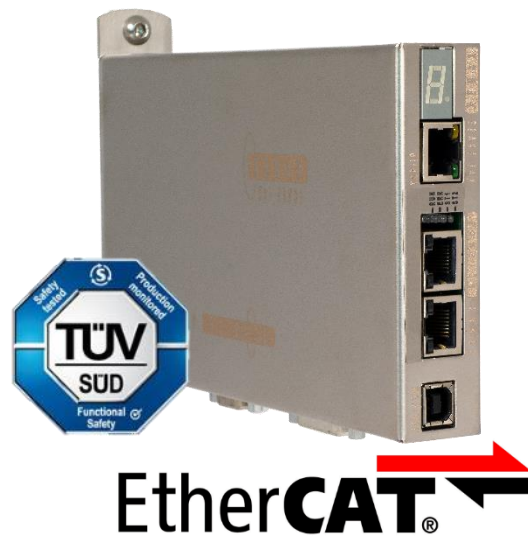**Instruction Manual
JS_MC_LIB with XENAX® Xvi
EtherCAT® andTwinCAT®3**

Version 3.1.6

Edition 16. November 2023

EtherCAT®

XENAX® Ethernet servo controller with EtherCAT® Busmodul

Functional Safety, TÜV certified
Force processes with „Force Limitation",
„Force Monitoring" and „Force Control"

General

This manual describes the **J**enny **S**cience **M**otion **C**ontrol **Lib**rary (JS_MC_Lib) for TwinCAT3. This library is designed after the PLCopen standard but also integrates Jenny Science specific features nicely. The library can be used without or with an NC-Axis according to user requirements.

Contents

1 Development Environment

1.1 Beckhoff

1.1.1 Programmable Logic Controller

Beckhoff control technology is scalable – from Industrial PCs to PLCs – and can be accurately adapted to your application. The automation software integrates real-time control with PLC, NC and CNC functions.
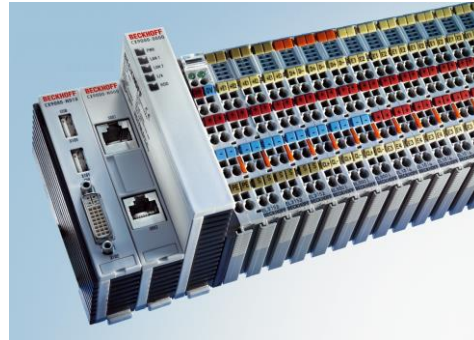
1.1.2 TwinCAT® Version 3

In order to program Beckhoff PLCs the development software for automation TwinCAT® 3 is required.
TwinCAT® 3 uses the Visual Studio Framework and all explanations in this instruction manual are based on it.

1.2 Jenny Science

1.2.1 XENAX® servo controller

XENAX® servo controller for Jenny Science Axis with integrated EtherCAT® bus module. The bus module is optional but it is required for this application. One XENAX® can control one axis. The XENAX® servo controller recognises all Jenny Science motors and configures the parameters correctly.

1.2.2 LINAX® Linear motors

The LINAX® linear motor axes are highly modular and can be flexibly combined amongst each other. Four different series are available.

Lxc = compact
Lxu = universal
Lxs = shuttle
Lxe = exclusive
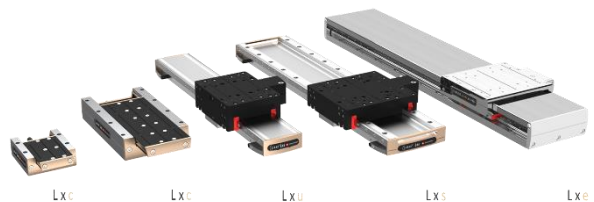
1.2.3 ELAX® Linear motor slides

Specifically designed for handling and Pick and Place tasks with strokes from 30mm up to 150mm. The configuration is extremely modular and there is only one cable to the XENAX® servo controller.
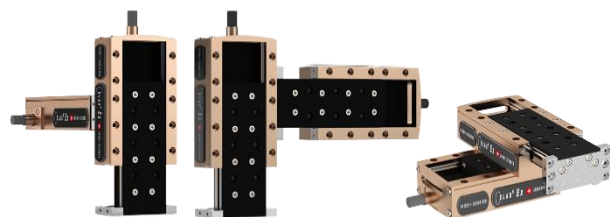
1.2.4 ROTAX® Rotary motor axes

Specifically designed for fast and precise assembly and handling tasks. It can be equipped with standard gripping tools which enables a 360° rotation and has a hollow shaft feedthrough for vacuum or compressed air.

Rxvp = vacuum pressure
Rxhq = high torque

1.2.5 WebMotion

This is the graphical user interface from Jenny Science. It is stored in the embedded Web server of the XENAX® servo controller. WebMotion® is launched with a web browser by entering the corresponding TCP/IP address of XENAX®.

LINAX® linear motor axes, ELAX® linear motor slides or ROTAX® rotary motor axes are automatically recognized. The corresponding controller parameters are saved and loaded automatically. With the Quick Start button, the linear motors can operate immediately. No user manual is needed.

Before the XENAX® controller can be used with the Beckhoff PLC via EtherCAT®, a set-up must be made via WebMotion®. This includes the set-up of a payload, soft limits, etc.
For further information on the set-up of a linear motor axis please refer to the instruction manual or the tutorial video that can be found on https://www.jennyscience.ch/en/applications/videos.

1.3 Status LED's of EtherCAT bus module

- 🟢 RUN
- 🔴 ERR
- 🟡 ST1
- 🟡 ST2

LED Status	RUN	ERR	ST1 (Jenny Science specific)	ST2 (Jenny Science specific)
<OFF>	Initialisation state or no power	Bus module operable	-	Bus module ready
<ON>	Operational state	State bus off	No application in the flash	-
<BLINK>	Pre-operational state	Internal Eeprom blank	-	Protocol download in progress

1.4 Additional Material

The following data is needed for a successful operation of the XENAX® servo controller with a EtherCAT bus module:

Filename	Description
Xenax_EtherCAT_Xvi_.xml	Jenny Science ESI-File/XML device description for TwinCAT. The ESI-file can be downloaded on our website www.jennyscience.ch „XENAX® Servo controller" and „Firmware Bus Module"
JS_MC_Lib.compiled-library	Jenny Science Motion Control Library as compiled library. The library is part of the folder in which this manual is located.
CANopen Ethernet Manual_.pdf	Manual describes the CANopen communication profile CiA DS301 as well as the device profile CiA DS402 including all available parameters.

1.5 Software Requirements

Software	Version
TwinCAT® 3 Automation Software	3.1.4020.0 or later
XENAX Firmware	V5.08 or later
EtherCAT Bus-Module	V2.66 or later

1.6 Cabling

The EtherCAT bus is connected with the XENAX®
servo controller (IN/OUT).
For easy commissioning and maintenance, a TCP/IP
connection to each servo controller is recommended
for access to the WebMotion®.

Switch
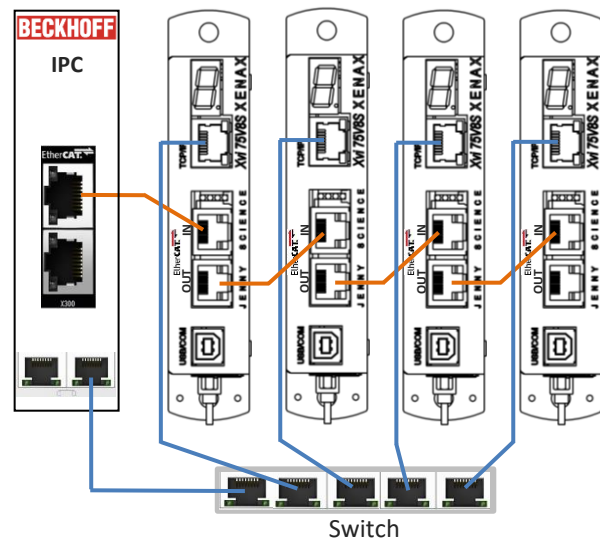
— EtherCAT Bus-Communication
— TCP/IP for WebMotion®

2 PLCopen Library (JsMcLib)

Jenny Science provides a PLCopen library for TwinCAT® 3. The PLCopen standard is easy to understand and includes basic movement functions as well as Jenny Science specific features.

2.1 Drive Modes: point to point or interpolated

The Jenny Science Motion Control Library supports two fundamental different drive modes.

1. **Point to point = Profile Position Mode:** The parameters distance, speed, acceleration and s-curve are fixed before a drive. The trajectory (driving curve) is calculated on the Xenax®. This driving mode is simpler to implement, but gives less control over the driving curve to the PLC. It is not possible drive a straight line with a XY-Axis since both Axis can be started at the same time but will reach their target at different times. It is also not possible to drive along a round curve because only the target position can be specified and not the path to the target location.

This mode fits a small PLC with low performance. There is **no** need for a **virtual nc-axis**.

XY-Axis Profile Position
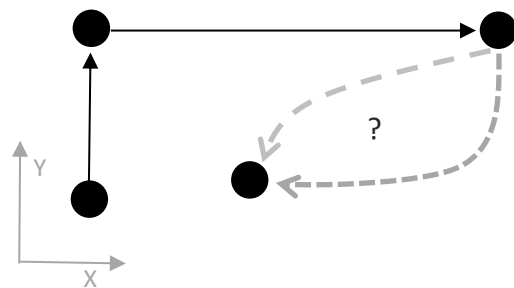
Limited control over driving path between two target positions with different X and Y coordinates. Furthermore, the speed and target position can not be changed during a drive. An Axis has to stop at every target position.

2. **Interpolated = Cyclic Synchronous Position Mode:** In the cyclic synchronous position mode, the target position is passed to the XENAX® servo controller at cyclic time intervals (for example every millisecond). The trajectory (driving curve) is calculated on the Beckhoff PLC. For this reason, a virtual Axis for each Axis is needed. This enables full control over the driving curve. Thanks to the **virtual nc-axis**, round curves or other complex driving paths are now possible.

XY-Axis Cyclic Synchronous Position
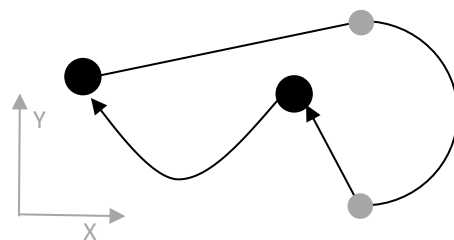
Full control over Axis movement. Two grey circles show a change in direction and speed without a stop.

The following diagram shows the state and the behaviour of the axis when multiple motion control function blocks are "simultaneously" active.

Each motion command is a transition that changes the state of the axis and, as a consequence, influences the method of calculation of the current movement.

All function blocks which do not appear in the state diagram, do not affect the state of the axis.

The current state of the axis can be determined with the function block "**JS_MC_ReadStatus**". If a function block is called where it is not allowed, the function block reports an error.

The notes describe the necessary conditions that must be met for a change in an axis state.

Important:
In the states "**Stopping**", "**ErrorStop**", "**Disabled**" and "**Reference**" no motion blocks can be called. In standstill condition, an axis must always be referenced before starting a movement.

2.2.1 Profile Position Mode
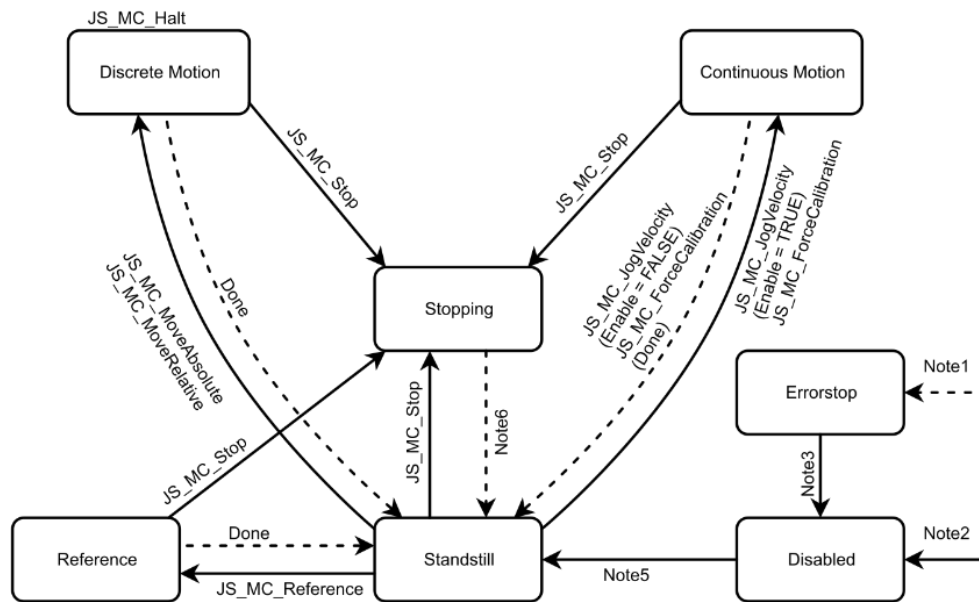
2.2.2 Cyclic Synchronous Position Mode

In the Cyclic Synchronized mode the JS_MC_ReadStatus will be in „Standstill" after the reference. To know which motion-commands can be used, the "MC_ReadStatus" of the **virtual Axis has to be executed.**
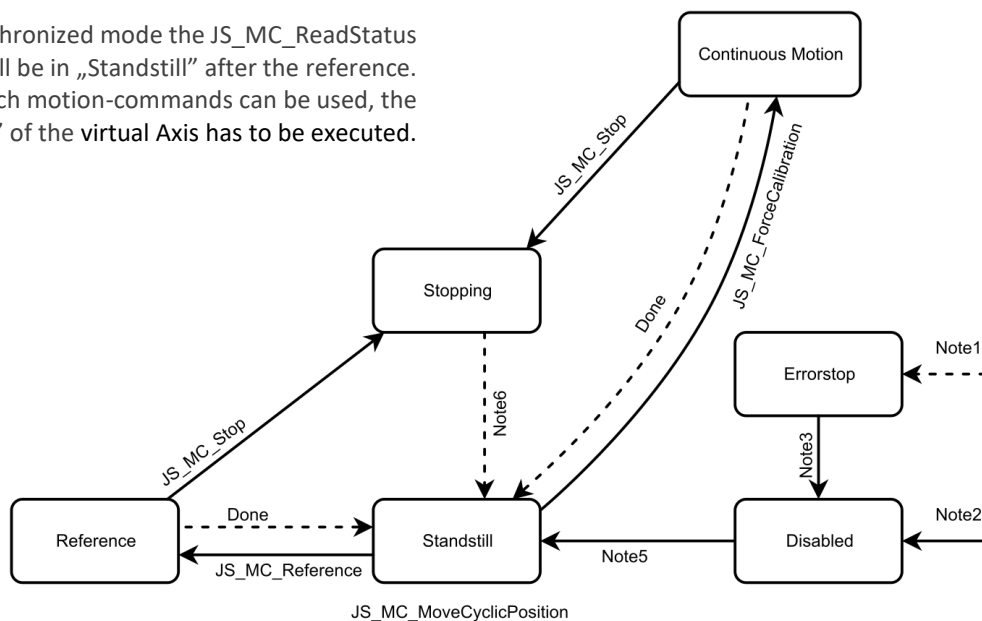
Note	Comment
Note 1	From any state in which an error occurs
Note 2	From any state if MC_Power.Enable = FALSE and the axis has no error
Note 3	MC_Reset and MC_Power.Status = FALSE
Note 4	MC_Reset and MC_Power.Status = TRUE and MC_Power.Enable = TRUE
Note 5	MC_Power.Status = TRUE and MC_Power.Enable = TRUE
Note 6	MC_Stop.Done= TRUE and MC_Stop.Execute = FALSE

2.3 Function blocks required for operation

2.3.1 JS_MC_Init

This function block must be called once at start up to initialize library variables. The block also provides a reference to the axis which is needed in all other JS_MC_Lib function blocks. Calling this function block a second time will only re-evaluate the Axis output reference.

Inputs	
OperationMode	Choose gcJS_MC.jsmc_MODE_CYCLIC_SYNC if you use a NC-Axis, otherwiede use gcJS_MC.jsmc_Mode_PROFILE_POSITION
EncorceReferenceDrive	Set this to TRUE if motor must execute a reference drive even though they are already referenced. Recommended = FALSE

Outputs	
pstAxis	The axis reference handle
Valid	A valid reference handle of the axis is available
Error	Error occurred within function block

```
                 JS_MC_Init
OperationMode                           pstAxis
SINT                          POINTER TO JS_MC_Init_IS
EnforceReferenceDrive                     Valid
BOOL                                      BOOL
                                          Error
                                          BOOL
```

2.3.2 JS_MC_CyclicIn

This function block has to be called at the start of the periodically called program.
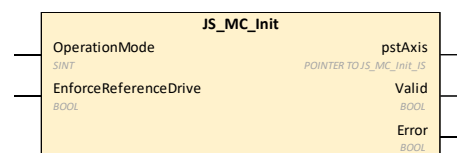Checks if communication with axis is valid.
Reads cyclic data from the fieldbus.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, cyclic data are received from the bus. (Must always be TRUE during operation)
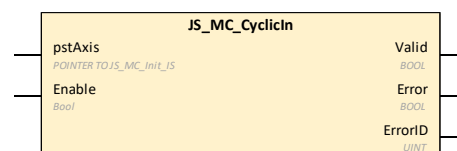
Outputs	
Valid	Cyclic realtime bus communication is valid (it is not allowed to enable or execute other function blocks unless valid is TRUE)
Error	Error occurred within function block
ErrorID	Error number

```
                 JS_MC_CyclicIn
pstAxis                                   Valid
POINTER TO JS_MC_Init_IS                   BOOL
Enable                                    Error
Bool                                       BOOL
                                          ErrorID
                                           UINT
```

2.3.3 JS_MC_CyclicOut

This function block has to be called as the last JSC_MC_Lib block in the periodically called program. Important: All other JSC_MC_Lib blocks must be called between CyclicIn and CyclicOut. Writes cyclic data to the fieldbus.

Inputs	
pstAxis	The axis reference handle

```
         JS_MC_CyclicOut
  pstAxis
  POINTER TO JS_MC_Init_IS
```

2.3.4 JS_MC_Power

This function block switches the power stage on and off.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the drive power Stage is enabled
Enable_Positive	Only in cyclic synchronous position mode. Advance movement enable in positive direction. Only takes effect if Enable = TRUE.
Enable_Negative	Only in cyclic synchronous position mode. Advance movement enable in negative direction. Only takes effect if Enable = TRUE.
Override	Only in cyclic synchronous position mode. Velocity override in % for all motion commands. (0 ≤ Override ≤ 100)
BufferMode	Only in cyclic synchronous position mode. This is evaluated when "Enable" is reset (see also MC_BufferMode). MC_Aborting mode leads to immediate deactivation of the axis enable.

```
              JS_MC_Power
  pstAxis                      Status
  POINTER TO JS_MC_Init_IS      BOOL
  Enable                        Valid
  BOOL                          BOOL
  Enable_Positive               Busy
  BOOL                          BOOL
  Enable_Negative              Active
  BOOL                          BOOL
  Override                      Error
  LREAL                         BOOL
  BufferMode                   ErrorID
  MC_BufferMode                 UINT
```

Outputs	
Status	Effective status of the power Stage
Valid	TRUE when the axis is ready for operation
Busy	Only in cyclic synchronous position mode. TRUE, as long as the function block is called with Enable = TRUE.
Active	Only in cyclic synchronous position mode. Indicates that the command is executed.
Error	Error occurred within function block
ErrorID	Error number

2.3.5 JS_MC_Reference

Performs a reference drive. The goal of the reference
drive is to find the absolute position of the axis.
The axis either drives to a mechanical stopper or to a
Z-Mark indicator on the scale.
The ReferenceMode influences the behaviour during
a reference drive.

Inputs	
pstAxis	The axis reference handle
Execute	Start reference at rising edge
ReferenceMode	Select behaviour during reference drive
ReferenceSpeedRot	Reference speed towards a reference switch [increment/s] (only for rotative drives)
ZMarkSpeedRot	Reference speed towards a Z-Mark on the scale [increment/s] (only for rotative drives)

Outputs	
Done	Reference procedure has finished successfully
Busy	The function block is not finished
CommandAborted	Function block is aborted by another command
Error	Error occurred within function block
ErrorID	Error number

```
              JS_MC_Reference
pstAxis                              Done
POINTER TO JS_MC_Init_IS             BOOL
Execute                              Busy
BOOL                                 BOOL
ReferenceMode              CommandAborted
USINT                                BOOL
ReferenceSpeedRot                   Error
UDINT                                BOOL
ZMarkSpeedRot                      ErrorID
UDINT                                UINT
```

2.3.6 JS_MC_Reset

This function block makes a transition in the State
Diagram from Errorstop to Standstill or Disabled by
resetting the axis error.

Inputs	
pstAxis	The axis reference handle
Execute	Reset the axis at the rising edge

Outputs	
Done	Standstill or Disabled state is reached
Busy	The function block is not finished
Error	Error occurred within function block
ErrorID	Error number

```
               JS_MC_Reset
pstAxis                              Done
POINTER TO JS_MC_Init_IS             BOOL
Execute                              Busy
BOOL                                 BOOL
                                    Error
                                     BOOL
                                   ErrorID
                                     UINT
```
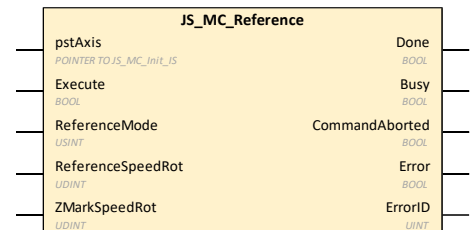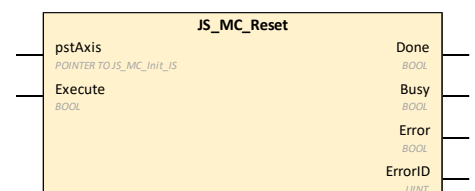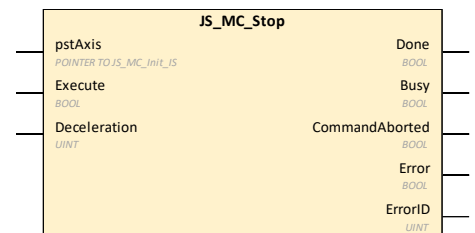
2.3.7 JS_MC_Stop

This function block commands a controlled motion stop and transfers the axis to the state Stopping. The axis stays in the Stopping state until execute is set back to FALSE. All move commands are blocked as long as the axis stays in the stopping state.

The axis does not have to be in motion to call MC_Stop. This means MC_Stop can be used to ensure an axis stays at the same position.

Inputs	
pstAxis	The axis reference handle
Execute	Reset the action at rising edge
Deceleration	Value of the deceleration [inc/s^2]

Outputs	
Done	Zero velocity is reached
Busy	The function block is not finished
CommandAborted	Command is aborted by switching off power (only possibility to abort)
Error	Error occurred within function block
ErrorID	Error number

```
              JS_MC_Stop
pstAxis                          Done
POINTER TO JS_MC_Init_IS         BOOL
Execute                          Busy
BOOL                             BOOL
Deceleration            CommandAborted
UINT                             BOOL
                                Error
                                 BOOL
                               ErrorID
                                 UINT
```

2.4 Additional function blocks for Forceteq®

2.4.1 JS_MC_ForceCalibration

Starts a Force Calibration. The axis moves from start- to end position and measures cogging force and friction. Those two forces are then compensated in future drives.

An active Force Calibration can only be stopped by calling JS_MC_STOP.
If the motor oscillates during the Force Calibration, set IterativeFcDisable. This will clear the old calibration data before a new calibration is started.

Inputs	
pstAxis	The axis reference handle
Execute	Start move at rising edge
StartPosition	Start position for the force calibration [Inc]
EndPosition	End position for the force calibration [Inc]
IterativeFcDisable	0 = Takes values from previous FC to improve calibration 1= Ignores calibration values from last FC

	JS_MC_ForceCalibration	
pstAxis		Done
POINTER TO JS_MC_Init_IS		BOOL
Execute		Busy
BOOL		BOOL
StartPosition		CommandAborted
DINT		BOOL
EndPosition		Error
DINT		BOOL
IterativeFcDisable		ErrorID
BOOL		UINT

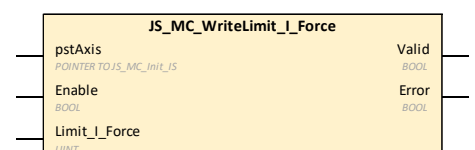| Outputs	
Done	Force Calibration finished successfully
Busy	Function block is not finished
CommandAborted	Function block is aborted by another command
Error	Error occurred within function block
ErrorID	Error number
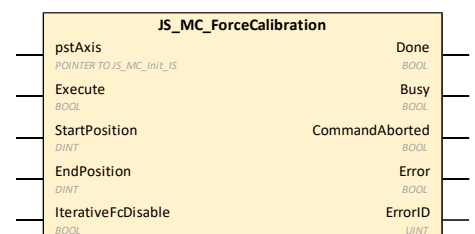
2.4.2 JS_MC_WriteLimit_I_Force

Sets the I_Force limitation in [10mA] with Forceteq® basic. Note that the PDO 0x6073 "Limit_I_Force" must be mapped and linked.

Inputs	
pstAxis	The axis reference handle
Enable	Writes Limit I_Force to XENAX when enabled
Limit_I_Force	I_Force limitation in [x10 mA], 20 = 200mA, 0 = no limitation.

	JS_MC_WriteLimit_I_Force	
pstAxis		Valid
POINTER TO JS_MC_Init_IS		BOOL
Enable		Error
BOOL		BOOL
Limit_I_Force		
UINT		

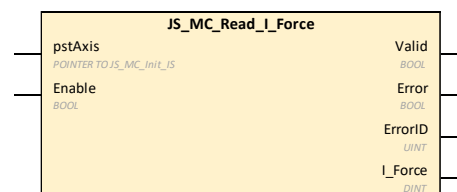| Outputs	
Valid	Write was successful
Error	Error occurred within function block

2.4.3 JS_MC_Read_I_Force

Reads the Force-proportional I_Force in mA
with Forceteq® basic. Note that the PDO 0x2005
"I_Force" must be mapped and linked.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the actual I_Force is read out continuously

Outputs	
Valid	A valid set of outputs is available at the function block
Error	Error occurred within function block
ErrorID	Error number
I_Force	Actual I_Force [mA]

```
            JS_MC_Read_I_Force
pstAxis                            Valid
POINTER TO JS_MC_Init_IS            BOOL
Enable                             Error
BOOL                                BOOL
                                  ErrorID
                                    UINT
                                  I_Force
                                    DINT
```

2.4.4 JS_MC_WriteLimit_Force

Sets the Force limitation in mN based on the value
measured by the Signateq® force sensor (only for
Forceteq® pro). Note that the PDO 0x2009 "Limit
Force" must be mapped and linked.

Inputs	
pstAxis	The axis reference handle
Enable	Writes LimitForce to XENAX when enabled
Limit_Force	Force limit [mN]

Outputs	
Valid	Last write was successful
Error	Error occurred within function block

```
            JS_MC_WriteLimit_Force
pstAxis                            Valid
POINTER TO JS_MC_Init_IS            BOOL
Enable                             Error
BOOL                                BOOL
Limit_Force
DINT
```
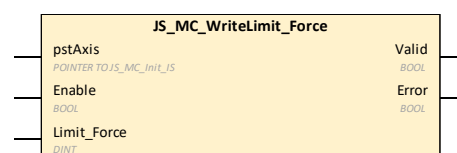
2.4.5 JS_MC_Read_Force

Reads the Force in mN measured by the Signateq®
force sensor (only with Forceteq® pro). Note that the
PDO 0x200A "Limit Force" must be mapped and
linked.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the actual force is read out continuously

Outputs	
Valid	A valid set of outputs is available at the function block
Error	Error occurred within function block
ErrorID	Error number
Force	Actual force measured by Signateq® [mN]

```
            JS_MC_Read_Force
pstAxis                            Valid
POINTER TO JS_MC_Init_IS            BOOL
Enable                             Error
BOOL                                BOOL
                                  ErrorID
                                    UINT
                                   Force
                                    DINT
```
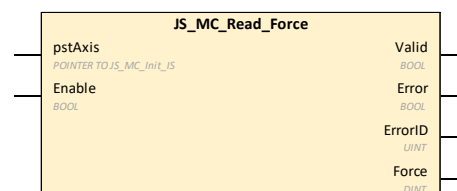
2.5 Additional function blocks for cyclic synchronous position mode

Use the MC_MoveAbsolute and MC_Halt function block from Beckhoff to operate the axis.

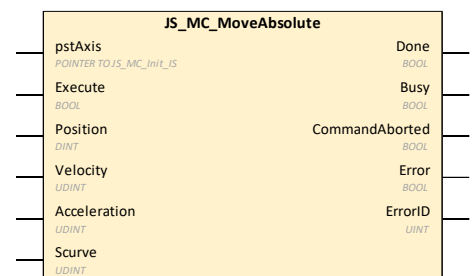JS_MC_STOP calls MC_STOP internally if the axis is driving in cyclic synchronous positon mode (csp). During a force calibration or a reference drive, the axis is not driving in csp mode. JS_MC_STOP will still correctly abort a non csp mode drive. This means that only JS_MC_STOP will abort a force calibration or a reference drive and MC_STOP will not.

2.6 Additional function blocks for Profile Position

2.6.1 JS_MC_MoveAbsolute

This function block drives to an absolute position.
Only for profile position mode. In cyclic synchronous
position mode, use MC_MoveAbsolute instead.

Inputs	
pstAxis	The axis reference handle
Execute	Start move at rising edge
Position	Target position for the motion [inc]
Velocity	Value of maximum velocity [inc/s] (not necessarily reached)
Acceleration	Value of maximum acceleration [inc/s^2] (not necessarily reached)
Scurve	Value of S-curve parameter during the acceleration [%]

```
            JS_MC_MoveAbsolute
─  pstAxis                           Done  ─
   POINTER TO JS_MC_Init_IS          BOOL
─  Execute                           Busy  ─
   BOOL                              BOOL
─  Position                 CommandAborted  ─
   DINT                             BOOL
─  Velocity                        Error  ─
   UDINT                            BOOL
─  Acceleration                   ErrorID  ─
   UDINT                             UINT
─  Scurve
   UDINT
```

Outputs	
Done	Commanded position reached
Busy	The function block is not finished
CommandAborted	Function block is aborted by another command
Error	Error occurred within function block
ErrorID	Error number

2.6.2 JS_MC_MoveRelative

This function block drives a distance.
Only for profile position mode. In cyclic synchronous
position mode, use MC_MoveRelative instead.

Inputs	
pstAxis	The axis reference handle
Execute	Start move at rising edge
Distance	Target distance for the motion [inc]
Velocity	Value of maximum velocity [inc/s] (not necessarily reached)
Acceleration	Value of maximum acceleration [inc/s^2] (not necessarily reached)
Scurve	Value of S-curve parameter during the acceleration [%]

```
            JS_MC_MoveRelative
─  pstAxis                           Done  ─
   POINTER TO JS_MC_Init_IS          BOOL
─  Execute                           Busy  ─
   BOOL                              BOOL
─  Distance                 CommandAborted  ─
   DINT                             BOOL
─  Velocity                        Error  ─
   UDINT                            BOOL
─  Acceleration                   ErrorID  ─
   UDINT                             UINT
─  Scurve
   UDINT
```
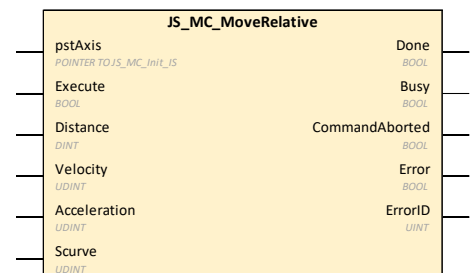
Outputs	
Done	Commanded distance reached
Busy	The function block is not finished
CommandAborted	Function block is aborted by another command
Error	Error occurred within function block
ErrorID	Error number

2.6.3 JS_MC_JogVelocity

This function block drives with a constant speed in positive or negative direction. Only for profile position mode. In cyclic synchronous position mode, use MC_JogVelocity instead.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the axis in in Continuous Motion mode (Jog drive possible)
Velocity	Value of maximum velocity [inc/s] Note: This value can also be changed while a movement is taking place
Acceleration	Value of maximum acceleration [inc/s^2] Note: This value can also be changed while a movement is taking place (new value is used at the next velocity change)
Deceleration	Value of maximum deceleration [inc/s^2] Note: This value can also be changed while a movement is taking place (new value is used at the next velocity change)
JogPositive	Executes a movement in the positive direction
JogNegative	Executes a movement in the negative direction

JS_MC_JogVelocity	
pstAxis POINTER TO JS_MC_Init_IS	Active BOOL
Enable BOOL	Busy BOOL
Velocity UDINT	CommandAborted BOOL
Acceleration UDINT	Error BOOL
Deceleration UDINT	ErrorID UINT
JogPositive BOOL	Jogging BOOL
JogNegative BOOL	
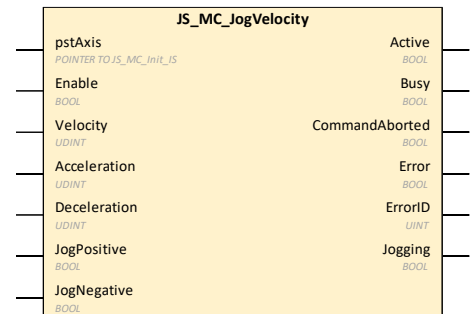
Outputs	
Active	The function block is active, possible to execute movements
Busy	The function block is not finished
CommandAborted	Function block is aborted by another command
Error	Error occurred within function block
ErrorID	Error number
Jogging	Movement being carried out

2.6.4 JS_MC_Halt

This function block stops an ongoing JS_MC_MoveAbsolute or a JS_MC_MoveRelative command and switches to the state standstill.

Inputs	
pstAxis	The axis reference handle
Execute	Start the action at rising edge
Deceleration	Value of the deceleration [inc/s^2]

JS_MC_Halt	
pstAxis POINTER TO JS_MC_Init_IS	Done BOOL
Execute BOOL	Busy BOOL
Deceleration UDINT	CommandAborted BOOL
	Error BOOL
	ErrorID UINT

Outputs	
Done	Zero velocity is reached
Busy	Function block is not finished
CommandAborted	Function block is aborted by another command
Error	Error occurred within function block
ErrorID	Error number

2.7 Function blocks for error handling

2.7.1 JS_MC_ReadAxisError

Describes general axis errors that are not related to function blocks. Use JS_MC_RESET to clear the axis error.

Inputs	
pstAxis	The axis reference handle
Enable	While TRUE, the output value provides the parameter value continuously for reading out.

Outputs	
Valid	Valid outputs are available
Error	Error occurred within function block
ErrorID	Error number
AxisError	Axis error has occured.
AxisErrorID	Axis error number, see XENAX manual

```
                JS_MC_ReadAxisError
pstAxis                               Valid
POINTER TO JS_MC_Init_IS               BOOL

Enable                                Error
BOOL                                   BOOL

                                     ErrorID
                                       UINT

                                    AxisError
                                       BOOL

                                   AxisErrorID
                                       UINT
```
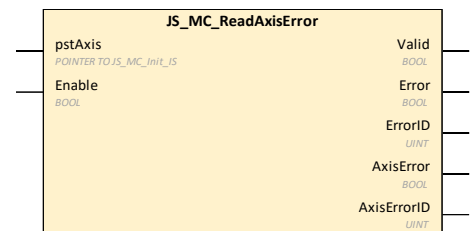
2.7.2 JS_MC_ReadLibraryError

This block can be used to handle all errors at a central point in the code. Note that this function block is completely optional.
Collect all errors from axis and JS_MC_Lib function blocks. All errors are collected in a queue. The first error in the queue is displayed in the output ErrorRecord.
Each error must be acknowledged to display the next error until the queue is empty.
The error should be handled first before it is acknowledged. This means to reset the enable or execute input of the
Function block which caused the error. Or to call JS_MC_RESET to clear the error caused by the axis.
An unhandled error which is acknowledged will be recollected and again saved in the queue.

JS_MC_ReadLibraryError	
pstAxis *POINTER TO JS_MC_Init_IS*	Valid *BOOL*
Enable *BOOL*	Busy *BOOL*
Acknowledge *BOOL*	Error *BOOL*
	ErrorID *UINT*
	ErrorRecordAvailable *BOOL*
	ErrorRecord *JS_MC_ErrorRecord*
	FunctionBlockErrorCount *UINT*
	AxisErrorCount *UINT*
	AxisWarningCount *UINT*

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the function block can be used to read out axis and function block errors
Acknowledge	Acknowledges the error record currently displayed

Outputs	
Valid	A valid set of outputs is available at the function block. This output is set to FALSE while an error is being acknowledged or error text is being read
Busy	New output data is to be expected. This output is set to TRUE while an error is being acknowledged or error text is being read
Error	Error occurred within this function block
ErrorID	Error number of this function block
ErrorRecordAvailable	Set if a new error record is displayed in the "ErrorRecord" output. FALSE if error queue is empty
ErrorRecord	Displays the first error in the queue including the ErrorNumber (see 2.10 Error Numbers ErrorType (see 2.11 Error Sources) ErrorSource (see 2.12 Error Type)
FunctionBlockErrorCount	Number of pending function block errors to display
AxisErrorCount	Number of pending axis errors to display
AxisWarningCount	Number of pending axis warnings to display
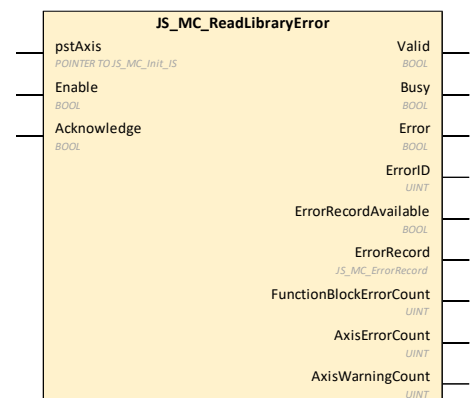
2.8 Optional function blocks

2.8.1 JS_MC_ReadActualPosition

Reads the position of the axis in increments. Note that PDO 0x6064 "Position Actual Value" must be mapped and linked.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the actual position is read continuously

Outputs	
Valid	A valid set of outputs is available at the function block
Error	Error occurred within function block
ErrorID	Error number
Postion	Actual position of the axis [Inc]

```
           JS_MC_ReadActualPosition
pstAxis                                    Valid
POINTER TO JS_MC_Init_IS                     BOOL
Enable                                     Error
BOOL                                         BOOL
                                         ErrorID
                                            UINT
                                        Position
                                            DINT
```
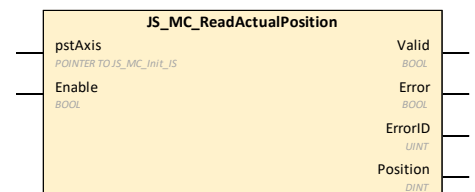
2.8.2 JS_MC_ReadDigitalInput

Reads digital inputs which are located in the XENAX socket. Note that PDO 0x60FD "Digital Inputs" must be mapped and linked.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the digital inputs are read continuously

Outputs	
Valid	A valid set of outputs is available at the function block
Error	Error occurred within function block
ErrorID	Error number
DigitalInput	The value of digital inputs (bit-coded)

```
           JS_MC_ReadDigitalInput
pstAxis                                    Valid
POINTER TO JS_MC_Init_IS                     BOOL
Enable                                     Error
BOOL                                         BOOL
                                         ErrorID
                                            UINT
                                     DigitalInput
                                            UINT
```

2.8.3 JS_MC_ReadDigitalOutput

Reads back digital outputs which are located in the XENAX socket. Note that PDO 0x60FE "Digital outputs" must be mapped and linked.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the digital outputs are read continuously

Outputs	
Valid	A valid set of outputs is available at the function block
Error	Error occurred within function block
ErrorID	Error number
DigitalOutput	The value of digital outputs (bit-coded)

```
           JS_MC_ReadDigitalOutput
pstAxis                                    Valid
POINTER TO JS_MC_Init_IS                     BOOL
Enable                                     Error
BOOL                                         BOOL
                                         ErrorID
                                            UINT
                                    DigitalOutput
                                           USINT
```
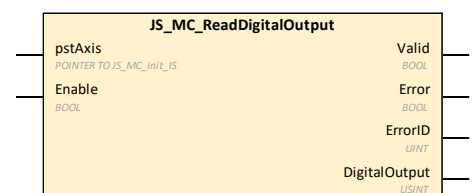
This block is used to read out CANopen parameter from the axis. All available CANopen parameters are described in the "CANOPEN_ETHERNET_MANUAL.pdf" (https://www.jennyscience.ch/en/products/download).

JS_MC_ReadParameter

pstAxis	Valid
POINTER TO JS_MC_Init_IS	BOOL
Enable	Busy
BOOL	BOOL
DataObject	Error
UINT	BOOL
SubID	ErrorID
USINT	UINT
DataLength	Value
USINT	DINT

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the desired data object is read out continuously
DataObject	Desired data object to be written (according CANopen communication profile)
SubID	SubID of the desired data object to be written
DataLength	Data length of the desired data object to be written in bytes

Outputs	
Valid	A valid set of outputs is available at the function block
Busy	The function block is not finished
Error	Error occurred within function block
ErrorID	Error number
Value	Value to be written to the desired data object
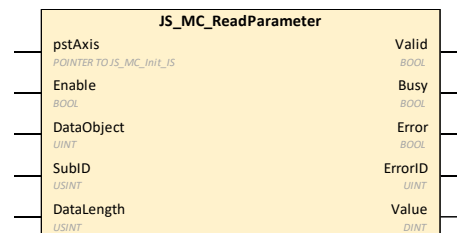
2.8.5 JS_MC_ReadPSR

Reads the Process Status Register (PSR). This register contains various information about the XENAX® servo controller. Note that PDO 0x2006 "Process Status Register" must be mapped and linked.

JS_MC_ReadPSR

pstAxis	Valid
POINTER TO JS_MC_Init_IS	BOOL
Enable	Error
BOOL	BOOL
	ErrorID
	UINT
	ProcessStatusRegister
	JS_MC_Init_IS_ProcStat

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the actual Process Status Register is read out continuously

Outputs	
Valid	A valid set of outputs is available at the function block
Error	Error occurred within function block
ErrorID	Error number
ProcessStatusRegister	Process Status Register of the XENAX controller (see data type JS_MC_Init_IS_ProcStat)
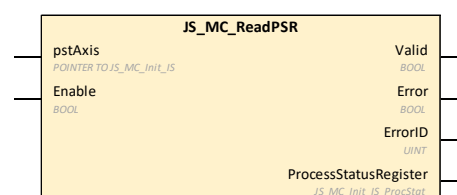
2.8.6 JS_MC_ReadStatus

Reads the current state of the PLCopen DS402 state machine.
Note: In the Cyclic Synchronized mode the state will be in „Standstill" after the reference.
To know which motion-commands can be used, the "MC_ReadStatus" of the virtual Axis has to be executed.

Inputs	
pstAxis	The axis reference handle
Enable	As long as "Enable" is TRUE, the axis status is read out continuously

Outputs	
Valid	A valid set of outputs is available at the function block
Error	Error occurred within function block
ErrorID	Error number
Errorstop	An error has occurred. Use JS_MC_Reset to acknowledge errors
Disabled	JS_MC_Power has not powered the axis, or an error has been acknowledged by JS_MC_Reset and the axis has been turned off
Stopping	JS_MC_Stop is active
Standstill	Motion is not active on the drive
DiscreteMotion	Axis is in motion due to one of the following function blocks: JS_MC_MoveAbsolute, JS_MC_Relative.
Reference	JS_MC_Reference has started referencing the axis
ContinuousMotion	Axis is in motion due to the following function block: JS_MC_JogVelocity, JS_MC_ForceCalibration

JS_MC_ReadStatus	
pstAxis	Valid
POINTER TO JS_MC_Init_IS	BOOL
Enable	Error
BOOL	BOOL
	ErrorID
	UINT
	Errorstop
	BOOL
	Disabled
	BOOL
	Stopping
	BOOL
	Standstill
	BOOL
	DiscreteMotion
	BOOL
	Reference
	BOOL
	ContinuousMotion
	BOOL
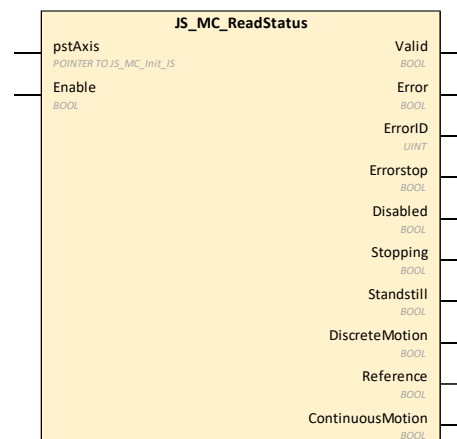
2.8.7 JS_MC_WriteDigitalOutput

This function block writes a digital output on the servo controller. The output is written once when "Execute" is set. Note that PDO 0x60FE "Digital Outputs" must be mapped and linked.

Inputs	
pstAxis	The axis reference handle
Execute	Writes the Digital value at the rising edge
DigitalOutput	The value of digital outputs (bit-coded)

Outputs	
Done	Digital outputs are written
Error	Error occurred within function block
ErrorID	Error number

JS_MC_WriteDigitalOutput	
pstAxis	Done
POINTER TO JS_MC_Init_IS	BOOL
Execute	Error
BOOL	BOOL
DigitalOutput	ErrorID
USINT	UINT
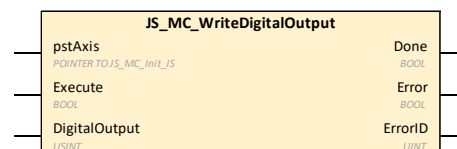
2.8.8 JS_MC_WriteParameter

This block is used to write CANopen parameter from the axis. All available CANopen parameters are described in the "CANOPEN_ETHERNET_MANUAL.pdf" (https://www.jennyscience.ch/en/products/download).

Inputs	
pstAxis	The axis reference handle
Execute	Writes the value at the rising edge
DataObject	Desired data object to be written (according CANopen communication profile)
SubID	SubID of the desired data object to be written
DataLength	Data length of the desired data object to be written in bytes
Value	Value to be written to the desired data object

Outputs	
Done	Value is written to the object
Busy	The function block is not finished
Error	Error occurred within function block
ErrorID	Error number

```
              JS_MC_WriteParameter
pstAxis                                    Done
POINTER TO JS_MC_Init_IS                   BOOL
Execute                                    Busy
BOOL                                       BOOL
DataObject                                 Error
UINT                                       BOOL
SubID                                      ErrorID
USINT                                      UINT
DataLength
USINT
Value
DINT
```
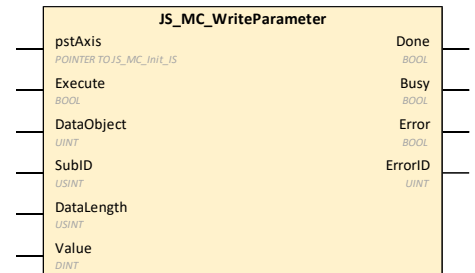
2.9 Minimum and Maximum Values of Function Blocks

Following minimum and maximum values of the function blocks should be adhered to.

Name	Datatype	Min value	Max value
Velocity linear	UDINT	10 inc/s	9'000'000 inc/s
Velocity rotative	UDINT	10 inc/s	100'000'000 inc/s
Deceleration	UDINT	2'000 inc/s^2	1'000'000'000 inc/s^2
Acceleration	UDINT	2'000 inc/s^2	1'000'000'000 inc/s^2
S-curve	UDINT	1 %	100 %

2.10 Error Numbers

The following ErrorIDs can be generated by the JsMcLib function blocks. Lower numbers than 50000 are Axis Error generated by the XENAX® servo controller. Please look up those errors in the XENAX® Manual.

Value	Name	Description	Correction
0	ERR_OK	FUB executed correctly with no errors	None
50000	jsmcERR_NIL_POINTER	No axis passed to FB	Ensure function block call only with correct axis passed.
50001	jsmcERR_DRIVE_NOT_READY	Servo controller is not ready to switch on	Check controller for errors
50002	jsmcERR_DRIVE_SWITCHED_OFF	Servo controller is switched off	Don't call function block when Servo controller is switched off
50004	jsmcERR_REFERENCE_WRONG_METHOD	Reference method is not correct for the motor	Check documentation for allowed reference methods for the motor
50006	jsmcERR_ACCE_TO_SMALL	Acceleration is too small	Use larger acceleration (>=2000 inc/s^2)
50008	jsmcERR_SCURVE_NOT_IN_RANGE	Scurve is not in allowed range	Use Scurve in allowed range (1...100%)
50010	jsmcERR_SDO_COMM_FAILURE	Failure during SDO communication	Check connection to the Servo Controller or for correct SDO object
50011	jsmcERR_POWER_UP_FAILURE	Failure during power up sequence	Check Servo controller for correct power supply
50012	jsmcERR_POWER_LOST	Power was turned off outside of JS_MC_Power control	Check and quit errors from other function blocks or axis, which caused the power off
50013	jsmcERR_WRONG_STATE_FOR_FB	The FB cannot be used in the current state	Check program to call FB's only in allowed states
50014	jsmcERR_WRONG_OP_MODE_FOR_FB	The FB cannot be used in the current mode of operation	Only use allowed FB's for the desired mode of operation (profile position or cyclic synchronized)
50015	jsmcERR_EXECUTION_ERROR	The FB failed during execution by an external error	Check and quit errors from other function blocks or axes, which caused the fault
50016	jsmcERR_BUFFER_TO_SMALL	The buffer for the error text string is too small	Put a pointer to a buffer for the error text string which size is at least 50 characters
50017	jsmcERR_TEXT_OBJ_NOT_FOUND	Error text object or function block text object not found	Enter correct name of the error text object and ensure, that the error text object (JsMcEtxDe/JsMcEtxEn) and the function block text object (JsMcFBtxEn) are present in the project

50018	jsmcERR_TEXT_READOUT_FAILURE	Error text or function block text could not be read successfully	Ensure that the error text object (JsMcEtxDe/JsMcEtxEn) and the function block text object (JsMcFBtxEn) are present in the project
50019	jsmcERR_WRONG_GENERAL_OP_MODE	General mode of operation not supported	Set a supported general mode of operation in JS_MC_Init (OperationMode = jsmcMODE_PROFILE_POSITION or jsmcMODE_CYCLIC_SYNC)
50020	jsmcERR_REF_SPEED_NOT_IN_RANGE	Reference speed for rotative motors is out of range	Use reference speed in allowed range (0...250'000 inc/s)
50021	jsmcERR_ZMARK_SPEED_NOT_IN_RANGE	Z-Mark speed for rotative motors is out of range	Use Z-Mark speed in allowed range (0...100'000 inc/s)
50022	jsmcERR_VELOCITY_NOT_IN_RANGE	Velocity is out of range	Use velocity in allowed range (10...9'000'000 inc/s for linear motor, 10...100'000'000 inc/s for rotative motor)
50023	jsmcERR_ACCE_TO_LARGE	Acceleration is too large	Use smaller acceleration (<= 1'000'000'000 inc/s^2)
50024	jsmcERR_CYCLE_TIME_FAILURE	Cycle time setting failure	Use correct cycle time setting (bus cycle time >= 200us and software task cycle time >= bus cycle time)
50025	jsmcERR_DECE_TO_SMALL	Deceleration is too small	Use larger deceleration (>= 2'000 inc/s^2)
50026	jsmcERR_DECE_TO_LARGE	Deceleration is too large	Use smaller deceleration (<= 1'000'000'000 inc/s^2)
50027	jsmcERR_FW_VERS_FAILURE	Firmware version failure	Use at least XENAX Firmware V3.64D
50028	jsmcERR_PDO_MAPPING_CHK_FAILURE	Failure during PDO mapping check	Check for correct PDO mapping
50029	jsmcERR_PDO_MAPPING_MISSING	Necessary PDO mapping missing	Check if all necessary PDOs are mapped
50030	jsmcERR_NO_DATA_ADDRESS_ASSIGNED	No data address for error text string assigned	Assign valid data address for error text string
50031	jsmcERR_SDO_ACCESS_FAILURE	Invalid SDO access	Check input values DataObject, SubID and DataLength and set correct values
50032	jsmcERR_CYCLIC_COMM_INTERRUPTED	Cyclic communication interrupted	Don't enable power until JS_MC_CyclicIn is valid and cyclic communication is running
50033	jsmcERR_SPAD_FAILURE	Wrong set point acknowledge setting	Use at least XENAX Firmware V3.68H
50034	jsmcERR_INDEX_NOTVALID	Index not valid	Check Index for validity
50035	jsmcERR_VALUE_OUTOFRANGE	Value not in range	Check Input parameter for validity
50036	jsmcERR_FC_INPUTS_NOTVALID	Force calibration inputs not valid	Check ForceCalibration value for validity
50037	jsmcERR_FC_NO_LINEAR	Force calibration only with Jenny Science motors	Check for JS motor recognizing or don't star Force Calibration for third party motor
50038	jsmcERR_FC_REF_ERROR	Force calibration: Error during reference	Check for possible reference drive
50039	jsmcERR_FC_MOTION_ERROR	Force calibration: Error during motion	Check whether the calibration movement can be performed by the motor
50040	jsmcERR_UNKNOWN_MOTORTYPE	Unknown motor type	Check if the firmware version supports the motor
50041	jsmcERR_VIRTUAL_AXIS_RESET_FAILURE	MC_Reset failed	Check ErrorID of MC_Reset

2.11 Error Sources

The error source block can be found in the ErrorRecord output of the JS_MC_ReadLibraryError block. The table associates sources number with the corresponding function block.

ErrorSource Nr.	Error source
1	Axis error or warning
2	JS_MC_CyclicIn
3	JS_MC_Power
4	JS_MC_Reference
5	JS_MC_MoveAbsolute
6	JS_MC_MoveRelative
7	JS_MC_MoveCyclicPosition
8	JS_MC_Stop
9	JS_MC_Halt
10	JS_MC_AxisErrorCollector
11	JS_MC_ReadLibraryError
12	JS_MC_ReadParameter
13	JS_MC_WriteParameter
14	JS_MC_JogVelocity
15	JS_MC_Read_I_Force
16	JS_MC_ReadDigitalInput
17	JS_MC_ReadDigitalOutput
18	JS_MC_WriteDigitalOutput
19	JS_MC_SetPDO
20	JS_MC_ForceCalibration

2.12 Error Type

The error type is important for error handling. Because of that, the error type is provided in the error record in an additional field.

ErrorTyp Nr.	ErrorTyp
1	Axis error
2	Axis warnung
3	Function block error

2.13 Release Notes

Version	Date	Changes
3.1.5	01.06.2023	added support for Lxs with absolute measurement system
3.1.4	30.03.2023	fixed JS_MC_Stop was stuck when internal virtual_MC_Stop.Error was true
3.1.3	07.02.2023	JS_MC_Stop works during reference drive (TwinCat 3.1.4024.7 required to use library)
3.1.2	16.05.2022	Intax Support JS_MC_Reference new reference method which uses settings from Webmotion JS_MC_CyclincIn changes DS402State only once per function call
3.1.1	11.04.2022	JS_MC_Power deadlock fixed when enable = 0 short after enable = 1
3.1.0	26.05.2021	JS_MC_STOP can be used in csp mode JS_MC_ReadAxisError renamed to JS_MC_ReadLibraryError New function block JS_MC_ReadAxisError to read only axis error
3.0.8	21.09.2020	JS_MC_POWER forwards Enable_Positive and Enable_Negative to MC_POWER from Beckhoff every cycle
3.0.7	28.07.2020	Signateq support
3.0.6	27.07.2020	JS_MC_INIT updates axis pointer in case code is changed during run
3.0.5	27.04.2020	JS_MC_LIB is listed under Motion/JennyScience instead of Miscellaneous
3.0.4	21.01.2020	JS_MC_STOP stops Force Calibration
3.0.3	02.12.2019	JS_MC_ForceCalibration option to delete old Force Calibration before starting new one
3.0.1	09.10.2019	Fixes error calling JS_MC_RESET when no error is pending
3.0.0	21.06.2019	Rotax Rxhq support
1.3.0	26.03.2019	PLCopen goes to Standstill after Jog drive
1.1.0	29.01.2018	ROTAX support
1.0.0	21.09.2017	Frist Release

This chapter describes how to put a Jenny Axis into operation. Example projects are used for this purpose. All examples are available in profile position without an NC-Axis for **Tc-PLC** or in cyclic synchronous position mode with an NC-Axis for **NC-PTP**.

3.1 List of Demo Applications

3.1.1 JSC_SimpleDemo

Simple demo application of an axis driving from start position to end position and back in an infinite loop.

3.1.2 JSC_Forceteq_basic_ForceLimit

This demo shows the force limitation part of Forceteq® basic.
The axis drives forward with a limited force. If an obstacle is in the forward path, the force limit will be reached and the axis moves back quickly to the starting position.
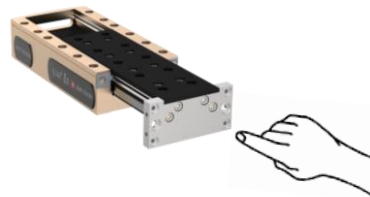
3.1.3 JSC_Forceteq_basic_ForceMonitoring

This is an extended version of the ForceLimit demo project. This example includes a demo of force monitoring where 3 sectors are defined. When the axis detects an obstacle in the forward path, it will evaluate the sectors and show in which sector the obstacle was.
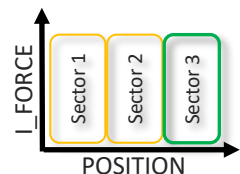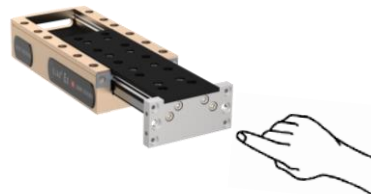
3.1.4 JSC_Forceteq_pro_ForceLimit

This demo application is similar to the ForceLimit demo with Forceteq® basic. The difference is the way force is measured. Forceteq® pro demo employs an external force sensor, while the Forceteq® basic demo measured the motor current. Only possible with XENAX® Xvi 75V8S and Signateq®.
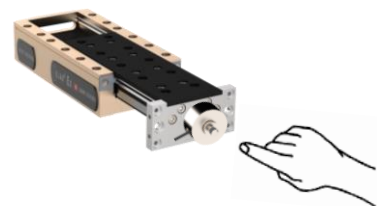
3.1.5 JSC_Forceteq_pro_ForceMonitoring

This is an extended version of the Forceteq® pro ForceLimit demo project. This example includes a demo of force monitoring where 3 sectors are defined. When the axis detects an obstacle in the forward path, it will evaluate the sectors and show in which sector the obstacle was.
Only possible with XENAX® Xvi 75V8S and Signateq®.
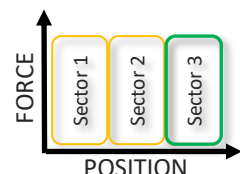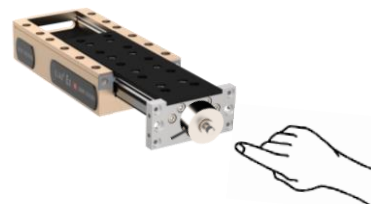
3.2 Open Solution

Note:

The demo projects were created for the XENAX®
Xvi 75V8, except the Forceteq® pro demos for the
Xvi 75V8S. If you are using a different controller than
the Xvi 75V8, please refer to chapter "5 Replacing Xvi
75V8 by Xvi 75V8S" after the import for the steps to
follow.

Start TwinCAT3, select "Open Solution from Archive",
choose the demo project and save it to your project
folder. It is recommended to start with the
"JSC_SimpleDemo" example project.
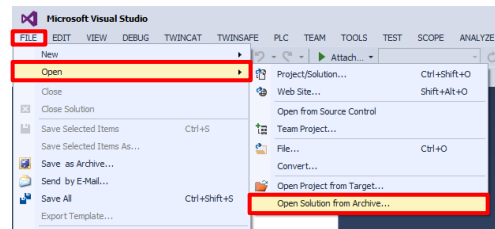
3.3 ESI XML Installation

The EtherCAT Slave Information XML for the XENAX
should always be completely unpacked into the ESI-
directory of the EtherCAT master. In TwinCAT 3 these
files are located in
\TwinCAT\3.1\Config\Io\EtherCAT. This ESI file can
be downloaded from www.jennyscience.ch under
"XENAX Servocontroller->Firmware Bus Module-
>EtherCAT".
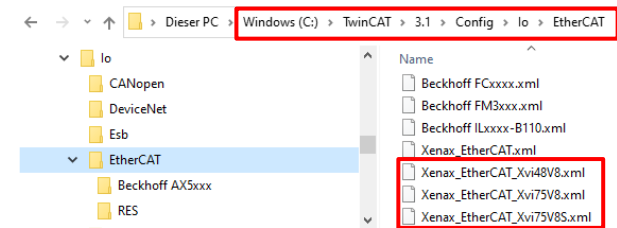
Load ESI file into TwinCAT.
„TwinCAT→EtherCAT Devices→Reload Device
Descriptions"
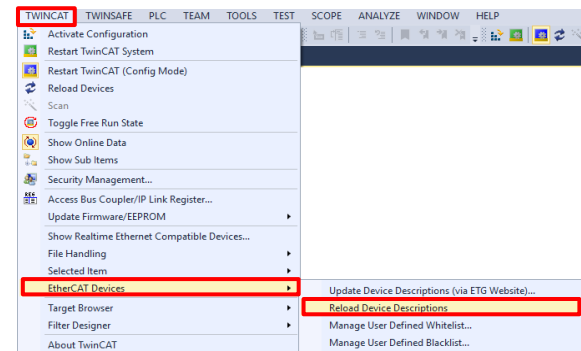
3.4 JS_MC_Lib Library Installation

"PLC→Library Repository→Install… "
Open the "JS_MC_Lib.compiled-library" from the
downloaded folder.
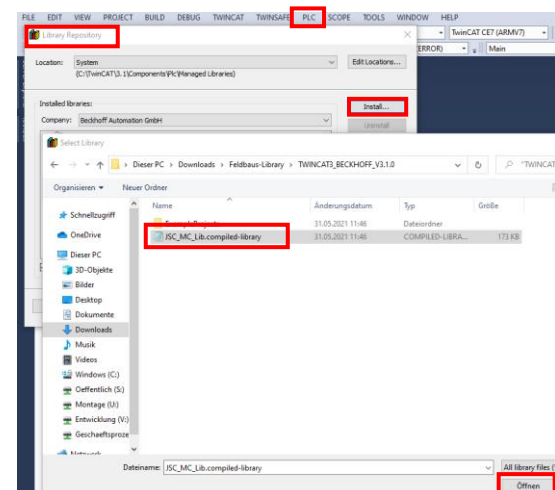
3.5 Choose Target System

Choose the target system to connect to a Beckhoff PLC.
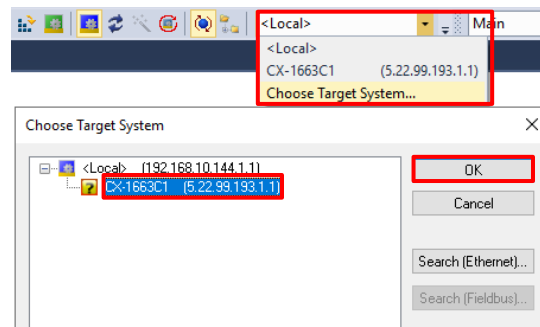
Set the system into Configuration Mode
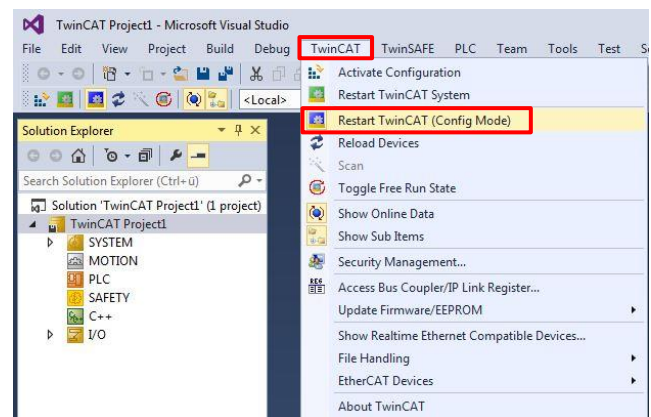
3.6 Scan for Devices

Save the Drive with all the PDO settings to your hard drive.
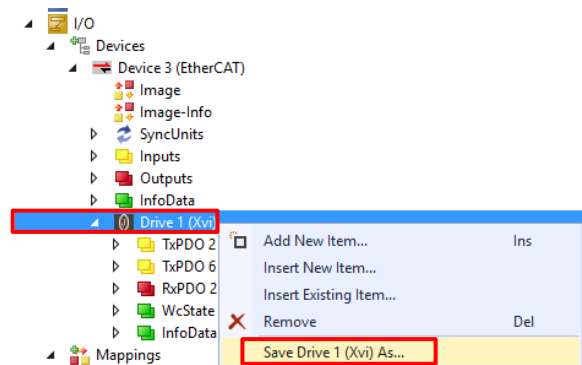
In the example project, the existing devices must be removed first. Right click on "Device X(EtherCAT)" and select remove.

33
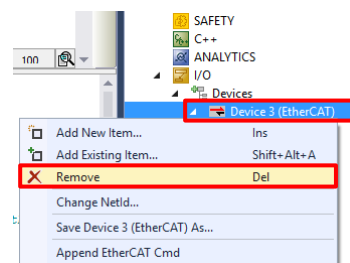
Then right click on Devices and select scan.
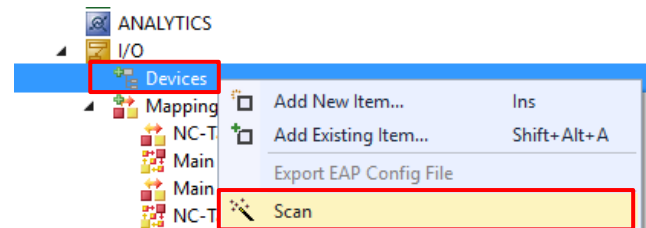
Select desired network interface card.
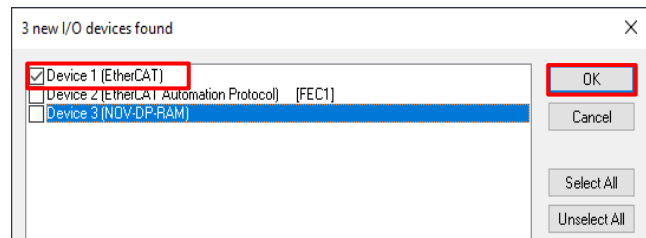
When „Scan for boxes" appears
→ Press YES (JA)
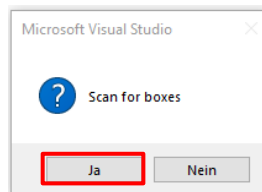
TwinCAT wants to add a Virtual Axis. Press cancel.
The example project already contains a virtual axis if
one is needed.
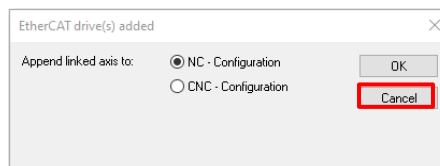
When „Activate Free Run" appears
→ Press No (Nein)
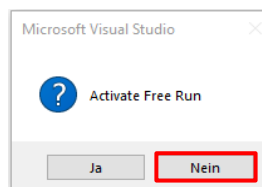
Remove the Drive which was found during the scan.
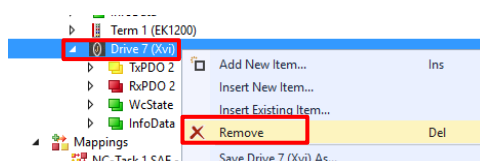
Add the Drive from your hard drive with the correct
PDO mapping.
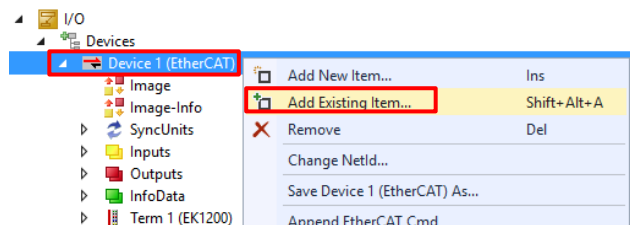
Reconnect imported links (Yes).
This should recover all links. If the links are not
correctly recovered, link it manually as described in
chapter "4.11 PDO-Linking".
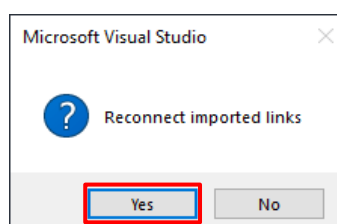
Double click Real-Time and press "Read from Target". This will assign the PLC tasks to a free CPU on the target device.
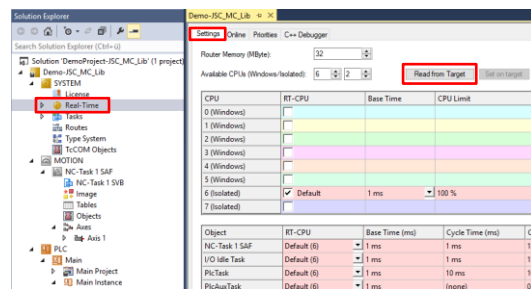
Compile project with „Build→Rebuild Solution" There must be 0 errors!
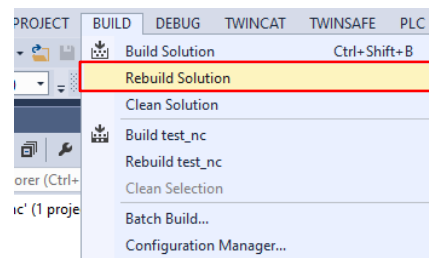
Sometimes, the project must be closed and opened again to apply the new realtime settings.

3.8 Launch Demo Project

Launch the example project by activating the configuration.
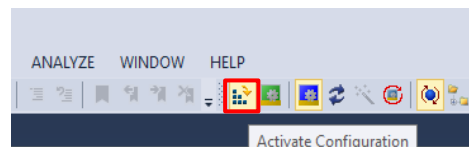
go online
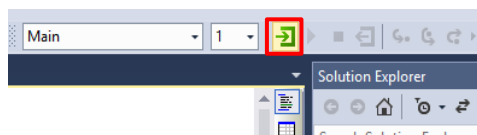
Start program
The Axis will start moving. Make sure that nothing can be damaged.
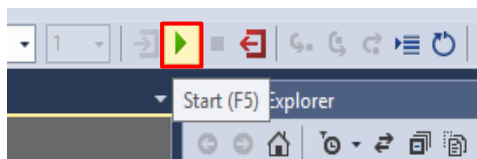
4 New Project in TwinCAT3

This chapter describes how to put a Jenny Science Axis into operation without a demo project. It is possible to create a new project or to add a Jenny Science axis into an existing project.

4.1 Create Project

File→New→Project…
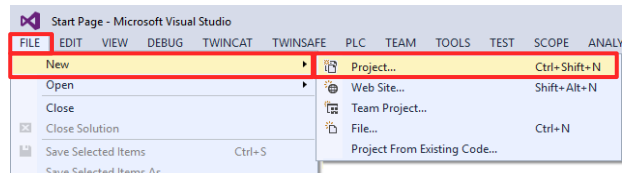
Create a new TwinCAT XAE Project.
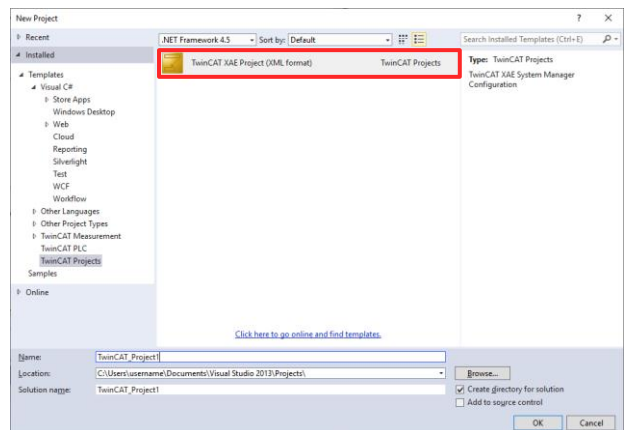
4.2 ESI XML Installation

The EtherCAT Slave Information XML for the XENAX should always be completely unpacked into the ESI-directory of the EtherCAT master. In TwinCAT 3 these files are located in **\TwinCAT\3.1\Config\Io\EtherCAT**. This ESI file can be downloaded from www.jennyscience.ch under "XENAX Servocontroller->Firmware Bus Module ->EtherCAT".
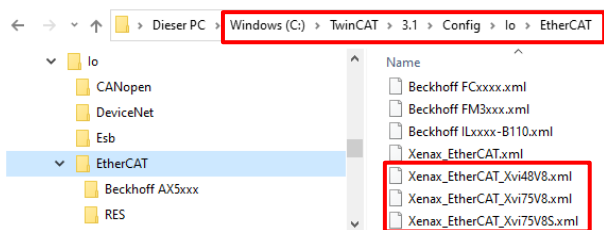
Load ESI file into TwinCAT. „TwinCAT→EtherCAT Devices→Reload Device Descriptions"
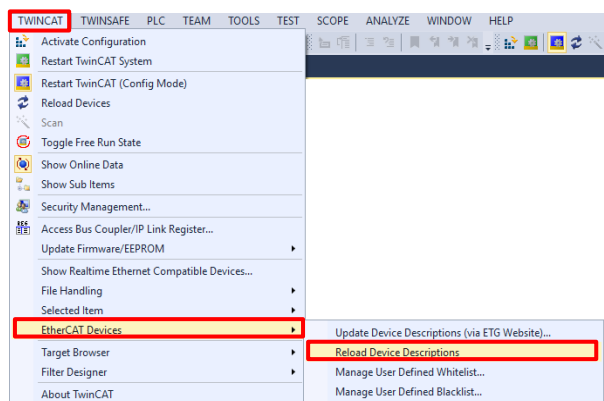
4.3 JS_MC_Lib Library Installation

"PLC→Library Repository→Install… "
Open the "JS_MC_Lib.compiled-library" from the downloaded folder.
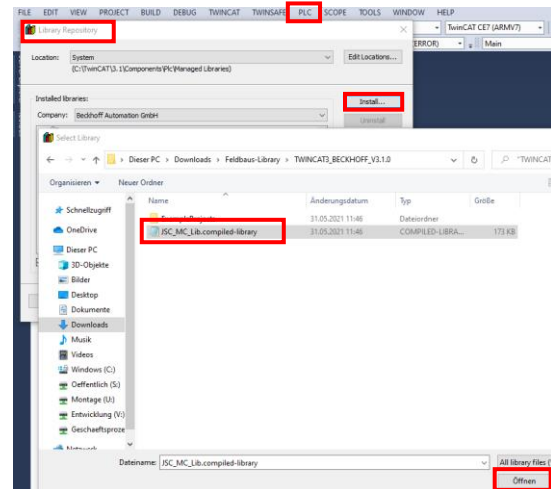
After the installation, you can include the library in the project.
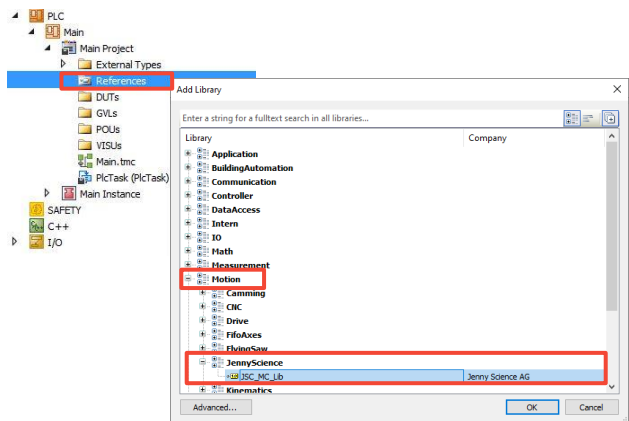
The Tc2_MC2 lib from Beckhoff is required too in cyclic synchronous position mode.
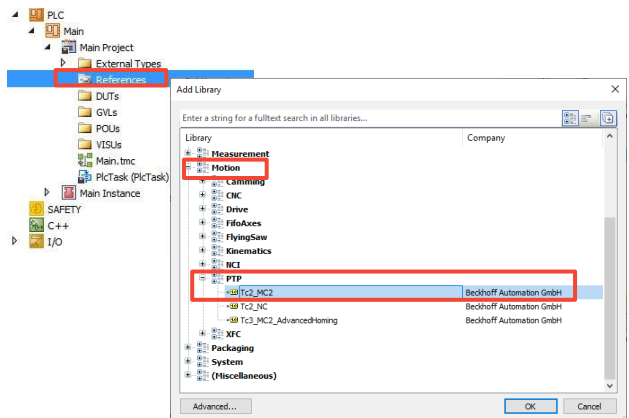
4.4 Choose Target System

Choose the target system to connect to a Beckhoff PLC.
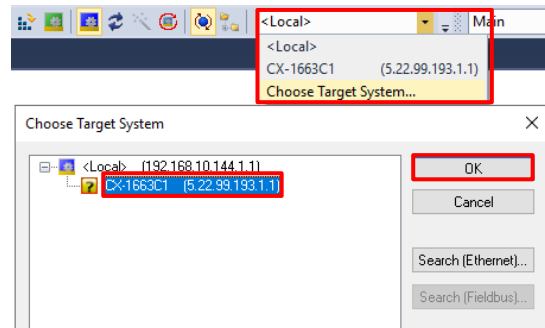
Set the system into Configuration Mode.
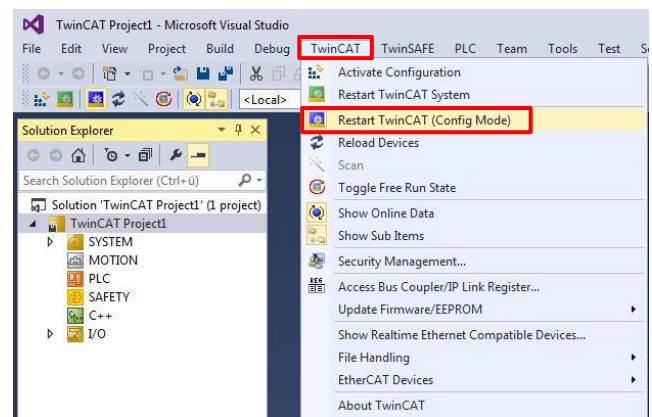
4.5 Scan for Devices

Right click on Devices and select "Scan".
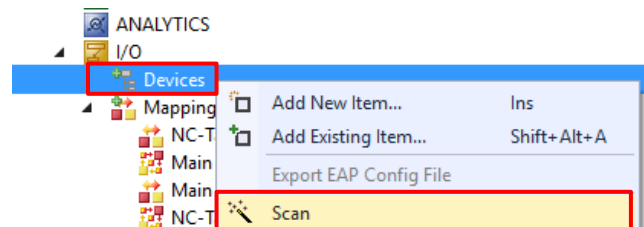
Select desired network interface card.

When „Scan for boxes" appears
→ Press YES (JA)
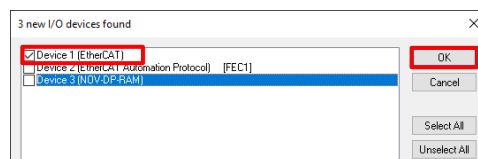
cyclic synchronous position mode
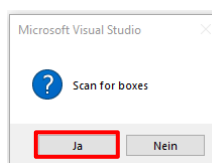
The next step depends on the chosen drive mode. More information about drive modes can be found in chapter "2.1 Drive Modes: point to point or interpolated". Cyclic synchronous position mode requires a virtual NC-Axis whereas profile position mode does not need one.
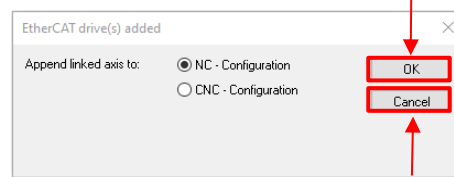
profile position mode
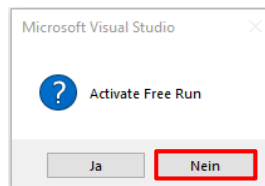
When „Activate Free Run" appears → Press No (Nein)

4.6 Disable Status Word Bit 12

With TwinCAT® build 4022 or later, Status Bit 12 must be disabled.
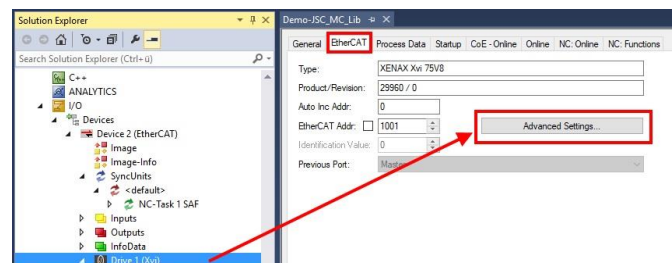
Under Drive n (Xvi) → EtherCAT → Advanced Settings → Mailbox → CoE enable the last option.
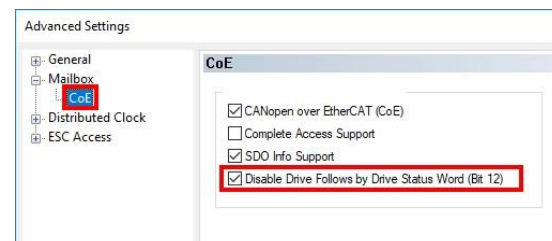
4.7 Cycle Time

In „SYSTEM→Real-Time" (Tab Settings) set the base time of the CPU.
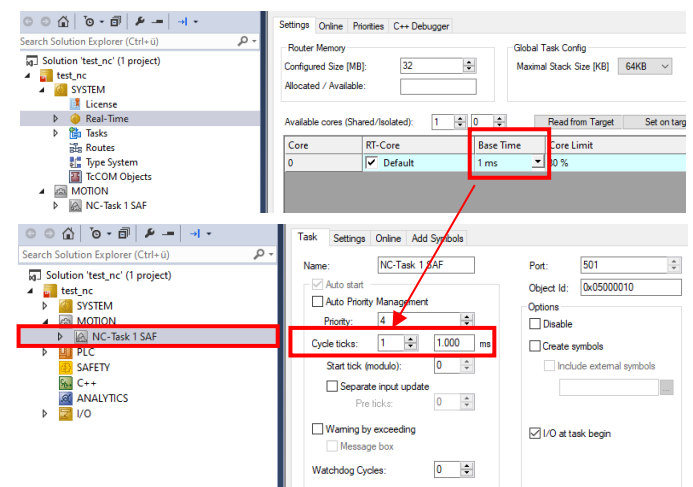
In the cyclic synchronous position mode, the cycle tick on the bus can be set in „MOTION→NC-Task 1 SAF". This time can only be greater than or equal to the base time. This is the objects transmission cycle.
Possible values are from 200us to 2ms.
Typical is 1ms.
Only multiples of 100us are allowed.

4.8 Virtual Axis Configuration (Only Nc-PTP)

This subchapter is only required for cyclic synchronous position mode where a virtual NC-axis is used.

Select the position measuring unit.

Linear Axis: **mm**
Rotary Axis: **Degree**
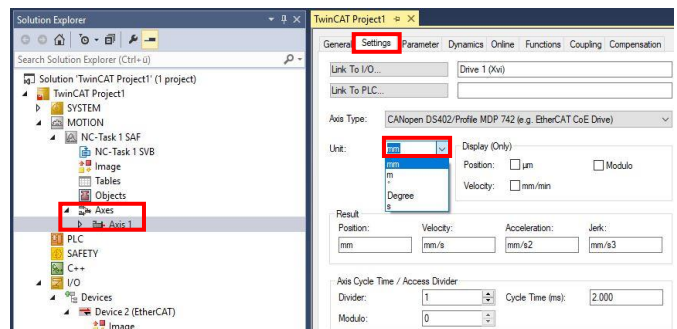
Set the velocity parameters according to the table below.
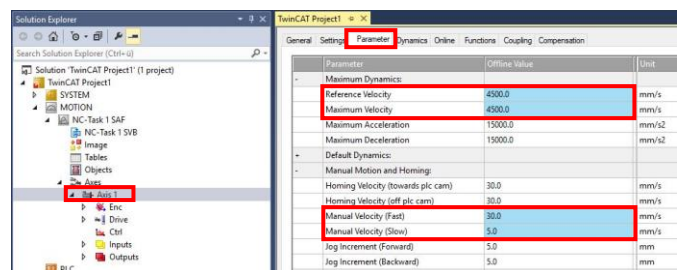
Motor typ	Increments per Revolution	Scale resolution	Reference Velocity [mm/s] or [Degree/s]	Maximum Velocity [mm/s] or [Degree/s]	Manual Velocity (Fast) [mm/s] or [Degree/s]	Manual Velocity (Slow) [mm/s] or [Degree/s]
LINAX®, ELAX®, INTAX®	-	1 um/inc	4500	4500	30	5
LINAX®	-	100 nm/inc	900	900	30	5
ROTAX® Rxvp	64'000	0.005625 deg/inc	9000	9000	90	15
ROTAX® Rxhq 50	120'000	0.003 deg/inc	14400	14400	90	15
ROTAX® Rxhq 110	120'000	0.003 deg/inc	5220	5220	90	15
ROTAX® Rxhq 110	162'000	8 arcsec/inc	5220	5220	90	15
ROTAX® Rxhq 110	648'000	2 arcsec/inc	5000	5000	90	15
ROTAX® Rxhq 110	2'592'000	0.5 arcsec/inc	1250	1250	90	15

The „Position Lag Monitoring" has to be disabled. Set "Position Lag Monitoring" to FALSE under Parameters.
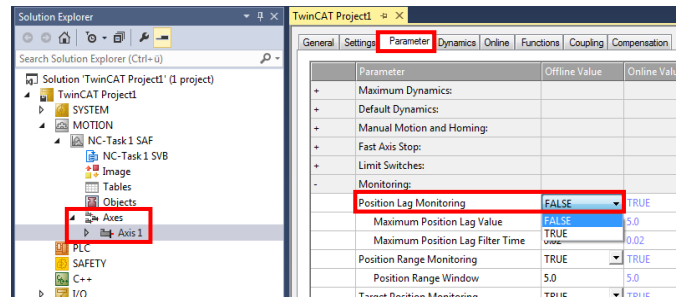
Adjust „Scaling Factor Numerator" to the resolution of the used motor.
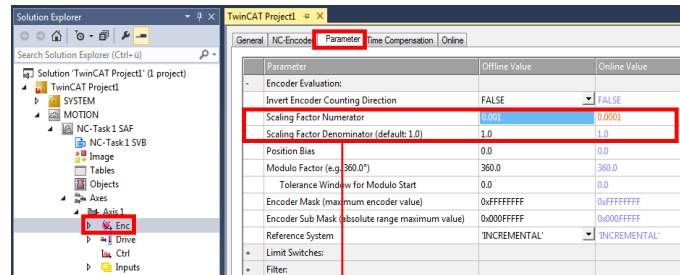
Motor typ	Increments per Revolution	Scale resolution	Scaling factor Numerator	Scaling factor Denumerator
LINAX®, ELAX®, INTAX®		1 um/inc	0.001	1
LINAX®		100 nm/inc	0.0001	1
ROTAX® Rxvp	64'000	0.005625 deg/inc	360	64'000
ROTAX® Rxhq 50	120'000	0.003 deg/inc	360	120'000
ROTAX® Rxhq 110	120'000	0.003 deg/inc	360	120'000
ROTAX® Rxhq 110	162'000	8 arcsec/inc	360	162'000
ROTAX® Rxhq 110	648'000	2 arcsec/inc	360	648'000
ROTAX® Rxhq 110	2'592'000	0.5 arcsec/inc	360	2'592'000

In Axis Drive set, „Following Error Calculation" to „Extern"
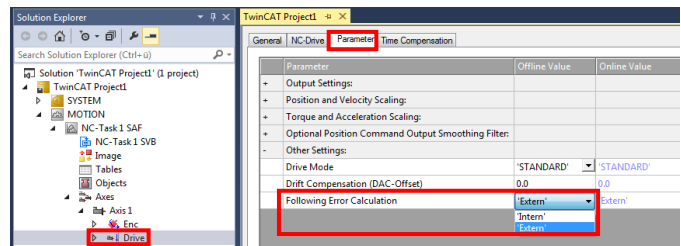
Since the XENAX® servo controller has an integrated NC-controller, the NC-internal controller has to be disabled by selecting „SERCOS controller (Position by SERCOS)".
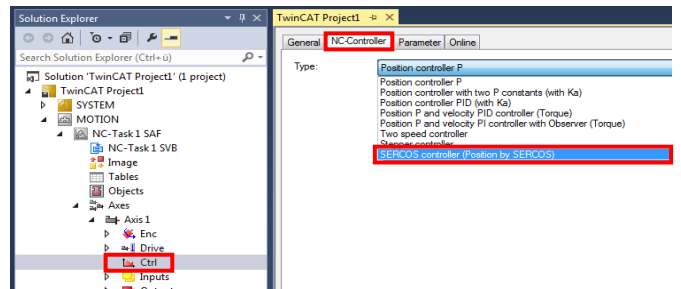
4.9 Manual Control

The configuration is activated by pressing on the highlighted icon.
The following two messages "Activate Configuration" and "Restart TwinCAT System in Run Mode" can be answered with OK.
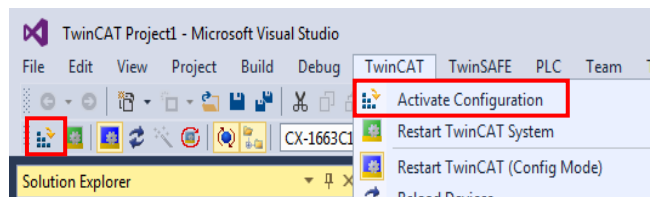
In "I/O→Devices→Device 1(EtherCAT)→Drive 1", tab: CoE – Online, double-click on object "6060".

When the window "Set Value Dialog" is opened, activate the reference mode
by putting the object "6060" on 6.
This reference mode has to be activated each time the XENAX® servo controller is restarted.
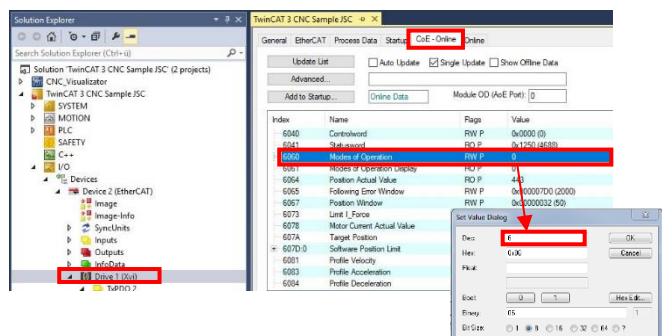
In "MOTION→NC-Task 1 SAF→Axes→Axis 1", tab: Online, the value (position) in the picture, should appear in black. If the value (position) appears in grey, there are communication problems between Beckhoff and XENAX® over EtherCAT.
Please check the connection, the power supply and the settings.
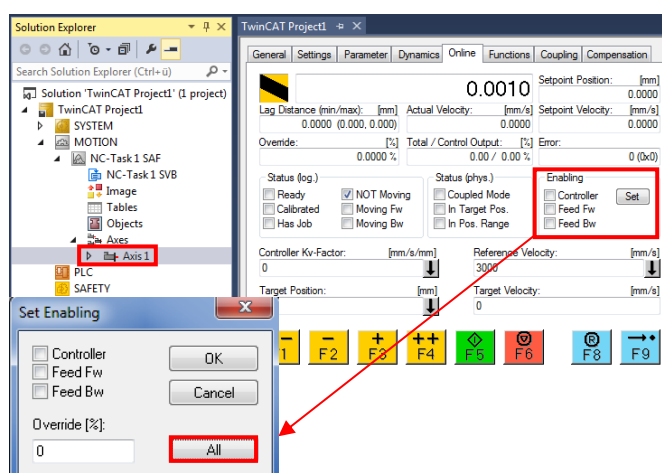
In "Enabling", press "Set". Then press "All" in the window that was opened. This window will be closed automatically. The motor will be unlocked and the reference will be executed.
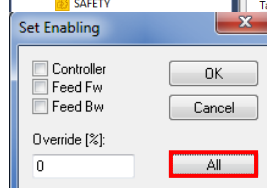
Please wait until the linear motor axis is again in standstill.
Transfer the absolute position of the axis to TwinCAT by pressing the button "Reset" (F8).
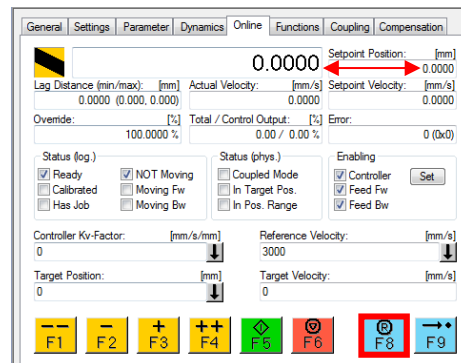
Activating "Cyclic Synchronous Position Mode": "I/O→Devices→Device 1 (EtherCAT)→Drive 1 (Xvi)", tab: „CoE – Online" double click on object "6060".
"Cyclic Synchronous Position Mode" is activated by setting the object 6060 on 8. The motor is now ready to drive.
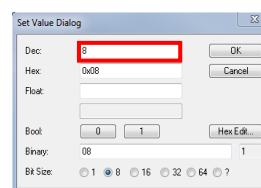
Under "MOTION→NC-Task 1 SAF→Axes→Axis 1", open the "Functions" tab. Select the start mode such as positive motion, negative motion, reversing sequence etc. Enter the remaining parameters such as target positions and set the LINAX® linear motor axis in motion by pressing "Start".
In the picture on the right side, the motor moves back and forth from the Target Position 1 to Target Position 2.
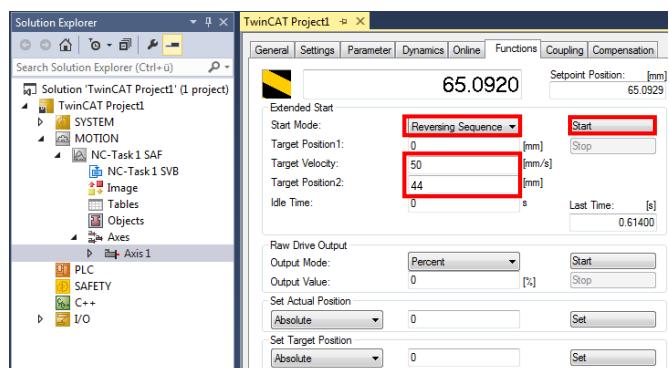
This is a function test to check the communication between Beckhoff and XENAX®.

4.10 PDO-Mapping

4.10.1 Required PDO Configuration

The required PDO configuration depends on the chosen drive mode from chapter "2.1 Drive Modes: point to point or interpolated".
Under Drive X→Process Data enable PDOs from the figures below.

4.10.1.1 Cyclic Synchronous Position Mode

Enable Output PDO 0x1600

Enable input PDO 0x1A00 and 0x1A01

4.10.1.2 Profile Position Mode

Enable Output PDO 0x1613

Enable Input PDO 0x1A12

4.10.2 Optional PDO's

Optional PDO's can be mapped to utilise the full scope of the Axis. Enable additional PDO's according to your needs.

Inputs:

PDO Entry Name	Parameter	PDO GROUP CSP*	PDO GROUP PP**	Description
I_Force Actual	0x2005	0x1A10	0x1A13	Provides the motor current in [mA], e.g. for monitoring a force process. This PDO is needed by the function block JS_MC_Read_I_Force
Force Actual	0x200A	-	-	Provides the actual Force value in [mA], e.g. for monitoring a force process. This PDO is needed by the function block JS_MC_Read_Force. (Only with Xvi 75V8S and Signateq® measuring amplifier)
Process Status Register	0x2006	0x1A01	0x1A12	Provides additional information about the Axis (See XENAX® Manual command "TPSR" for detailed information). This PDO is needed by the function block JS_MC_ReadPSR
Following Error Actual Value	0x60F4	0x1A00	-	Provides the actual deviation of calculated trajectory position and measured position on encoder.
Digital Inputs	0x60FD	0x1A10	0x1A13	Provides the digital input status of the XENAX®. This PDO is needed by the function block JS_MC_ReadDigitalInput

Outputs:

PDO Entry Name	Parameter	PDO GROUP CSP*	PDO GROUP PP**	Description
Limit I_Force	0x6073	0x1611	0x1614	Limits the motor current in [x10mA] which corresponds to the motor force or torque. This PDO is needed by the function block JS_MC_WriteLimit_I_Force
Limit Force	0x2009	-	-	Force limitation in [mN] based on the value measured by the Signateq® measuring amplifier. This PDO is needed by the function block JS_MC_WriteLimit_Force. (Only with Xvi 75V8S and Signateq® measuring amplifier)
Torque Offset	0x60B2	0x1612	-	Sends the target acceleration along with the target position and velocity, resulting in an even smoother and more accurate drive. See more details in chapter "4.10.2.2 Torque (Acceleration) Offset".
Target Velocity	0x60FF	0x1612	-	Sends the target velocity along with the target position resulting in a smoother and more accurate drive. See more details in chapter "4.10.2.1 Target Velocity".
Physical Outputs	0x60FE	0x1611	0x1614	Control the digital output signals of the XENAX®.

CSP*: Cyclic Synchronous Position Mode
PP**: Profile Position Mode

Enable optional PDO's

4.10.2.1 Target Velocity

Hint:

Target Velocity is optional. It sends the target velocity along with the target position to the XENAX®. This results in a smoother drive and a lower deviation from the target position.

Make sure that PDO 0x1612 is enabled.
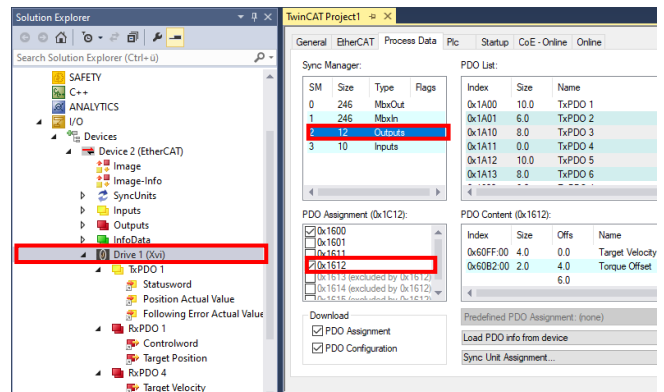
The link is automatically made to nDatadOut2 of the corresponding Axis.
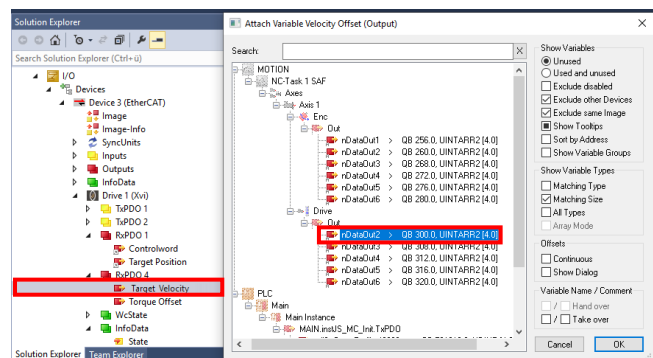
Under Axis→Drive set the Ouput Scaling Factor (Velocity) to 125.0. This parameter does not depend on the motor type.
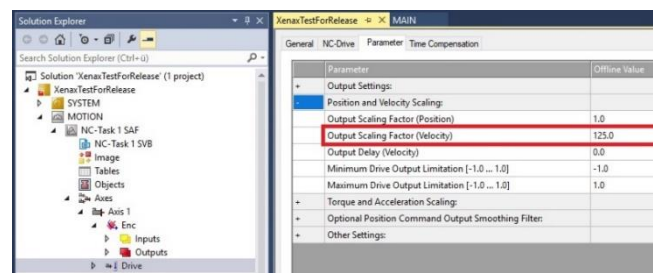
Attention:

This link often gets lost when another PDO is added or removed. An unlinked Target Velocity PDO increases deviation from the target position and results in a rougher drive. An unlinked Torque Offset PDO has similar results.

4.10.2.2 Torque (Acceleration) Offset

Hint:

Acceleration feed forward is optional. It sends the target acceleration along with the target position and target velocity to the XENAX®. This is necessary if the Axis must follow a contour path with high precision.

The link is automatically made to nDataOut3[0] of the corresponding Axis.
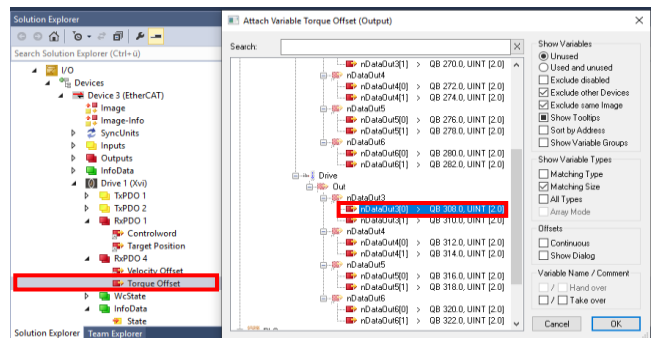
Adjust „Ouput Scaling Factor (Acceleration)" to the resolution of the used motor.
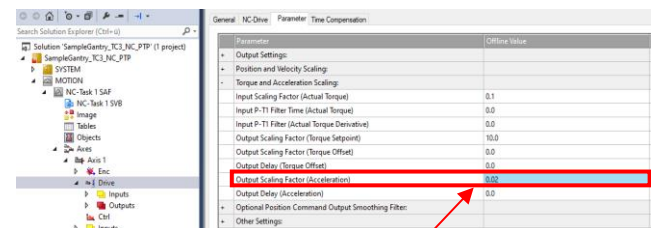
Motor typ	Increments per Revolution	Scale resolution	Output Scaling Factor
LINAX®, ELAX®, INTAX®		1 um/inc	0.02
LINAX®		100 nm/inc	0.2
ROTAX® Rxvp	64'000	0.005625 deg/inc	0.00355
ROTAX® Rxhq 50	120'000	0.003 deg/inc	0.00666
ROTAX® Rxhq 110	120'000	0.003 deg/inc	0.00666
ROTAX® Rxhq 110	162'000	8 arcsec/inc	0.009
ROTAX® Rxhq 110	648'000	2 arcsec/inc	0.036
ROTAX® Rxhq 110	2'592'000	0.5 arcsec/inc	0.144

Attention:

This link often gets lost when another PDO is added or removed. An unlinked Torque Offset PDO increases deviation from the target position and results in a rougher drive. Note that the Torque Offset PDO requires the Target Velocity PDO to be mapped and linked.

4.11 PDO-Linking

The configured PDO's must be linked to their variable. Follow the instructions of chapter "4.11.1 for profile position" or chapter "4.11.2 for cyclic synchronous position mode".

4.11.1 Profile Position Mode

Links between PDO and PLC must be done manually. Double click on a PDO and select a variable which should be linked.

Link Statusword 0x6041 to "uiStatusword_I6041"
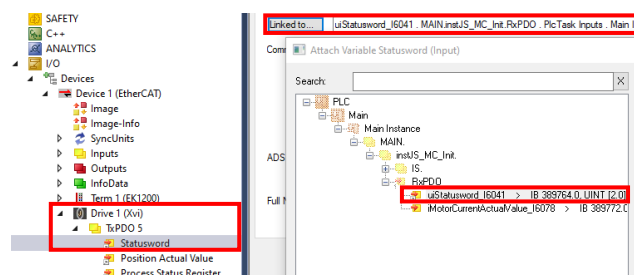
Link Position Actual Value 0x6064 to "diPositionActualValue_I6064"
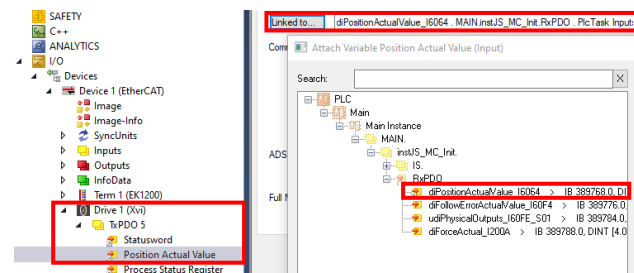
Link Process Status register 0x2006 to "diProcessStatusRegister_I2006"
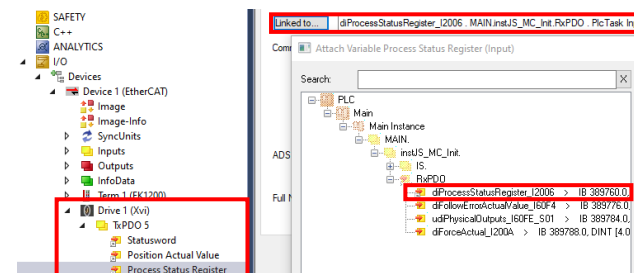
Link Controlword 0x6040 to "uiControlword_I6040"
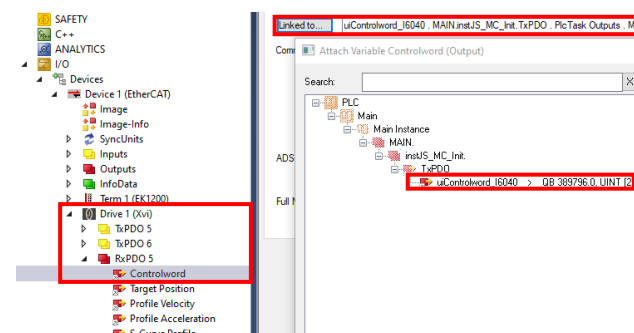
Link Target Position 0x607A to "diTargetPosition_I607A"
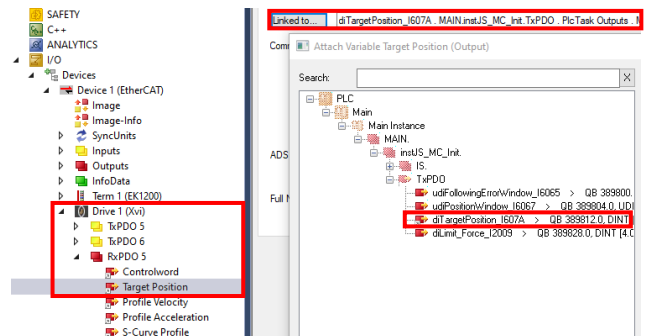
Link Profile Velocity 0x6081 to "udiProileVelocity_I6081"
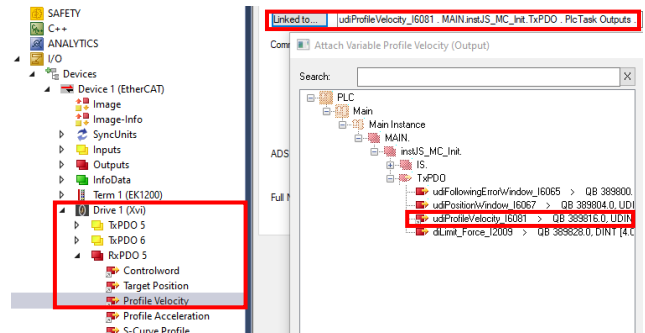
Link Profile Acceleration 0x6083 to "udiProfileAcceleration_I6083"
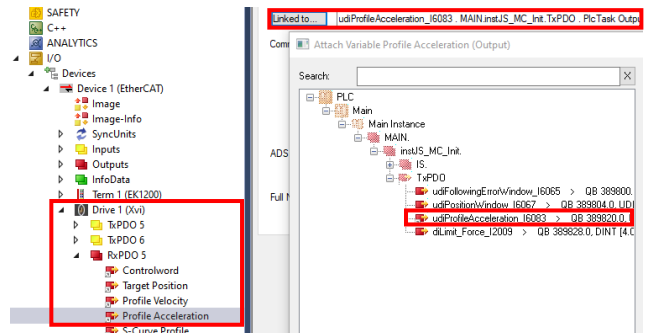
Link S-Curve Profile 0x2000 to "udiS_CurceProfile_I2000"
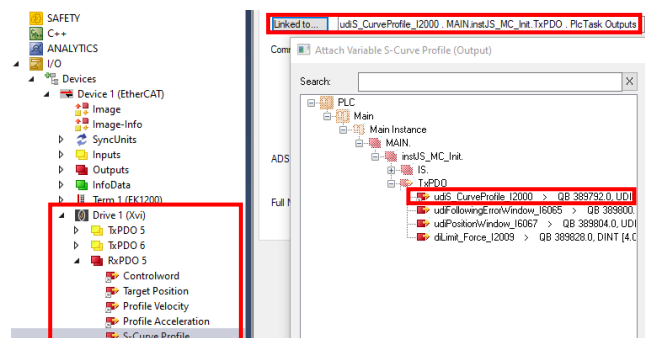
Link WcState to "bWcState"
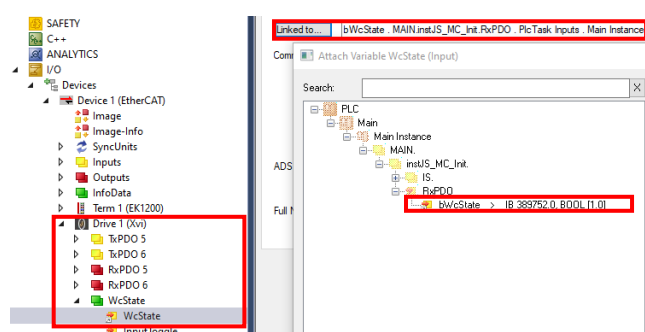
Link netId
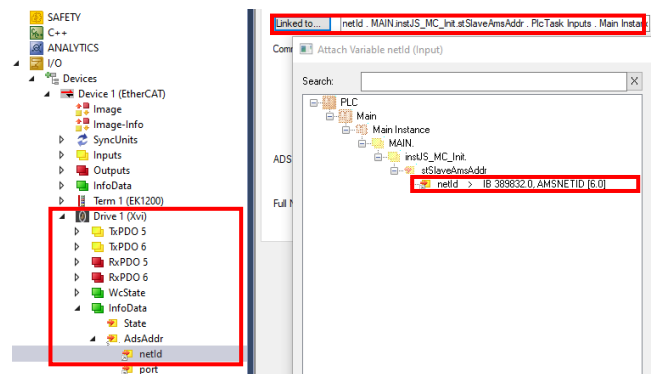
Link port
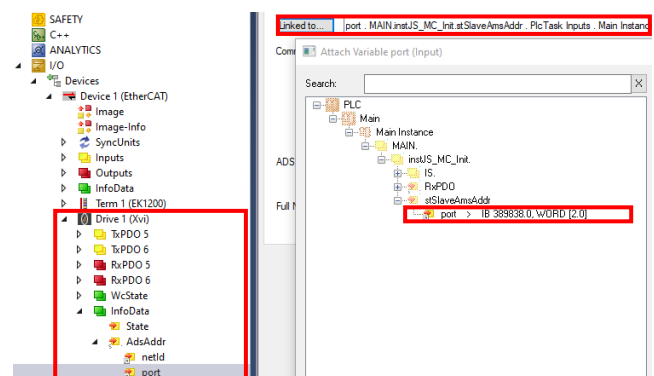
Link also all optional PDO's to the PLC if there are any.

4.11.2 Cyclic Synchronous Position Mode

At least an instance of the JS_MC_Init in the PLC program is required for the following steps. It is recommended to copy the source code from the JSC_SimpleDemo example project.

Building the PLC project will update the interface between PLC and NC-Axis.
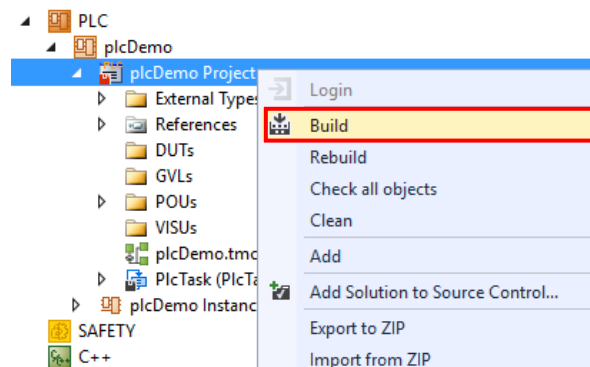
Make sure that the NC-Axis is linked to the drive and to the PLC.

TxPDO 1 and RxPDO 1 are mapped automatically to the NC-Axis. Links to the PLC must be done manually. Double click on a PDO and select a variable which should be linked.
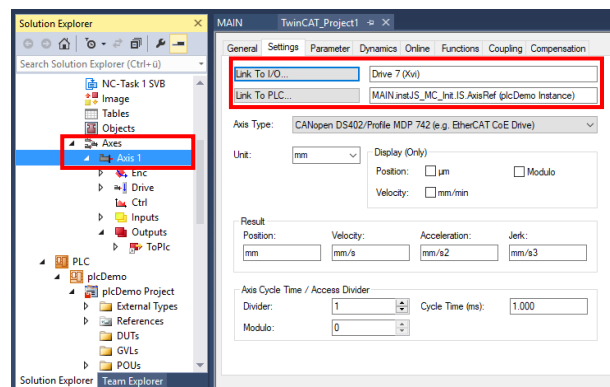
Link Statusword 0x6041 from TxPDO 2 to "uiStatusword_I6041" from the MAIN.
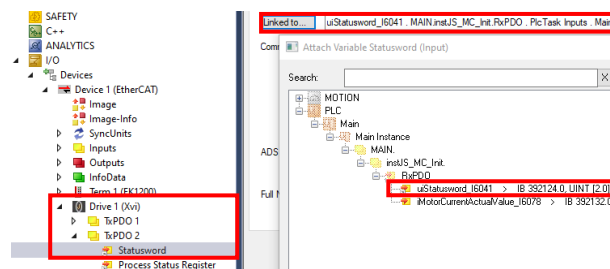
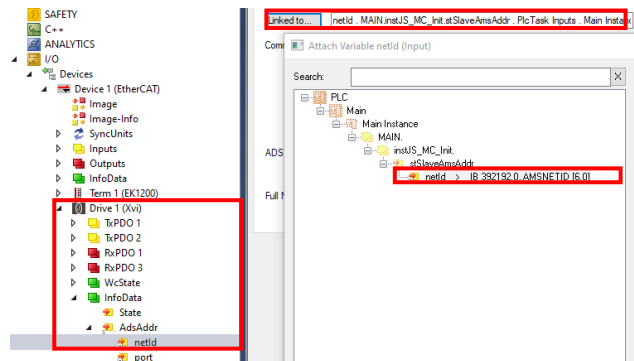Link Process Status Register 0x2006 from TxPDO 2 to "diProcessStatusRegister_I2006" from the Main.
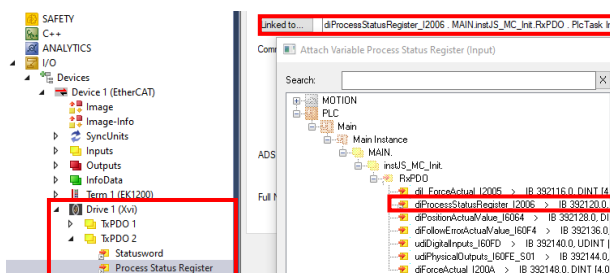
The WcState must be linked to three variables. Use CTRL and click to add a link to "bWcState". Note that the third link often gets lost when the PDO mapping gets changed.
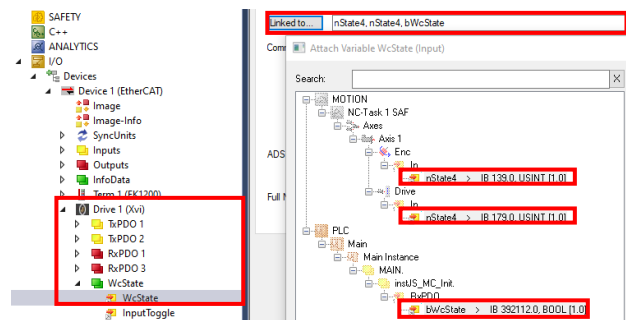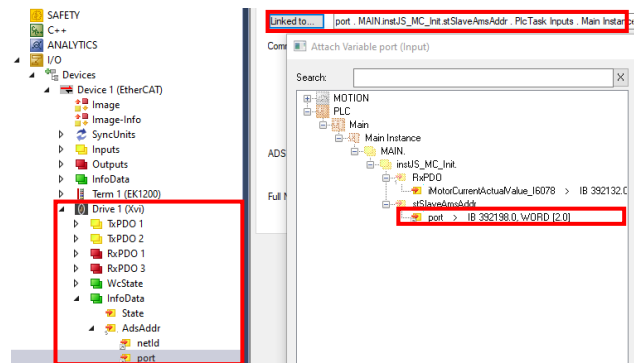

Click OK two times here.
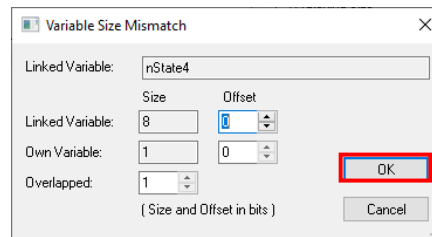
Link netId to "netId" of the Main.


Link port to "port" of the Main.

Note:
Link all optional PDOs to the PLC if there are any. Make sure that "Target Velocity", "Torque Offset" and "WcState" are still linked.

5 Replacing Xvi 75V8 by Xvi 75V8S

To replace a XENAX® Xvi 75V8 with an Xvi 75V8S in an existing project, the following steps must be done.

5.1 ESI XML Installation

The EtherCAT Slave Information XML for the XENAX Xvi 75V8S should always be completely unpacked into the ESI-directory of the EtherCAT master. In TwinCAT 3 these files are located in **\TwinCAT\3.1\Config\Io\EtherCAT**. This ESI file can be downloaded from www.jennyscience.ch under "XENAX Servocontroller->Firmware Bus Module->EtherCAT".
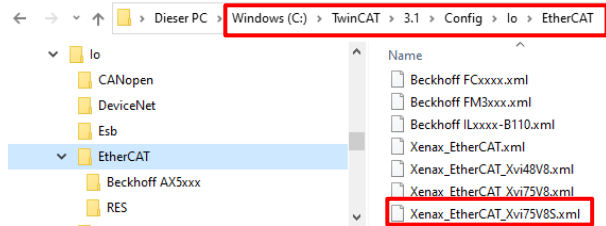
Load ESI file into TwinCAT. „TwinCAT→EtherCAT Devices→Reload Device Descriptions"
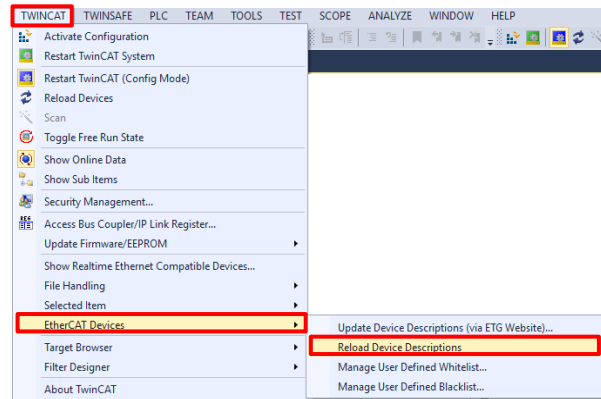
Change the type of the selected XENAX® controller „Change to Compatible Type…", then choose the right entry and press OK.
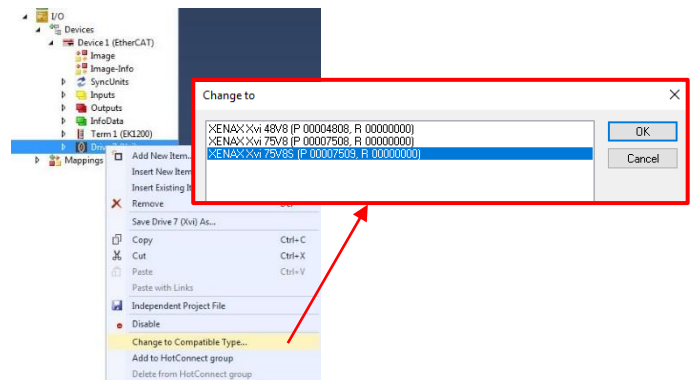
Notes

This instruction manual contains copyright protected information. All rights are reserved. This document may not be entirely or partially copied, duplicated or translated without the prior consent of Jenny Science AG.

Jenny Science AG grants no guarantee on, or will be held responsible for, any incidents resulting from false information.

Information in this instruction manual might be subject to change.

Jenny Science AG
Sandblatte 11
CH-6026 Rain

Tel +41 (0) 41 255 25 25

www.jennyscience.ch
info@jennyscience.ch