

Running MIS as NC axis in TwinCAT:

The MIS34x motor support the TwinCAT NC axis. The NC axis enables you to control the motor with PLC open standard.

NC axis support synchronized axis, 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 variant in behavior.

This sample will guide you in the following

Setup the motor for CanOpen

Create a new TwincCAT project.

Connect to your PLC hardware.

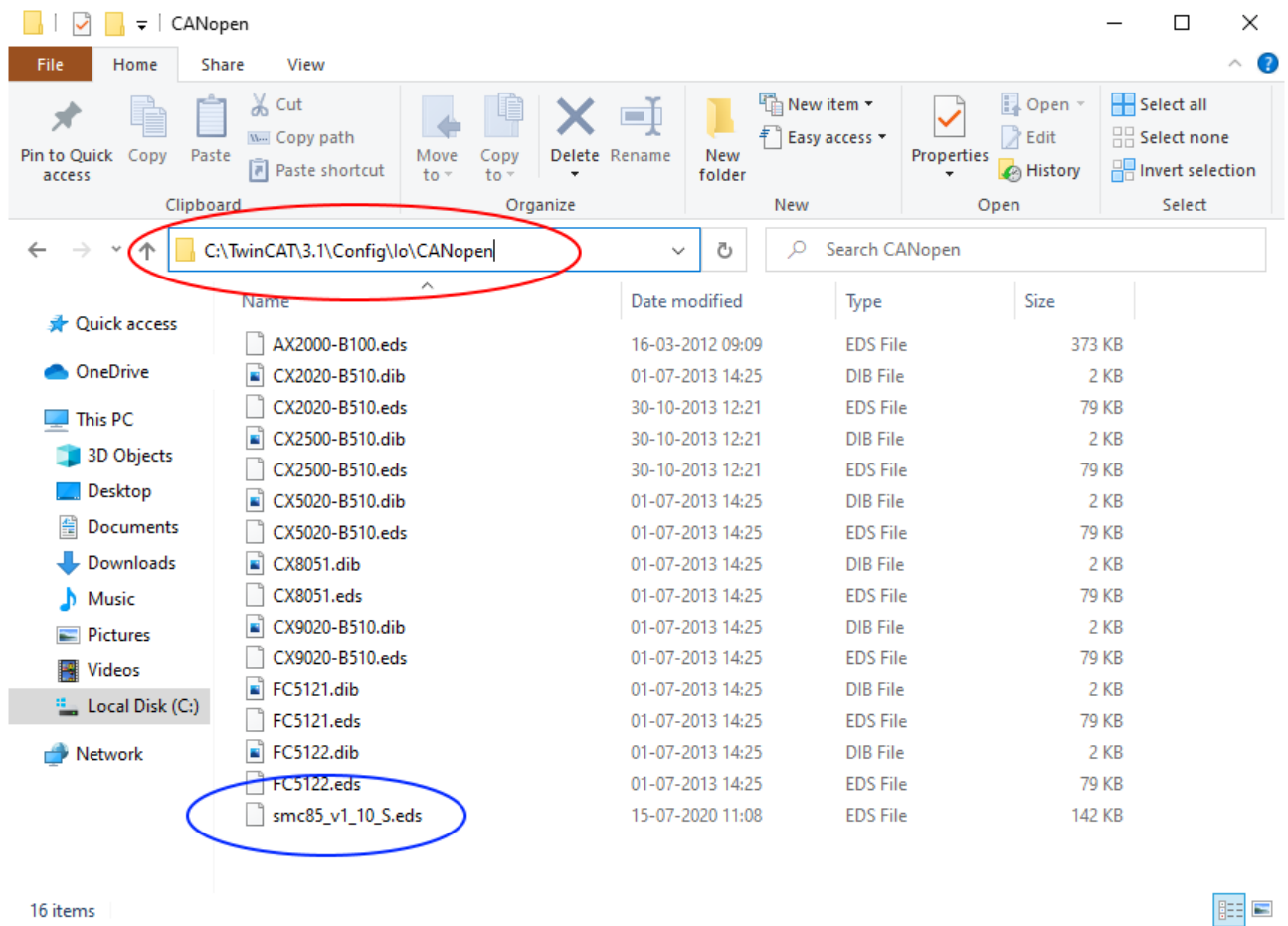
Scan your Hardware.

Setup the connected motor for NC axis support.

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.



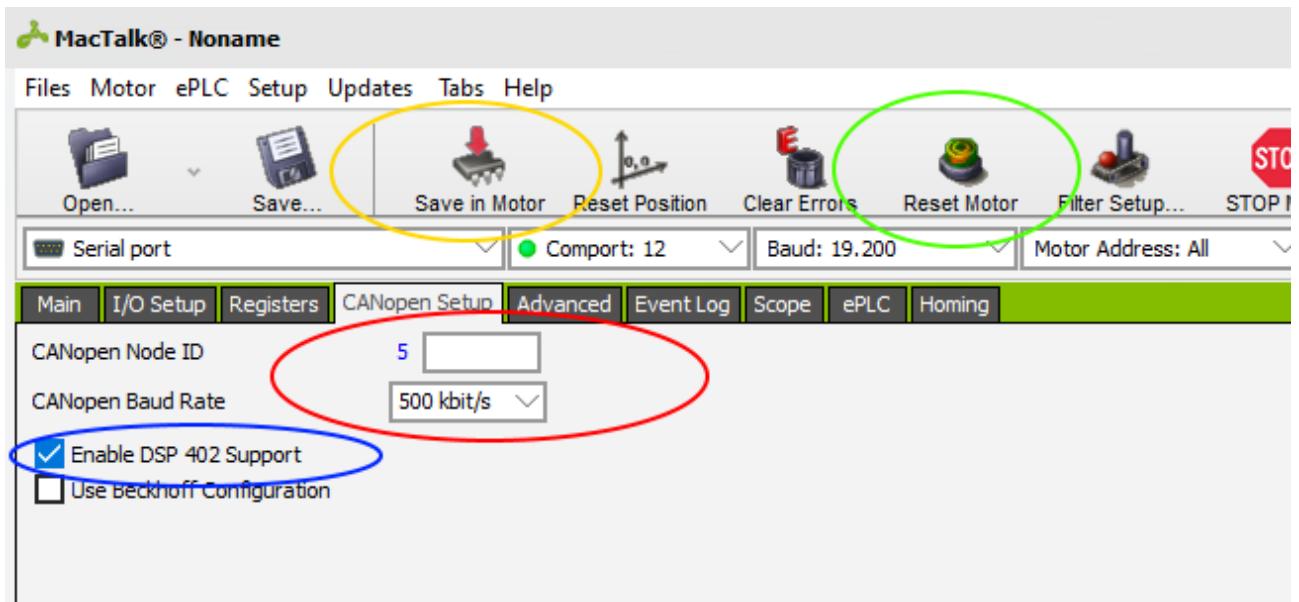
#Open the TwinCAT folder.

#Copy the EDS file to this location. I am using EDS file version 1.10 in this sample

Setup motor

Connect to the MIS34x motor with Mactalk.

Open the CANopen setup page.



#Choose you node ID and Baud rate. Node id 5 and 500kbits/second is default.

#Make sure Enable DSP 402 support Is checked

#Save the settings in the motor.

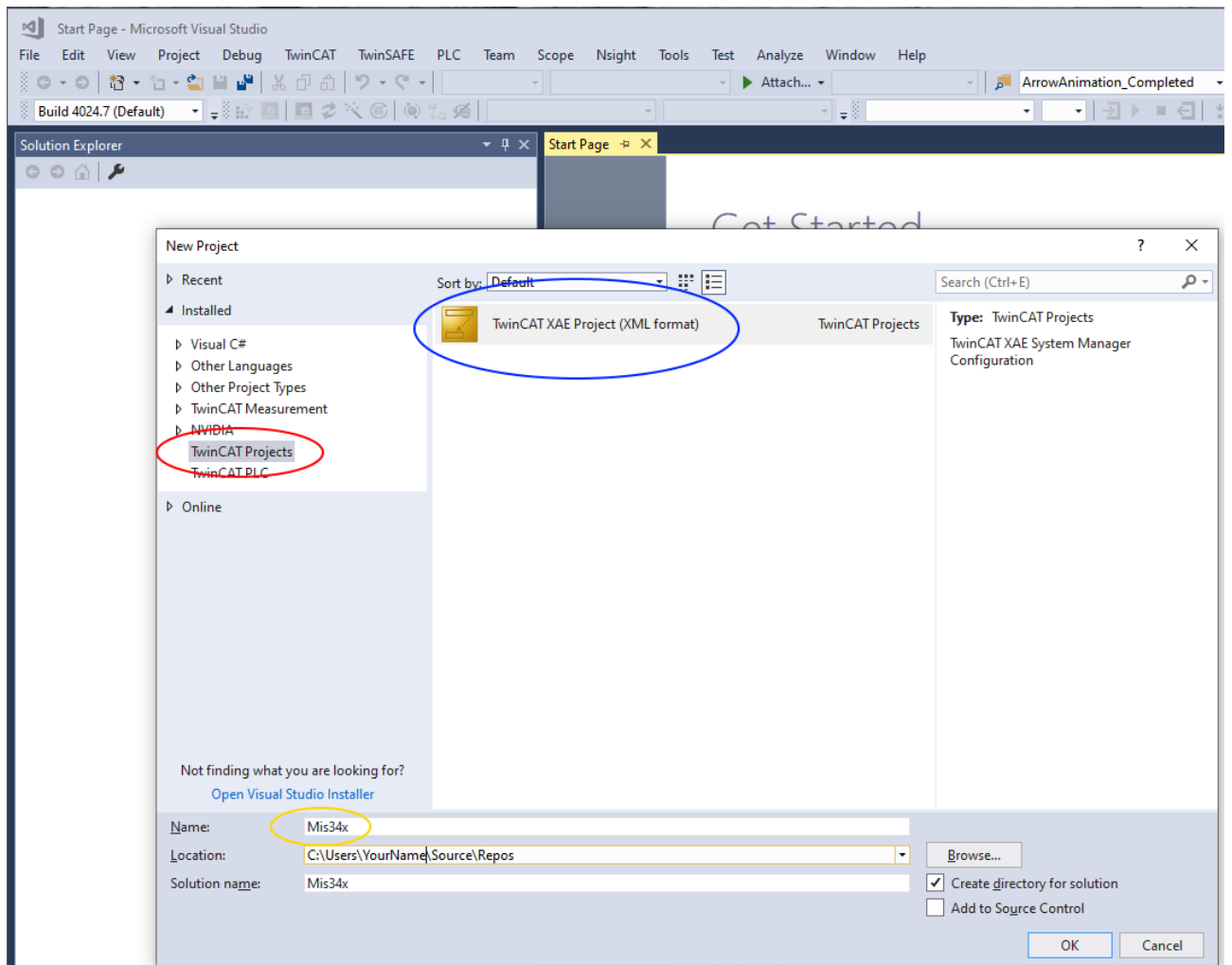
#Reset the motor

Create 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 sample is done in Visual Studio.

Click File->New project.



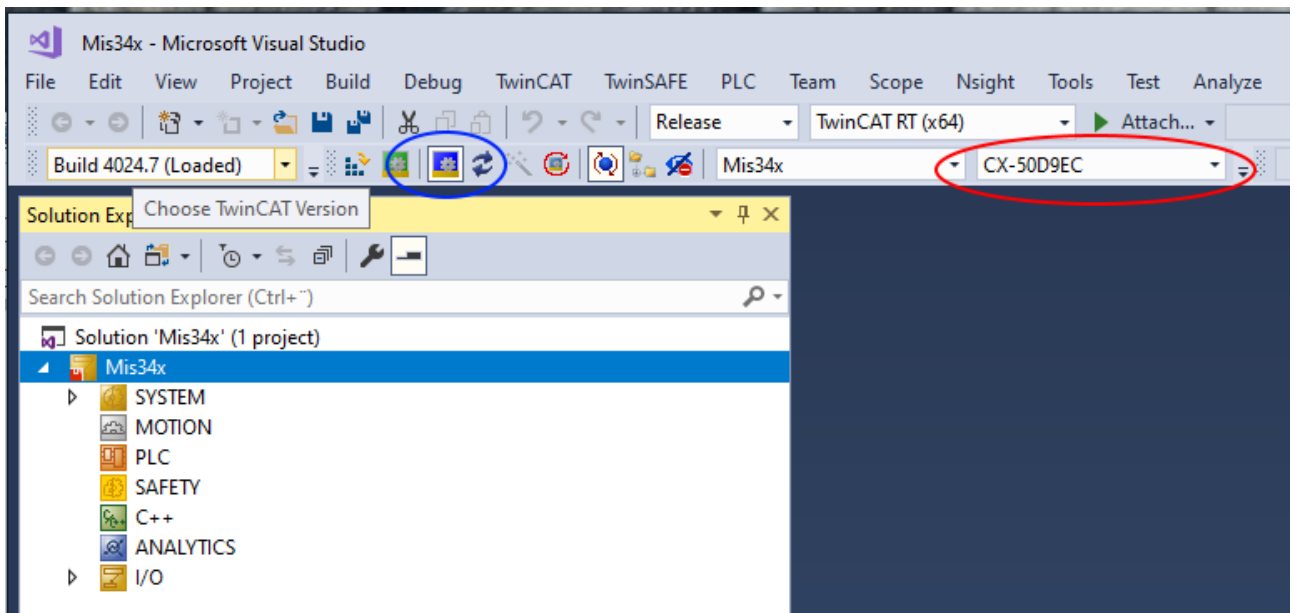
#In the new project window, select TwinCAT project.

Select the twinCAT XAE project

Give your project a name and click “ok” button.

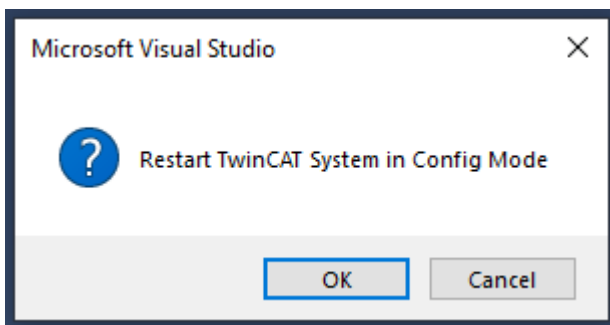
Connect to PLC in Config mode

Connect to PLC to scan connected hardware. This will also detect node attached to the Canbus. Make sure that all other nodes are disconnected, to get the same result as this sample.



#Select the ADS connection for your PLC.

#Click the Restart TwinCAT in config mode 

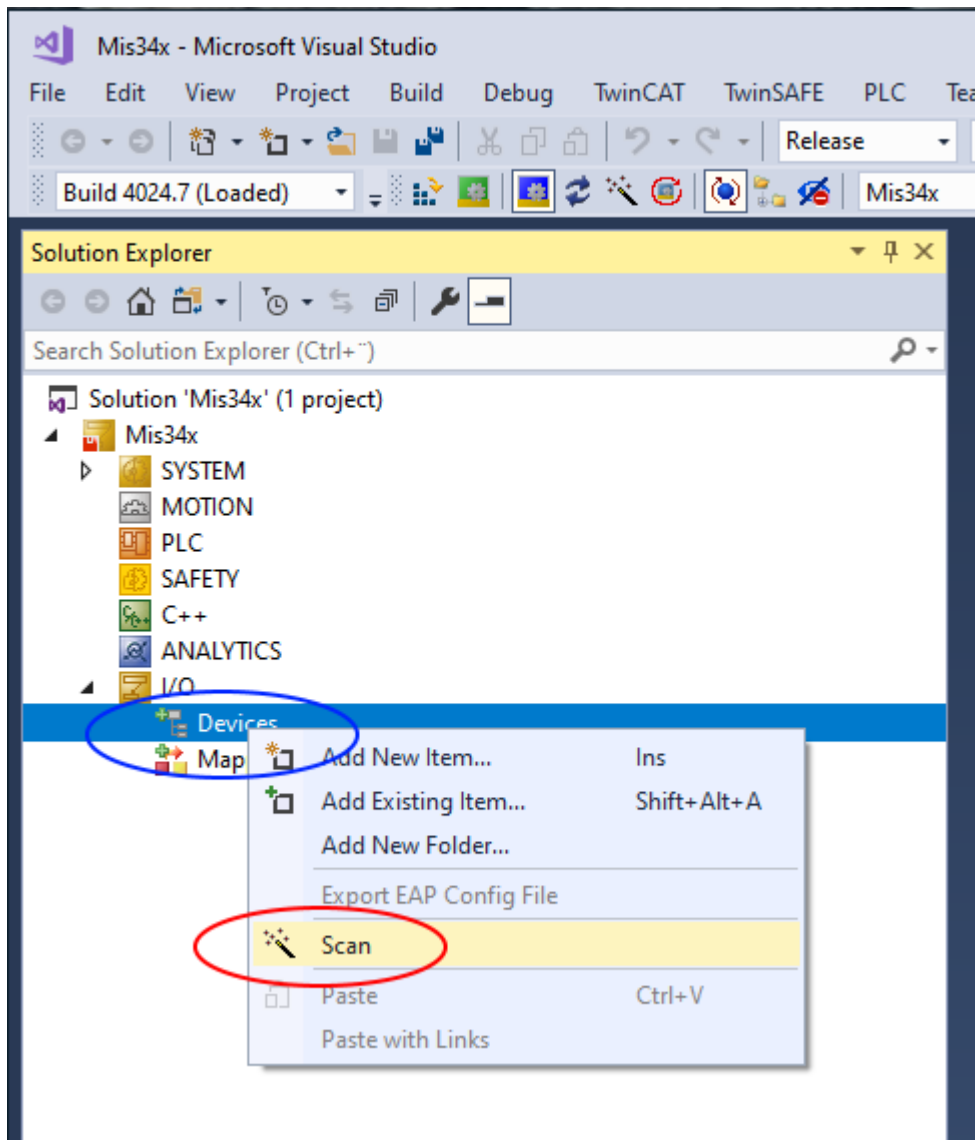


Click OK to allow TwinCAT to restart.

You should be connected to your PLC.

Scan for hardware

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

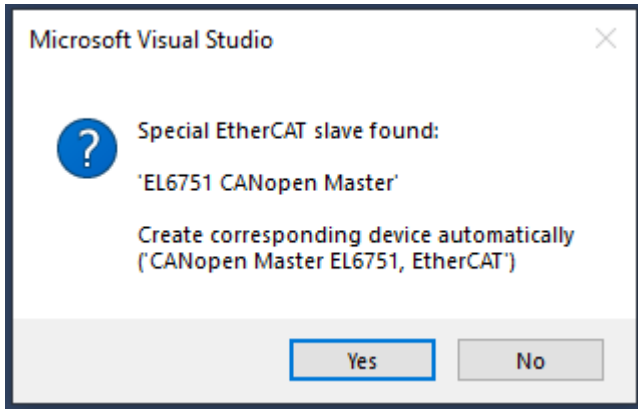


Right click IO->Devices in your solution tree.

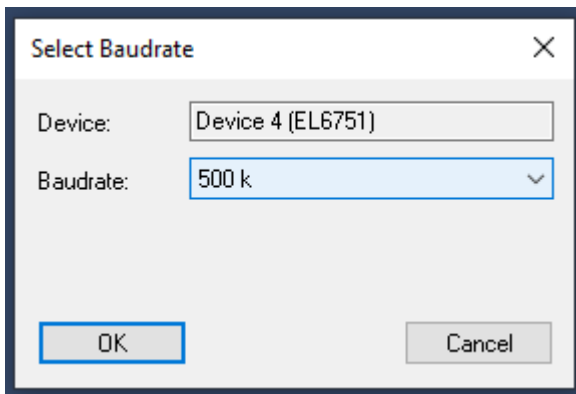
Select Scan

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 my case, the Device1 Ethercat.



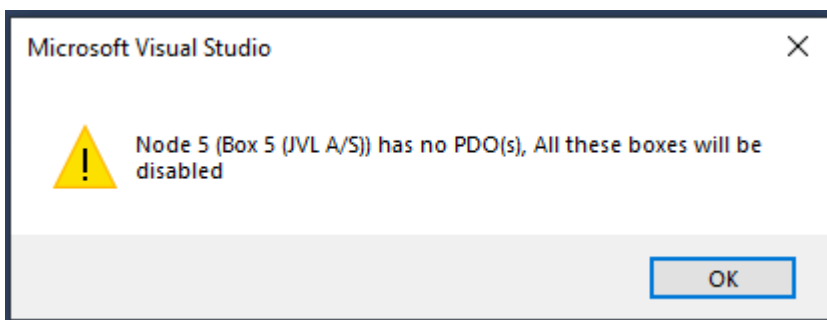
When the master is found, you will be prompted to create CanOpen devices automatically. Click "Yes". I am using the EL6751 CanOpen Master module.



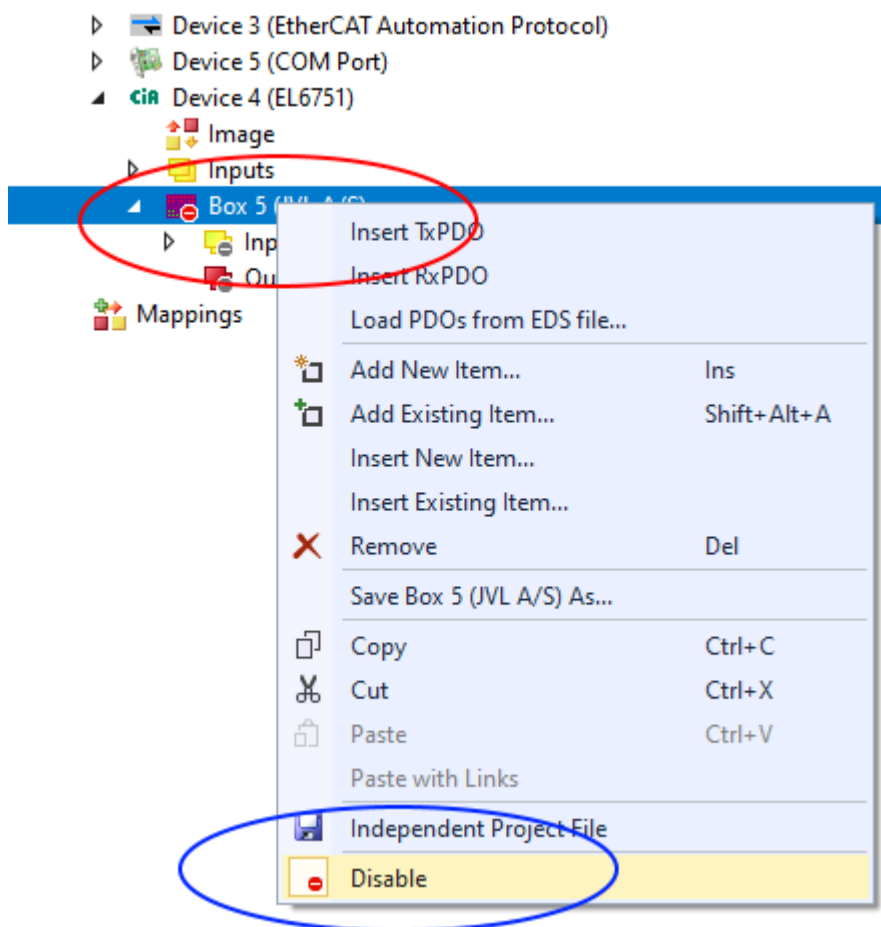
Select the baudrate you selected in macktalk, to scan your CanOpen bus with and Click "OK"

If the scan was able to detect the PDO automatically, continue to the next paragraph "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.

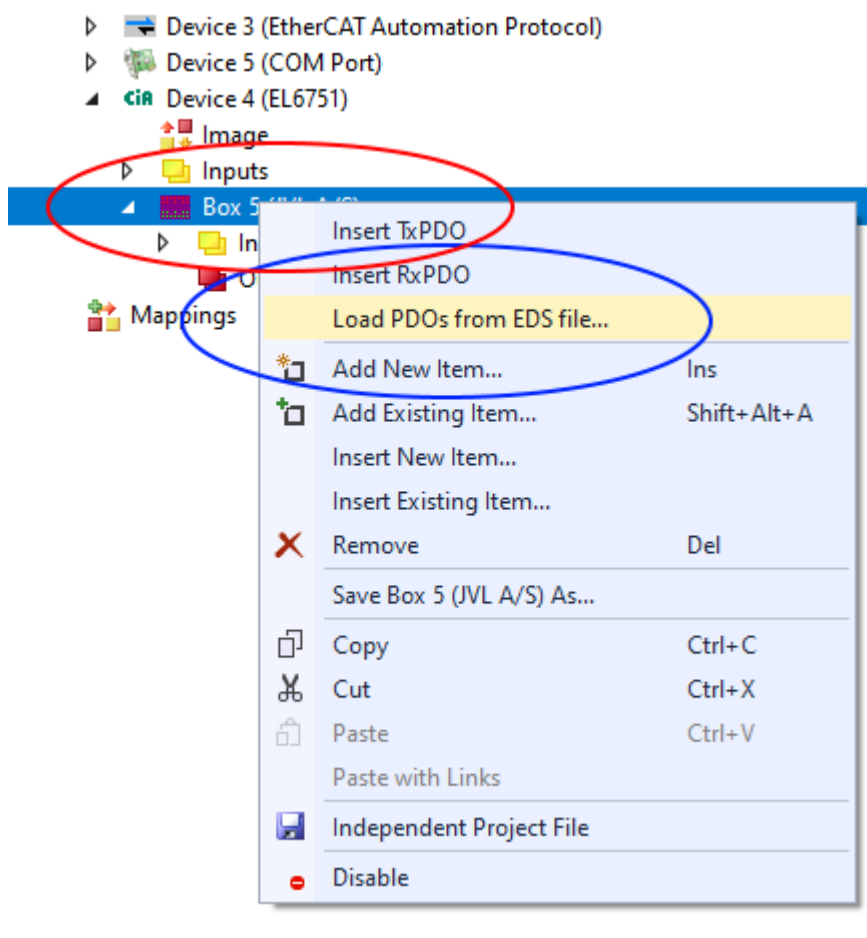


Find the Motor box under your CanOpen master device.

#Right click the motor Box

#Click Disable, to re enable the node

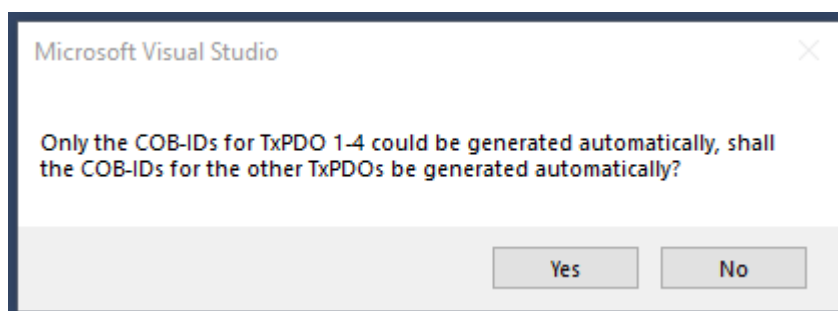
Load the PDOs from EDS file.



#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 COBid's for now.

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.

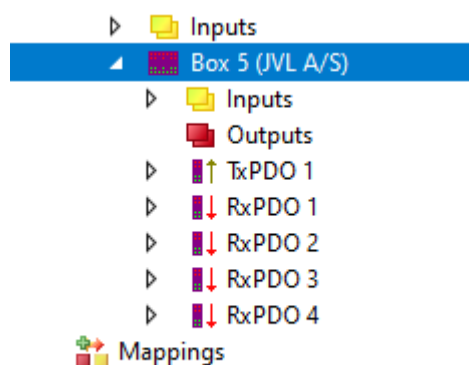
The scan can generate up to 24 RX PDOs, and 24 TX PDO's

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.

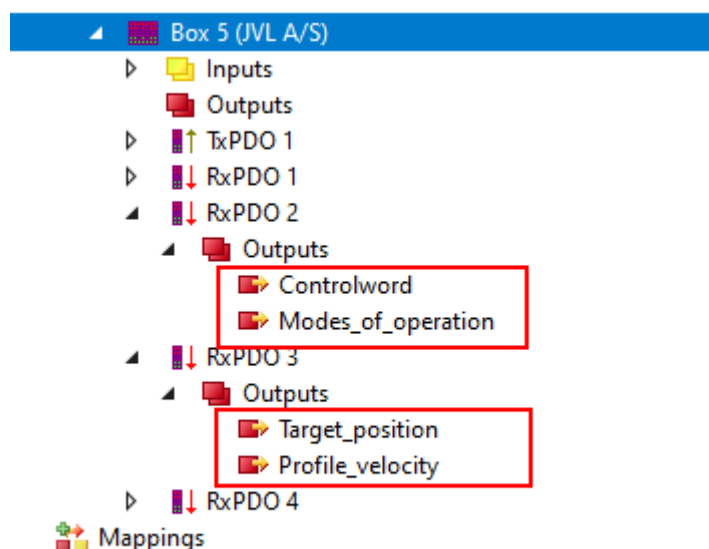
Delete TXpdo2 – Txpdo24 and RXpdo5 to RXpdo24

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.

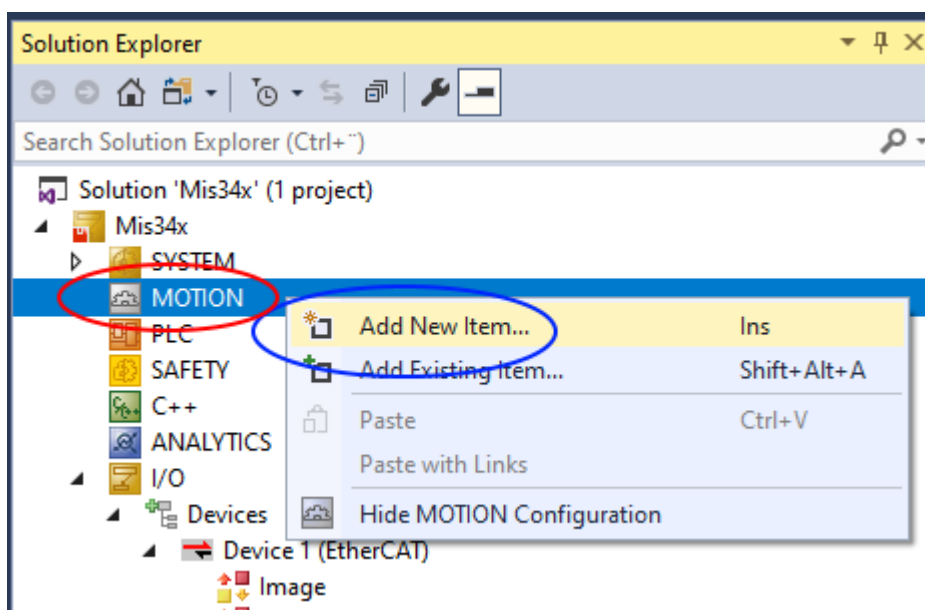
Create NC axis

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 motor behave the same when mapped to a NC axis.

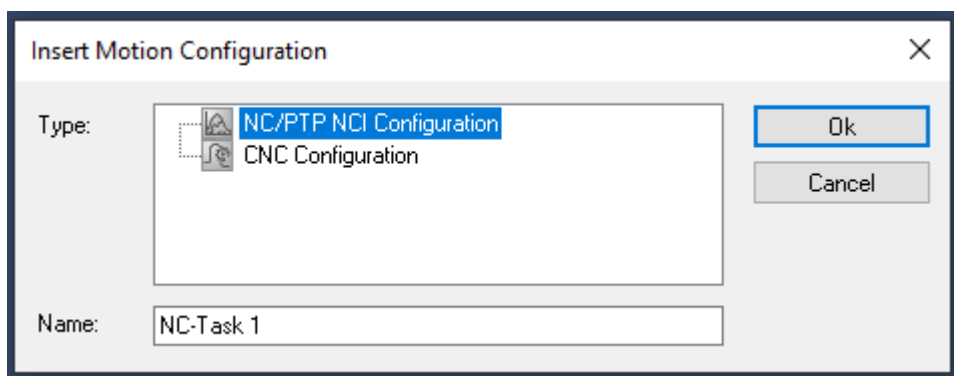
Start by creating a NC task.



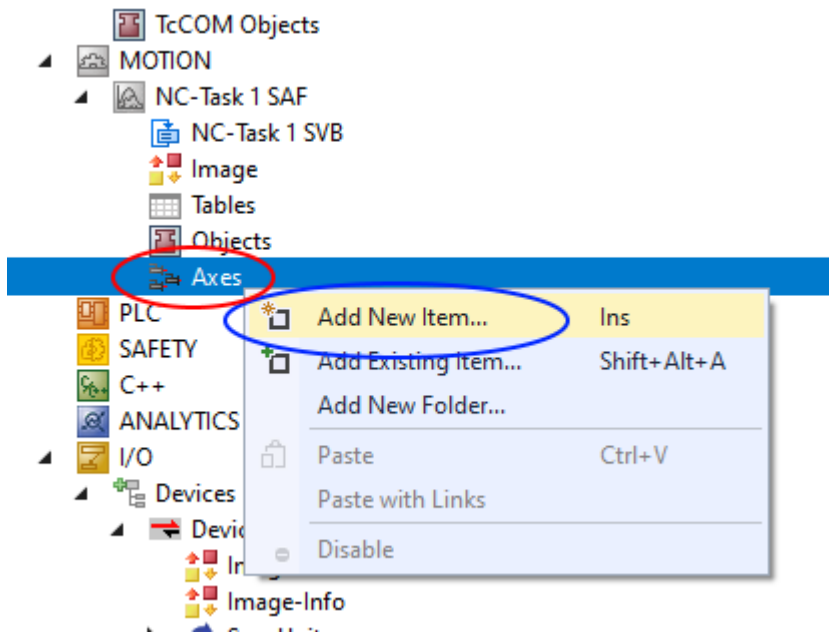
Right click on the Motion icon,

select "Add new Item"

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



Add the axis to the new task.



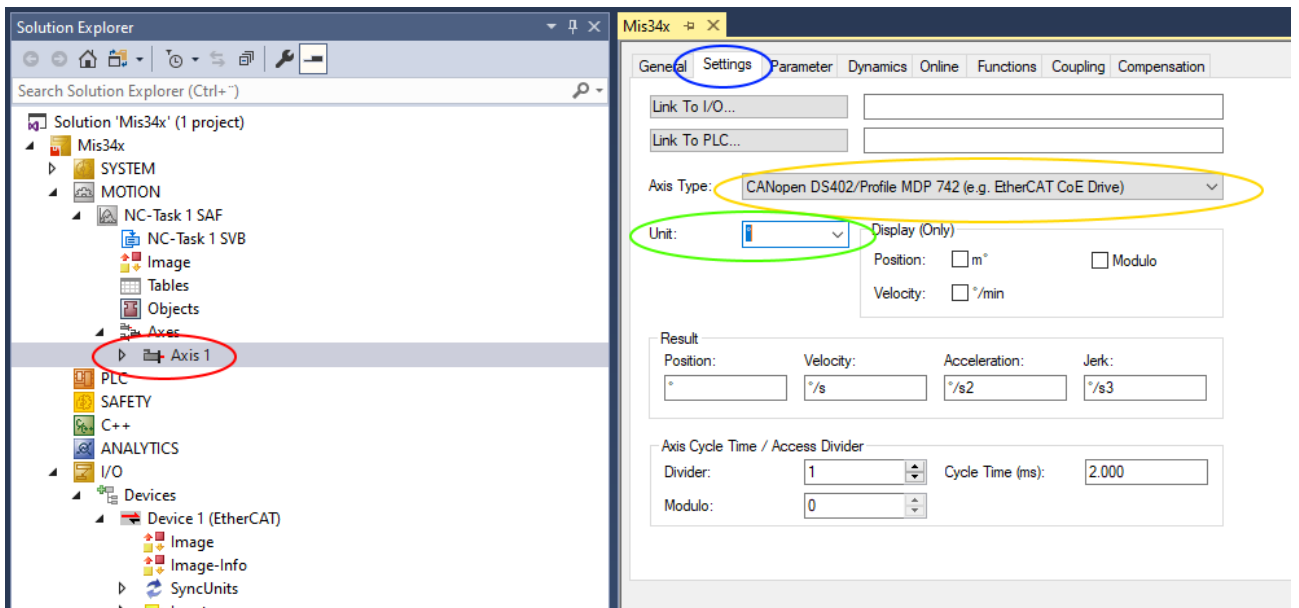
Right click "Axes"

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

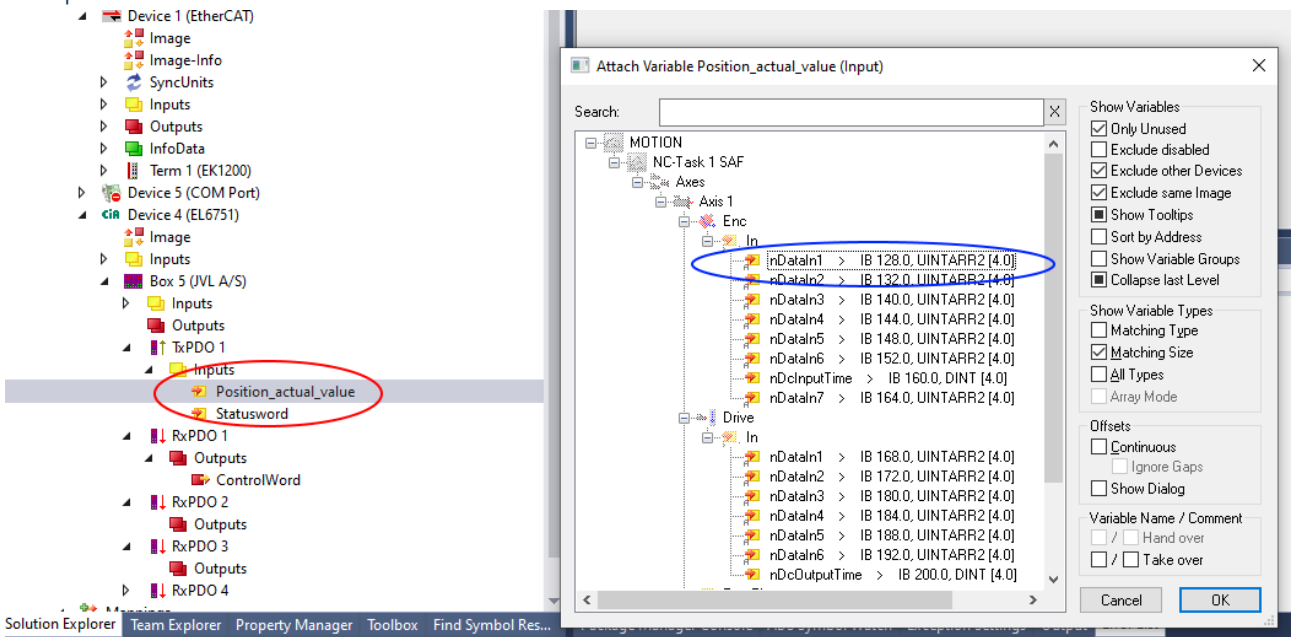
#Under axis type, select CanOpen DS402....

#Units: This example are using angle as units. This could also be a linear axis and could use millimeters instead.

Map PDO data

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

Link position from motor to axis



Link status word from motor to axis

The screenshot shows the SIMATIC Manager interface with the variable declaration tree on the left and the 'Attach Variable Statusword (Input)' dialog box on the right.

Variable Declaration Tree (Left):

- ANALY IILS
 - I/O
 - Devices
 - Device 1 (EtherCAT)
 - Device 5 (COM Port)
 - Device 4 (EL6751)
 - Image
 - Inputs
 - Box 5 (JVL A/S)
 - Inputs
 - Inputs
 - Position_actual_value
 - Statusword

'Attach Variable Statusword (Input)' Dialog Box (Right):

- Search:** [Empty]
- Tree:** MOTION > NC-Task 1 SAF > Axes > Axis 1 > Drive > nDataIn2[1] > IB 174.0, UINT [2.0]
- Selected Variables:**
 - nState1 > IB 176.0, USINT [1.0]
 - nState2 > IB 177.0, USINT [1.0]
- 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
 - Away Mode
- Offsets:**
 - Continuous
 - Ignore Gaps
 - Show Dialog
- Variable Name / Comment:**
 - Hand over
 - Take over
- Buttons:** Cancel, OK

The 'Variable Size Mismatch' dialog box shows the following configuration for the variable `nState1`:

| Field | Value |
|------------------|---------|
| Linked Variable: | nState1 |
| Size | 8 |
| Offset | 0 |
| Own Variable: | 16 |
| Overlapped: | 8 |

(Size and Offset in bits)

Buttons: OK, Cancel

The 'Variable Size Mismatch' dialog box shows the following configuration for the variable `nState2`:

| Field | Value |
|------------------|---------|
| Linked Variable: | nState2 |
| Size | 8 |
| Offset | 0 |
| Own Variable: | 16 |
| Overlapped: | 8 |

(Size and Offset in bits)

Buttons: OK, Cancel

Link Controlword from axis to motor

The screenshot shows the variable declaration tree on the left and the 'Attach Variable ControlWord (Output)' dialog on the right.

Variable Declaration Tree (Left):

- Device 1 (EtherCAT)
 - Image
 - Image-Info
 - SyncUnits
 - Inputs
 - Outputs
 - InfoData
 - Term 1 (EK1200)
 - Device 5 (COM Port)
 - Device 4 (EL6751)
 - Image
 - Inputs
 - Box 5 (JVL A/S)
 - Inputs
 - RxPDO 1
 - Inputs
 - Position_actual_value
 - Statusword
 - Outputs
 - ControlWord
 - RxPDO 2
 - Outputs
 - RxPDO 3
 - Outputs
 - RxPDO 4
 - Outputs

Attach Variable ControlWord (Output) Dialog (Right):

The dialog shows a tree view of variables under 'MOTION'. The following variables are highlighted with yellow circles:

- nCtrl1 > QB 304.0, USINT [1.0]
- nCtrl2 > QB 305.0, USINT [1.0]
- nCtrl3 > QB 306.0, USINT [1.0]

The 'Show Variable Types' section on the right has the following settings:

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

The 'Show Variable Types' section has the following settings:

- Matching Type
- Matching Size
- All Types
- Array Mode

The 'Offsets' section has the following settings:

- Continuous
- Ignore Gaps
- Show Dialog

The 'Variable Name / Comment' section has the following settings:

- / Hand over
- / Take over

The 'Variable Size Mismatch' dialog for 'nCtrl1' shows the following settings:

| Field | Value |
|-------------------------|--------|
| Linked Variable: | nCtrl1 |
| Linked Variable: Size | 8 |
| Linked Variable: Offset | 0 |
| Own Variable: Size | 16 |
| Own Variable: Offset | 0 |
| Overlapped: | 8 |

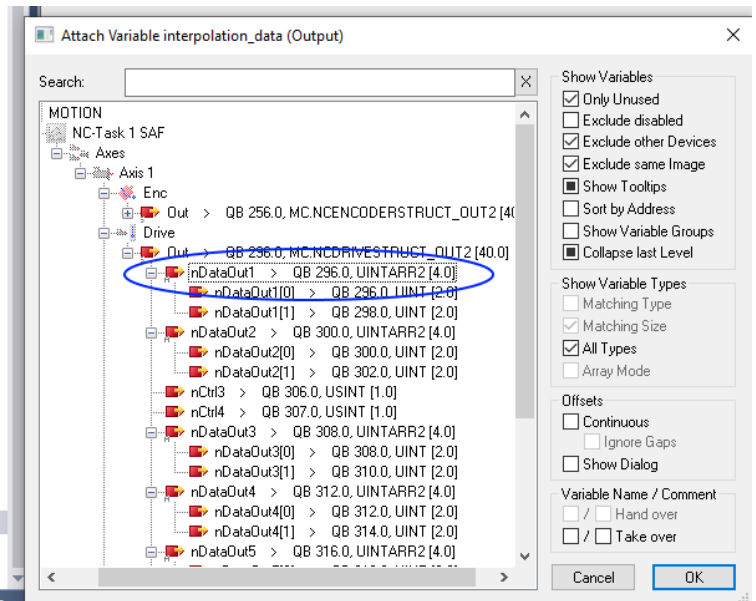
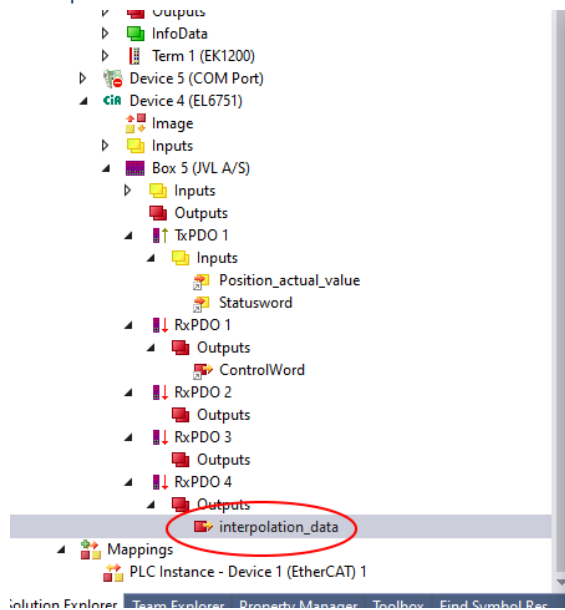
(Size and Offset in bits)

The 'Variable Size Mismatch' dialog for 'nCtrl2' shows the following settings:

| Field | Value |
|-------------------------|--------|
| Linked Variable: | nCtrl2 |
| Linked Variable: Size | 8 |
| Linked Variable: Offset | 0 |
| Own Variable: Size | 16 |
| Own Variable: Offset | 8 |
| Overlapped: | 8 |

(Size and Offset in bits)

Link position from axis to motor

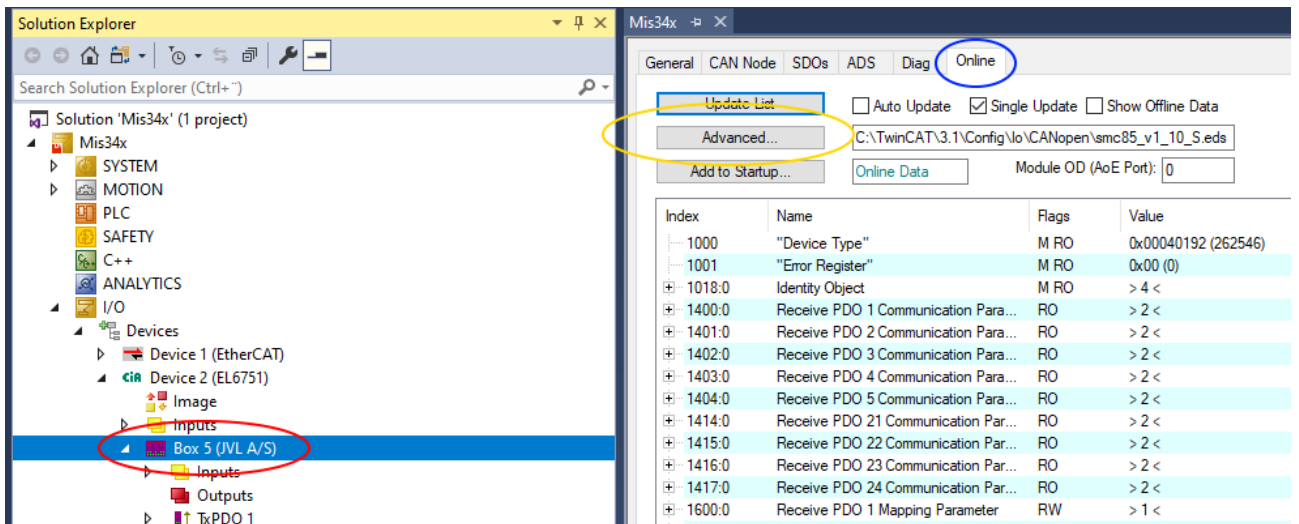


Set SDO data

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:

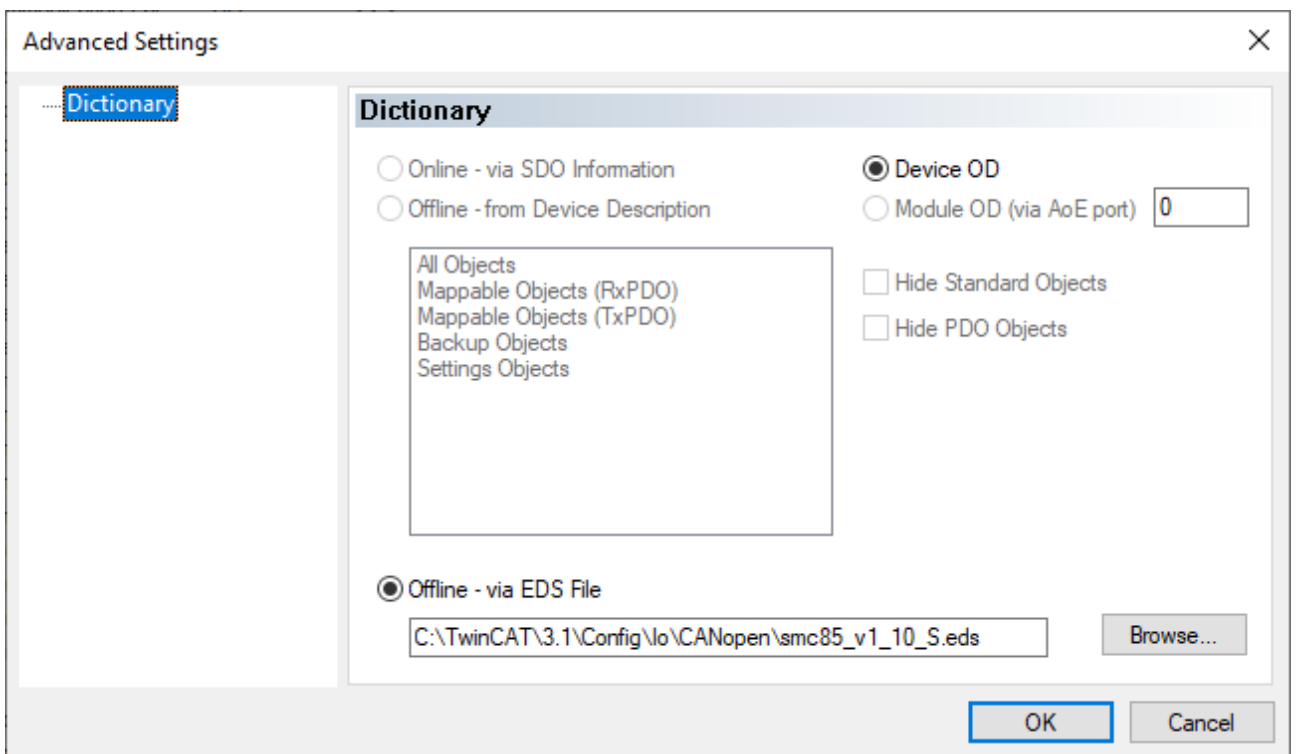
Set motor "mode of operation" to 7 : Interpolated position mode. This is the operation mode the NX axis is using to control the motor. The NC axis expect the motor to be in this operation mode when its available and ready.

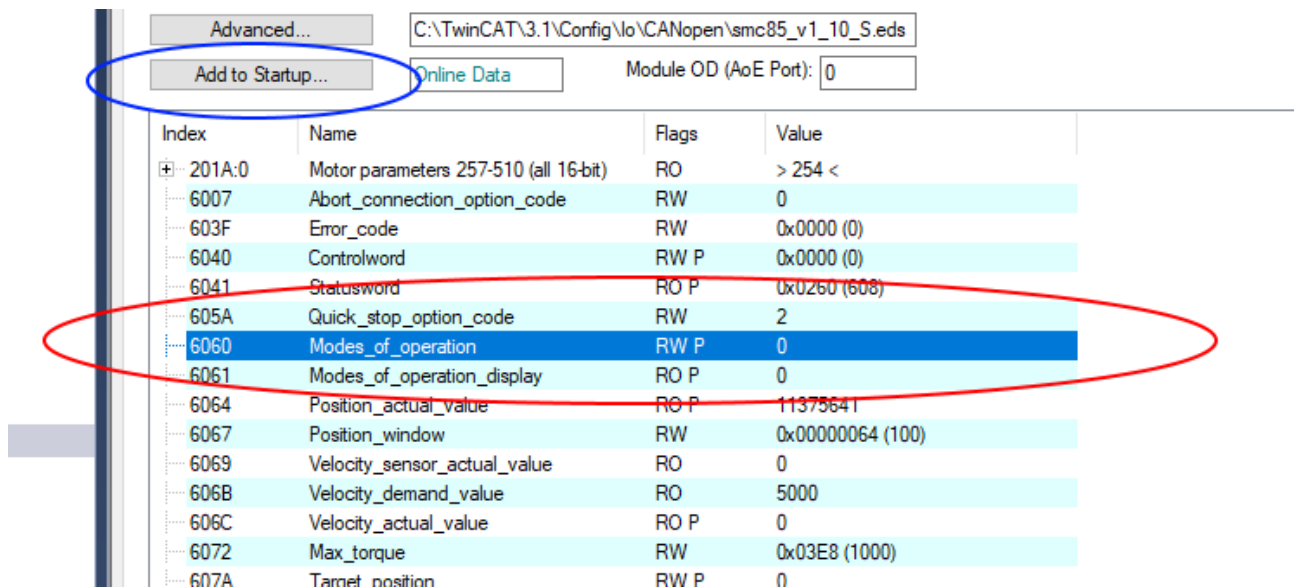


#Click on Motor Box.

#Select the online tab

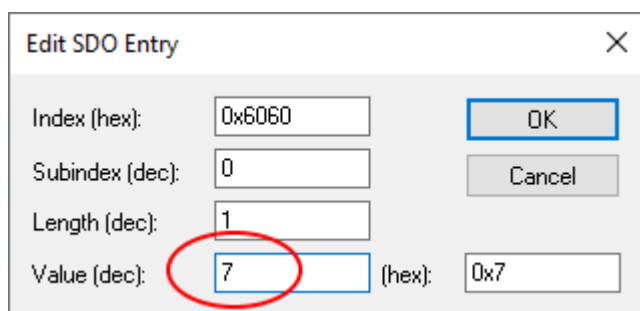
#If list is empty. Click the Advanced button. And select the EDS file manually. (Window below)





Select object 6060 Modes of operation from the list of objects.

Click the button “Add to startup”



Change value to 7 and press OK.

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 02 is used to tell the motor what timing is used in the PLC.

Select the object 60C2:02 interpolated_time_ms

| | | | |
|---------|---------------------------|------|----|
| 60C1:0 | interpolation_data_record | RO | |
| 60C2:0 | interpolation_time_period | RO | |
| 60C2:01 | interpolation_data units | RO | -- |
| 60C2:02 | interpolation_time_ms | RW | -- |
| 60FD | Digital_inputs | RO P | |
| 60FE:0 | Digital_outputs | RO | |
| 60FF | Target velocity | RW P | |

Click the button “Add to startup”

Index (hex): 0x60C2

Subindex (dec): 2

Length (dec): 1

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

OK Cancel

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

| 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 | 2 | 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...

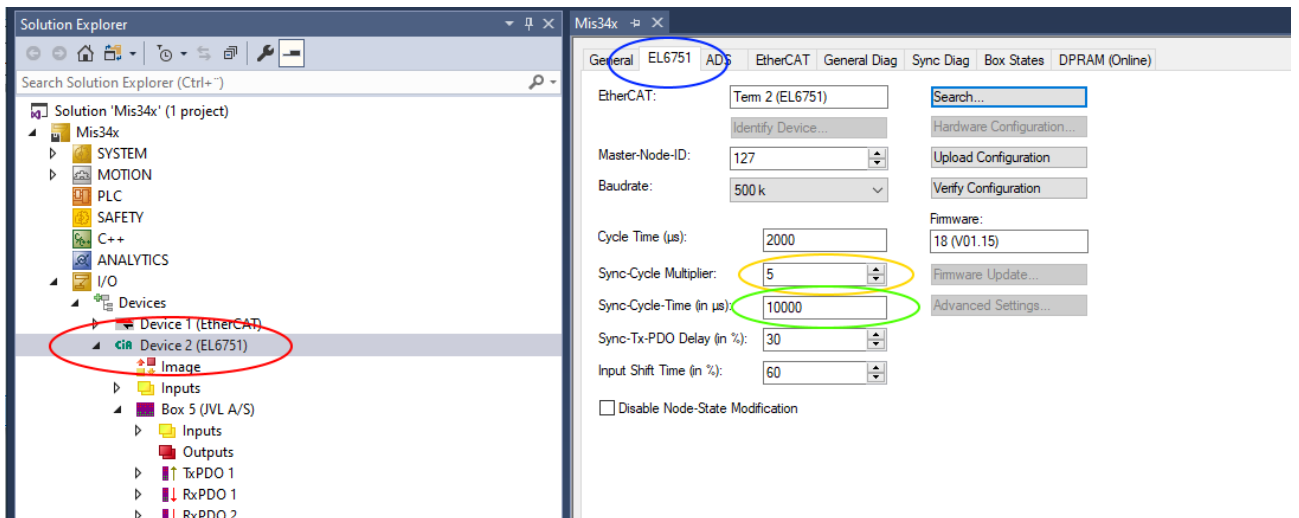
Sync interval

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 60C2:02



#Select the Cia device

#Click the EL6751 (Beckhoff CanOpen master)

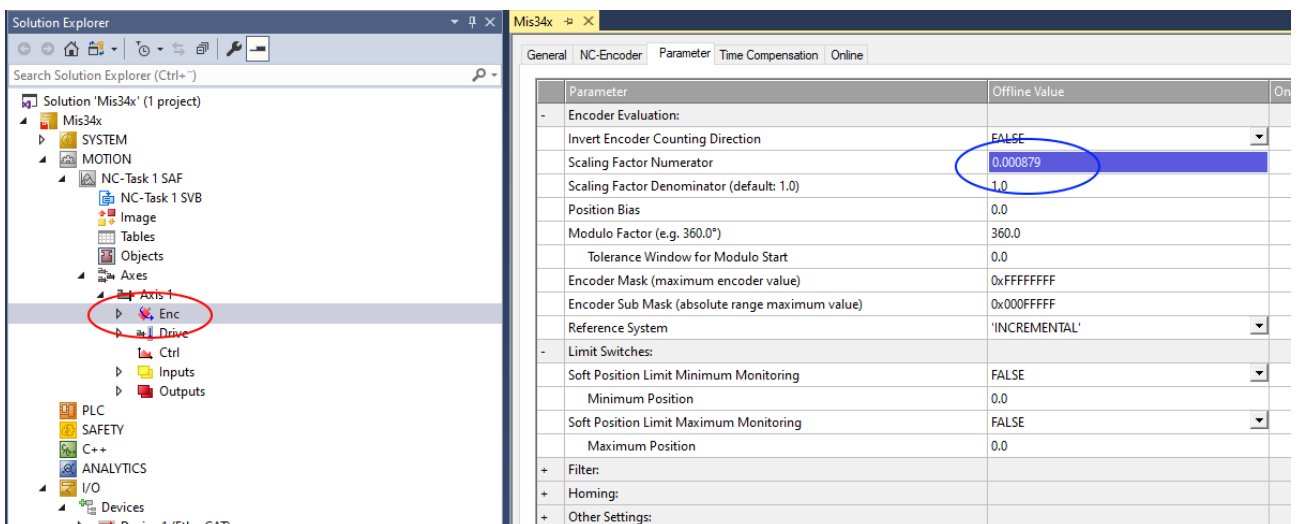
#Adjust the multiplier so that Cycle time times multiplier is the Sync interval.

#Se sync interval updates on next build.

Set gear factor

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

Set scaling factor in degrees for an example.



Select the encoder for the NC axis

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

Sample calculation:

Count pr. revolution = 409600

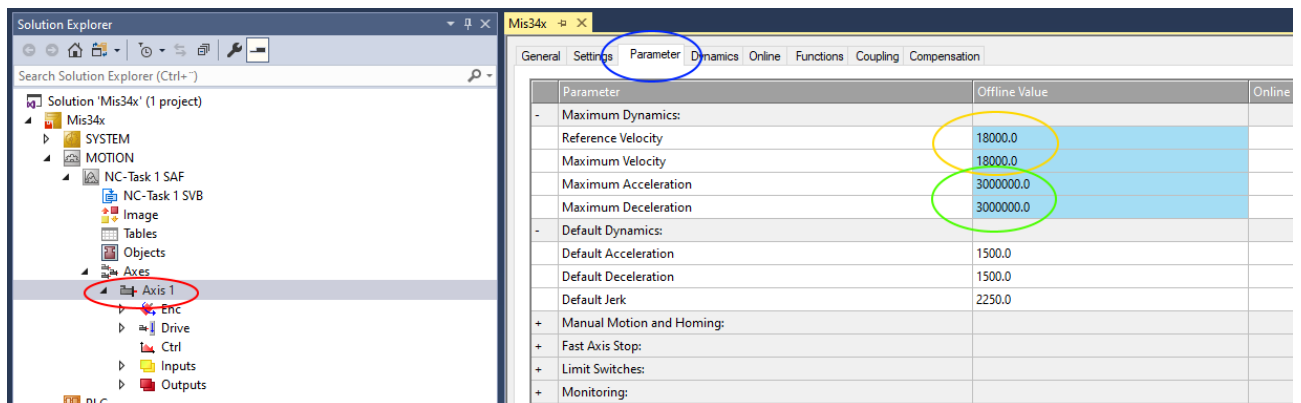
Degrees pr. revolution = 360

Scaling Factor Numerator:

Degrees pr. count = Degrees pr. revolution / Count pr. revolution = $360/409600 = 0,000879$

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

Maximum Velocity #

Enter the calculated values. Sample calculation below.

Sample calculation:

Max rpm = 3000

Seconds pr. minute = 60

Degrees pr. revolution = 360

Degrees pr. second = Max rpm * degrees pr. revolution / Seconds pr. Minute = $3000 * 360 / 60 = 18000$

Max acceleration #

Enter the calculated values. Sample calculation below.

Sample calculation:

Max acceleration rpm= 500000

Seconds pr. minute = 60

Degrees pr. revolution = 360

Degrees in seconds² = Max acceleration rpm * degrees pr. revolution / Seconds pr. Minute = 500000 * 360 / 60= 3000000

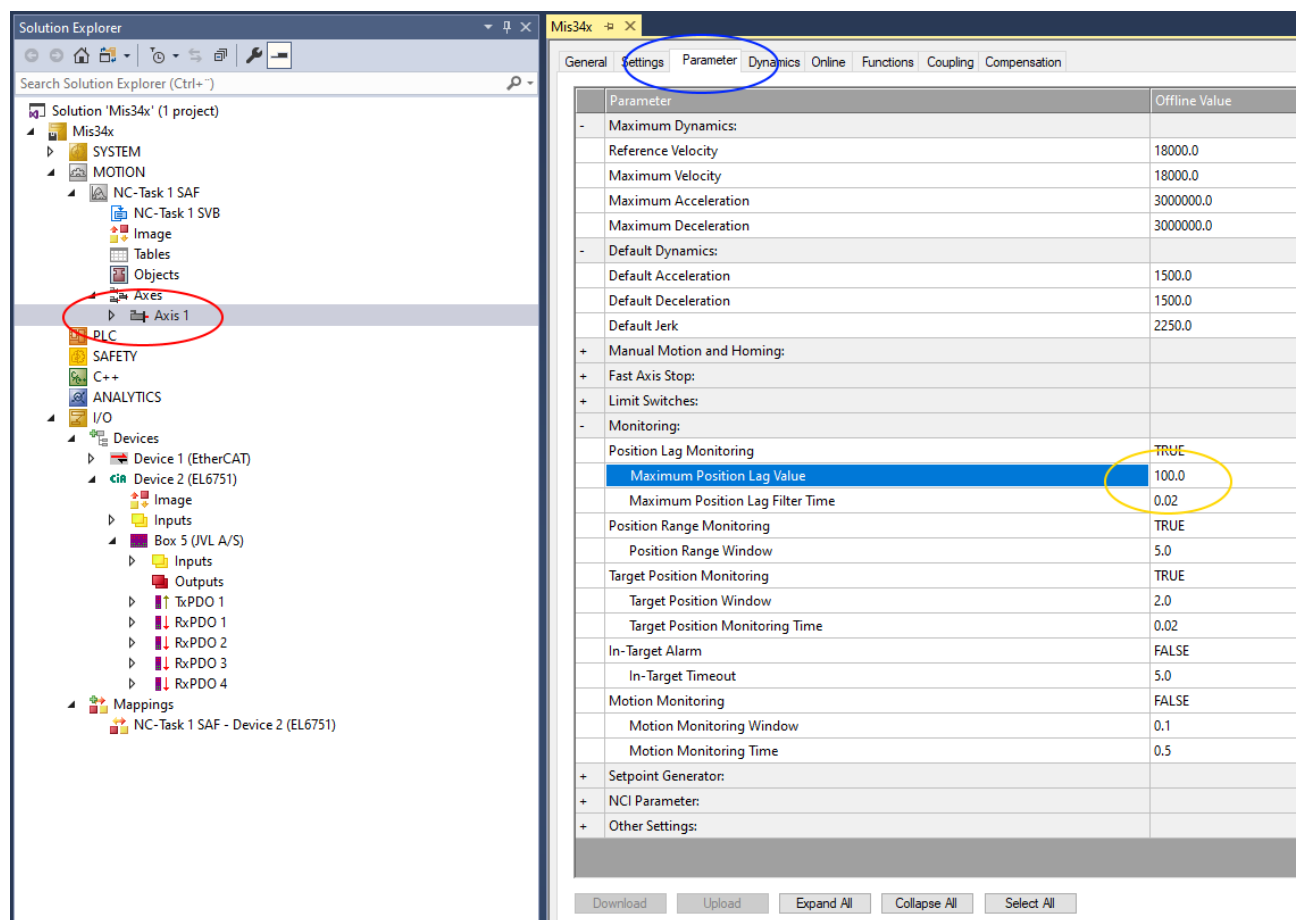
Following error monitoring

The NC axis monitors the following 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.

Its recommended to adjust the following error limit to your suit the application.

TwinCAT call following error “Position lag Value” monitoring.

In this sample it is set to 100degrees.



Select the motor axis

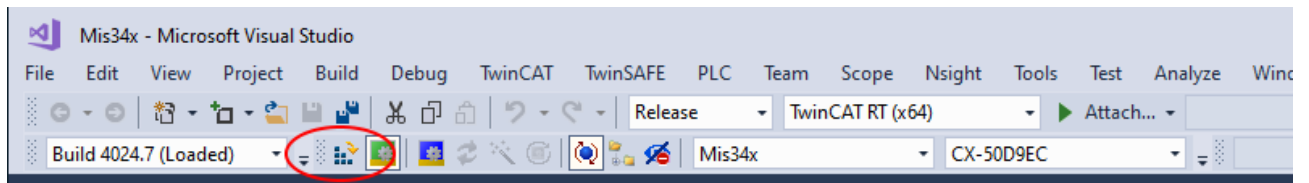
select parameters

enter the following error limit in the “Maximum position Lag value” field.

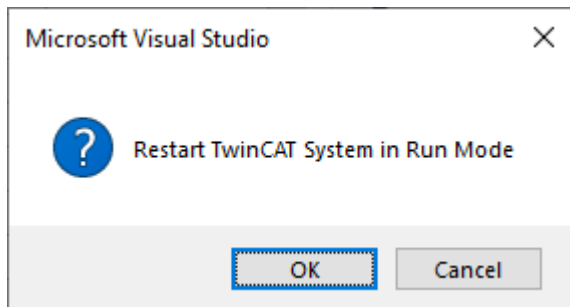
Store settings in PLC

To use the new settings in the PLC project , the configuration need 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"

The NC axis is ready to control your JVL motor.