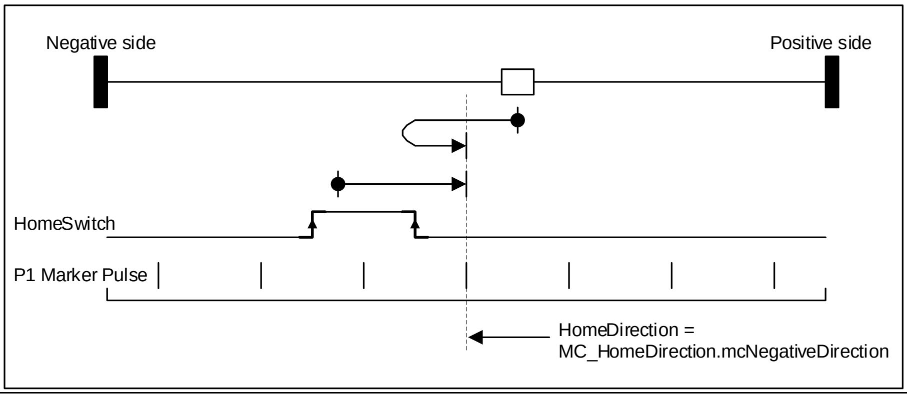
FB Parameter	**HomeMode.HomeAtCurrentPosition (0)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home at current position** The axis remains stationary, and the position counter is reset to the value present at the HomeResetPosition input. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. 

FB Parameter	**HomeMode.HomeToFbHomingSwitch (1)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to FB homing swich** The axis will perform an initial movement in the direction set by HomeDirection input using the velocity specified by HomeSearchSpeed. Once the HomeSwitch signal has been detected, a direction reversal is performed and the velocity changes to that specified at the HomeSearchBackoffSpeed input. The home position is set once the HomeSwitch signal is no longer detected where the axis is brought to a complete stop. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input. The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcNegativeDirection:  The diagram below shows the resulting homing sequence when CFG_HomeDirection = MC_HomeDirection.mcPositiveDirection: 

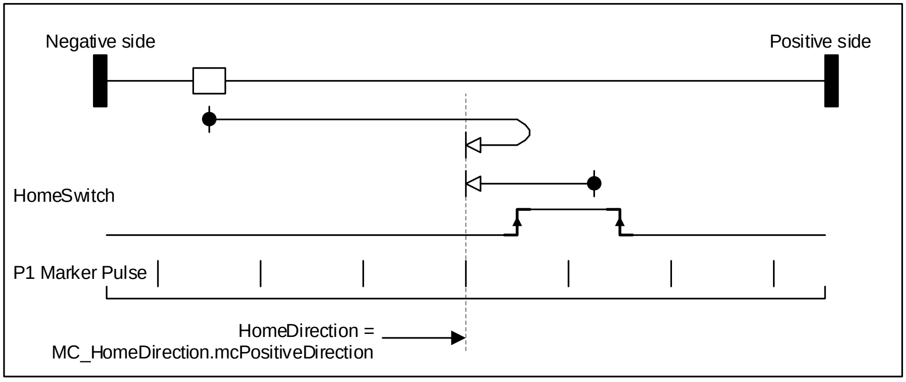
FB Parameter	**HomeMode.HomeToDriveHomingSwitch (2)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to drive homing switch** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. Once the drive digital input signal has been detected, a direction reversal is performed and the velocity changes to that specified at the HomeSearchBackoffSpeed input. The home position is set once the drive digital input signal is no longer detected where the axis is brought to a complete stop. The drive digital input source can be selected via the HomeDriveDigitalInput input where the data type is of MC_CT_DriveDigitalInputs (ENUM). For more information see section **5.5 MC_CT_DriveDigitalInputs**. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input. The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcNegativeDirection:  The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcPositiveDirection: 

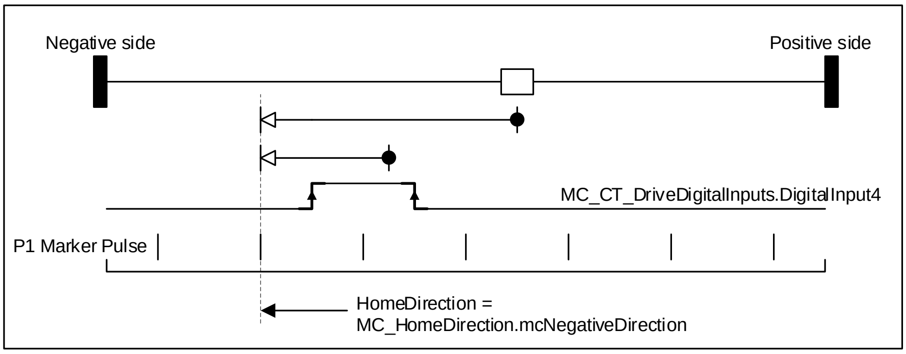
FB Parameter	**HomeMode. HomeToFbHomingSwitchThenIndexMarkerPulse (3)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to FB homing switch then index marker pulse** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. Once the HomeSwitch signal has been detected, the axis continues to move in the same direction, but the velocity will change to that specified at the HomeSearchBackoffSpeed input. Once the HomeSwitch signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input. The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcNegativeDirection: 

	The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcPositiveDirection: 

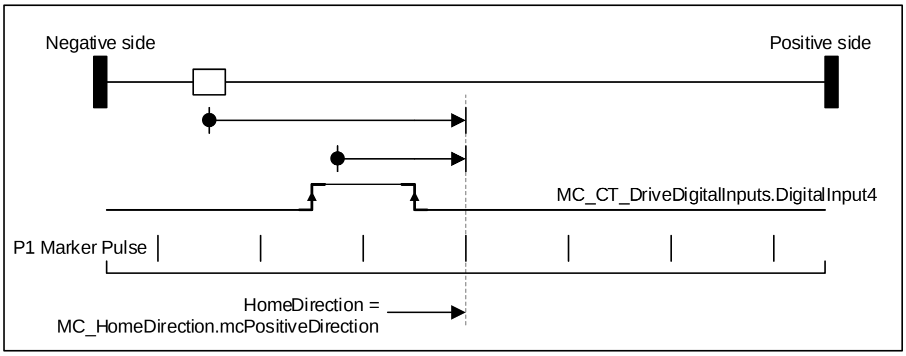
FB Parameter	**HomeMode.HomeToFbHomingSwitchThenReverseToIndexMarkerPulse (4)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to FB homing switch then reverse to index marker pulse** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. Once the HomeSwitch signal has been detected, a direction reversal is performed and the velocity changes to that specified at the HomeSearchBackoffSpeed input. Once the HomeSwitch signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input. The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcNegativeDirection: 

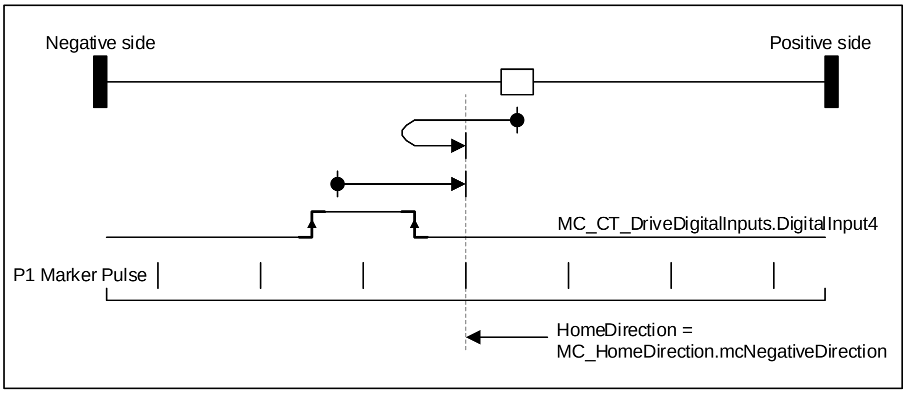
The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcPositiveDirection:



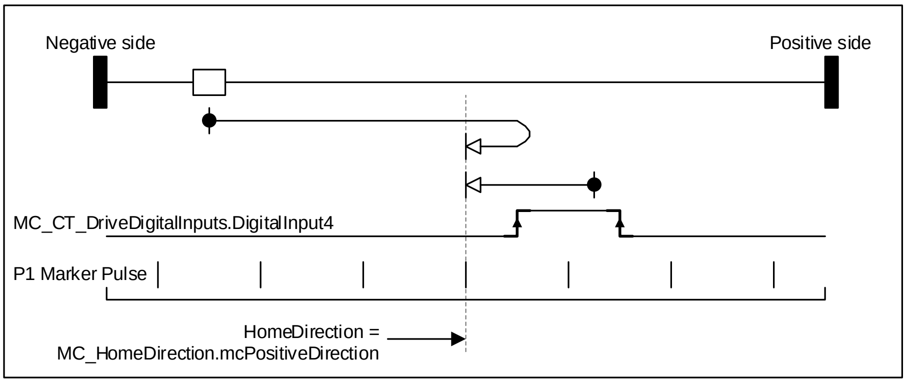
FB Parameter	**HomeMode. HomeToDriveHomingSwitchThenIndexMarkerPulse (5)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to drive homing switch then index marker pulse** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. Once the drive digital input signal has been detected, the axis continues to move in the same direction, but the velocity will change to that specified at the HomeSearchBackoffSpeed input. Once the drive digital input signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The drive digital input source can be selected via the HomeDriveDigitalInput input where the data type is of MC_CT_DriveDigitalInputs (ENUM). For more information see section **5.5 MC_CT_DriveDigitalInputs**. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input. The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcNegativeDirection: 

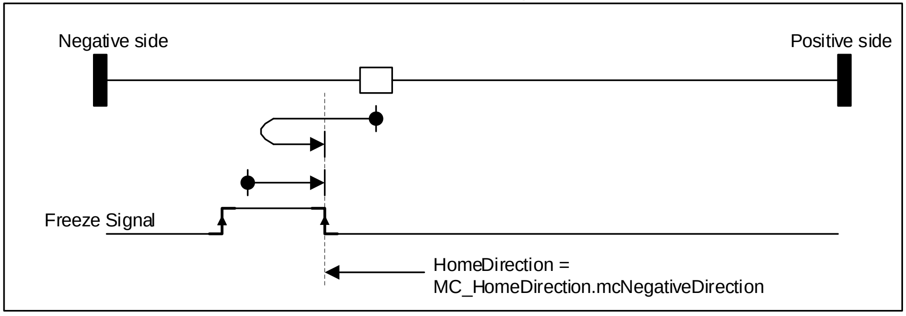
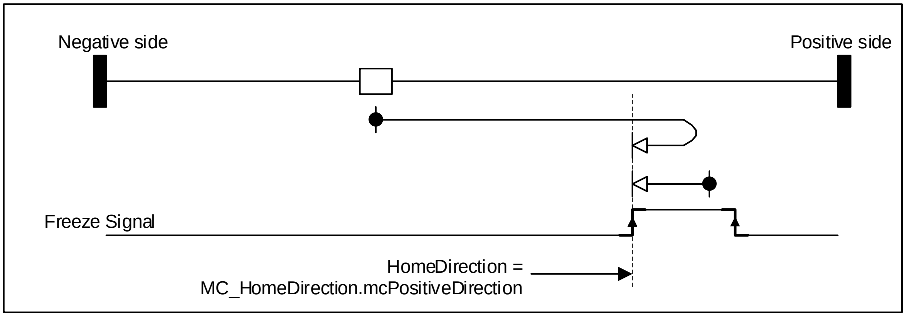
The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcPositiveDirection:



FB Parameter	**HomeMode. HomeToDriveHomingSwitchThenReverseIndexMarkerPulse (6)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to drive homing switch then reverse to index marker pulse** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. Once the drive digital input signal has been detected, a direction reversal is performed and the velocity changes to that specified at the HomeSearchBackoffSpeed input. Once the drive digital input signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The drive digital input source can be selected via the HomeDriveDigitalInput input where the data type is of MC_CT_DriveDigitalInputs (ENUM). For more information see section **5.5 MC_CT_DriveDigitalInputs**. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input. The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcNegativeDirection: 

The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcPositiveDirection



FB Parameter	**HomeMode.HomeToIndexFreeze (7)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to index freeze** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. The axis uses the Drive F1 freeze input (Digital input 4 by default) freeze mechanism as the home switch input, to complete a 1 stage (search only) homing routine. Once the freeze signal has been detected a direction reversal is performed where the axis moves to the absolute position specified at the HomeOffsetPosition input (0 by default), where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs Once homing has completed, the axis position is set to the value present at the HomeResetPosition input. The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcNegativeDirection:  The diagram below shows the resulting homing sequence when HomeDirection = MC_HomeDirection.mcPositiveDirection: 

FB Parameter	**HomeMode.HomeToTorqueThreshold (8)**
Applies To	**MC_CT_Homing**
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to torque threshold** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. The axis uses a Percentage Load >= HomeTorqueThreshold condition to satisfy the homing switch input. Once this homing switch condition has been detected, the axis will perform a reversal where the velocity changes to that specified at the HomeSearchBackoffSpeed to perform a 2 stage (Search and Backoff) homing routine. Once the Percentage Load >= HomeTorqueThreshold condition is no longer detected, the axis is brought to a complete stop and the home position is set. The rates for this homing mode are defined at the HomeAccel, HomeDecel and HomeJerk inputs The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input.

FB Parameter	**HomeMode. HomeToTorqueThresholdThenReverseIndexMarkerPulse (9)**		
Applies To	**MC_CT_Homing**		
Description	Specifies homing mode to be used when Execute is set to TRUE. Data type is of the MC_CT_HomingModes ENUM. **Home to torque threshold then index marker pulse** The axis will perform an initial movement in the direction set by HomeDirection at the velocity specified by HomeSearchSpeed. The axis uses a Percentage Load >= HomeTorqueThreshold condition to satisfy the homing switch input. Once this homing switch condition has been detected, the axis will perform a reversal where the velocity changes to that specified at the HomeSearchBackoffSpeed to perform a 2 stage (Search and Backoff) homing routine. Once the Percentage Load >= HomeTorqueThreshold condition is no longer detected, the home position is set at the rising edge of the next index marker pulse where the axis is then is brought to a complete stop. are defined at the HomeAccel, HomeDecel and HomeJerk inputs The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the HomeOffsetPosition input, where the maximum speed for this movement can be specified via the HomeOffsetMaxSpeed input. Once homing has completed, the axis position is set to the value present at the HomeResetPosition input.		
Usage	Input	**Data Type**	MC_CT_HomingModes
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	9

<table>
<tr><td>FB Parameter</td><td colspan="3">HomeDirection</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_Homing</td></tr>
<tr><td>Description</td><td colspan="3">Specifies the axis direction when homing begins.
For example, to start the homing routine in the forward direction, CFG_HomeDirection = MC_Home_Direction.mcPositiveDirection or to start the homing routine in the reverse direction, CFG_HomeDirection = MC_Home_Direction.mcNegativeDirection.
HINT: Although this input accepts the type MC_Home_Direction (ENUM), the only values accepted at this input are “MC_HomeDirection.mcNegativeDirection” and “MC_HomeDirection.mcPositiveDirection”.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>MC_Home_Direction</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>mcPositiveDirection</td><td>Maximum</td><td>mcNegativeDirection</td></tr>
</table>

<table>
<tr><td>FB Parameter</td><td colspan="3">HomeSwitchLogic</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_Homing</td></tr>
<tr><td>Description</td><td colspan="3">Specifies the polarity of the home switch logic using the MC_CT_SwitchLogic (ENUM) where:
MC_CT_SwitchLogic.PositiveLogic
The home switch input must be set TRUE to active the home switch.
MC_CT_SwitchLogic.NegativeLogic
The home switch input must be set FALSE to activate the home switch.
For more information see section 5.18 MC_CT_SwitchLogic.</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>MC_CT_SwitchLogic</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>PositiveLogic</td><td>Maximum</td><td>NegativeLogic</td></tr>
</table>

FB Parameter	**HomeTorqueThreshold**		
Applies To	**MC_CT_Homing**		
Description	Specifies the percentage load threshold for the “**MC_CT_HomingModes.HomeToTorqueThreshold**” and “**HomeToTorqueThresholdThenReverseIndexMarkerPulse**” homing modes. A drive percentage load value greater than or equal to the CFG_HomeTorqueThreshold value results in a home switch TRUE condition for the home to torque threshold homing modes. During homing the current limit for the drive is set to this value so that excessive torque is not applied to the motor when bumping into the end stops.		
Usage	Input	**Data Type**	LREAL
Unit	%	**Initial Value**	20.0
Minimum	>0.0	**Maximum**	100.0

FB Parameter	**HomeResetPosition**		
Applies To	**MC_CT_Homing**		
Description	Specifies the position that the axis will be reset to once homing completes.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeSearchSpeed**		
Applies To	**MC_CT_Homing**		
Description	Specifies the speed used for the home switch search phase of a homing routine.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeSearchBackoffSpeed**		
Applies To	**MC_CT_Homing**		
Description	Specifies the speed used for backing from the home switch, this should typically be 10% of the velocity specified for CFG_HomeSearchSpeed.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	100.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeOffsetPosition**		
Applies To	**MC_CT_Homing**		
Description	Specifies the final absolute resting position of the axis used during the offset phase of a homing routine.		
Usage	Input	**Data Type**	LREAL
Unit	Application units	**Initial Value**	0.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeOffsetMaxSpeed**		
Applies To	**MC_CT_Homing**		
Description	Specifies the speed used for moving to the offset position specified at the CFG_HomeOffsetPosition input used during the offset phase of a homing routine.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeAccel**		
Applies To	**MC_CT_Homing**		
Description	Specifies the maximum acceleration rate to be used when homing, in Application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeDecel**		
Applies To	**MC_CT_Homing**		
Description	Specifies the maximum deceleration rate to be used when homing, in Application units/s².		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s²	**Initial Value**	1000.0
Minimum	>0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeJerk**		
Applies To	**MC_CT_Homing**		
Description	Specifies the jerk rate to be used when homing, in Application units/s³. A value of 0 will result in linear ramps.		
Usage	Input	**Data Type**	LREAL
Unit	Application units/s³	**Initial Value**	0.0
Minimum	0.0	**Maximum**	LREAL MAX

FB Parameter	**HomeSwitch**		
Applies To	**MC_CT_Homing**		
Description	Homing switch input that can be used in conjunction with the following homing modes as defined by CFG_HomeMode: **CFG_HomeMode.HomeToFbHomingSwitch (1)** Home to FB homing switch. **CFG_HomeMode.HomeToFbHomingSwitchThenIndexMarkerPulse (3)** Home to FB homing switch then index marker pulse. **CFG_HomeMode.HomeToFbHomingSwitchThenReverseIndexMarkerPulse (4)** Home to FB homing switch then reverse to index marker pulse. FALSE = Homing switch is not activated TRUE = Homing switch is activated		
Usage	Input	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

<table>
<tr><td>FB Parameter</td><td colspan="3">HomeDriveDigitalInput</td></tr>
<tr><td>Applies To</td><td colspan="3">MC_CT_Homing</td></tr>
<tr><td>Description</td><td colspan="3">

Selects which drive signal is used as the drive homing switch used in homing modes “**HomeToDriveHomingSwitch**”, “**HomeToDriveHomingSwitchThenIndexMarkerPulse**” and “**HomeToDriveHomingSwitchThenReverseIndexMarkerPulse**”.

Data type is of the MC_CT_DriveDigitalInputs (ENUM) where:

MC_CT_DriveDigitalInputs.DigitalInOut1
Digital In Out 1 is selected.

MC_CT_DriveDigitalInputs.DigitalInOut2
Digital In Out 2 is selected.

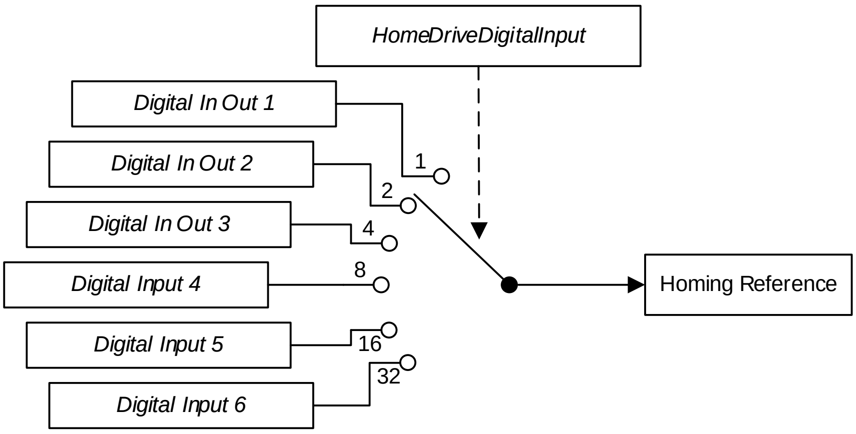
MC_CT_DriveDigitalInputs.DigitalInOut3
Digital In Out 3 is selected.

MC_CT_DriveDigitalInputs.DigitalInput4
Digital Input 4 is selected.

MC_CT_DriveDigitalInputs.DigitalInput5
Digital Input 5 is selected.

MC_CT_DriveDigitalInputs.DigitalInput6

Digital Input 6 is selected.



For more information see section **5.5 MC_CT_DriveDigitalInputs**.

</td></tr>
<tr><td>Usage</td><td>Input</td><td>Data Type</td><td>MC_CT_DriveDigitalInputs</td></tr>
<tr><td>Unit</td><td>N/A</td><td>Initial Value</td><td>N/A</td></tr>
<tr><td>Minimum</td><td>DigitalInOut1</td><td>Maximum</td><td>DigitalInput6</td></tr>
</table>

4.6.11.4 Outputs

FB Parameter	**Done**		
Applies To	**MC_CT_Homing**		
Description	Indicates that homing has completed, where: FALSE = The homing routine has not completed or is in progress. TRUE = The homing routine has completed successfully.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Busy**		
Applies To	**MC_CT_Homing**		
Description	Displays when the function block is running, where: FALSE = The function block is not running. TRUE = Indicates the function block is running and must continue to be called.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**CommandAborted**		
Applies To	**MC_CT_Homing**		
Description	Indicates that execution has been interrupted by another function block instance operating on the same axis.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**Error**		
Applies To	**MC_CT_Homing**		
Description	Indicates that the function block has entered the error state, where: FALSE = No error. TRUE = Indicates that the function block has encountered an error, further diagnostic error information will be present at the “ErrorID” output.		
Usage	Output	**Data Type**	BOOL
Unit	N/A	**Initial Value**	FALSE
Minimum	FALSE	**Maximum**	TRUE

FB Parameter	**ErrorID**		
Applies To	**MC_CT_Homing**		
Description	Indicates the NC error when Error = TRUE. The datatype is MC_CT_NC_Error (ENUM). This ENUM collates NC axis errors from multiple categories into one enumeration and contains strings for each possible error number that can be returned by the internal NC function blocks. For more information see section **5.14 MC_CT_NC_Error**.		
Usage	Output	**Data Type**	MC_CT_NC_Error
Unit	N/A	**Initial Value**	0
Minimum	0	**Maximum**	19536

5 Enumerations

This section contains descriptions of the various Enumerations (ENUMs) used in the Control Techniques NC Axis Interface Library.

5.1 MC_CT_ActiveAlarms

MC_CT_ActiveAlarms is an enumeration of possible drive alarms generated at the “STS_DriveActiveAlarm” output of the MC_CT_EGB_Axis, MC_CT_PositionAxis and MC_CT_SpeedAxis function blocks.

Enumeration	Value	Description
NoAlarm	0	Indicates that no alarm is present.
BrakeResistor	1	Indicates that a brake resistor alarm is present. The braking IGBT is active, and the braking resistor thermal accumulator is greater than 75%.
MotorOverload	2	Indicates that motor overload alarm is present. The motor overload accumulator is above 75% and the output current is greater than the rated current.
RegenInductorOverload	3	Indicates that a regen inductor overload alarm is present. The inductor overload accumulator is above 75% and the output current is greater than the rated current.
DriveOverload	4	Indicates that a drive overload alarm is present. The temperature of the drive is greater than 90% of the over temperature trip level.
Autotune	5	Indicates that an autotune test is in progress.
LimitSwitch	6	Indicates that a limit switch is active.
FireMode	7	Indicates that fire mode is active.
LowLoad	8	Indicates that a low load condition has been detected.
OptionSlot1	9	The option module in slot 1 has indicated an alarm.
OptionSlot2	10	The option module in slot 2 has indicated an alarm.
OptionSlot3	11	The option module in slot 3 has indicated an alarm.
OptionSlot4	12	The option module in slot 4 has indicated an alarm.

5.2 MC_CT_AxisMode

MC_CT_AxisMode is an enumeration that defines the type of axis reset that will take place and is consumed at the “AxisResetMode” input of the MC_CT_AxisFilter and MC_CT_AxisSelector function blocks.

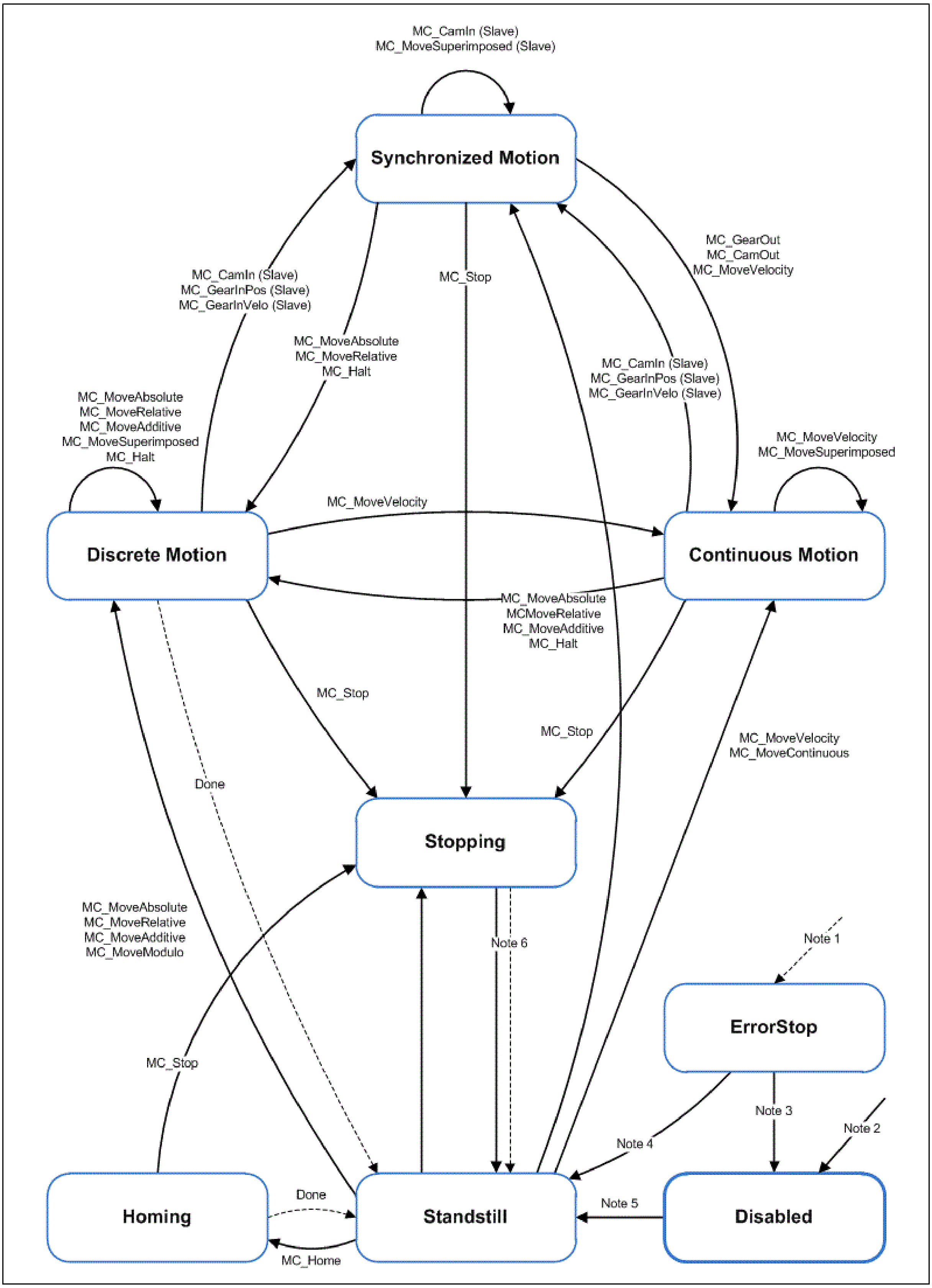
Enumeration	Value	Description
Absolute	0	Absolute means that the output position integrator will be reset to the selected master absolute position + ResetOffset when AxisReset = TRUE (Rising Edge).
Relative	1	Relative means that the output position integrator will be reset to 0 + ResetOffset when AxisReset = TRUE (Rising Edge).

5.3 MC_CT_AxisStates

MC_CT_AxisStates is an enumeration that is used to display the current PLCopen state of an NC Axis. It is used to show the PLC opens states concisely when compared with MC_AxisStates in the NC axis library, where all states have the prefix “MC_AXISSTATE_” making them hard to read in an LD diagram program.

The “MC_CT_EGB_Axis”, “MC_CT_PositionAxis”, “MC_CT_SpeedAxis” and “MC_CT_VirtualAxis” function blocks included with the Control Techniques NC Axis Interface Library have an output of “STS_PLCopenState” which displays the current PLCopen state for the NC axis.

Enumeration	Value	Description
Undefined	0	The axis state is “Undefined”.
Disabled	1	The axis is in the “Disabled” state, check the diagram and notes below for more information.
Standstill	2	The axis is in the “Standstill” state, check the diagram and notes below for more information.
Errorstop	3	The axis is in the “Errorstop” state, check the diagram and notes below for more information.
Stopping	4	The axis is in the “Stopping” state, check the diagram and notes below for more information.
Homing	5	The axis is in the “Homing” state, check the diagram and notes below for more information.
DiscreteMotion	6	The axis is in the “DiscreteMotion” state, check the diagram and notes below for more information.
ContinuousMotion	7	The axis is in the “ContinuousMotion” state, check the diagram and notes below for more information.
SynchronisedMotion	8	The axis is in the “SynchronisedMotion” state, check the diagram and notes below for more information.



NOTE 1: From any state. An error in the axis has occured.

NOTE 2: From any state. MC_Power.Enable = FALSE . There is no error in the axis.

NOTE 3: MC_Reset and MC_Power.Status = FALSE.

NOTE 4: MC_Reset and MC_Power.Status = TRUE and MC_Power.Enable = TRUE.

NOTE 5: MC_Power.Enable = TRUE and MC_Power.Status = TRUE.

NOTE 6::MC_Stop.Done = TRUE and MC_Stop.Execute = FALSE.

5.4 MC_Direction

MC_Direction is an enumeration contained within the TwinCAT NC library and is used with many of the “MC_CT” function blocks included with the Control Techniques NC Axis Interface Library.

Enumeration	Value	Description
MC_Positive_Direction	0	Moves in positive direction.
MC_Shortest_Way	1	Selects the direction according to the shortest distance (only modulo axes).
MC_Negative_Direction	2	Moves in negative direction.
MC_Current_Direction	3	Keeps the current direction to reach the target (only modulo axes).

5.5 MC_CT_DriveDigitalInputs

MC_CT_DriveDigitalInputs is an enumeration for the possible drive inputs that can be selected when using the drive homing switch as a homing method. CT_DriveDigitalInputs can be specified at the “CFG_HomeDriveDigitalInput” input of MC_CT_EGB_Axis and MC_CT_PositionAxis.

Enumeration	Value	Description
DigitalInOut1	1	Digital In Out 1 is selected.
DigitalInOut2	2	Digital In Out 2 is selected.
DigitalInOut3	4	Digital In Out 3 is selected.
DigitalInput4	8	Digital Input 4 is selected.
DigitalInput5	16	Digital Input 5 is selected.
DigitalInput6	32	Digital Input 6 is selected.

5.6 MC_CT_EGB_Modes

MC_CT_EGB_Modes is an enumeration for the possible EGB lock types specified at the “CFG_EGB_Mode” input of MC_CT_EGB_Axis.

Enumeration	Value	Description
NonRigidLock	0	The EGB slave axis will catch up with the speed of the master axis position reference and then become locked.
RigidLock	1	The EGB slave axis will catch up with the speed and relative position of the master axis position reference and then become locked.
Registration	2	A EGB master axis freeze event will cause the EGB slave axis to catch up with the speed and relative position of the master axis position reference that was captured when the freeze event took place and then become locked.

5.7 MC_CT_ErrorCategories

MC_CT_ErrorCategories is an enumeration for the possible error categories generated at the "STS_ErrorFB_Category" output of MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis and MC_CT_VirtualAxis function blocks.

Enumeration	Value	Description
NoErrorCategory	0	No axis error is present or the current error is not categorised.
CurrentLimit	1	Indicates that the current error relates to the current limit functionality of the axis.
EGB	2	Indicates that the current error relates to the EGB functionality of the axis.
SlaveFreeze	3	Indicates that the current error relates to the freeze functionality of the axis being controlled.
Homing	4	Indicates that the current error relates to the homing functionality of the axis.
Jogging	5	Indicates that the current error relates to the jogging functionality of the axis.
Limits	6	Indicates that the current error relates to the limit functionality of the axis.
Positioning	7	Indicates that the current error relates to the positioning functionality of the axis.
Power	8	Indicates that the current error relates to the power functionality of the axis.
Reset	9	Indicates that the current error relates to the reset functionality of the axis.
Status	10	Indicates that the current error relates to the status functionality of the axis.
Stopping	11	Indicates that the current error relates to the stopping functionality of the axis.
FollowingErrorStop	12	Indicates that the current error relates to the following error stop functionality of the axis.
Speed	13	Indicates that the current error relates to the speed functionality of the axis.

5.8 MC_CT_ErrorList

MC_CT_ErrorErrorlist is an enumeration for the possible error categories generated at the “AdditionalErrorID” output of a range of MC_CT function blocks.

Enumeration	Value	Description
NoErrorCategory	0	No axis error is present or the current error is not categorised.
CurrentLimit	1	Indicates that the current error relates to the current limit functionality of the axis.
EGB	2	Indicates that the current error relates to the EGB functionality of the axis.
SlaveFreeze	3	Indicates that the current error relates to the freeze functionality of the axis being controlled.
Homing	4	Indicates that the current error relates to the homing functionality of the axis.
Jogging	5	Indicates that the current error relates to the jogging functionality of the axis.
Limits	6	Indicates that the current error relates to the limit functionality of the axis.
Positioning	7	Indicates that the current error relates to the positioning functionality of the axis.
Power	8	Indicates that the current error relates to the power functionality of the axis.
Reset	9	Indicates that the current error relates to the reset functionality of the axis.
Status	10	Indicates that the current error relates to the status functionality of the axis.
Stopping	11	Indicates that the current error relates to the stopping functionality of the axis.
FollowingErrorStop	12	Indicates that the current error relates to the following error stop functionality of the axis.
Speed	13	Indicates that the current error relates to the speed functionality of the axis.

5.9 MC_CT_ErrorModes

MC_CT_ErrorModes is an enumeration for the possible PLCopen error capturing modes specified at the “CFG_ErrorMode” input of MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis and MC_CT_VirtualAxis.

Enumeration	Value	Description
ContinuousUpdate	0	NC Axis errors will be captured and displayed continuously.
OneShotTrigger	1	NC Axis errors will be captured and displayed on a one-shot basis. Once an error has been captured, the CMD_ErrorReportReset input will need to be set TRUE to reset the error outputs.

5.10 MC_CT_FB_Types

MC_CT_FB_Types is an enumeration for the possible function block types generated at the “STS_ErrorFB_Type” output of MC_CT_EGB_Axis, MC_CT_PositionAxis, MC_CT_SpeedAxis and MC_CT_VirtualAxis function blocks.

Enumeration	Value	Description
NotAssigned	0	The function block is not in error, or the internal function block in error is not assigned to a type.
MC_Power	1	The internal instance of MC_Power is in error.
MC_Jog	2	The internal instance of MC_Jog is in error.
MC_MoveVelocity	3	The internal instance of MC_MoveVelocity is in error.
MC_MoveRelative	4	The internal instance of MC_MoveRelative is in error.
MC_MoveAbsolute	5	The internal instance of MC_MoveAbsolute is in error.
MC_Halt	6	The internal instance of MC_Halt is in error.
MC_Stop	7	The internal instance of MC_Stop is in error.
MC_Homing	8	The internal instance of MC_Homing is in error.
MC_GearIn	9	The internal instance of MC_GearIn is in error.
MC_ReadActualVelocity	10	The internal instance of MC_ReadActualVelocity is in error.
MC_ReadActualPosition	11	The internal instance of MC_ReadActualPosition is in error.
MC_Reset	12	The internal instance of MC_Reset is in error.

MC_TouchProbe	13	The internal instance of MC_TouchProbe is in error.
SMC_SetSoftwareLimits	14	The internal instance of MC_SetSoftwareLimits is in error.
MC_SetPosition	15	The internal instance of MC_SetPosition is in error.
MC_MoveSuperimposed	16	The internal instance of MC_MoveSuperimposed is in error.
MC_HaltSuperImposed	17	The internal instance of MC_HaltSuperimposed is in error.
MC_StepAbsoluteSwitchDetection	18	The internal instance of MC_StepAbsoluteSwitchDetection is in error.
MC_StepAbsoluteSwitch	19	The internal instance of MC_StepAbsoluteSwitch is in error.
MC_StepReferencePulse	20	The internal instance of MC_StepReferencePulse is in error.
MC_HomeDirect	21	The internal instance of MC_HomeDirect is in error.
MC_MoveModulo	22	The internal instance of MC_MoveModulo is in error.

5.11 MC_CT_FB_FilterTypes

MC_CT_FilterTypes is an enumeration for the possible filter types specified at the “FilterType” input of MC_CT_AxisFilter.

Enumeration	Value	Description
NoFilter	0	The no filter is used.
WindowingAverage	1	The windowing average filter will be used.
WeightedAverage	2	The weighted average filter will be use.

5.12 MC_CT_FreezeCaptureSource

MC_CT_FreezeCaptureSource is an enumeration for the possible freeze sources specified at the “CFG_FreezeSource” input of MC_CT_EGB_Axis and MC_CT_PositionAxis.

Enumeration	Value	Description
DigitalInput4	0	Drive digital input 4 will be used as the freeze source.
DigitalInput5	1	Drive digital input 5 will be used as the freeze source.
P1Marker	2	Drive P1 marker pulse will be used as the freeze source.
P2Marker	3	Drive P2 marker pulse will be used as the freeze source.
CommonFreezeBus	4	The drives common freeze bus will be used as the freeze source.
P1Zero	5	Drive P1 zero pulse will be used as the freeze source.
P2Zero	6	Drive P2 zero pulse will be use as the freeze source.

5.13 MC_CT_HomingModes

MC_CT_HomingModes is an enumeration for the possible homing modes specified at the "CFG_HomeMode" input of MC_CT_EGB_Axis and MC_CT_PositionAxis.

Enumeration	Value	Description
HomeAtCurrentPosition	0	The axis remains stationary, and the position counter is reset to the value present at the CFG_HomeSearchPosition input. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.
HomeToFbHomingSwitch	1	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the CMD_HomeSwitch signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. The home position is set once the CMD_HomeSwitch signal is no longer detected where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.

HomeToDriveHomingSwitch	2	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the drive digital input 4 signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. The home position is set once the drive digital input 4 signal is no longer detected where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.
HomeToFbHomingSwitch ThenIndexMarkerPulse	3	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the CMD_HomeSwitch signal has been detected, the axis continues to move in the same direction, but the velocity will change to that specified at the CFG_HomeSearchBackoffSpeed input. Once the CMD_HomeSwitch signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.

HomeToFbHomingSwitch ThenReverseToIndexMarkerPulse	4	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the CMD_HomeSwitch signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. Once the CMD_HomeSwitch signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.
HomeToDriveHomingSwitch ThenIndexMarkerPulse	5	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the drive digital input 4 signal has been detected, the axis continues to move in the same direction, but the velocity will change to that specified at the CFG_HomeSearchBackoffSpeed input. Once the drive digital input 4 signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.

HomeToDriveHomingSwitch ThenReverseIndexMarkerPulse	6	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. Once the drive digital input 4 signal has been detected, a direction reversal is performed and the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed input. Once the drive digital input 4 signal is no longer detected, homing completes on the next encoder index marker pulse where the axis is brought to a complete stop. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs. The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.
HomeToIndexFreeze	7	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. The axis uses the Drive F1 freeze input (Digital input 4 by default) freeze mechanism as the home switch input, to complete a 1 stage (search only) homing routine. Once the freeze signal has been detected a direction reversal is performed where the axis moves to the absolute position specified at the CFG_HomeOffsetPosition input (0 by default), where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input. The freeze source can be modified via the CFG_FreezeSource function block input. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs

HomeToTorqueThreshold	8	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. The axis uses a Percentage Load >= CFG_HomeTorqueThreshold condition to satisfy the homing switch input. Once this homing switch condition has been detected, the axis will perform a reversal where the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed to perform a 2 stage (Search and Backoff) homing routine. Once the Percentage Load >= CFG_HomeTorqueThreshold condition is no longer detected, the axis is brought to a complete stop, and the home position is set. The rates for this homing mode are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.
HomeToTorqueThresholdThenReverseIndexMarkerPulse	9	The axis will perform an initial movement in the direction set by CFG_HomeDirection at the velocity specified by CFG_HomeSearchSpeed. The axis uses a Percentage Load >= CFG_HomeTorqueThreshold condition to satisfy the homing switch input. Once this homing switch condition has been detected, the axis will perform a reversal where the velocity changes to that specified at the CFG_HomeSearchBackoffSpeed to perform a 2 stage (Search and Backoff) homing routine. Once the Percentage Load >= CFG_HomeTorqueThreshold condition is no longer detected, the home position is set at the rising edge of the next index marker pulse where the axis is then is brought to a complete stop. are defined at the CFG_HomeAccel, CFG_HomeDecel and CFG_HomeJerk inputs The axis may be moved to an absolute offset position after homing has completed. This absolute offset position can be specified via the CFG_HomeOffsetPosition input, where the maximum speed for this movement can be specified via the CFG_HomeOffsetMaxSpeed input.

5.14 MC_CT_NC_Error

MC_CT_NC_Error is a large enumeration that collates NC axis errors from multiple categories into one enumeration.

The text strings have been created to aid in error diagnoses – the default enumerations in the NC library contain the hexadecimal error code with no plain text description. The MC_CT_NC_Error text strings contain the error description followed by the hexadecimal error code, e.g. "Internal_Error_4000" is displayed instead of "4000".

More information on TwinCAT NC error codes can be found by using the following link and navigating to "TwinCAT 2 > TX12xx | TwinCAT NC > NC Errorcodes > Overview of NC Errors":

Beckhoff Information System - English

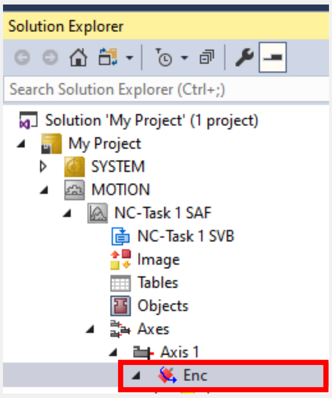
5.15 MC_CT_PositionMotionTypes

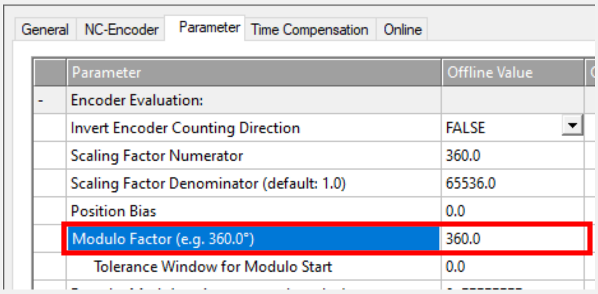
MC_CT_PositionMotionTypes is an enumeration for the possible motion types used when positioning, specified at the "CFG_PositionMotionType" input of MC_CT_EGB_Axis, MC_CT_PositionAxis and MC_CT_VirtualAxis.

Enumeration	Value	Description
Absolute	0	When the CMD_Position input is set TRUE the axis will move to the absolute position specified at the CFG_Position input.
Relative	1	On a rising edge of the CMD_Position input, the axis will move incrementally from its current position by the distance specified at the CFG_Position input, where a positive value will result in a forward movement and a negative value in a backward movement.
AutoIndex	2	When the CMD_Position input is set TRUE the axis will continually index by the distance specified at the CFG_Position input, where a positive value will result in a forward movement and a negative value in a backward movement. **HINT:** The axis will be brought to a stop as soon as the CMD_Position input is set FALSE meaning the axis may stop before the current index has completed. For indexing applications, is it recommended to use Relative mode.

AutoReciprocate	3	When the CMD_Position input is set TRUE the axis will continually reciprocate by the distance specified at the CFG_Position input, where a positive value will result in an initial forward movement and a negative value in an initial backward movement. **HINT:** The axis will be brought to a stop as soon as the CMD_Position input is set FALSE meaning the axis may stop before the current movement has completed.
Modulo	4	When the CMD_Position input is set TRUE, the axis will move to the modulo position specified at the CFG_Position input using the modulo direction specified at the CFG_PositionModuloDirection input.

HINT: The modulo rollover factor is defined in the encoder parameter area which can be reached via the project tree under "**Project Name**" > "**Motion**" > "**NC-Task SAF**" > "**Axes**" > "**Axis**" > "**Enc**" and then selecting the "**Parameter**" tab:





	Parameter	Offline Value
-	Encoder Evaluation:	
	Invert Encoder Counting Direction	FALSE
	Scaling Factor Numerator	360.0
	Scaling Factor Denominator (default: 1.0)	65536.0
	Position Bias	0.0
	Modulo Factor (e.g. 360.0°)	360.0
	Tolerance Window for Modulo Start	0.0

5.16 MC_CT_SDO_ErrorList

MC_CT_SDO_ErrorList is a enumeration that collates FoE (file over EtherCAT) mailbox errors codes, ADS error codes and CoE mailbox error codes onto one enumeration.

Enumeration	Value	Description
NoError	0	*FoE mailbox protocol error codes*
NotFound	1	
Access	2	
DiskFull	3	
Illegal	4	
PackENO	5	
Exists	6	
NoUser	7	
BootStrapOnly	8	
NotInBootStrap	9	
InvalidPassword	10	
UnknwonAmsCommand	11	*ADS Errors*
Win32Error	12	
PortNotConnected	13	
InvalidAmsLength	14	
InvalidAmsNetID	15	
LowInstallLevel	16	
NoDebug	17	
PortDisabled	18	
PortConnected	19	
AmsSyncWin32Error	20	
SyncTimeOut	21	

AmsSyncAmsError	22	
AmsSyncNoIndexMap	23	
InvalidAmsPort	24	
NoMemory	25	
TCPSendError	26	
HostUnreachable	27	
NoLockedMemory	1280	
MailboxFull	1282	
DeviceError	1792	
DeviceServiceNotSupported	1793	
DeviceInvalidGroup	1794	
DeviceInvalidIndexOrParameter	1795	
DeviceInvalidAccess	1796	
DeviceInvalidSize	1797	
DeviceInvalidData	1798	
DeviceNotReady	1799	
DeviceBusy	1800	
DeviceInvalidContext	1801	
DeviceNoMemory	1802	
DeviceInavlidParam	1803	
DeviceNotFound	1804	
DeviceSyntaxError	1805	
DeviceIncompatible	1806	
DeviceSymbolNotFound	1808	

DeviceSymbolVersionInvalid	1809	
DeviceInvalidState	1810	
DeviceTransModeNotSupported	1811	
DeviceNotifyHandleInvalid	1812	
DeviceClientUnknown	1813	
DeviceNoMoreHandles	1814	
DeviceInvalidWatchsize	1815	
DeviceNotInitialized	1816	
DeviceTimeOut	1817	
DeviceNoInterface	1818	
DeviceInvalidInterface	1819	
DeviceInvalidCLSID	1820	
DeviceInvalidObjectID	1821	
ClientError	1856	
ClientInvalidParameter	1857	
ClientListEmpty	1858	
ClientVaraiableInUse	1859	
ClientDuplicateInvokeID	1860	
ClientSyncTimeOut	1861	
ClientW32OR	1862	
ClientTimeoutInvalid	1863	
ClientPortNotOpen	1864	
ClientNoAmsAddr	1865	
ClientSyncInternal	1872	

ClientAddHash	1873	
ClientRemoveHash	1874	
ClientNoMoreSymbols	1875	
ClientSyncResInvalid	1876	
ClientSyncPortLocked	1877	
ClientQueueFull	32768	
Toggle	16#05030000	*CoE mailbox protocol error codes* Toggle bit not alternated.
Timeout	16#05040000	SDO protocol timed out.
CCS_SCS	16#05040001	Client/server command specifier not valid or unknown.
InvalidBlockSize	16#05040002	Invalid block size (block mode only).
InvalidSequenceNumber	16#05040003	Invalid sequence number (block mode only).
CRC_Error	16#05040004	CRC error (block mode only).
OutOfMemory	16#05040005	Out of memory.
UnsupportedAccess	16#06010000	Unsupported access to an object.
WriteOnly	16#06010001	Attempt to read a write only object.
ReadOnly	16#06010002	Attempt to write a read only object.
IndexDoesNotExist	16#06020000	Object does not exist in the object dictionary.
PDO_Map	16#06040041	Object cannot be mapped to the PDO.
PDO_Length	16#06040042	The number and length of the objects to be mapped would exceed PDO length.
P_Incompatibility	16#06040043	General parameter incompatibility reason.
GeneralIncompatibility	16#06040047	General internal incompatibility in the device.
HardwareError	16#06060000	Access failed due to an hardware error.
DataSize	16#06070010	Data type does not match, length of service parameter does not match.
DataSizeTooHigh	16#06070012	Data type does not match, length of service parameter too high.

DataSizeTooLow	16#06070013	Data type does not match, length of service parameter too low.
SubIndexDoesNotExist	16#06090011	Sub-index does not exist.
WriteRangeExceeded	16#06090030	Value range of parameter exceeded (only for write access).
ValueWrittenIsTooHigh	16#06090031	Value of parameter written too high.
ValueWrittenIsTooLow	16#06090032	Value of parameter written too low.
MaxLessThanMin	16#06090036	Maximum value is less than minimum value.
GeneralError	16#08000000	General error.
CantTransferData	16#08000020	Data cannot be transferred or stored to the application.
CantTransferDataLocalControl	16#08000021	Data cannot be transferred or stored to the application because of local control.
CantTransferDataDeviceState	16#08000022	Data cannot be transferred or stored to the application because of the present device state.
ObjectDictionaryError	16#08000023	Object dictionary dynamic generation fails, or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error).

5.17 MC_CT_StopOnFollowingErrorModes

MC_CT_StopOnFollowingErrorModes is an enumeration for the possible stop on following error modes, specified at the “CFG_StopOnFollowingErrorModes” input of MC_CT_EGB_Axis, MC_CT_PositionAxis and MC_CT_SpeedAxis.

Enumeration	Value	Description
Disabled	0	The stop on following error feature is disabled.
PositionFollowingError	1	Stop if the position following error is greater than CFG_PositionFollowingErrorLimit for the duration specified at CFG_StopOnFollowingErrorTime.
SpeedFollowingError	2	Stop if the speed following error is greater than CFG_SpeedFollowingErrorLimit for the duration specified at CFG_StopOnFollowingErrorTime.
PositionAndSpeedFollowingError	3	Stop if the position following error is greater than CFG_PositionFollowingErrorLimit or if the speed following error is greater than CFG_SpeedFollowingErrorLimit for the duration specified at CFG_StopOnFollowingErrorTime.

5.18 MC_CT_SwitchLogic

MC_CT_SwitchLogic is an enumeration for the possible home switch logic modes, specified at the “CFG_HomeSwitchLogic” input of MC_CT_EGB_Axis and MC_CT_PositionAxis.

Enumeration	Value	Description
PositiveLogic	0	When the homing switch is activated by the axis, the homing switch input must be set to TRUE.
NegativeLogic	1	When the homing switch is activated by the axis, the homing switch input must be set to FALSE.

5.19 MC_CT_SyncMode

Enumeration	Value	Description
InstantLock	0	The master and slave should start from 0 speed, and then the axes are gear locked together. The Velocity, Acceleration, Deceleration, and Jerk inputs have no infulence.
MatchSpeed	1	The slave axis and the master axes move with synchronised velocity. Changes in the gear ratios during the journey are immediately calculated internally. The Velocity, Acceleration, Deceleration, and Jerk inputs only limit the slave response dynamics before InGear = TRUE. After GearIn = TRUE the Acceleration, Deceleration, and Jerk inputs are ignored and replaced by the axis Maximum Dynamics allowing much larger rates to be used while locked, where Jerk is set to the greater of the maximum acceleration or maximum deceleration value.
MatchSpeedProfile	2	The slave axis and the master axes move with a synchronised velocity. In the case of a dynamic limitation, the slave establishes a position difference that is not caught up again. The Velocity, Acceleration, Deceleration, and Jerk inputs limit the slave response dynamics, even when locked.
MatchPosition	3	The slave axis and the master axes move with synchronised velocity and position. In the case of a dynamic limitation, the slave establishes a position difference that is then caught up at a later time before synchronisation. Changes in the gear ratios during the journey are immediately calculated internally and the slave continuously synchronises itself to the new position. The Velocity, Acceleration, Deceleration, and Jerk inputs only limit the slave response dynamics before InGear = TRUE. After InGear = TRUE the Acceleration, Deceleration, and Jerk inputs are ignored and replaced by the axis Maximum Dynamics allowing much larger rates to be used while locked, where Jerk is set to the greater of the maximum acceleration or maximum deceleration value.
MatchPositionProfile	4	The slave axis and the master axes move with synchronised velocity and position. In the case of a dynamic limitation, the slave establishes a position difference that is then caught up at a later time using the profile constraints even after InGear = TRUE. Changes in the gear ratios during the journey are immediately calculated internally and the slave continuously synchronises itself to the new position. The Velocity, Acceleration, Deceleration, and Jerk inputs limit the slave response dynamics, even when locked.

5.20 MC_CT_TripStrings

MC_CT_TripStrings is an enumeration of possible drive alarms generated at the “STS_DriveLastTrip” output of the MC_CT_EGB_Axis, MC_CT_PositionAxis and MC_CT_SpeedAxis function blocks.

Enumeration	Value	Description
None	0	No trip.
Reserved001	1	Reserved.
OverVolts	2	DC bus voltage has exceeded the peak level or maximum continuous level for a period of time.
OIac	3	Instantaneous output over current detected.
OI_Brake	4	Braking IGBT over current detected or short circuit protection for the braking IGBT activated.
PSU	5	Internal power supply fault.
ExternalTrip	6	External Trip has occurred.
OverSpeed	7	Motor over speed has occurred.
Inductance	8	Inductance measurement out of range or motor saturation not detected.
PSU_24V	9	24V internal power supply overload.
ThBrakeRes	10	Brake resistor over temperature.
Autotune1	11	Position feedback did not change or required speed could not be reached during an autotune.
Autotune2	12	Position feedback direction incorrect during an autotune.
Autotune3	13	Measured inertia has exceeded the parameter range or commutation signals changed in wrong direction during an autotune.
Autotune4	14	Drive encoder U commutation signal fail during an autotune.
Autotune5	15	Drive encoder V commutation signal fail during an autotune.
Autotune6	16	Drive encoder W commutation signal fail during an autotune.
Autotune7	17	Motor number of poles / position feedback resolution set incorrectly.
AutotuneStopped	18	Autotune test was stopped before completion.
BrakeR_TooHot	19	Braking resistor thermal protection overload timed out.

MotorTooHot	20	Motor thermal protection overload timed out.
OHtInverter	21	Inverter over temperature based on thermal model.
OHtPower	22	Power stage over temperature.
OHtControl	23	Control stage over temperature.
Thermistor	24	Motor thermistor over temperature.
ThShortCircuit	25	Motor thermistor short circuit.
IO_Overload	26	Digital output overload.
OHt_dc_bus	27	DC bus over temperature based on thermal model.
AnInput1Loss	28	Analog input 1 current loss.
AnInput2Loss	29	Analog input 2 current loss.
Watchdog	30	Control word watchdog has timed out.
EEPROM_Fail	31	Non-volatile memory issue detected causing default parameter values to be loaded.
PhaseLoss	32	Phase loss detected at the supply input to the drive.
Resistance	33	Stator resistance measurement failed or value exceeds the allowed range.
KeypadMode	34	Keypad has been removed when the drive is in keypad mode.
ControlWord	35	Trip initiated from the Control Word.
UserSave	36	Error detected in the user save parameter data saved in non-volatile memory.
PowerDownSave	37	Error detected in the power down save parameter data saved in non-volatile memory.
LowLoad	38	The load on the drive has fallen below the Low Load Detection Level.
LineSync	39	Synchronization to the power supply has been lost.
UserTrip40	40	User generated trip.
UserTrip41	41	User generated trip.
UserTrip42	42	User generated trip.
UserTrip43	43	User generated trip.

UserTrip44	44	User generated trip.
UserTrip45	45	User generated trip.
UserTrip46	46	User generated trip.
UserTrip47	47	User generated trip.
UserTrip48	48	User generated trip.
UserTrip49	49	User generated trip.
UserTrip50	50	User generated trip.
UserTrip51	51	User generated trip.
UserTrip52	52	User generated trip.
UserTrip53	53	User generated trip.
UserTrip54	54	User generated trip.
UserTrip55	55	User generated trip.
UserTrip56	56	User generated trip.
UserTrip57	57	User generated trip.
UserTrip58	58	User generated trip.
UserTrip59	59	User generated trip.
UserTrip60	60	User generated trip.
UserTrip61	61	User generated trip.
UserTrip62	62	User generated trip.
UserTrip63	63	User generated trip.
UserTrip64	64	User generated trip.
UserTrip65	65	User generated trip.
UserTrip66	66	User generated trip.
UserTrip67	67	User generated trip.

UserTrip68	68	User generated trip.
UserTrip69	69	User generated trip.
UserTrip70	70	User generated trip.
UserTrip71	71	User generated trip.
UserTrip72	72	User generated trip.
UserTrip73	73	User generated trip.
UserTrip74	74	User generated trip.
UserTrip75	75	User generated trip.
UserTrip76	76	User generated trip.
UserTrip77	77	User generated trip.
UserTrip78	78	User generated trip.
UserTrip79	79	User generated trip.
UserTrip80	80	User generated trip.
UserTrip81	81	User generated trip.
UserTrip82	82	User generated trip.
UserTrip83	83	User generated trip.
UserTrip84	84	User generated trip.
UserTrip85	85	User generated trip.
UserTrip86	86	User generated trip.
UserTrip87	87	User generated trip.
UserTrip88	88	User generated trip.
UserTrip89	89	User generated trip.
PowerComms	90	Communications problem within the power system of the drive.
User24V	91	User 24V supply not present on control terminals when required.

OI_Snubber	92	Over-current condition detected in the rectifier snubber circuit.
InductorTooHot	93	Regen inductor thermal protection overload timed out.
RectifierSetUp	94	A rectifier has not been set-up correctly in a multi-power module system.
ActiveRectifier	95	The active input rectifier has tripped.
UserProgTrip	96	Trip generated by an onboard user program.
DataChanging	97	Drive parameter data is changing and the drive has been commanded to enable.
OutPhaseLoss	98	Phase loss been detected at the output of the drive.
CAM	99	Advanced motion controller cam failure.
Reset	100	This is not a valid trip number as this value is used in User Trip (10.038) to reset the drive.
OHtBrake	101	Braking IGBT over-temperature.
Cloning	102	Failure of the cloning function.
InterConnect	103	Multi-module drive interconnection cable error.
Reserved104	104	Phase imbalance has been detected (Regen mode only).
Reserved105	105	Reserved.
Reserved106	106	Reserved.
Reserved107	107	Reserved.
Reserved108	108	Reserved.
OIdc	109	Drive output stage short circuit detected.
Undefined	110	Power system indicated a trip but did not identify the reason.
Configuration	111	Number of power modules or rectifiers installed is different from the number expected.
UserTrip112	112	User generated trip.
UserTrip113	113	User generated trip.
UserTrip114	114	User generated trip.
UserTrip115	115	User generated trip.

UserTrip116	116	User generated trip.
UserTrip117	117	User generated trip.
UserTrip118	118	User generated trip.
UserTrip119	119	User generated trip.
UserTrip120	120	User generated trip.
UserTrip121	121	User generated trip.
UserTrip122	122	User generated trip.
UserTrip123	123	User generated trip.
UserTrip124	124	User generated trip.
UserTrip125	125	User generated trip.
UserTrip126	126	User generated trip.
UserTrip127	127	User generated trip.
UserTrip128	128	User generated trip.
UserTrip129	129	User generated trip.
UserTrip130	130	User generated trip.
UserTrip131	131	User generated trip.
UserTrip132	132	User generated trip.
UserTrip133	133	User generated trip.
UserTrip134	134	User generated trip.
UserTrip135	135	User generated trip.
UserTrip136	136	User generated trip.
UserTrip137	137	User generated trip.
UserTrip138	138	User generated trip.
UserTrip139	139	User generated trip.

UserTrip140	140	User generated trip.
UserTrip141	141	User generated trip.
UserTrip142	142	User generated trip.
UserTrip143	143	User generated trip.
UserTrip144	144	User generated trip.
UserTrip145	145	User generated trip.
UserTrip146	146	User generated trip.
UserTrip147	147	User generated trip.
UserTrip148	148	User generated trip.
UserTrip149	149	User generated trip.
UserTrip150	150	User generated trip.
UserTrip151	151	User generated trip.
UserTrip152	152	User generated trip.
UserTrip153	153	User generated trip.
UserTrip154	154	User generated trip.
UserTrip155	155	User generated trip.
UserTrip156	156	User generated trip.
UserTrip157	157	User generated trip.
UserTrip158	158	User generated trip.
UserTrip159	159	User generated trip.
Island	160	Island supply condition detected in regen mode.
Reserved161	161	Selected device type is not supported by the drive.
Encoder12	162	Encoder could not be identified during auto-configuration.
Encoder13	163	Data read from the encoder is out of range during auto-configuration.

Encoder14	164	Data in the additional configuration parameter is out of range.
Reserved165	165	Reserved.
Reserved166	166	Reserved.
Reserved167	167	Reserved.
Reserved168	168	Reserved.
VoltageRange	169	Supply voltage is outside of the defined range in Regen mode.
Reserved170	170	Reserved.
Reserved171	171	Reserved.
Reserved172	172	Reserved.
Reserved173	173	Reserved.
CardSlot	174	Communications issue between the drive and option module during an NV media card action.
CardProduct	175	NV media card is from another product and is not compatible.
NamePlate	176	Electronic nameplate transfer has failed.
CardUserProg	177	Onboard user program is write-only or does not exist, and cannot be transferred to an NV media card.
CardBusy	178	NV media card cannot be accessed as it is being accessed by an option module.
CardDataExists	179	NV media card file already contains data.
CardOption	180	Option modules installed are different between source drive data on the NV media card and destination drive.
CardReadOnly	181	NV media card is read-only.
CardError	182	Failure to access the NV media card.
CardNoData	183	Requested file does not exist on the NV media card.
CardFull	184	NV media card is full.
CardFileError	185	NV media card header file is incorrect.
CardRating	186	Voltage rating different between the source drive parameter file on the NV media card and the destination drive.

CardFileData	187	Drive mode, required defaults or product type in the parameter file on the NV media card is not compatible with the destination drive.
CardDerivative	188	Derivative different between the source drive parameter file on the NV media card and the destination drive.
Encoder1	189	Drive position feedback interface power supply overload.
Encoder2	190	Wire break between the drive and the position feedback device.
Encoder3	191	Incorrect position feedback phase offset angle detected while running.
Encoder4	192	Encoder communications has timed out.
Encoder5	193	Encoder communications checksum or CRC error.
Encoder6	194	Encoder has indicated an error.
Encoder7	195	Set-up parameters for the position feedback device have changed.
Encoder8	196	Encoder communications time exceeds 250us.
Encoder9	197	Selected motor control feedback source does not contain position feedback category option module.
PhasingError	198	Incorrect phase offset angle in RFC-S mode.
Destination	199	The destination of two or more functions are writing to the same parameter.
Slot1HF	200	Fault with the option module in slot 1 that means that this module cannot operate.
Slot1Watchdog	201	Watchdog service error with the option module in slot 1.
Slot1Error	202	Option module in slot 1 has indicated an error.
Slot1NotFitted	203	Option module in slot 1 has been removed.
Slot1Different	204	Option module in slot 1 has changed.
Slot2HF	205	Fault with the option module in slot 2 that means that this module cannot operate.
Slot2Watchdog	206	Watchdog service error with the option module in slot 2.
Slot2Error	207	Option module in slot 2 has indicated an error.
Slot2NotFitted	208	Option module in slot 2 has been removed.
Slot2Different	209	Option module in slot 2 has changed.

Slot3HF	210	Fault with the option module in slot 3 that means that this module cannot operate.
Slot3Watchdog	211	Watchdog service error with the option module in slot 3.
Slot3Error	212	Option module in slot 3 has indicated an error.
Slot3NotFitted	213	Option module in slot 3 has been removed.
Slot3Different	214	Option module in slot 3 has changed.
OptionDisable	215	Option module did not acknowledge during drive mode changeover.
SlotAppMenu	216	Application menu customisation conflict error.
AppMenuChanged	217	Customisation table for an application menu has changed.
TempFeedback	218	Fault with a thermistor in the drive.
AnOutputCalib	219	Analog output calibration failed.
PowerData	220	Power system configuration/rating data error.
StoredHF	221	Hardware trip occurred prior to last power down.
Reserved222	222	Reserved.
RatingMismatch	223	Voltage or current rating mismatch between power modules in a multi-module drive.
DriveSize	224	Unrecognized drive size/rating.
CurrentOffset	225	Current feedback offset error.
Reserved226	226	Reserved.
SubArrayRAM	227	RAM allocation error.
Reserved228	228	Reserved.
Reserved229	229	Reserved.
Reserved230	230	Reserved.
Reserved231	231	Reserved.
Reserved232	232	Reserved.
Reserved233	233	Reserved.

Reserved234	234	Reserved.
Reserved235	235	Reserved.
Reserved236	236	Reserved.
Reserved237	237	Reserved.
Reserved238	238	Reserved.
Reserved239	239	Reserved.
Reserved240	240	Reserved.
Reserved241	241	Reserved.
Reserved242	242	Reserved.
Reserved243	243	Reserved.
Reserved244	244	Reserved.
Reserved245	245	Reserved.
Reserved246	246	Reserved.
DerivativeID	247	Derivative identification error.
DerivativeImage	248	Derivative product image error.
UserProgram	249	Onboard user program error.
Slot4HF	250	Fault with the option module in slot 4 that means that this module cannot operate.
Slot4Watchdog	251	Watchdog service error with the option module in slot 4.
Slot4Error	252	Option module in slot 4 has indicated an error.
Slot4NotFitted	253	Option module in slot 4 has been removed.
Slot4Different	254	Option module in slot 4 has changed.
ResetLogs	255	This is not a valid trip number as this value is used in User Trip (10.038) to reset the trip logs.

5.21 MC_CT_WriteModes

MC_CT_WriteModes is an enumeration for the possible SDO write modes, specified at the “Mode” input of CT_SDO_WriteCoE2 and CT_SDO_WriteParameter2.

Enumeration	Value	Description
WriteWhenInputChanges	0	The value will be written when a change is detected at the “Value” input.
WriteContinuous	1	The value will be continuously written at the repeat rate specified at the “Update Period” input.

6 Additional Information

6.1 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.

ESI = The communications description file for a EtherCAT device.

PLC = Programmable Logic Controller.

EoE = Ethernet over EtherCAT.

ADS = Automation Device Specification.

AoE = ADS over EtherCAT.

IPC = Industrial Personal Computer.

EGB = Electronic Gear Box.

CoE = CANopen over EtherCAT.

PDO = Process Data Object.

SDO = Service Data Object.

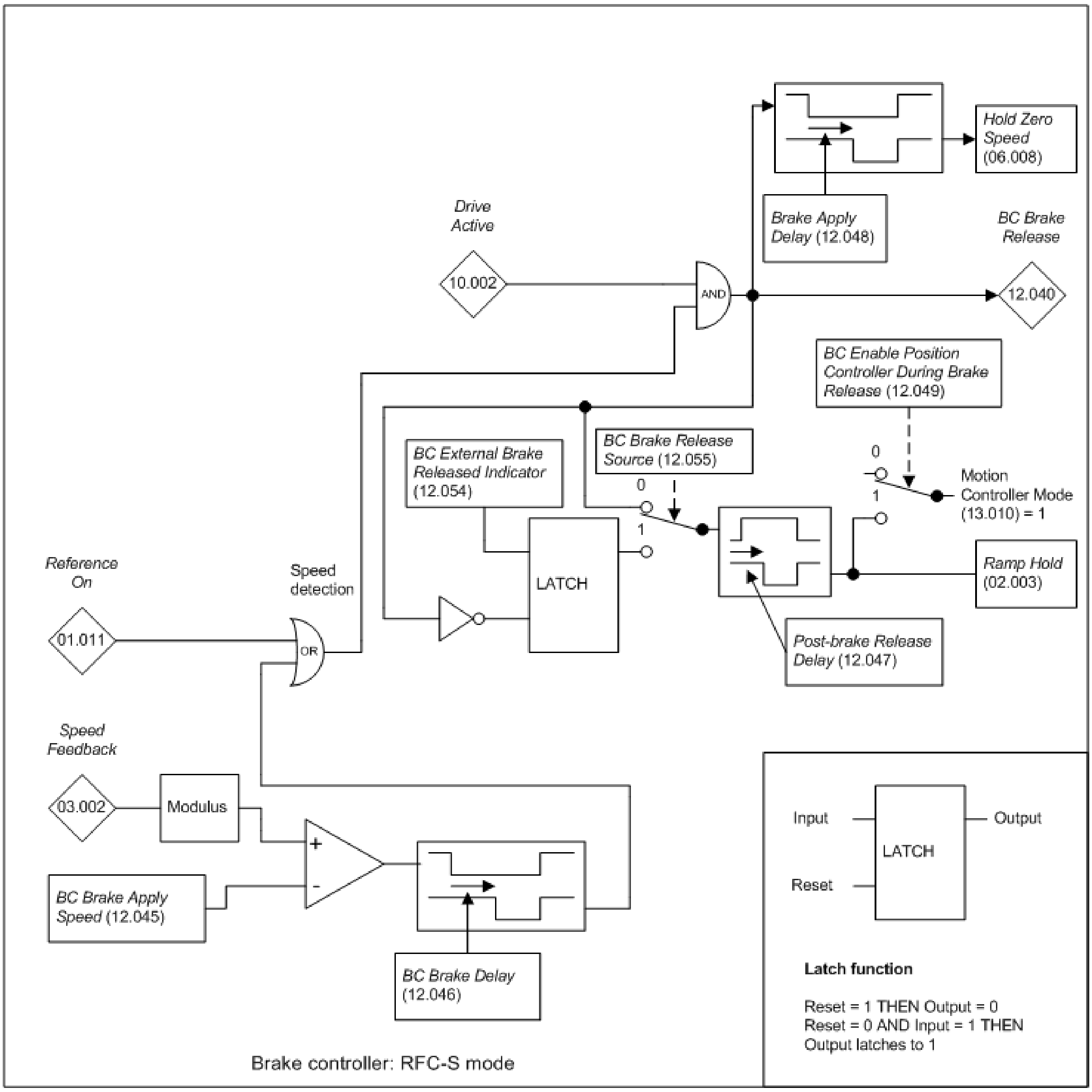
CoE = CAN over EtherCAT

Freeze Capture = This is high speed position capture technology in Control Techniques drives that provides the data for Touch Probe as defined in DS402 and PLCopen.

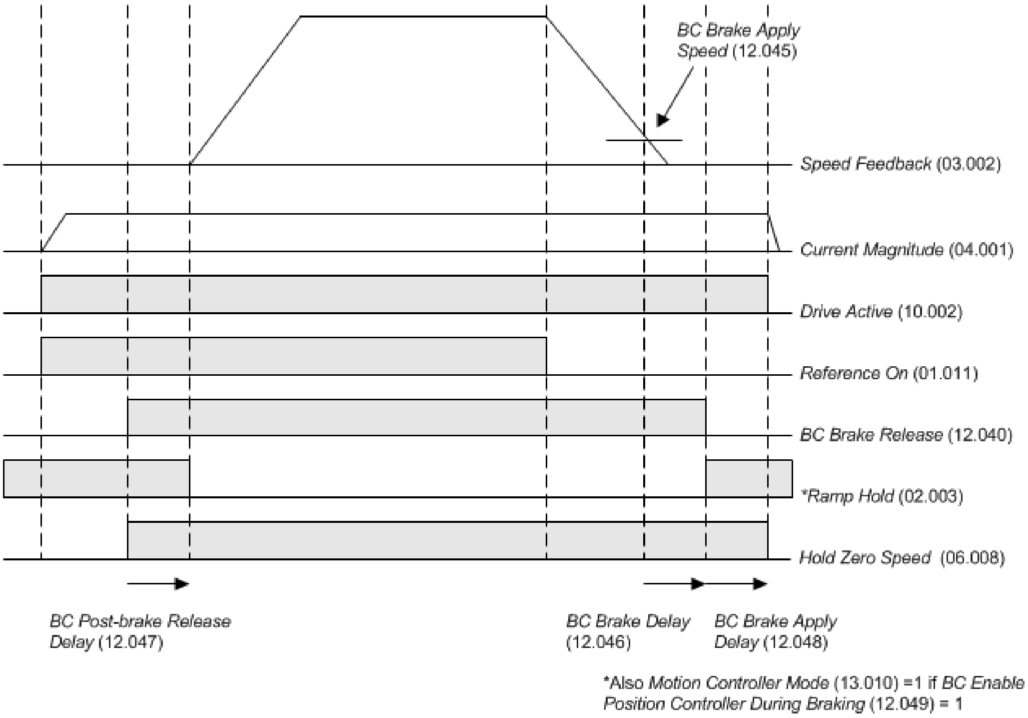
6.2 Mechanical brake controller logic

6.2.1 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.



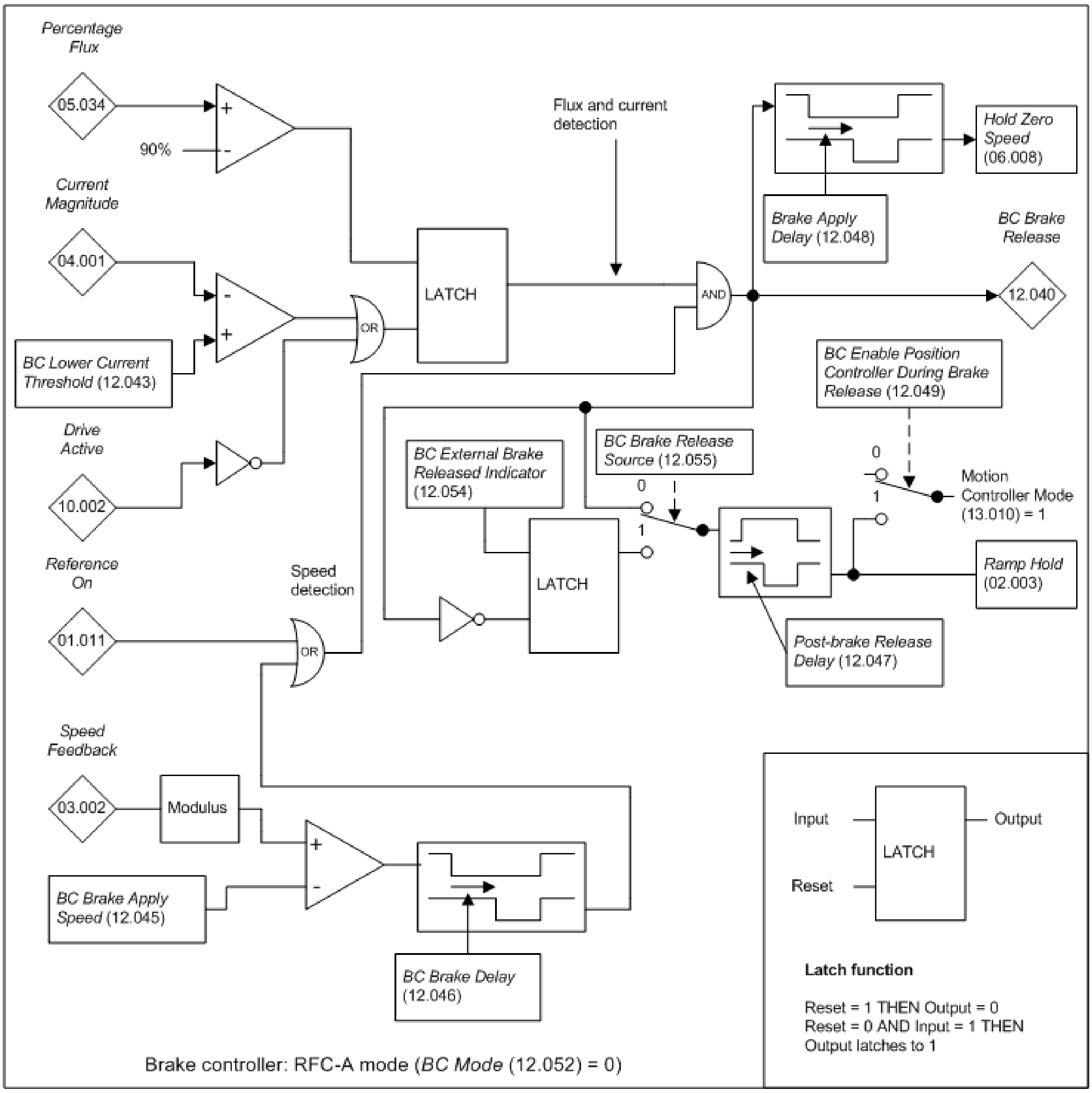
Brake controller: RFC-S mode



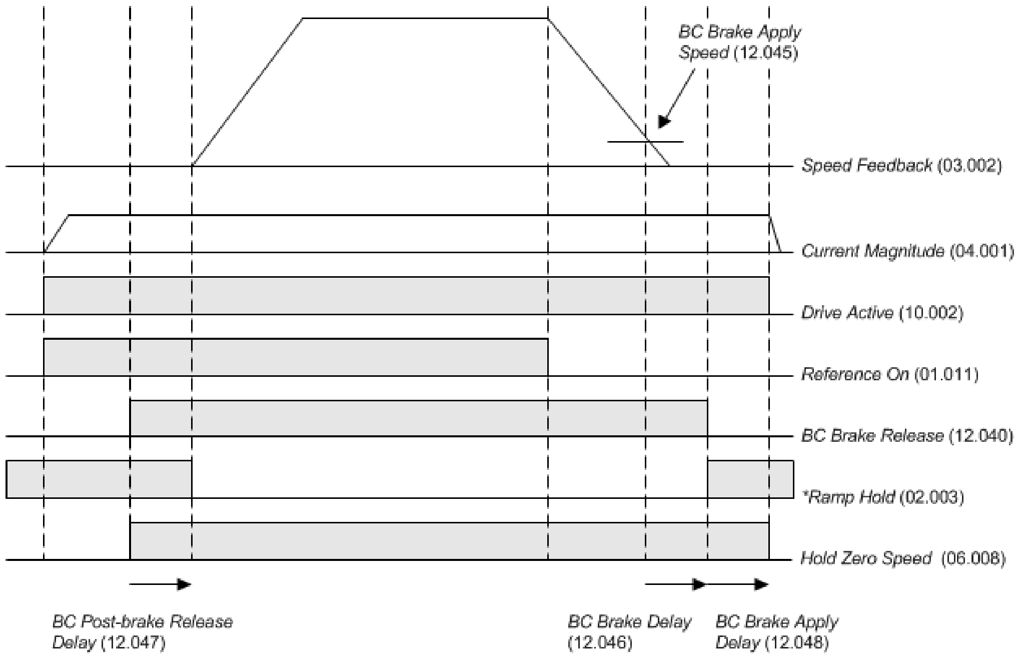
Brake controller sequence: RFC-A mode (*BC Mode* (12.052) = 0) and RFC-S mode

6.2.2 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.



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



*Also *Motion Controller Mode* (13.010) =1 if *BC Enable Position Controller During Braking* (12.049) = 1

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