



Running ServoStep™ - MIS motors with CANopen as NC axis in TwinCAT

The ServoStep™ series:

- MIS17x, MIS23x, MIS34x and MIS43x integrated stepper motors
- SMC66 and SMC85 stepper motor controllers

with **built-in CANopen** modules (e.g. MIS232S1P6H266 and SMC66B1-P6AABX1) all support the TwinCAT NC axis. The NC axis enables you to control the motor via the PLC Open Standard.

NC axis supports synchronized axes with interpolation.

This document describes how to set up the MIS motor to run as NC axis in TwinCAT3 with Visual Studio 2017, on a CX55130 PLC with EL 6751 CANopen master module.

Other variants of IDE or Hardware may vary in behavior.

This guide will help you with the following:

- Setup the motor for CANopen
- Create a new TwinCAT project
- Connect to your PLC hardware
- Scan your Hardware
- Setup the connected motor for NC axis support

Contents

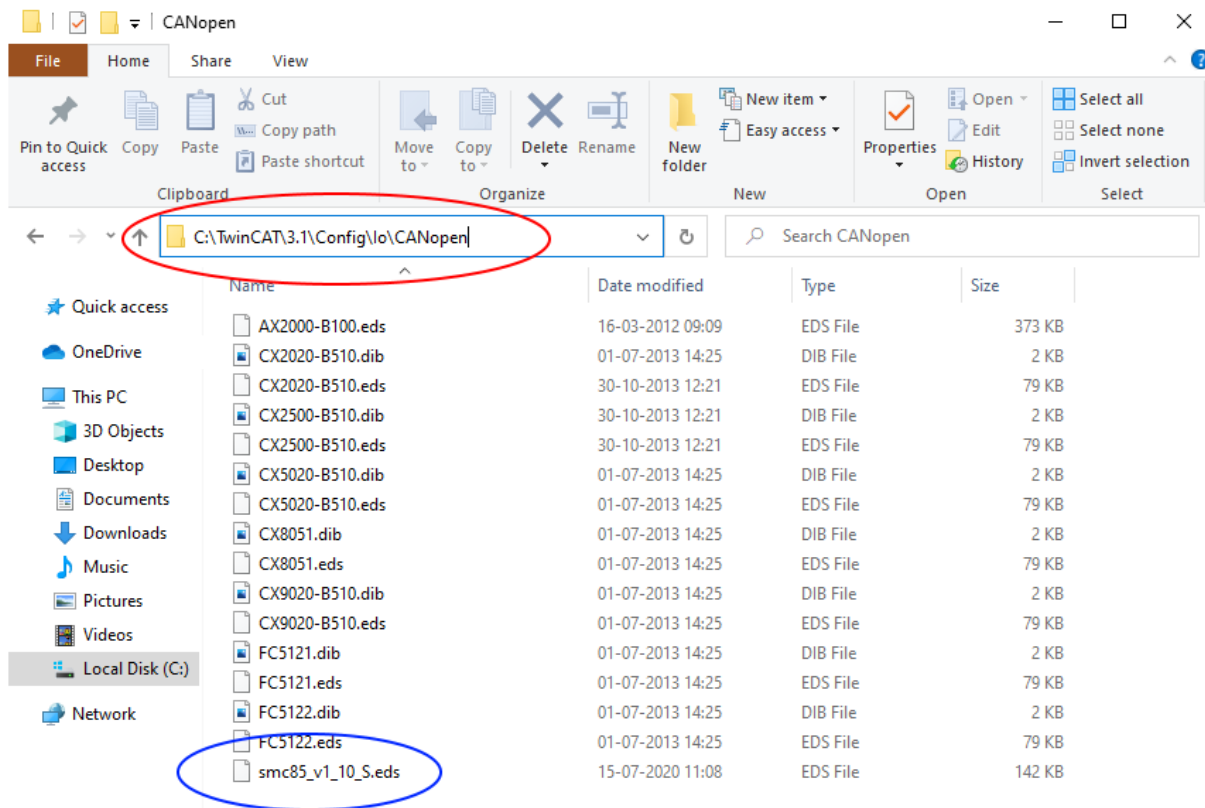
1.	Copy EDS file to TwinCAT folder.....	2
2.	Motor Setup.....	2
3.	Create a new TwinCAT project.....	3
4.	Connect to PLC in Config mode.....	3
5.	Scan for hardware.....	4
6.	Configure PDO mapping.....	8
7.	Create NC axis.....	9
8.	Map PDO data.....	11
9.	Set SDO data.....	14
10.	Sync interval.....	17
11.	Set gear factor.....	17
12.	Set motor parameters.....	18
13.	Follow Error Monitoring.....	18
14.	Store settings in PLC.....	19



1. Copy EDS file to TwinCAT folder

Download the EDS file for MIS34 from JVL homepage.

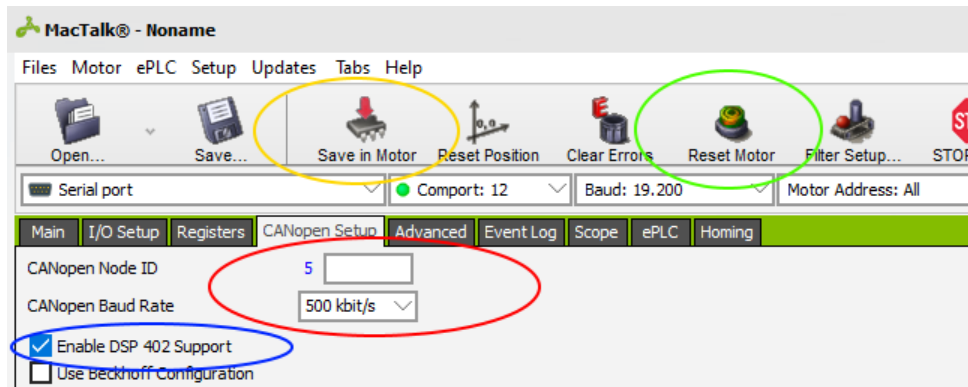
The EDS file is an Electronic Data sheet, that describes the interface between the PLC and the motor. This enables the PLC to map the right registers in the motor.



- # Open the TwinCAT folder.
- # Copy the EDS file to this location. Here an EDS file version 1.10 is used. Current version is 1.42

2. Motor Setup

Connect your ServoStep™ motor with MacTalk. Open the CANopen Setup tab.



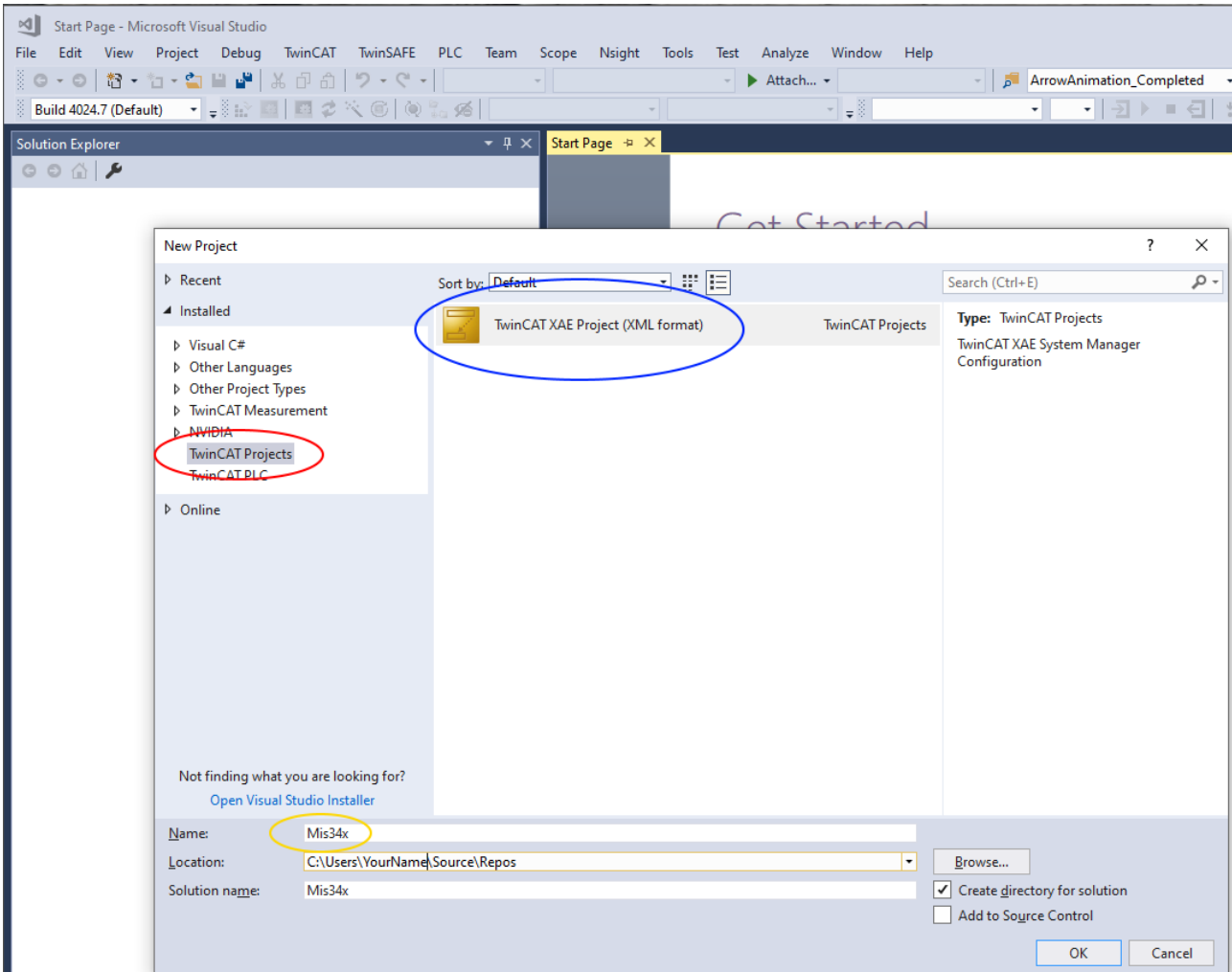
- # Choose you node ID and Baud rate. Node ID 5 and 500 kbits/sec is default.
- # Make sure Enable DSP 402 Support is checked.
- # Save the settings in the motor.
- # Reset the motor

3. Create a new TwinCAT project

Open Visual studio 2017 with TwinCAT3 integration. If TwinCAT is not installed on the computer, the TwinCAT project will not be available.

The rest of this example is done in Visual Studio.

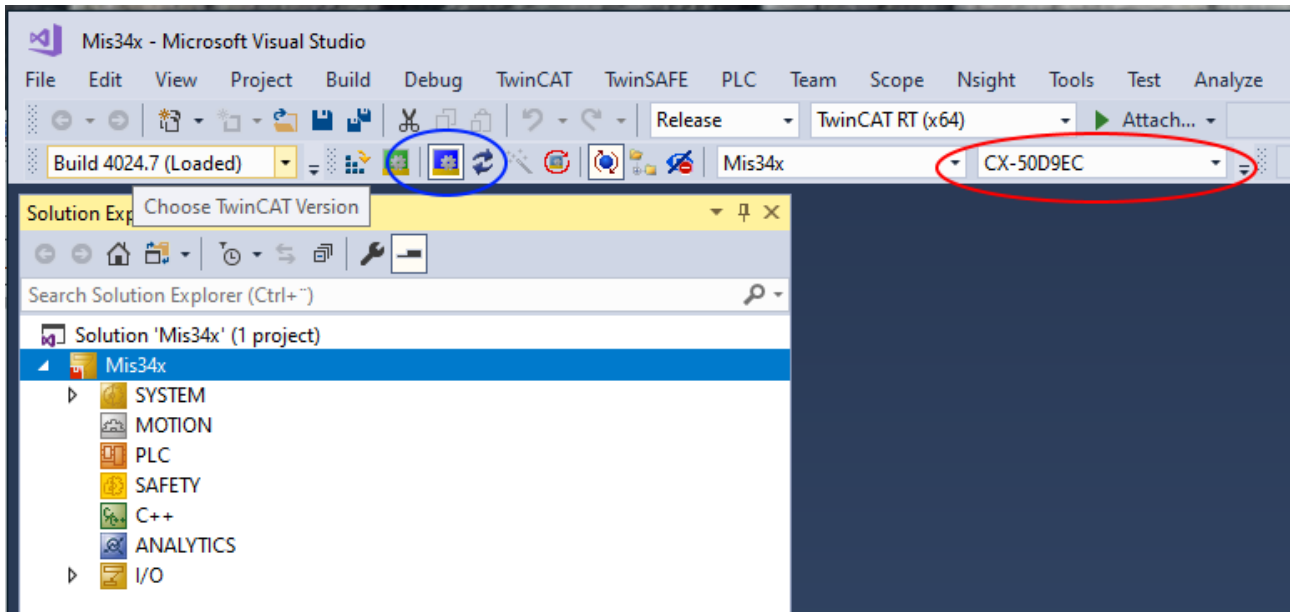
Click File -> New project.



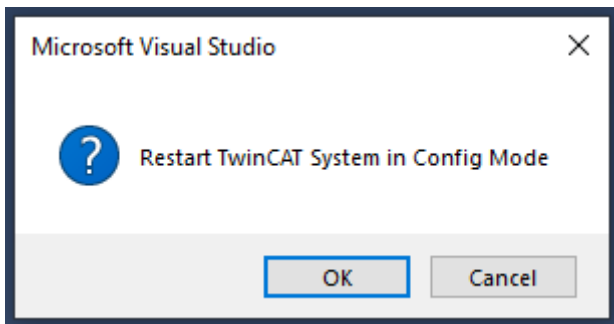
- # In the new project window, select TwinCAT project.
- # Select the TwinCAT XAE project
- # Give your project a name and click “OK” button.

4. Connect to PLC in Config mode

Connect to your PLC to scan the connected hardware. This will also detect nodes attached to the CAN bus. Make sure that all other nodes are disconnected, to get the same result as in this example.



- # Select the ADS connection for your PLC.
- # Click the Restart TwinCAT in config mode

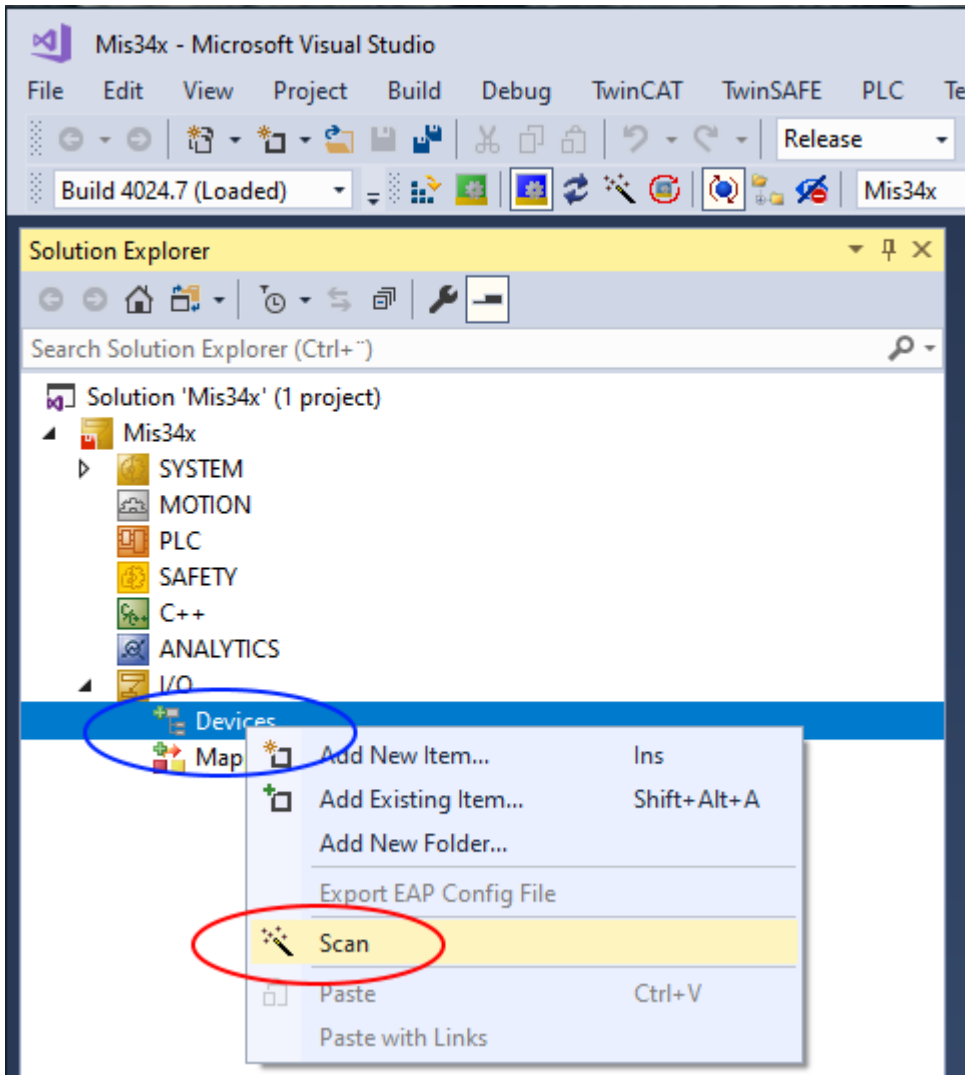


Click OK to allow TwinCAT to restart.

You should now be connected to your PLC.

5. Scan for hardware

Make sure your motor is connected to the Beckhoff CANopen master module and ready for communication with the master.

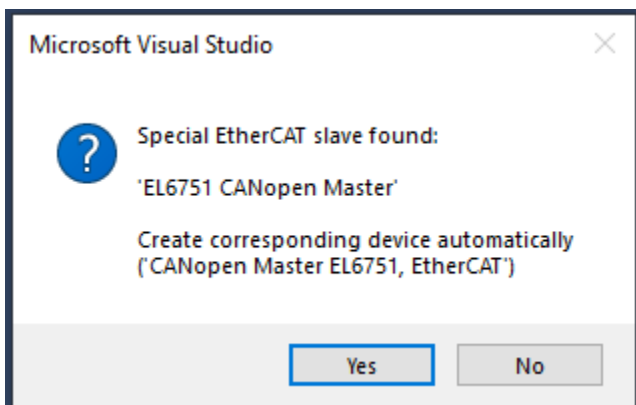


Right click IO->Devices in your solution tree.

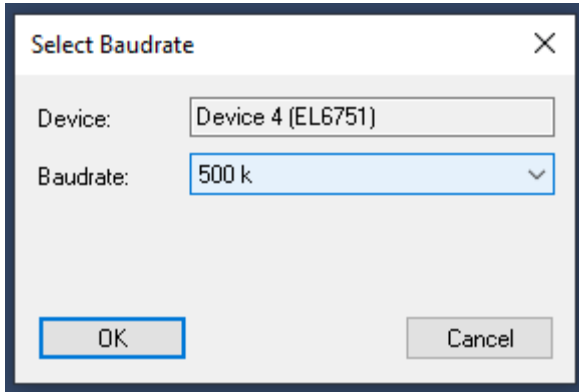
Select Scan

This will find all hardware on your PLC: IO's, busses, etc.

Sometime the Scan function does not find the CANOpen master. In this case just scan the bus where the master is connected. In the actual case the Device1 EtherCAT.



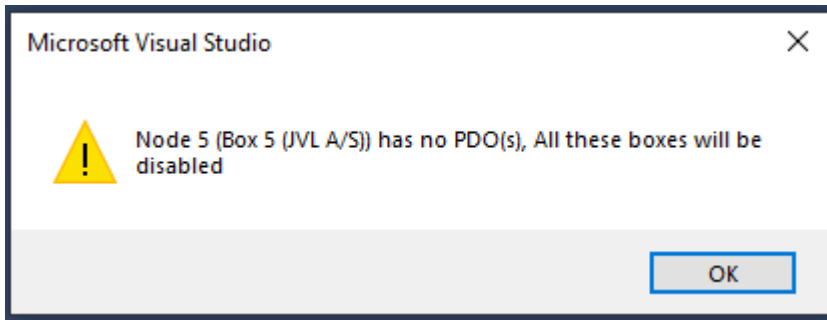
When the master is found, you will be prompted to create CANOpen devices automatically. Click "Yes". In this example the EL6751 CANOpen Master module is used.



Select the baud rate you selected in MacTalk, to scan your CANopen bus with and Click “OK”

If the scan was able to detect the PDO automatically, continue to the next heading **6** “Configure PDO mapping”.

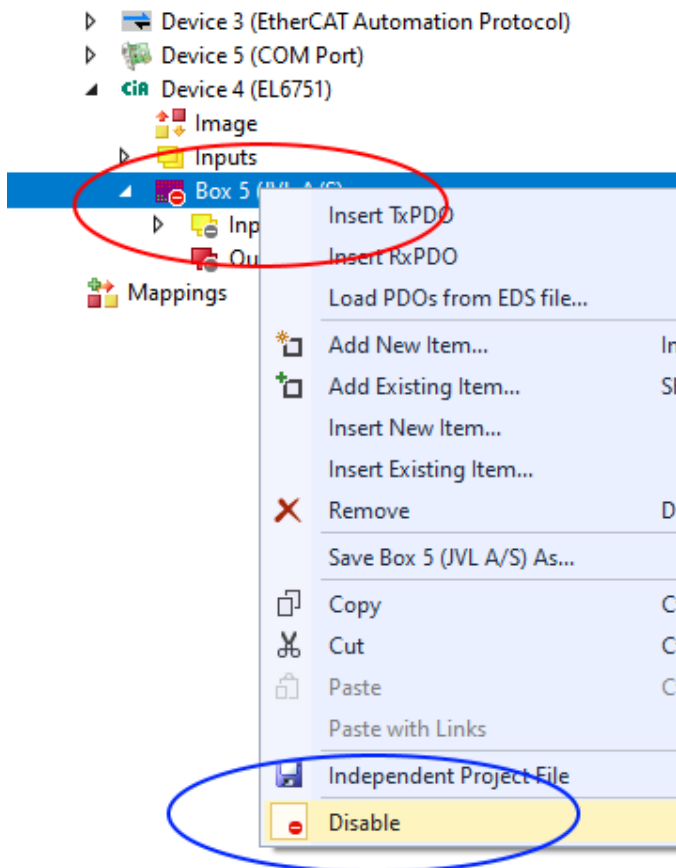
If you get this message below, the PLC did not recognize the motor as the one in the EDS file.



Continue with enabling the node and load the PDOs manually from file.

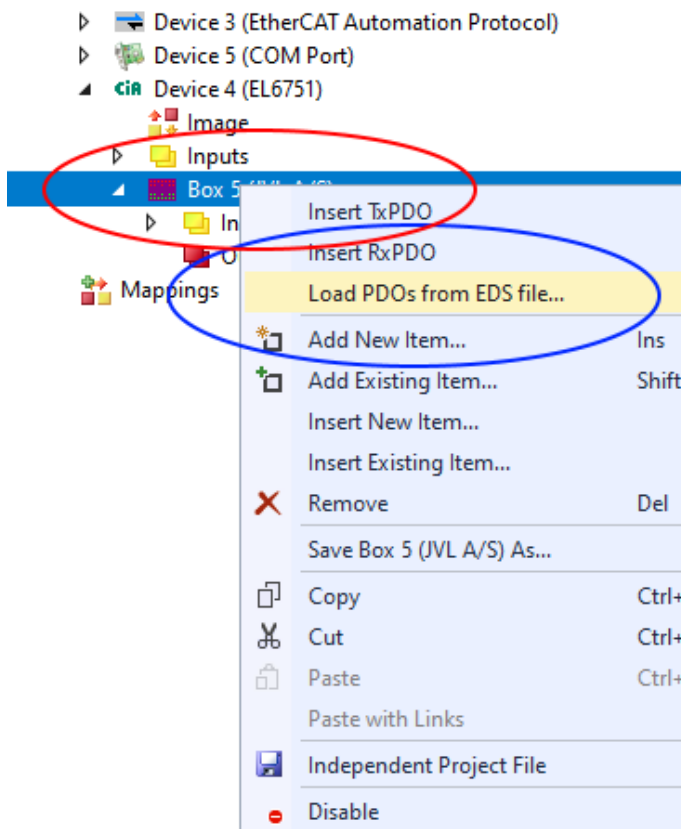


Find the Motor Box under your CANopen master device.



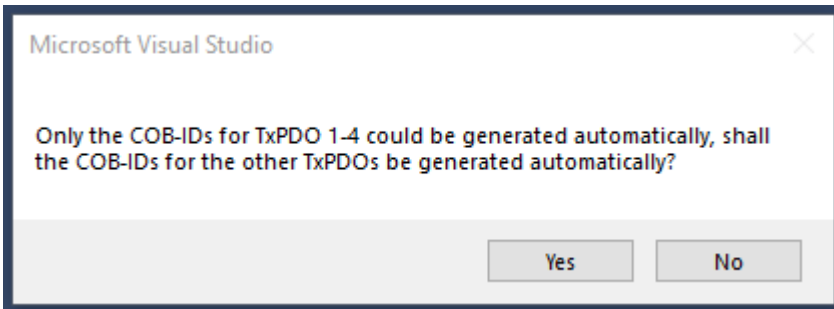
- # Right click the motor Box
- # Click Disable, to re-enable the node

Load the PDOs from EDS file.



- # Right click the motor Box.
- # Select Load PDOs from EDS file.

Select the EDS file you copied to the EDS folder earlier.



TwinCAT will generate COB-IDs for unused PDO's.

Click "Yes" for both RX and TX PDO's.

Unused PDO's are removed later in the guide but allow automatically generated COB-IDs for now.

6. Configure PDO mapping

The PDO is the data send cyclic to and from the node.

We need to send (RX PDO) Controlword and requested position to the motor, and we need to receive (TX PDO) Status word and actual position from the motor. This is easy to setup when the EDS file does most of the work.

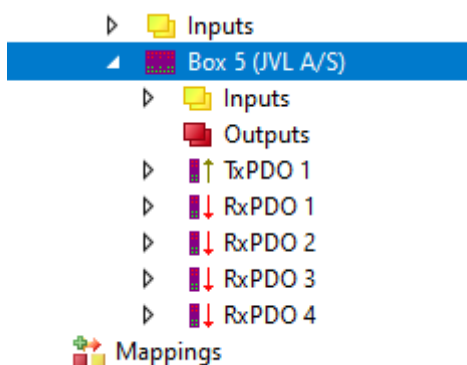
The scan can generate up to 24 RX PDOs, and 24 TX PDO's.

We only need TxPDO1 and RxPDO1 and RxPDO4

To limit the communication to what is essential for NC axis, all PDO's larger than the one used can be deleted.

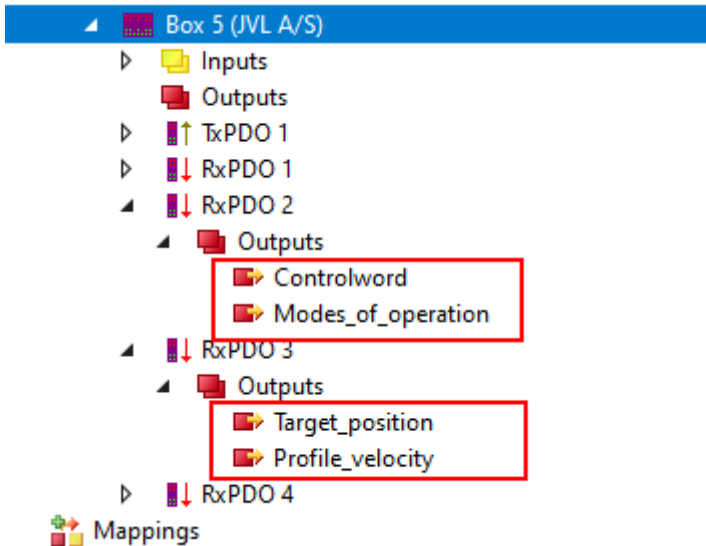
Delete TxPDO2 – TxPDO24 and RxPDO5 to RxPDO24

You should end up with something like this



We do not need RxPDO2 and RxPDO3, but we cannot delete them without destroying the mapping of RxPDO4, but we can remove the mapping inside them.

Expand RxPDO2 and RxPDO3 by clicking on the white triangle.



Delete the mapped object marked with red squares.

Select them with mouse and use the “Delete” key to remove them.

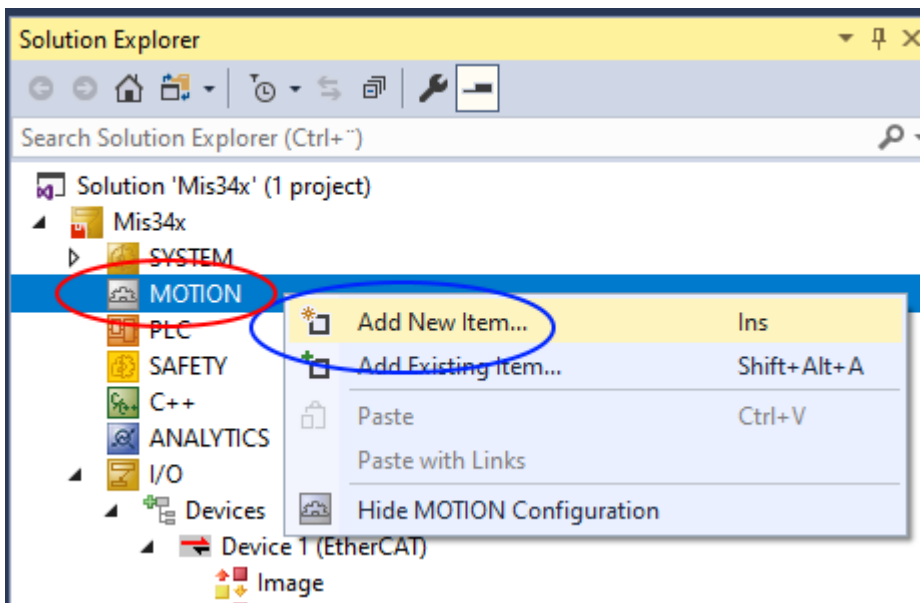
7. Create NC axis

The NC axis is a PLC motion component that represent a virtual axis. You can use a NC axis without any hardware and use it to simulate a moving axis.

In this case, we map the virtual axis to a physical motor. The motor will be a slave of the virtual axis. The NC axis will handle all control and monitoring of the physical motor, when mapped correctly to the axis.

This way the PLC program does not need any information about what kind of hardware that is mapped to the axis. All motors behave the same way when mapped to a NC axis.

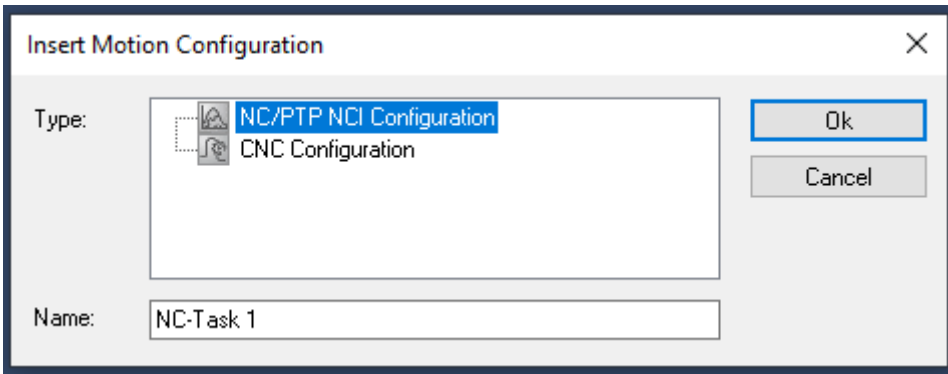
Start by creating a NC task.



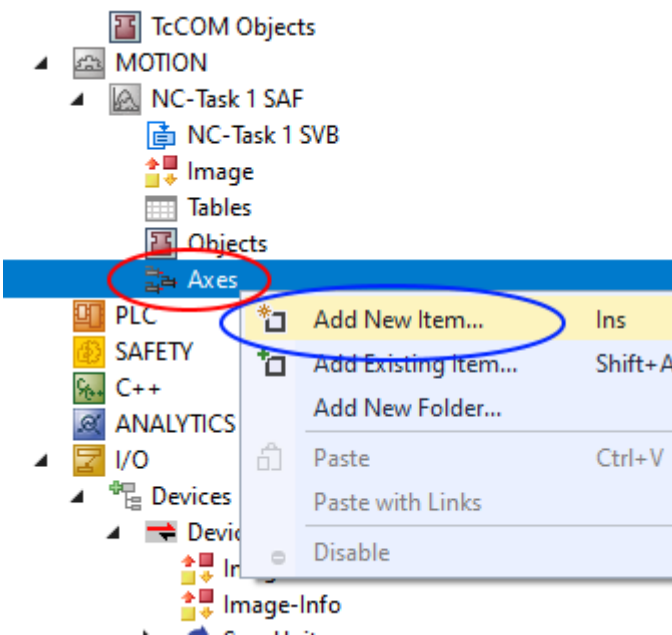
Right click on the MOTION icon,

Select “Add new Item”

Select "NC/PTP NCI configuration" and give the motion configuration a name

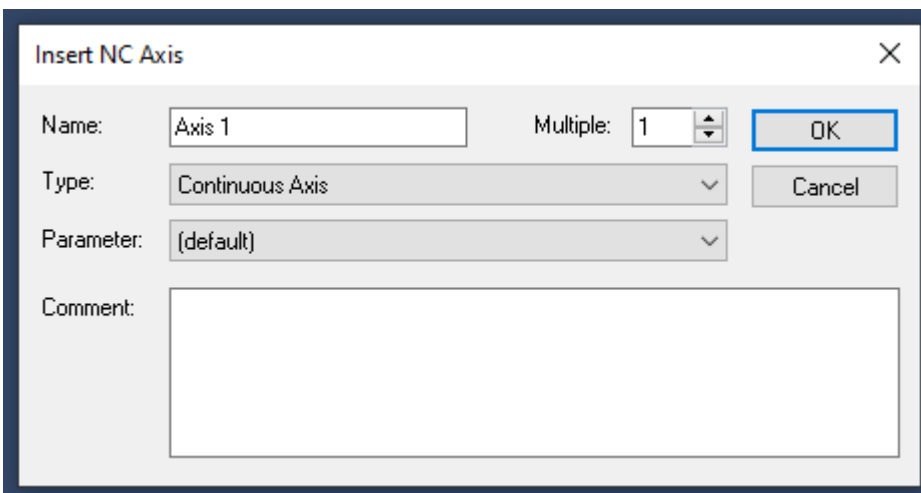


Add the axis to the new task.

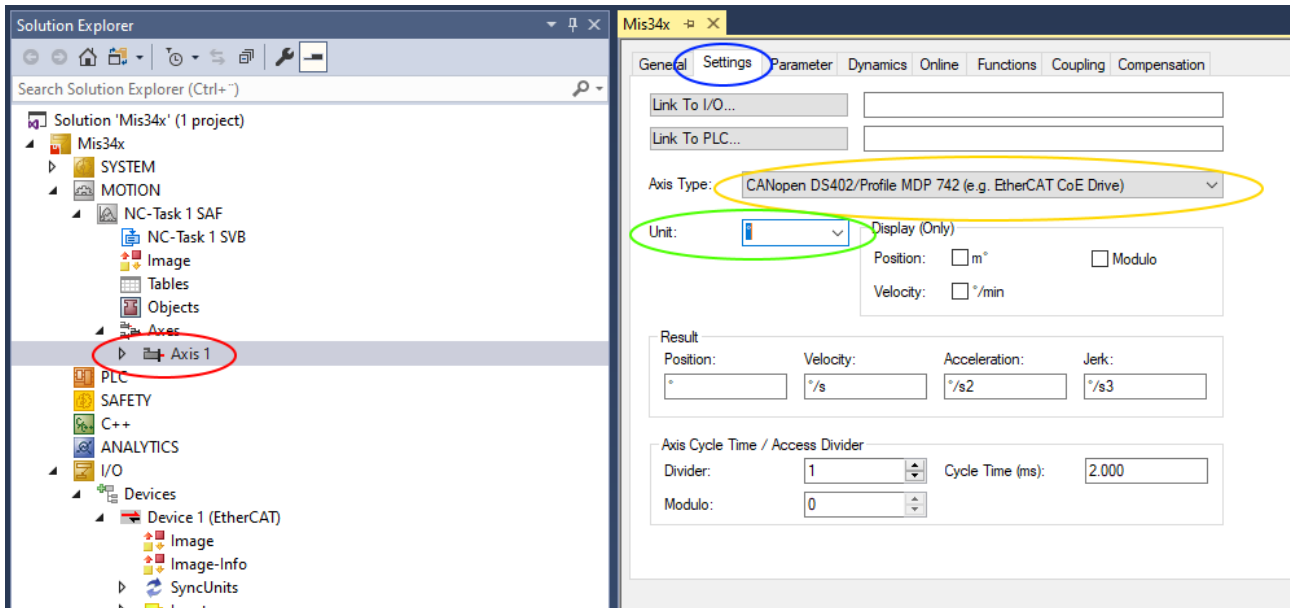


Right click "Axes"
Click "Add new item"

Give the axis a name, and click "OK"



Set the Axis to the right type and with the right units.

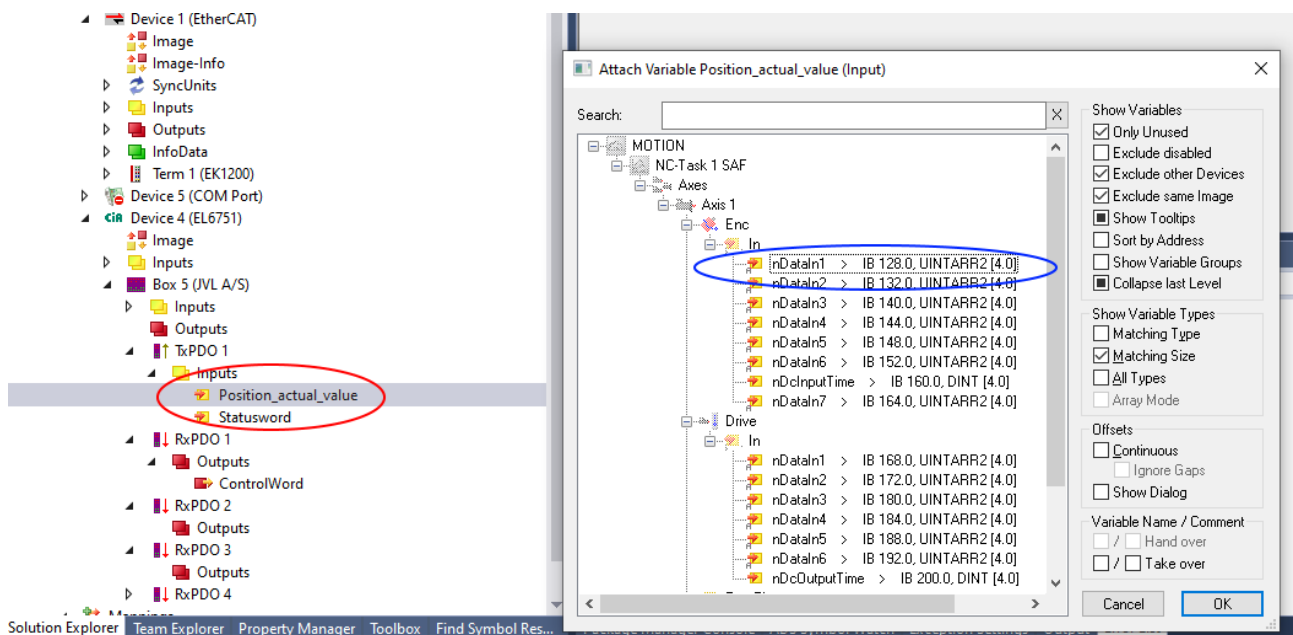


- # Double click on Axis.
- # Select the settings tab
- # Under Axis Type, select CANopen DS402....
- # Units: This example are using angle as units. This could also be a linear axis and could use millimeters instead.

8. Map PDO data

Mapping is where data is mapped from the virtual axis to the physical motor.

8.1 Link position from motor to axis



8.2 Link status word from motor to axis

Attach Variable Statusword (Input)

Search: []

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

Cancel OK

Variable Size Mismatch

Linked Variable: nState1

Linked Variable	Size	Offset
nState1	8	0
Own Variable	16	0
Overlapped	8	

(Size and Offset in bits)

OK Cancel

Variable Size Mismatch

Linked Variable: nState2

Linked Variable	Size	Offset
nState2	8	0
Own Variable	16	8
Overlapped	8	

(Size and Offset in bits)

OK Cancel

8.3 Link ControlWord from axis to motor

Attach Variable ControlWord (Output)

Search: []

MOTION

- NC-Task 1 SAF
 - Axis 1
 - Enc > QB 256.0, MC.NCENCODERSTRUCT_OUT2 [40.0]
 - Drive
 - Out > QB 296.0, MC.NCDRIVESTRUCT_OUT2 [40.0]
 - nDataOut1 > QB 296.0, UINTEARR2 [4.0]
 - nDataOut2 > QB 300.0, UINTEARR2 [4.0]
 - nCtrl1 > QB 304.0, USINT [1.0]
 - nCtrl2 > QB 305.0, USINT [1.0]
 - nCtrl3 > QB 306.0, USINT [1.0]
 - nCtrl4 > QB 307.0, USINT [1.0]
 - nDataOut3 > QB 308.0, UINTEARR2 [4.0]
 - nDataOut4 > QB 312.0, UINTEARR2 [4.0]
 - nDataOut5 > QB 316.0, UINTEARR2 [4.0]
 - nDataOut6 > QB 320.0, UINTEARR2 [4.0]
 - nCtrl5 > QB 324.0, USINT [1.0]
 - nCtrl6 > QB 325.0, USINT [1.0]
 - nCtrl7 > QB 326.0, USINT [1.0]
 - nCtrl8 > QB 327.0, USINT [1.0]
 - ToPlc > QB 0.0, MC.NCTOPLC_AXIS_REF [256.0]

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

Cancel OK

Variable Size Mismatch

Linked Variable: nCtrl1

Size	Offset
8	0
Own Variable: 16	0
Overlapped: 8	

(Size and Offset in bits)

OK Cancel

Variable Size Mismatch

Linked Variable: nCtrl2

Size	Offset
8	0
Own Variable: 16	8
Overlapped: 8	

(Size and Offset in bits)

OK Cancel

8.4 Link position from axis to motor

Attach Variable interpolation_data (Output)

Search: []

MOTION

- NC-Task 1 SAF
 - Axis 1
 - Enc > QB 256.0, MC.NCENCODERSTRUCT_OUT2 [40.0]
 - Drive
 - Out > QB 296.0, MC.NCDRIVESTRUCT_OUT2 [40.0]
 - nDataOut1 > QB 296.0, UINTEARR2 [4.0]
 - nDataOut1[0] > QB 296.0, UINT [2.0]
 - nDataOut1[1] > QB 298.0, UINT [2.0]
 - nDataOut2 > QB 300.0, UINTEARR2 [4.0]
 - nDataOut2[0] > QB 300.0, UINT [2.0]
 - nDataOut2[1] > QB 302.0, UINT [2.0]
 - nCtrl3 > QB 306.0, USINT [1.0]
 - nCtrl4 > QB 307.0, USINT [1.0]
 - nDataOut3 > QB 308.0, UINTEARR2 [4.0]
 - nDataOut3[0] > QB 308.0, UINT [2.0]
 - nDataOut3[1] > QB 310.0, UINT [2.0]
 - nDataOut4 > QB 312.0, UINTEARR2 [4.0]
 - nDataOut4[0] > QB 312.0, UINT [2.0]
 - nDataOut4[1] > QB 314.0, UINT [2.0]
 - nDataOut5 > QB 316.0, UINTEARR2 [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

Cancel OK

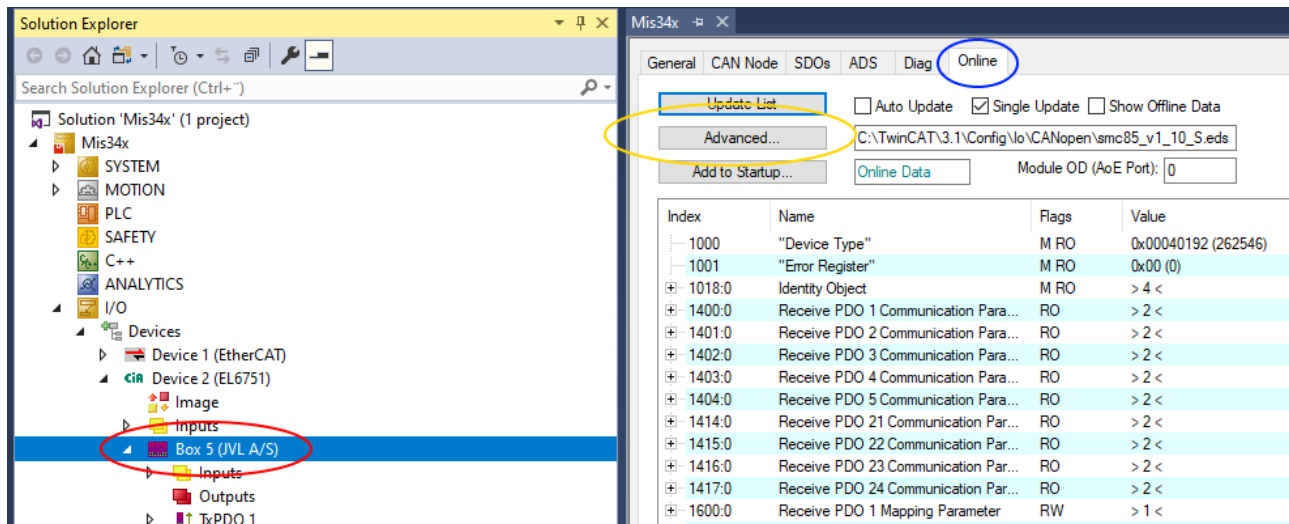


9. Set SDO data

SDO is data send at startup to do some pre-use configuration, to enable the motor to respond correctly to the PDO data.

A few parameters in the motor need to be setup:

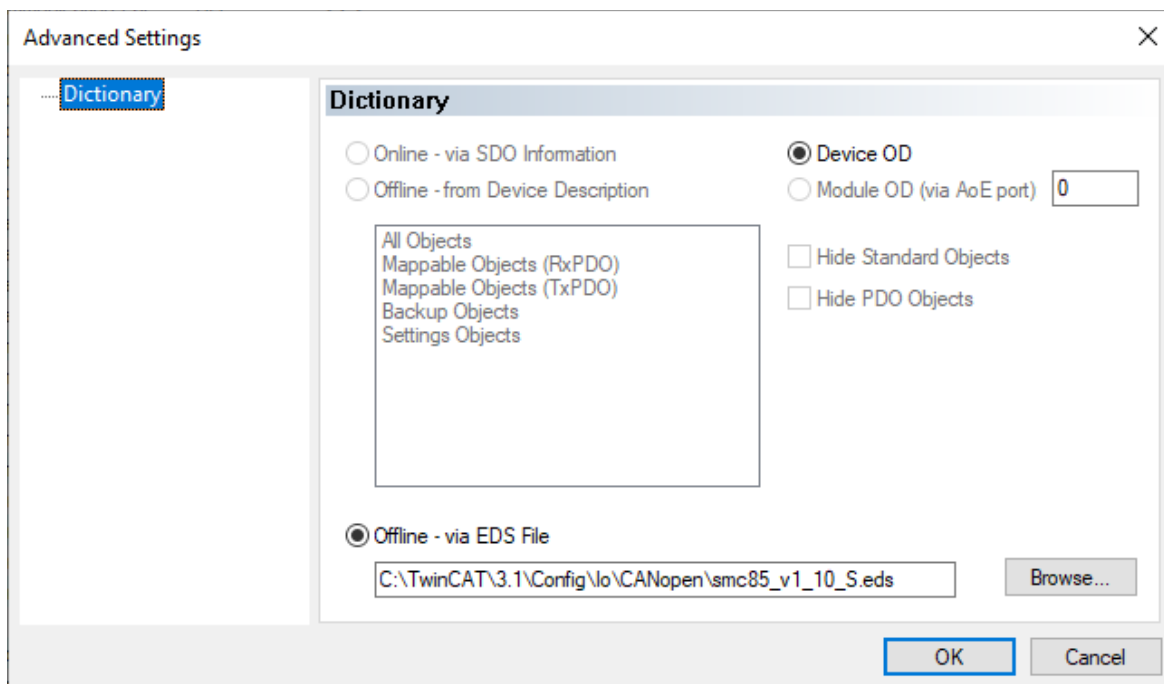
Set motor “mode of operation” to 7 : Interpolated position mode. This is the operation mode the NX axis is using to control the motor. The NC axis expect the motor to be in this operation mode when its available and ready.



Click on Motor Box.

Select the online tab

If list is empty. Click the Advanced button. And select the EDS file manually. (Window below)





Index	Name	Flags	Value
201A:0	Motor parameters 257-510 (all 16-bit)	RO	> 254 <
6007	Abort_connection_option_code	RW	0
603F	Error_code	RW	0x0000 (0)
6040	Controlword	RW P	0x0000 (0)
6041	Statusword	RO P	0x0260 (608)
605A	Quick_stop_option_code	RW	2
6060	Modes_of_operation	RW P	0
6061	Modes_of_operation_display	RO P	0
6064	Position_actual_value	RO P	11379641
6067	Position_window	RW	0x00000064 (100)
6069	Velocity_sensor_actual_value	RO	0
606B	Velocity_demand_value	RO	5000
606C	Velocity_actual_value	RO P	0
6072	Max_torque	RW	0x03E8 (1000)
607A	Target position	RW P	0

- # Select object 6060 Modes of operation from the list of objects.
- # Click the button “Add to startup”

Dialog box: Edit SDO Entry

Index (hex): 0x6060

Subindex (dec): 0

Length (dec): 1

Value (dec): 7 (hex): 0x7

Buttons: OK, Cancel

- # Change value to 7 and press OK.

9.1 Set the synchronous timing with SDO

The motor and the PLC need to be synchronized. The PLC used the timing to transmit Sync messages to the motor, and the motor uses the timing to calculate when to expect the next sync message. It is important that the motor and the PLC is set up to use the same timing.

The object 60C2 sub 01 is used to tell the motor what timing is used in the PLC.

Select the object 602:01 ip_time_Period



General | CAN Node | SDOs | ADS | Diag | Online

Update List Auto Update Single Update Show Offline Data

Advanced... C:\TwinCAT\3.1\Config\Io\CANopen\smc85_v1_42_S.edc

Add to Startup... Online Data Module OD (AoE Port): 0

Index	Name	Flags	Value
609A	Homing_acceleration	RW	0x00001388 (5000)
60C1:0	interpolation_data_record	RO	> 2 <
60C2:0	interpolation_time_period	RO	> 2 <
60C2:01	ip_time_Period	RW	10
60C2:02	ip_time_index	RW	-3
60FD	Digital_inputs	RO P	0x00000000 (0)
60FE:0	Digital_outputs	RO	> 2 <
60FF	Target_velocity	RW P	5000

Click the button “Add to startup”

Edit SDO Entry

Index (hex): 0x60C2 OK

Subindex (dec): 1 Cancel

Length (dec): 1

Value (dec): 10 (hex): 0xA

Set value for the sync time in milliseconds. The default is 10ms. Lower synchronization can be used for better precision.

Make sure this value is the same set by the CANopen master

Both custom SDO values can be viewed and edited under the SDO tab

General | CAN Node | SDOs | ADS | Diag | Online

Obj. idx	Sub. idx	Length	Value (dec)	Value (hex)
<0x1400>	1	4	517	0x205
<0x1400>	2	1	255	0xFF
<0x1401>	1	4	773	0x305
<0x1401>	2	1	255	0xFF
<0x1402>	1	4	1029	0x405
<0x1402>	2	1	255	0xFF
<0x1403>	1	4	1285	0x505
<0x1403>	2	1	255	0xFF
0x6060	0	1	7	0x7
0x60c2	1	1	10	0xA

Restart Node when no TxPDOs are received for 10s after Start Node

max. SDOs in Send Queue: 5 max. Boot-Up Timeout (s): 0

max. SDO Timeout (ms): 2000

Append... Insert... Delete... Edit...



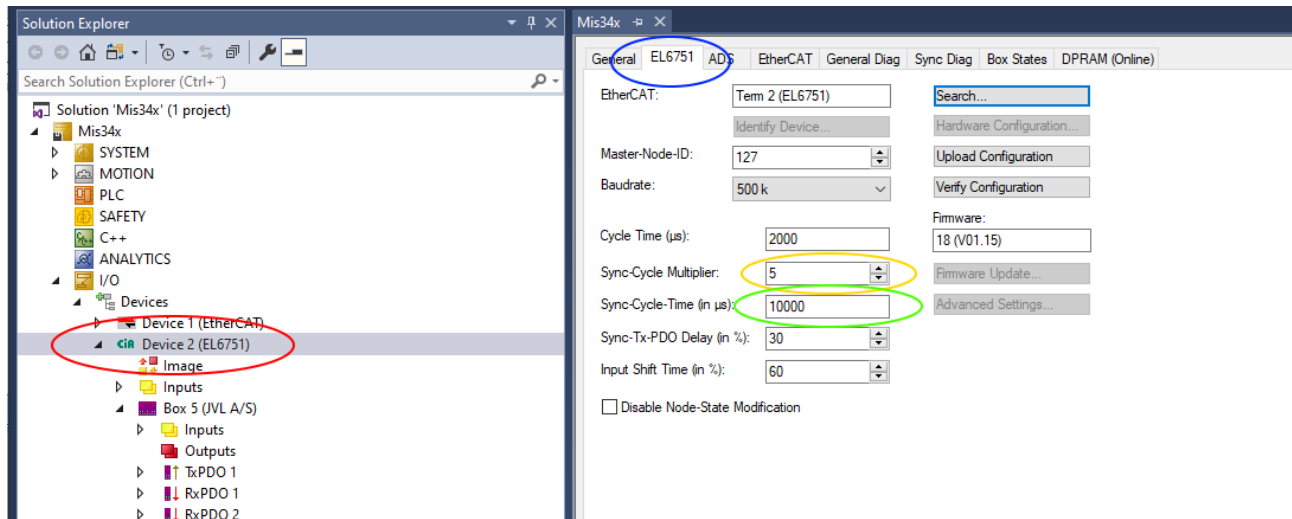
10. Sync interval

This is where the CANopen master timing is configured.

The cycle time is setup automatically to the PLC task that uses the fastest timing. In this case we only have a NC task, which by default uses 2ms cycle time. We can adjust the Sync time by multiplying the cycle time.

The default cycle time for JVL motor is 10ms, which need a multiplier of 5.

Set the CANopen master Sync interval. The sync interval should match the object 60C2:01

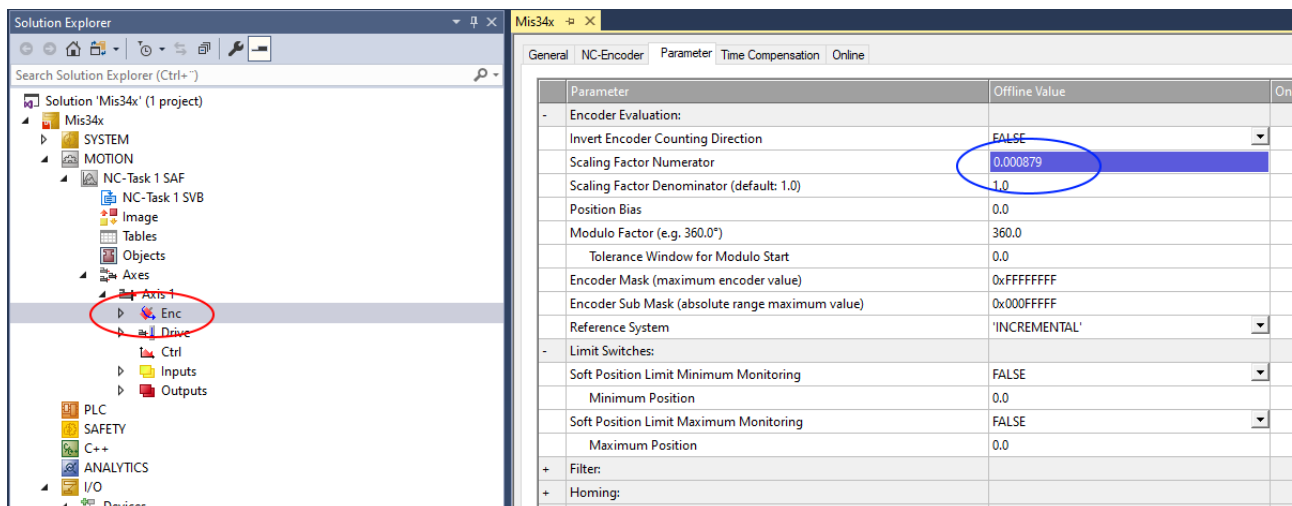


- # Select the CiA device
- # Click the EL6751 (Beckhoff CANopen master)
- # Adjust the Multiplier so that Cycle Time times Multiplier is the Sync interval.
- # Set Sync interval updates on next build.

11. Set gear factor

The gear factor converts the virtual axis units to a physical unit. In this example we use angle. Thus 360 units is 1 revolution on the motor.

11.1 Set scaling factor in degrees (example)



- # Select the Encoder (Enc) for the NC axis



Enter the calculated unit for the motor. See example calculation below.

Example calculation

Count per revolution = 409600

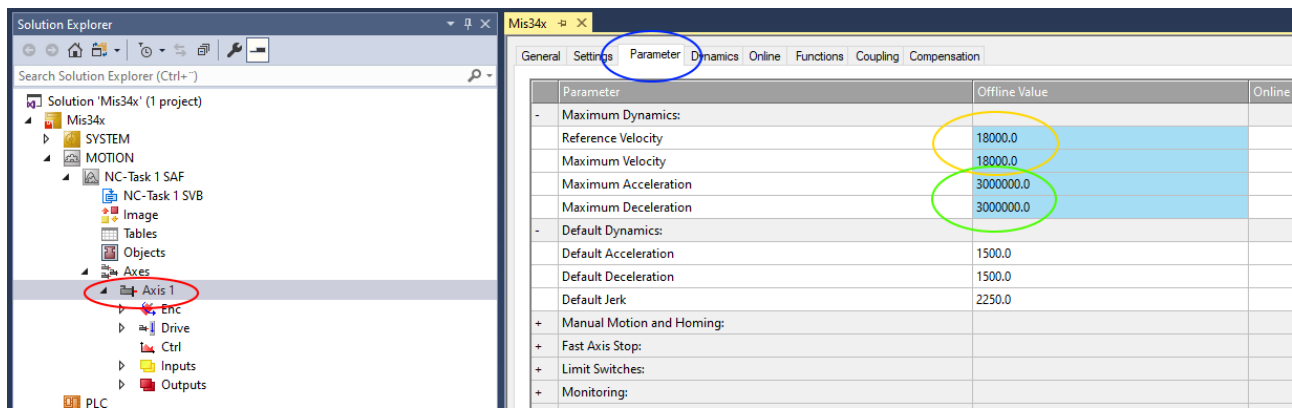
Degrees per revolution = 360

Scaling Factor Numerator:

Degrees per Count = Degrees per Revolution / Count per Revolution = 360/409600 = 0,000879

12. Set motor parameters

Maximum velocity and maximum acceleration should also be entered to get the best performance from the motor.



Select the motor axis

Select Parameters

12.1 Maximum Velocity

Enter the calculated values. Example calculation below.

Example calculation:

Max RPM = 3000

Seconds per Minute = 60

Degrees per Revolution = 360

Degrees per Second = Max RPM * Degrees per Revolution / Seconds per Minute = 3000 * 360 / 60 = 18000

12.2 Max acceleration

Enter the calculated values. Example calculation below.

Example calculation:

Max Acceleration RPM= 500000

Seconds per Minute = 60

Degrees per Revolution = 360

Degrees in Seconds² = Max Acceleration RPM * Degrees per Revolution / Seconds per Minute = 500000 * 360 / 60 = 3000000

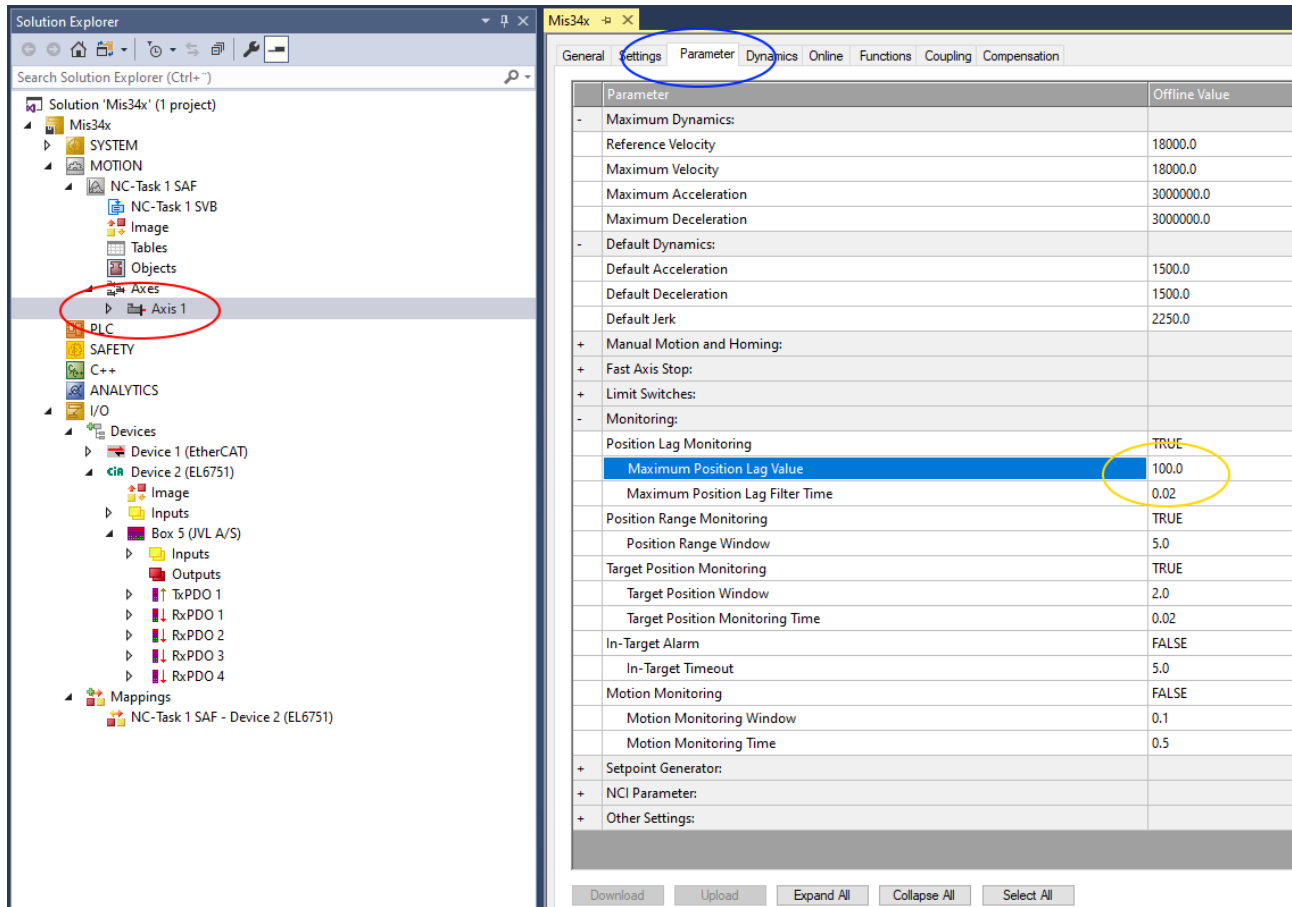
13. Follow Error Monitoring

The NC axis monitors the Follow Error. This is how far behind the motor is at any given time. The motor will always be behind at least the time it takes to move between two sync cycles. With full speed that is significant.

It is recommended to adjust the Follow error limit to your suit the application.

TwinCAT's terminology for Follow Error is "Position Lag Value".

In this example it is set to 100.0 degrees.

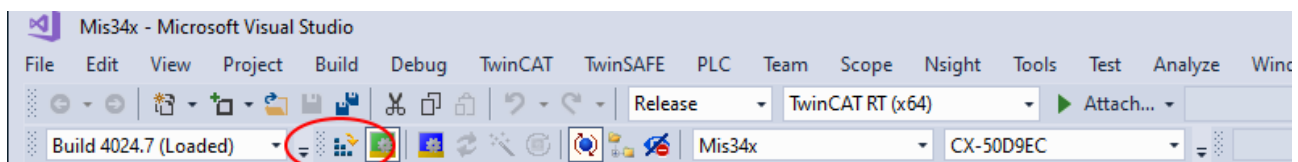


- # Select the motor Axis
- # Select Parameters
- # Enter the Follow Error limit in the "Maximum Position Lag value" field.

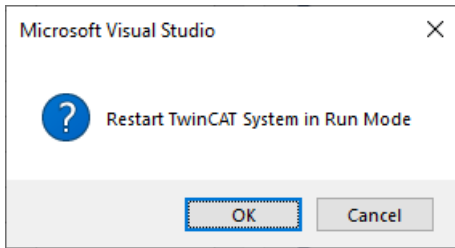
14. Store settings in PLC

To use the new settings in the PLC project , the configuration needs to be activated.

This is done with the "Activate configuration" button



- # Click the Activate configuration



Allow TwinCAT to restart in Run mode by clicking "OK".

Now the NC axis is ready to control your JVL motor!