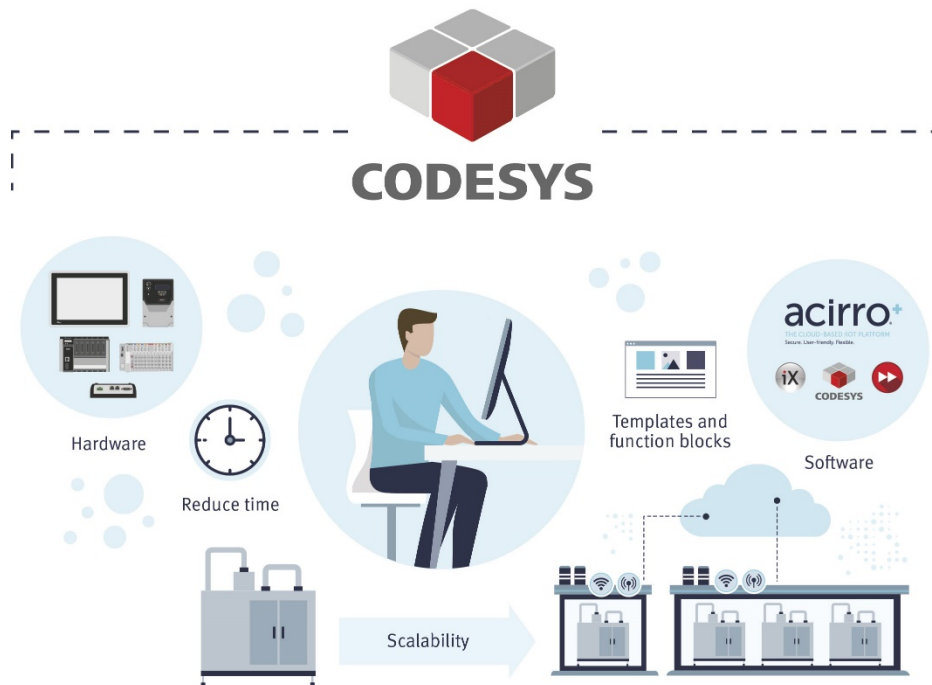


# Quick start guide

## SoftMotion Lite library for X2 control

SER0018 - SoftMotion Lite



## 1 Function and area of use

This document provides guidelines when working with the SoftMotion Lite library for X2 control. This document explains the use of this library, and the contents therein.

## 2 About this document

This quick start document should not be considered as a complete manual. It is an aid to be able to startup a normal application quickly and easily.

### **Copyright © Beijer Electronics, 2022**

*This documentation (below referred to as 'the material') is the property of Beijer Electronics. The holder or user has a non-exclusive right to use the material. The holder is not allowed to distribute the material to anyone outside his/her organization except in cases where the material is part of a system that is supplied by the holder to his/her customer.*

*The material may only be used with products or software supplied by Beijer Electronics. Beijer Electronics assumes no responsibility for any defects in the material, or for any consequences that might arise from the use of the material. It is the responsibility of the holder to ensure that any systems, for whatever applications, which is based on or includes the material (whether in its entirety or in parts), meets the expected properties or functional requirements. Beijer Electronics has no obligation to supply the holder with updated versions.*

Use the following hardware, software, drivers and utilities in order to obtain a stable application:

### **In this document we have used following software and hardware**

- Codesys 3.5.SP13 Patch3 (EtherCAT library >= SP13 essential)
- X2 control (any)

### **For further information refer to**

- [Beijer Electronics knowledge database, HelpOnline](#)

This document and other quick start documents can be obtained from our homepage. Please use the address [support.europe@beijerelectronics.com](mailto:support.europe@beijerelectronics.com) for feedback.

### 3 Table of Contents

1	Function and area of use.....	2
2	About this document.....	2
3	Table of Contents.....	3
4	Motion Lite with X2 control .....	4
5	Preparing your editor .....	4
5.1	<i>Installation of the library to your editor</i> .....	4
5.2	<i>Add the library into your project</i> .....	5
6	EtherCAT configuration .....	6
6.1	<i>Adding the EtherCAT master to the X2 control</i> .....	6
6.2	<i>Adding drives to the EtherCAT network</i> .....	7
7	Programming.....	9
7.1	<i>Enabling a servo</i> .....	9
7.2	<i>Homing an axis</i> .....	10
7.3	<i>Bump-homing</i> .....	11
7.4	<i>Position Moves</i> .....	12
7.5	<i>Speed Moves</i> .....	13
7.6	<i>Torque Control</i> .....	14
7.7	<i>Access the drive's I/O</i> .....	15
7.8	<i>Access the drive's data</i> .....	17
7.9	<i>Profiling</i> .....	18
7.10	<i>Servo direction</i> .....	20
7.11	<i>Stopping a drive</i> .....	20
7.12	<i>Interrupt feeding (cut-to-length)</i> .....	21
7.13	<i>Touchprobe (AKA registration input)</i> .....	22
8	Recommendations .....	24
9	About Beijer Electronics .....	26
9.1	<i>Contact us</i> .....	26

## 4 Motion Lite with X2 control

This library takes advantage of two things:

1. The EtherCAT master available in the X2 control
2. The Bejer range of servo drives (L7NH) all support 'Profile' Modes in addition to 'Cyclic Synchronous' modes. This is not the case for all drives.

With this library, servo control can be easily achieved with little or no motion knowledge. It's not intended to replace Bejer's full motion offering, which is the full CODESYS motion library.

This library is not to be confused with the SoftMotionLight SL library available from the CODESYS store.

For synchronised motion, Bejer provide the BoX2 pro motion which is equipped with a distributed clock for deterministic EtherCAT operation and the full CODESYS SoftMotion licence with PLCOpen function blocks. This is out of the scope of this SER/manual

## 5 Preparing your editor

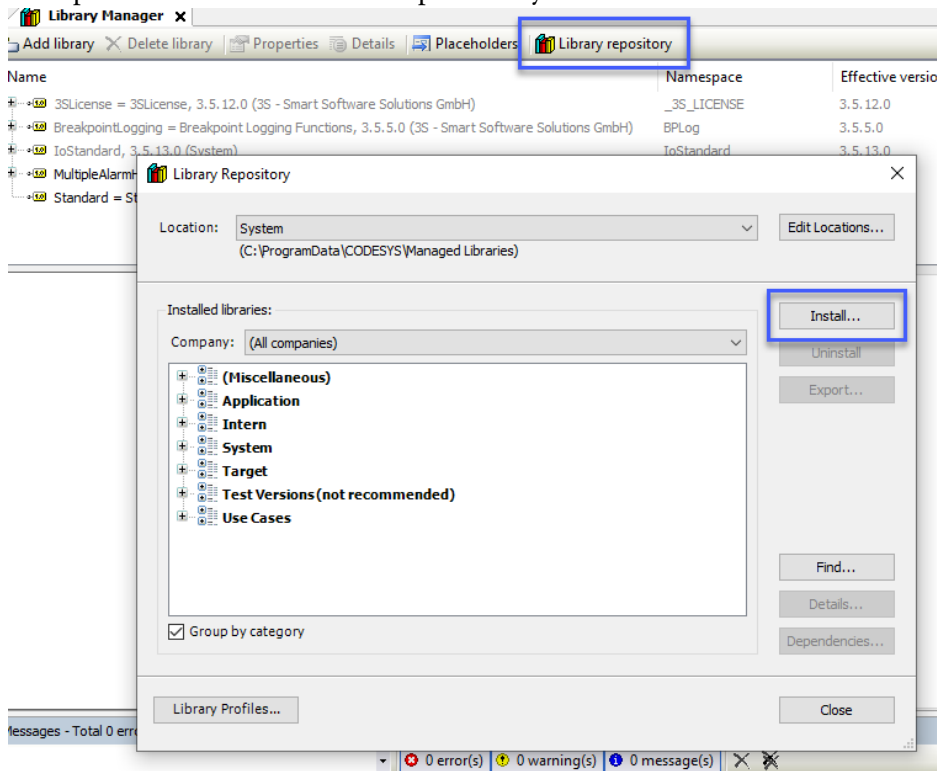
The following chapter describes important procedures and settings needed for a well- functioning system.

### 5.1 Installation of the library to your editor

The \*.compiled-library needs to be made available in your system so it can be included in projects. This is done by accessing the 'Library Manager' → 'Library Repository' then 'Install'

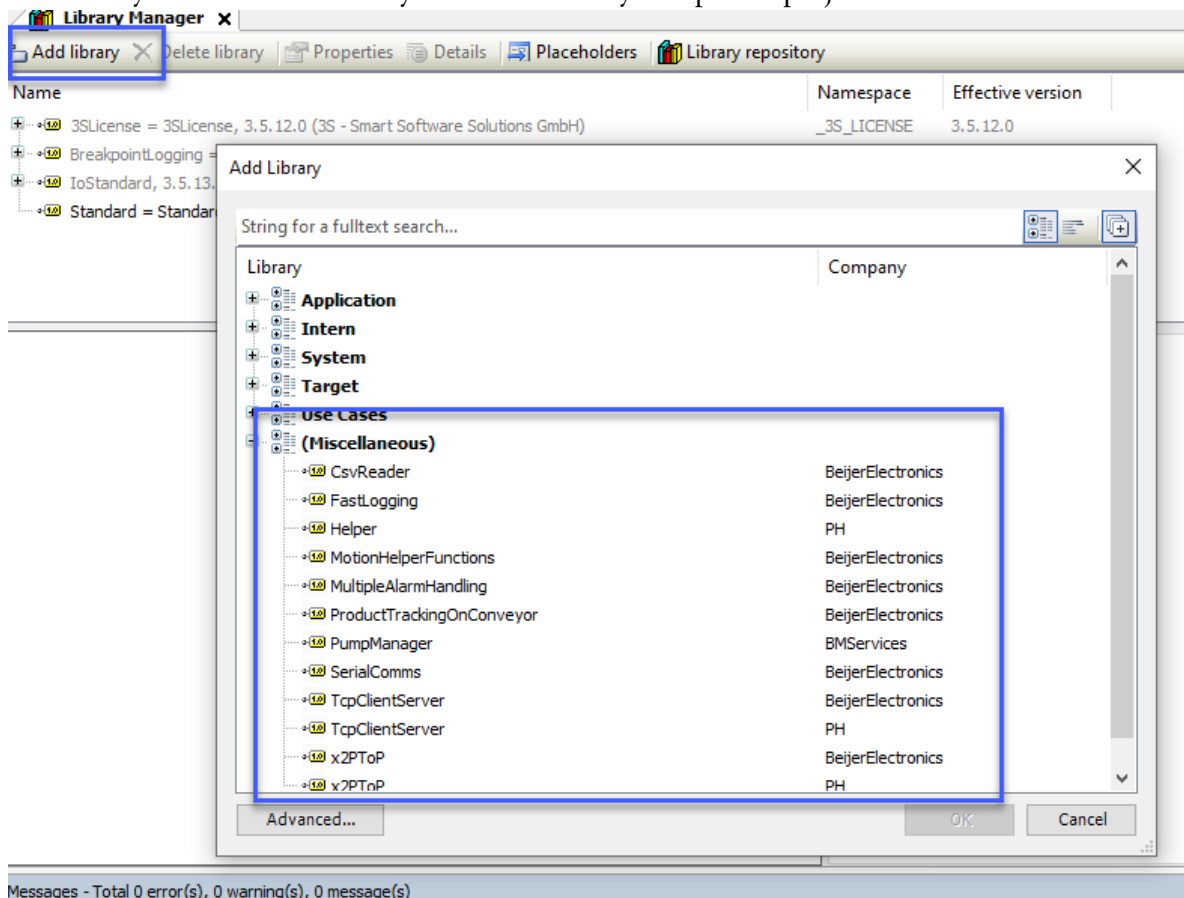
Navigate to the folder where you have put the \*.compiled-library.

This procedure will need to be repeated if you use a new PC.



## 5.2 Add the library into your project

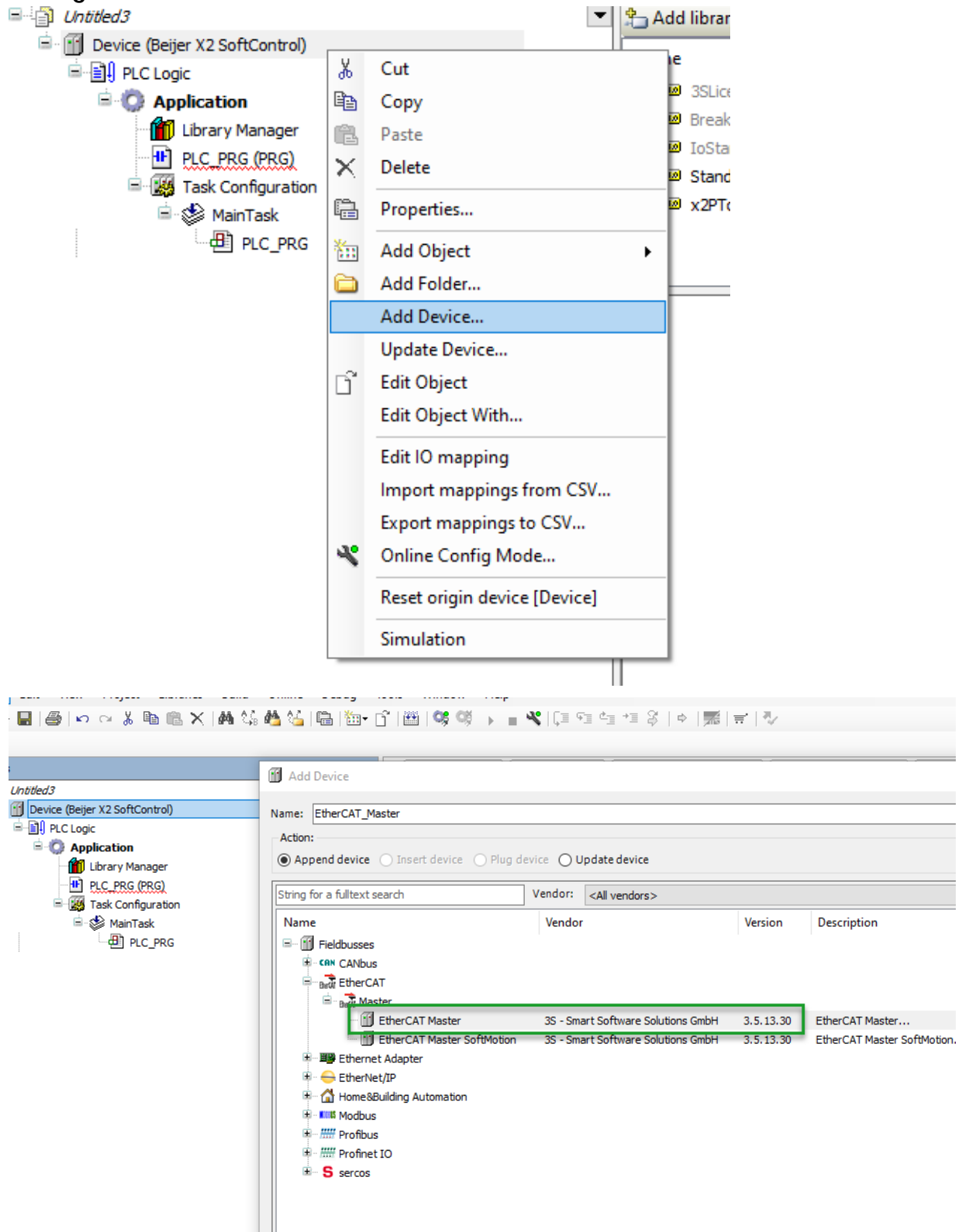
The library is now available for you to include in your specific project.



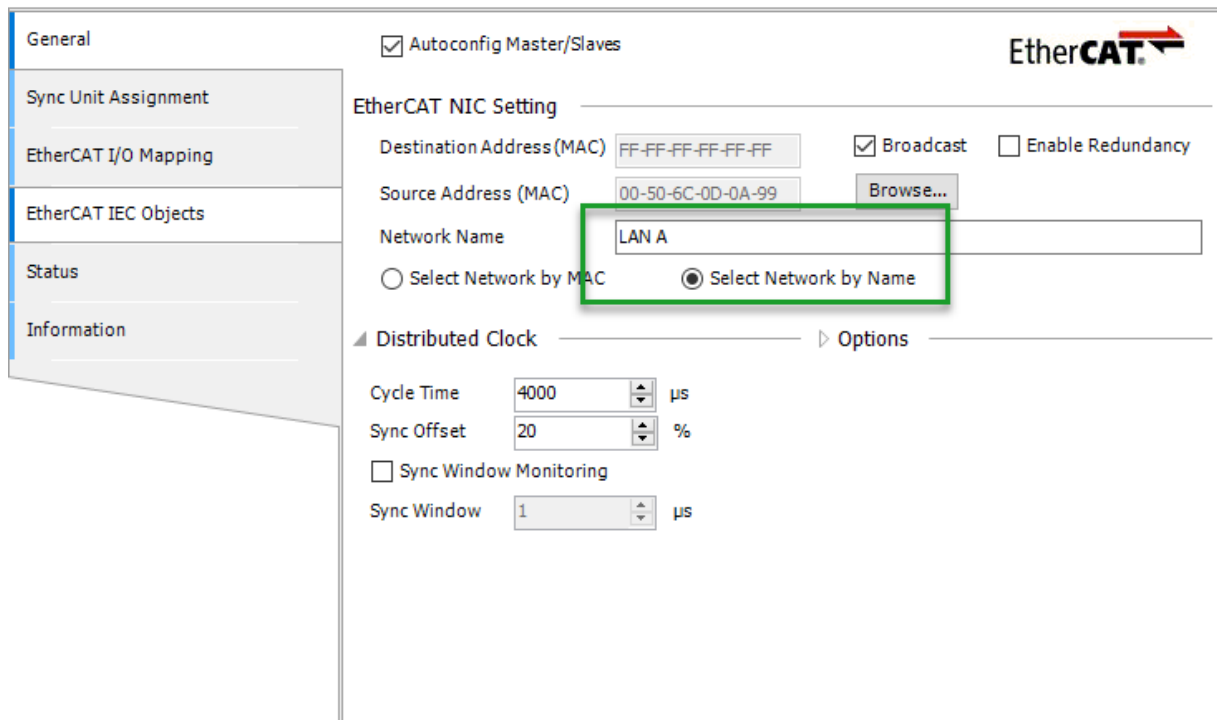
The selected library is now visible in the Library Manager. Its public objects and supplementary help is available here.

## 6 EtherCAT configuration

### 6.1 Adding the EtherCAT master to the X2 control



Within the settings of this master, let's make the following settings:



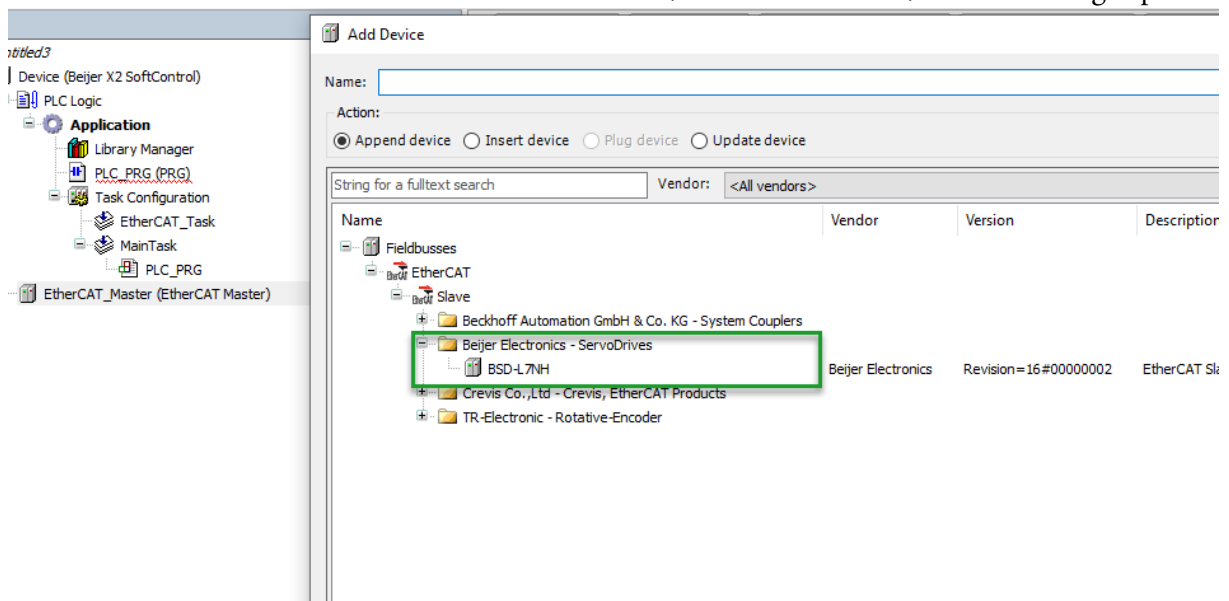
Using Autoconfig Master/Slaves in mandatory

Using LAN A for the EtherCAT network is recommended although not essential; both LAN A and LAN B work. Simply select **LAN B** in this dialog instead.

## 6.2 Adding drives to the EtherCAT network

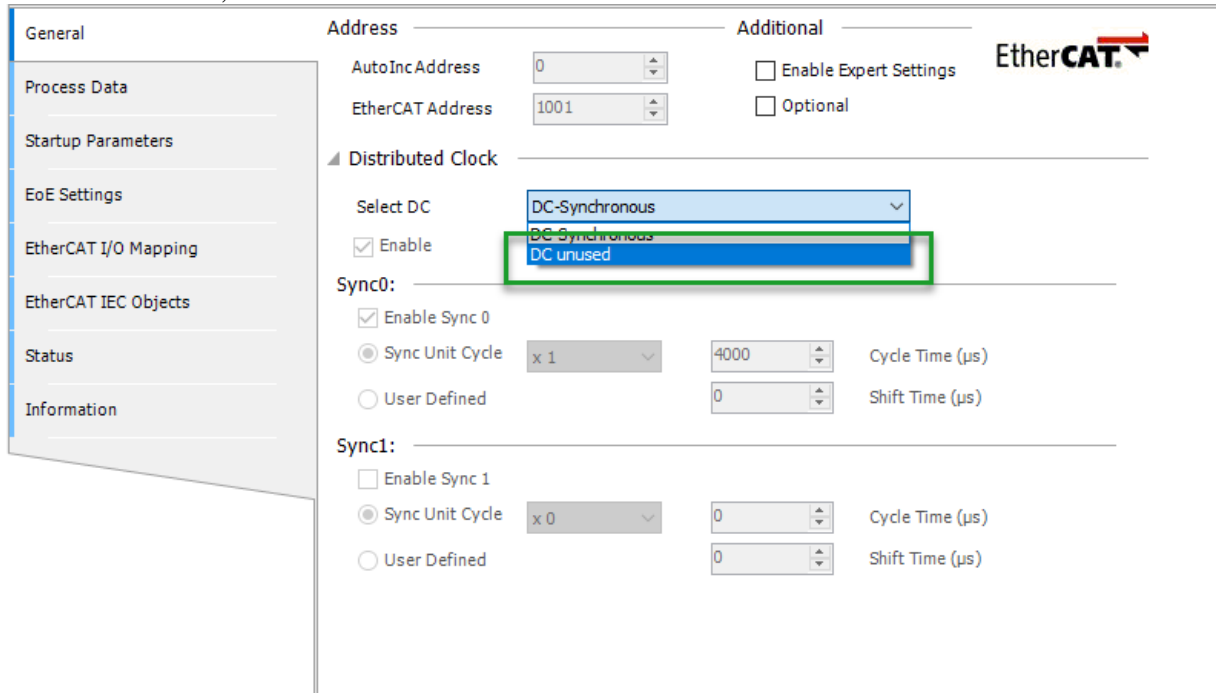
We now have an EtherCAT master, and EtherCAT slaves can be added to it. This device description is available at our website, and is included in the library. Installation of device descriptions is not included in this procedure.

From the context menu of the new EtherCAT master, choose add device, and this dialog is presented:

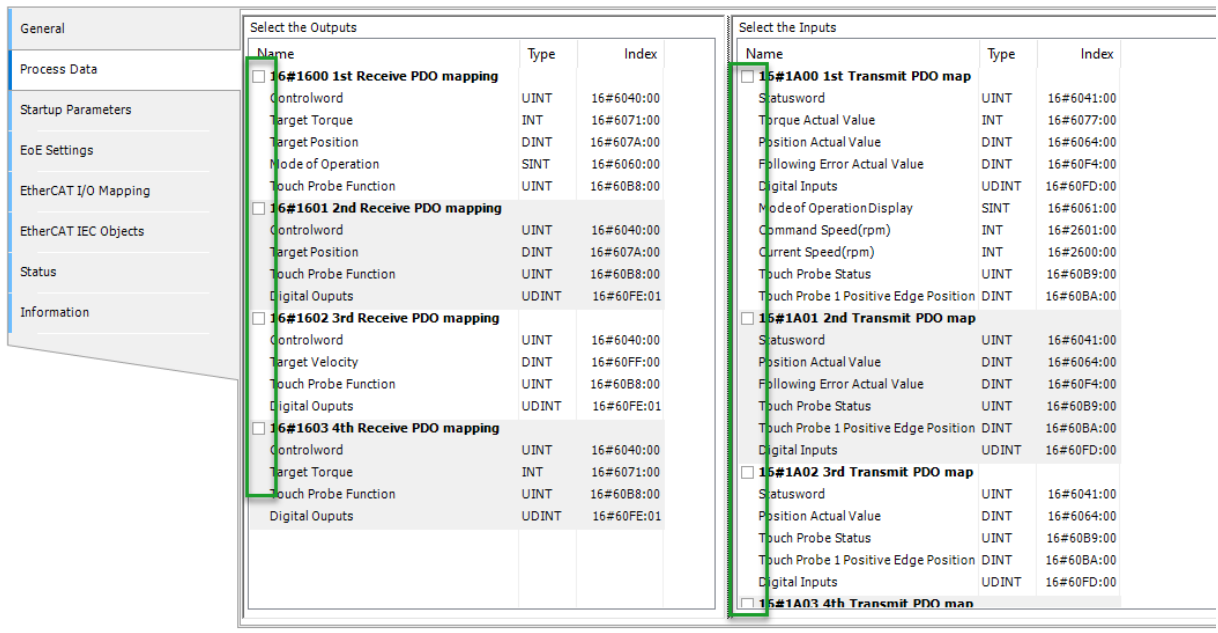


Add the drive and rename to suit your application. Standard symbol-naming rules apply.

The device description is generic, so we need to make some changes. X2 control is not equipped with a distributed clock, so let's disable this:



All EtherCAT transactions are managed by the library, so disable all PDO communications. Expect the unexpected otherwise.



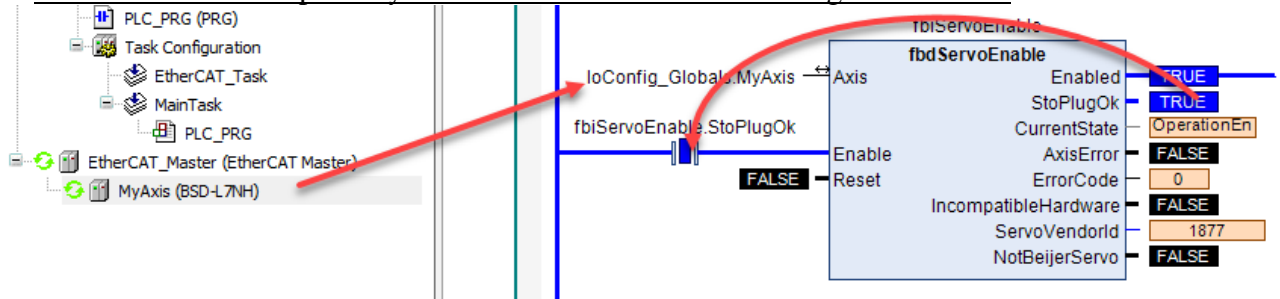
You are now at a stage to start programming.



## 7 Programming

### 7.1 Enabling a servo

Each drive needs the Function Block definition **fbdServoEnable** invoking. The drive's **Enabled** status is shared transparently to all other function blocks acting on this drive



Enabling the drive when the drive's STO is healthy puts the safety squarely in the Safety Programmer's realm. If the STO is healthy, the drive *could* move the motor.

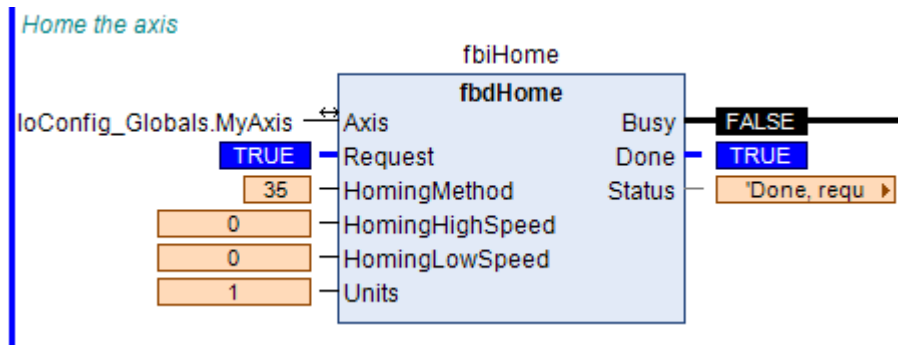
Name	Scope	Type	Description
EtherCATSlave	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Enable	VAR_IN	BOOL	The drive will be enabled whilst high
Reset	VAR_IN	BOOL	Edge-detected reset pulse to the drive
Enabled	VAR_OUT	BOOL	Reflects the enabled status of the drive (in Run mode)
StoPlugOk	VAR_OUT	BOOL	Whether the drive's STO input is healthy
CurrentState	VAR_OUT	eStateMachine	Copy of the CiA402 status
AxisError	VAR_OUT	BOOL	True if the drive is reporting an error
ErrorCode	VAR_OUT	WORD	The drive's error code, Refer to the L7NH manual
IncompatibleHardware	VAR_OUT	BOOL	Not X2Control hardware
ServoVendorId	VAR_OUT	DWORD	Servo VendorId
NotBeijerServo	VAR_OUT	BOOL	Connected servo is not from Beijer. Functionality is limited

Now you can add any of the other function blocks that your application requires.

## 7.2 Homing an axis

**fbdHome** is your friend when you need to home an axis.

The homing methods are defined in the L7NH servo manual ([L7NHA manual](#))

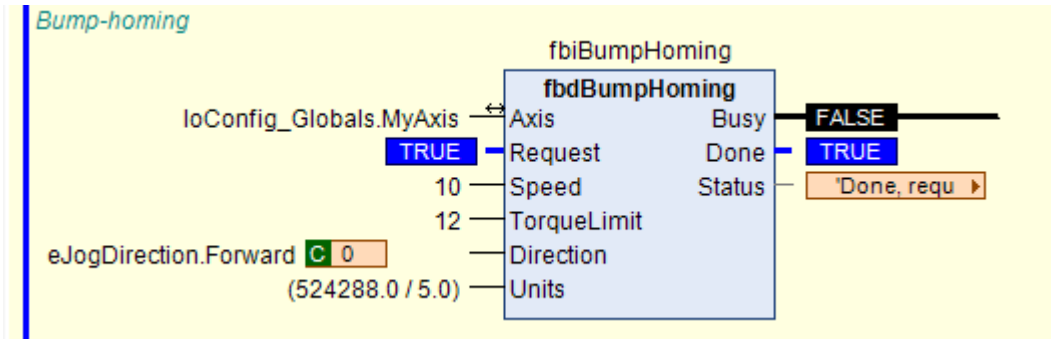


Name	Scope	Type	Description
EtherCATSlave	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Request	VAR_IN	BOOL	Performs the required homing method. Edge-detected internally
HomingMethod	VAR_IN	SINT	The homing method defined in CiA402
HomingHighSpeed	VAR_IN	REAL	Some homing methods required a high-speed setting. Entered in user-units
HomingLowSpeed	VAR_IN	REAL	Some homing methods required a low-speed setting. Entered in user-units
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Busy	VAR_OUT	BOOL	High whilst finding the home position
Done	VAR_OUT	BOOL	High for at least one scan after the homing has completed
Status	VAR_OUT	STRING	Plain text status of the block's progress

### 7.3 Bump-homing

**fbdBumpHome** offers an alternative to **fbdHome**. Some axes can be homed by performing a torque-limited speed move to a solid object.

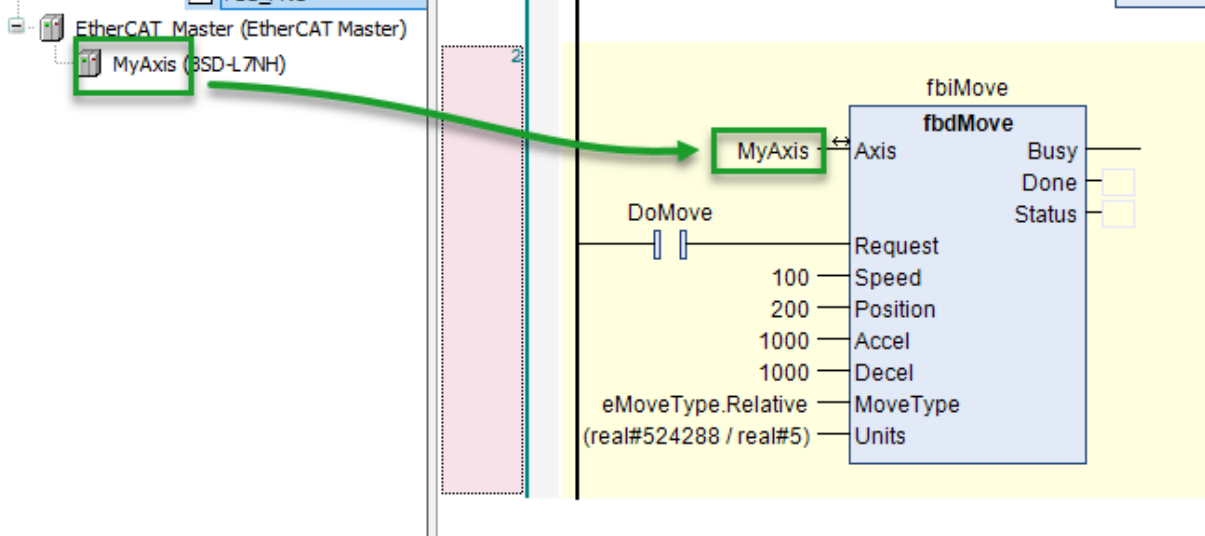
This position is defined as home (position zero). Not all axes are suitable for this method.



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Request	VAR_IN	BOOL	Initiate homing on the rising edge
Speed	VAR_IN	REAL	User-units
TorqueLimit	VAR_IN	REAL	Percent of motor's maximum torque
Direction	VAR_IN	eDirection	Which direction
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Busy	VAR_OUT	BOOL	Homing in progress
Done	VAR_OUT	BOOL	High for at least one scan after successful completion
Status	VAR_OUT	STRING	Plain text status of the block's progress

### 7.4 Position Moves

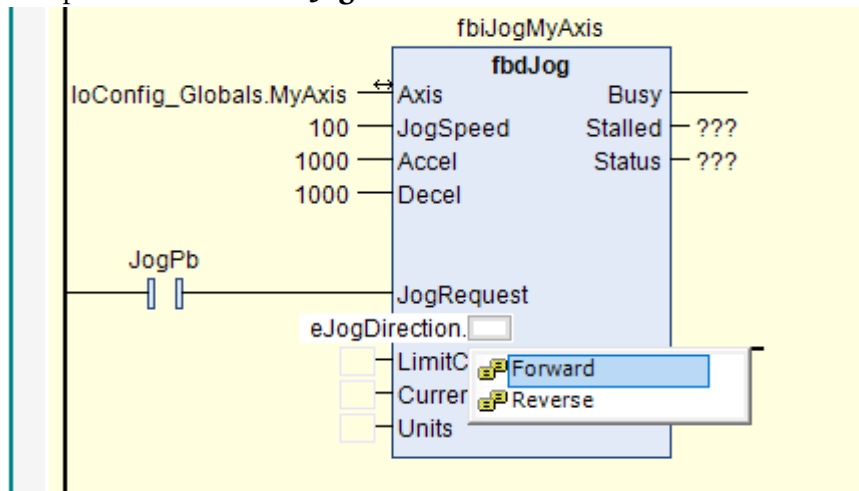
For moves (Absolute or Relative), use **fbdMove**



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Request	VAR_IN	BOOL	Initiates a move on the rising edge
Speed	VAR_IN	REAL	Command speed, in user-units/second
Position	VAR_IN	REAL	Command position in user-units
Accel	VAR_IN	REAL	Command acceleration in user-units/second/second
Decel	VAR_IN	REAL	Command deceleration in user-units/second/second
MoveType	VAR_IN	eMoveType	Enumeration to determine if the move is Absolute, Relative, or BlendedAbsolute
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Busy	VAR_OUT	BOOL	High whilst the move is in progress.
Done	VAR_OUT	BOOL	High for at least one scan after the move has completed/ Qualified with the drive reporting in-position
Status	VAR_OUT	STRING	Plain text status of the block's progress

## 7.5 Speed Moves

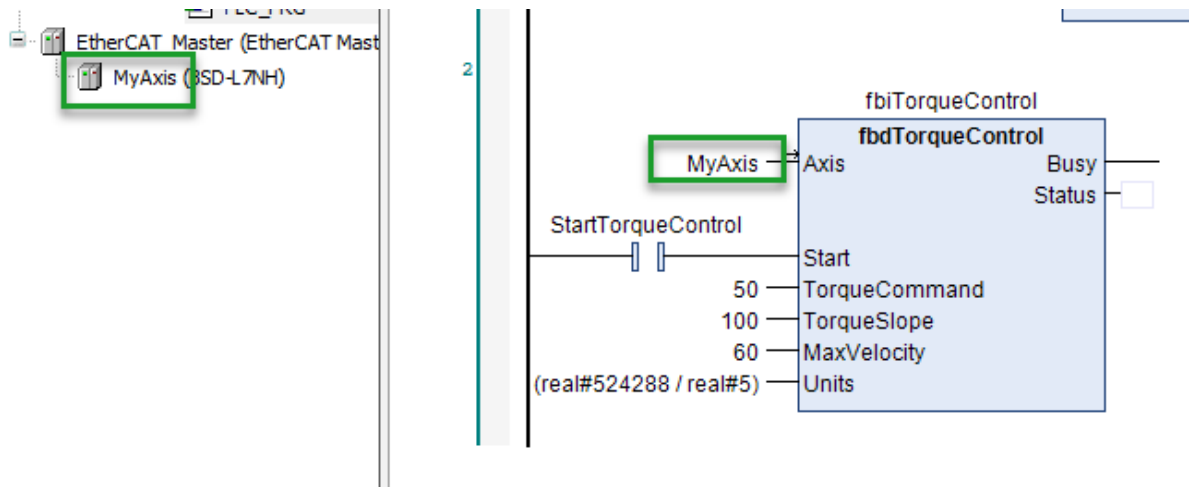
For speed moves, use **fbJog**



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
JogSpeed	VAR_IN	REAL	Command speed, in user-units/second
Accel	VAR_IN	REAL	Command acceleration in user-units/second/second
Decel	VAR_IN	REAL	Command deceleration in user-units/second/second
JogRequest	VAR_IN	BOOL	Runs in speed mode as long as this input is high
Direction	VAR_IN	eDirection	Determines the jog direction
LimitCurrent	VAR_IN	BOOL	If high, the drive runs with a current limit set in the parameter <b>CurrentLimit</b>
CurrentLimit	VAR_IN	REAL	Determines the deliverable torque. Measured as a percentage of the motor's maximum. Relevant if <b>LimitCurrent</b> input is high
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Busy	VAR_OUT	BOOL	High during a speed move
Stalled	VAR_OUT	BOOL	High if the drive has reached the maximum allowable torque
Status	VAR_OUT	STRING	Plain text

## 7.6 Torque Control

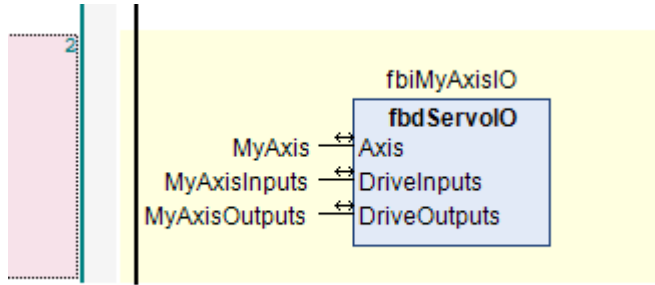
For torque moves, use **fbdTorque**. This will run the axis with a torque command. The axis' maximum speed will be governed by the MaxVelocity input, which is user units.



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Start	VAR_IN	BOOL	Initiates a torque profile
TorqueCommand	VAR_IN	REAL	Percentage of the motor's maximum the drive will deliver
TorqueSlope	VAR_IN	REAL	Percentage per second increase to reach the torque command
MaxVelocity	VAR_IN	REAL	Maximum achievable speed, in user-units/seconds
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Busy	VAR_OUT	BOOL	High whilst the drive is in torque mode
Status	VAR_OUT	STRING	Plain text

### 7.7 Access the drive's I/O

To access the drive's IO, uses **fbdServoIO**



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
DriveInputs	VAR_IN_OUT	s0x211F	Structure containing the drive's inputs
DriveOutputs	VAR_IN_OUT	s0x2121	Structure containing the drive's outputs

Device.Application.Gv			
Expression	Type	Value	Prepared
[-] MyAxisInputs	s0x211F		
PosOt	BIT	FALSE	
NegOt	BIT	FALSE	
Home	BIT	FALSE	
Stop	BIT	FALSE	
PCon	BIT	FALSE	
Gain2	BIT	FALSE	
PCL	BIT	FALSE	
NCL	BIT	FALSE	
Probe1	BIT	FALSE	
Probe2	BIT	FALSE	
Emg	BIT	FALSE	
ARst	BIT	FALSE	
ServoOn	BIT	FALSE	
Lvsf1	BIT	FALSE	
Lvsf2	BIT	FALSE	
Bit15	BIT	FALSE	
[-] MyAxisOutputs	s0x2121		
Brake	BIT	FALSE	
Alarm	BIT	FALSE	
Ready	BIT	TRUE	
ZSpd	BIT	TRUE	
InPos1	BIT	TRUE	
TLimit	BIT	FALSE	
VLimit	BIT	FALSE	
InSpd	BIT	TRUE	
Warn	BIT	FALSE	
TGOn	BIT	FALSE	
InPos2	BIT	TRUE	
Bit11	BIT	FALSE	
Bit12	BIT	FALSE	
Bit13	BIT	FALSE	
Bit14	BIT	FALSE	
Bit15	BIT	TRUE	

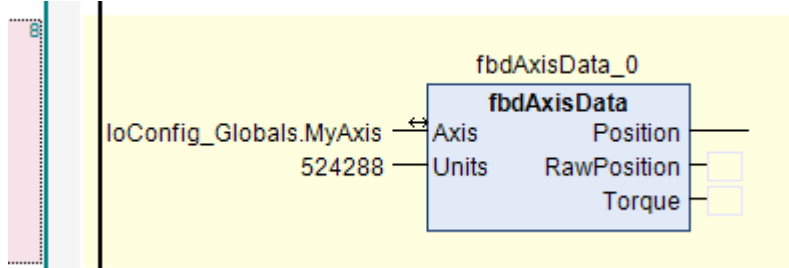
write (only)  
to the drive

read (only)  
from the  
drive



### 7.8 Access the drive's data

The axis' data (position and torque demand) can be read with the **fbdAxisData** block. Create one instance per drive.



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Position	VAR_OUT	REAL	Current position in user-units
RawPosition	VAR_OUT	DINT	Current position in encoder edges
Torque	VAR_OUT	REAL	Percentage of motor's maximum torque being delivered

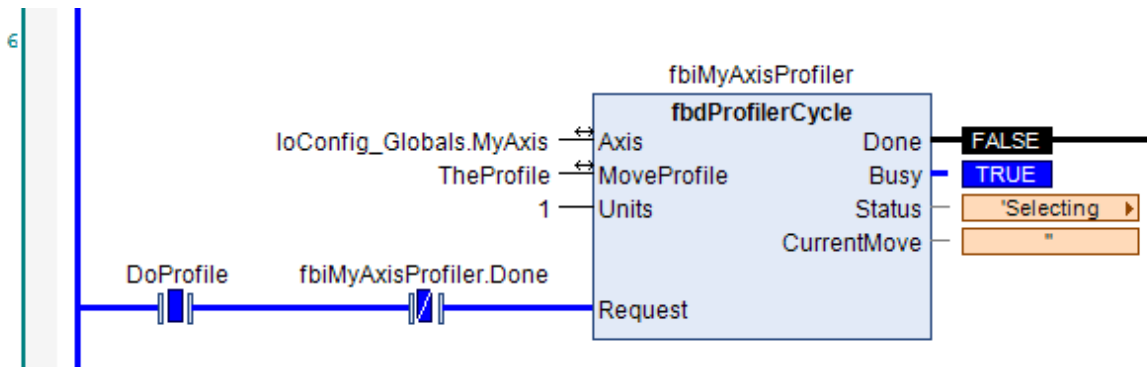
## 7.9 Profiling

Profiles can be achieved with the **fbdProfiler** function block. This will cycle through a series of moves. At the end of a move, the move can either

1. Dwell
2. Wait for another trigger
3. Blend moves together

These selections are made via ENUMs

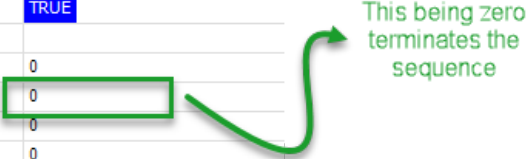
Some motion understanding is required. Use meaningful speeds and accelerations/decelerations to avoid unexpected results, especially with the **Blend** option.



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
MoveProfile	VAR_IN_OUT	ARRAY[0..9] of sProfile	The parameters for the 10 moves. See below
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Request	VAR_IN	BOOL	Initate the profile on the rising-edge
Done	VAR_OUT	BOOL	High for at least one scan after the profile has completed
Busy	VAR_OUT	BOOL	High whilst busy
Status	VAR_OUT	STRING	
CurrentMove	VAR_OUT	STRING	The name of the current move

The **sProfile** datatype is ARRAYed and passed to the block. Its structure is like this:

TheProfile		ARRAY [0..9] OF sP...
TheProfile[0]		sProfile
Position	REAL	1000
Speed	REAL	500
Acceleration	REAL	5000
Deceleration	REAL	5000
Next	ENEXTMOVEOPTION	Dwell
Dwell	REAL	3
Description	STRING	'Move to pick postion'
InProgress	BOOL	FALSE
TheProfile[1]		sProfile
Position	REAL	0
Speed	REAL	500
Acceleration	REAL	5000
Deceleration	REAL	5000
Next	ENEXTMOVEOPTION	Dwell
Dwell	REAL	0
Description	STRING	'Move to place'
InProgress	BOOL	TRUE
TheProfile[2]		sProfile
Position	REAL	0
Speed	REAL	0
Acceleration	REAL	0
Deceleration	REAL	0
Next	ENEXTMOVEOPTION	Dwell
Dwell	REAL	0
Description	STRING	"
InProgress	BOOL	FALSE

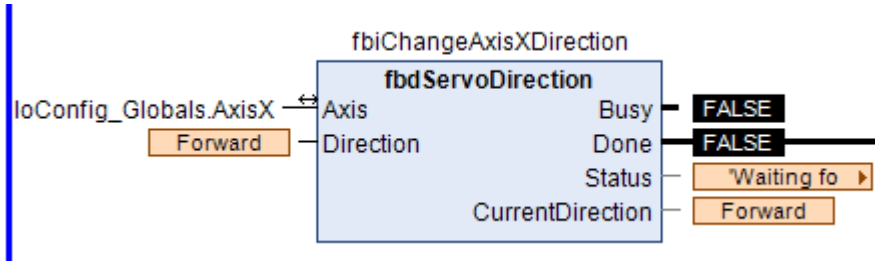


The sequence terminates when the *next* speed is seen to be zero or all 10 moves have completed  
 All positions in are in absolute co-ordinates.

### 7.10 Servo direction

A servo forward direction is the shaft rotates anti-clockwise (looking at the shaft). It's sometime required to change this so the drive interprets a forward command to rotate clockwise. BSDTools can be used to make this change, but the FB **fbdServoDirection** can make the change for you.

If the **Direction** input is different to the **CurrentDirection**, the new direction is written. A drive reboot is required to effect the change.



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Direction	VAR_IN	eDirection	Select forward or reverse from the ENUM
Busy	VAR_OUT	BOOL	Indicates the block is processing. The block executes asynchronously.
CurrentDirection	VAR_OUT	eDirection	The drive's current direction
Done	VAR_OUT	BOOL	Indicates the block has finished processing. This flag stays high for 2 seconds
Status	VAR_OUT	STRING	Plain text description of the block's progress

### 7.11 Stopping a drive

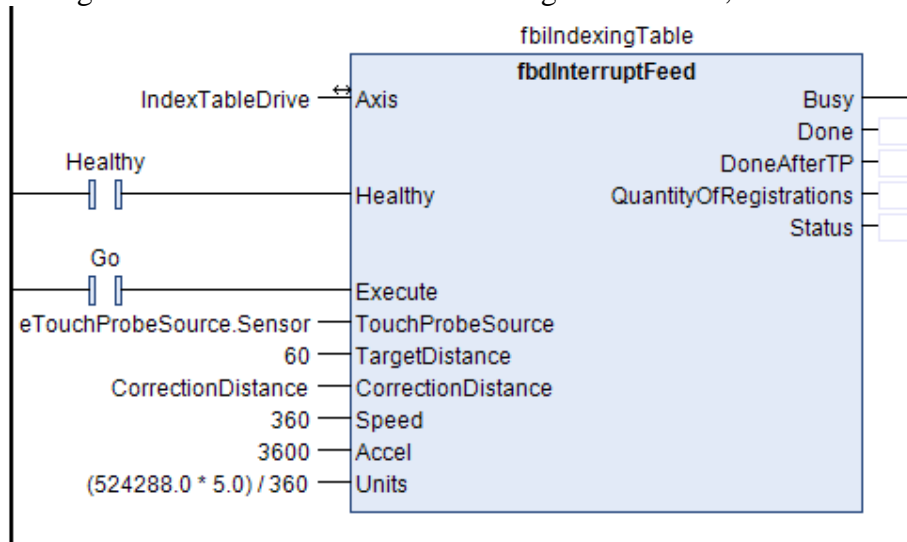
Use **fbdHalt**.

## 7.12 Interrupt feeding (cut-to-length)

The axis can be positioned at a set distance from a registration mark, for example a cut-to-length operation.

The block is given the 'ideal' distance to travel (so missed registration marks are handled).

If a registration mark is encountered during this distance, the move is modified.



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Healthy	VAR_IN	BOOL	
Go	VAR_IN	BOOL	A rising edge triggers the move
TouchProbeSource	VAR_IN	eTouchProbeSource	Z phase of the motor or a physical sensor
TargetDistance	VAR_IN	REAL	Theoretical perfect index
CorrectionDistance	VAR_IN	REAL	Distance to move after the registration mark
Speed	VAR_IN	REAL	Speed in user-units/second
Accel	VAR_IN	REAL	Command acceleration in user-units/second/second
Status	VAR_OUT	STRING	
Done	VAR_OUT	BOOL	Indicates completion of the operation. High for at least one scan
DoneAfterTP	VAR_OUT	BOOL	Indicates completion after a modified move. High for at least one scan
QuantityOfRegistrations	VAR_OUT	UDINT	For information
RegPosRaw	VAR_OUT	DINT	Captured position in encoder edges

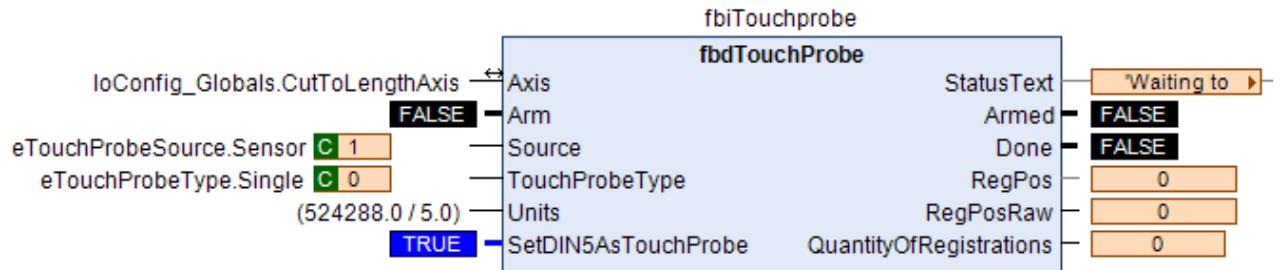
## 7.13 Touchprobe (AKA registration input)

To use the servo's touchprobe inputs (AKA registration inputs), the block fbdTouchprobe is included.

The touchprobe source can be either the Z-phase, or an input wired directly to the drive. This is selected with an ENUM.

The touchprobe can be programmed to repeat automatically or wait for re-arming. Again, this is selected via an ENUM.

The drive's touchprobe position (DINT) is converted to a REAL, there WILL BE inaccuracy at high values.



Name	Scope	Type	Description
Axis	VAR_IN_OUT	ETCSlave	Enter the name you have given to the drive in the etherCAT tree
Arm	VAR_IN	BOOL	In 'Continuous', registration is armed whilst this flag is high. In 'Single', this flag must toggle false-to-true to re-arm
Units	VAR_IN	REAL	Conversion between encoder edges and user-units
Source	VAR_IN	eTouchProbeSource	Z phase or a sensor
TouchProbeType	VAR_IN	eTouchProbeType	Single shot or continuous
SetDIN5AsTouchProbe	VAR_IN	BOOL	This will cause the appropriate parameter to be set in the L7NH drive to change DIN 5 (pin 13) functionality to 'TouchProbe'
StatusText	VAR_OUT	STRING	
Armed	VAR_OUT	BOOL	High whilst waiting for a trigger
Done	VAR_OUT	BOOL	
RegPos	VAR_OUT	REAL	Captured position, in user-units
RegPosRaw	VAR_OUT	DINT	Captured position in encoder edges
QuantityOfRegistrations	VAR_OUT	UDINT	Number of registrations captured

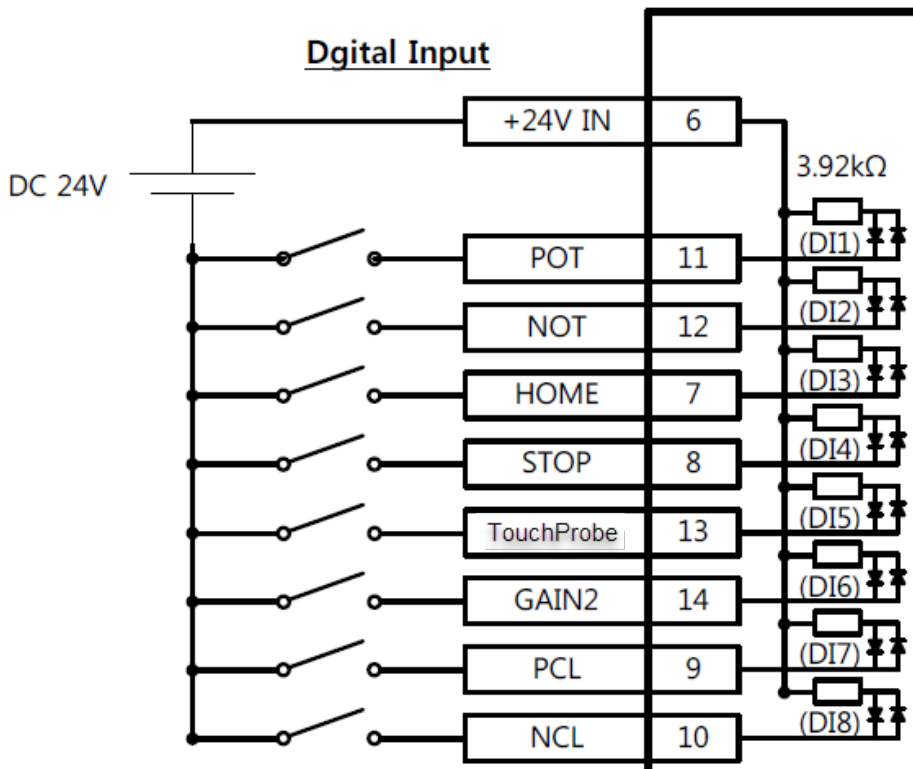
### Tip!

Retrieving a registration position from the drives takes approximately 20ms. Arming faster than this may cause lost registrations.

### Tip!

If a sensor is used as the TouchProbe source, in addition to selecting it with the eTouchProbeSource enumeration, we need to allocate one of the drive's digital inputs to the TouchProbe function. This can be

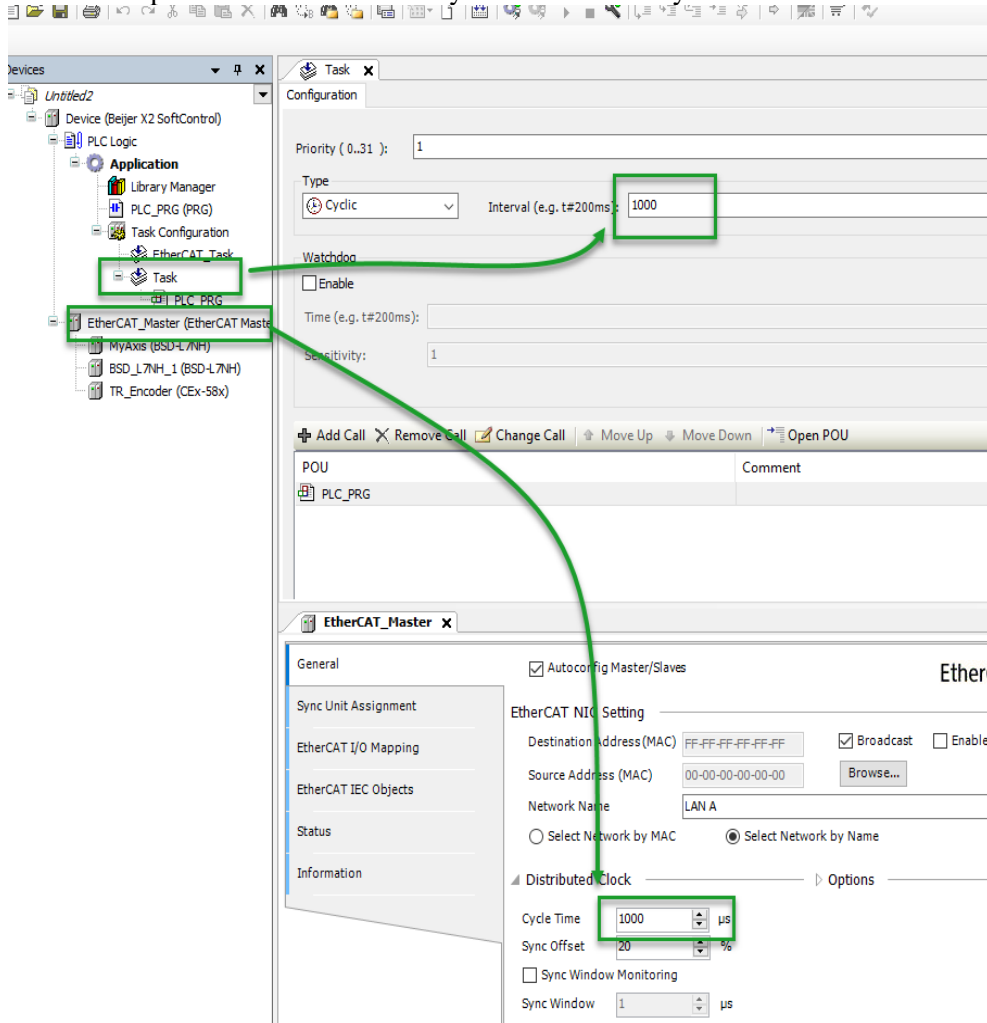
done with the BSD Tools servo programming software or alternatively the input SetDIN5AsTouchProbe is provided. This will make the appropriate setting in the drive and perform an EEPROM save. The wiring will be configured like this (assuming other digital inputs are left as default)



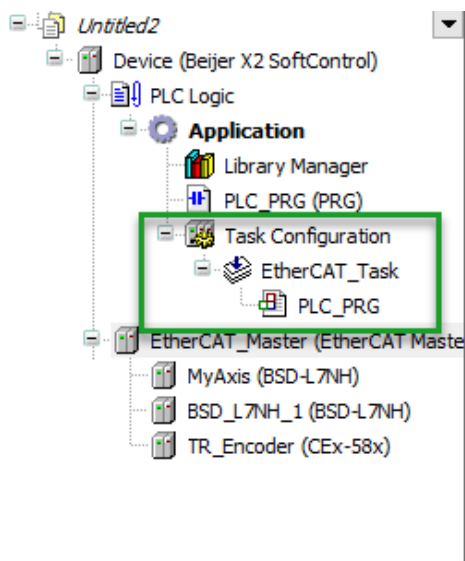
The drive requires a reboot after changing a digital input's function.

## 8 Recommendations

1. Use LAN A for the EtherCAT network
2. Keep the EtherCAT task and your main task in synch

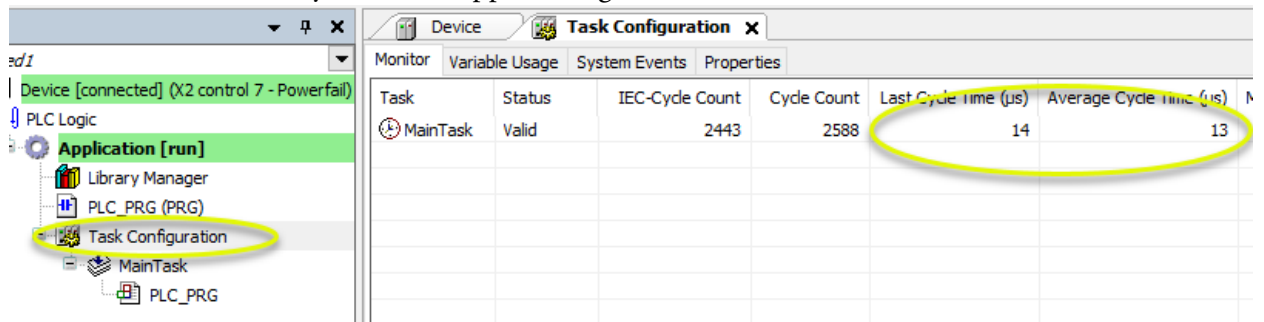


Alternatively, call your Program from the EtherCAT task, thus:



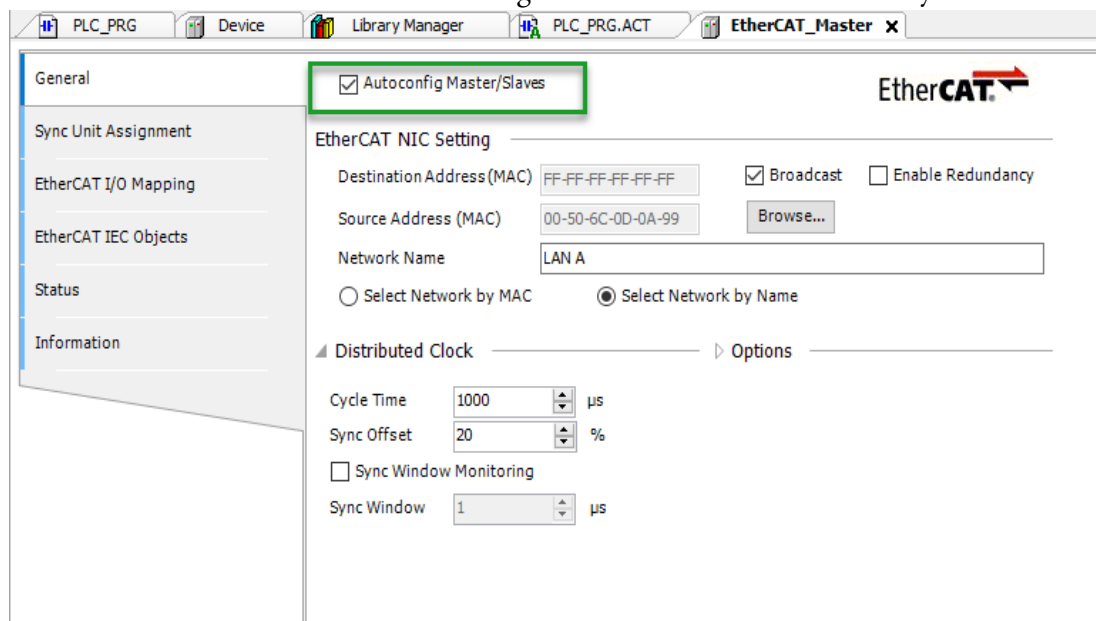


3. Aim for a task time of 1ms, but certainly not essential. The execution time of the task can be monitored here. If the Cycle time is approaching the task interval, increase the task interval.



Task	Status	IEC-Cycle Count	Cycle Count	Last Cycle time (µs)	Average Cycle Time (µs)
MainTask	Valid	2443	2588	14	13

4. Although there are no axis limits, be aware that reaction times will drop off as more axes are added
5. Allow the EtherCAT master to autoconfigure the slaves. This is mandatory.



Autoconfig Master/Slaves

**EtherCAT NIC Setting**

Destination Address (MAC) FF-FF-FF-FF-FF-FF  Broadcast  Enable Redundancy

Source Address (MAC) 00-50-6C-0D-0A-99

Network Name LAN A

Select Network by MAC  Select Network by Name

**Distributed Clock**

Cycle Time 1000 µs

Sync Offset 20 %

Sync Window Monitoring

Sync Window 1 µs

6. Most block have a **Units** input, which is so speed, distances, accels and decels can be entered in engineering units.
- All Beijer motors have 524288 ppr, except FAL01 which has 262144 ppr
  - A 5mm ballscrew with 10:1 gearbox and a 400W motor would have units:
    - $(524288 * 10) / 5 \text{ mm}$
  - A rotary table with a 40:1 gearbox and 800W motor would have units:
    - $(524288 * 40) / 360 \text{ degrees}$

## 9 About Beijer Electronics

Beijer Electronics is a multinational, cross-industry innovator that connects people and technologies to optimize processes for business-critical applications. Our offer includes operator communication, automation solutions, digitalization, display solutions and support. As experts in user-friendly software, hardware and services for the Industrial Internet of Things, we empower you to meet your challenges through leading-edge solutions.

Beijer Electronics is a BEIJER GROUP company. Beijer Group has a sale over 1.6 billion SEK in 2021 and is listed on the Nasdaq Stockholm Main Market under the ticker BELE.

[www.beijergroup.com](http://www.beijergroup.com)

### 9.1 Contact us

[Global offices and distributors](#)