Instruction Manual JS_MC_LIB with XENAX® Xvi, EtherCAT® Busmodule and Indraworks.

Version 1.0.0

Edition 13. January 2020

XENAX® Ethernet servo controller with EtherCAT® Busmodul

Introduction

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

Contents

1 Development Environment

1.1 Bosch Rexroth

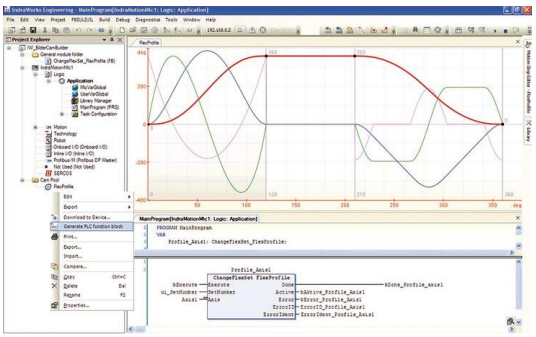
1.1.1 Programmable Logic Controller

Rextroth PLC with EtherCAT interface.

The programming of the machine control is operated with IndraWorks.

1.1.2 IndraWorks

In order to program Rexroth PLCs the development software IndraWorks is required.



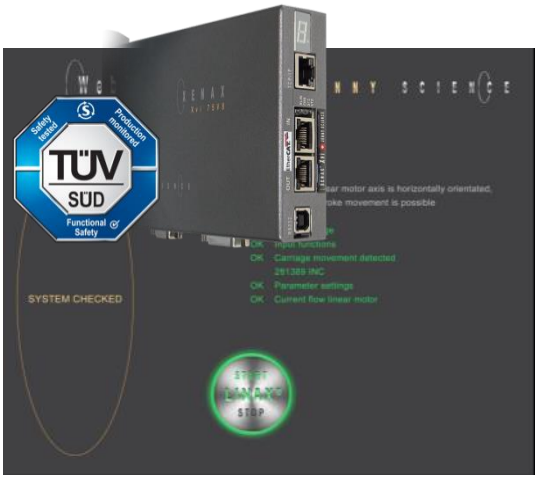
1.2 Jenny Science

1.2.1 WebMotion

The proprietary graphical user interface for servo controllers 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 and ROTAX® rotary axes are automatically recognized. The corresponding controller parameters are saved and loaded automatically. With the Quick Start button, the linear motors can easily and immediately be operated. No other user manual is needed.



1.2.2 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.

1.2.3 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

1.2.4 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.

1.2.5 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

1.3 Required Resources

The following resources are needed for the successful operation of the XENAX® servo controller with an EtherCAT bus module.

1.3.1 ESI EtherCAT Slave Information

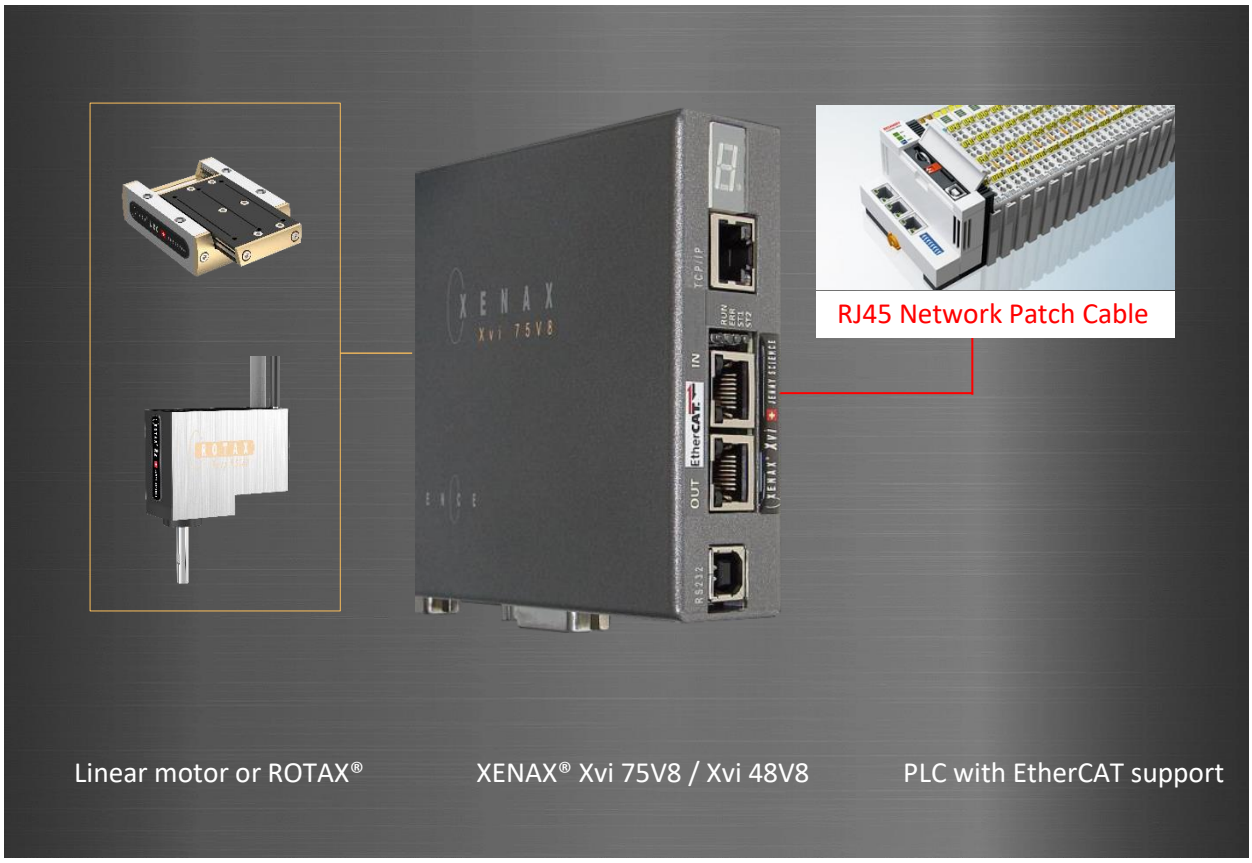
The ESI XML file is available on our website https://www.jennyscience.ch/en/products/download under XENAX® → Firmware Bus Module.

1.3.2 JS_MC_Lib

The Jenny Science Motion Control library is included in the zip file of this manual or it can be downloaded from our website https://www.jennyscience.ch/en/products/download under XENAX® → PLCopen Library.

1.3.3 RJ45 Patch Cable

Configuration to use the example projects:



1.4 Software Requirements

Software	Version
IndraWorks Software	14V22 (older versions should work, but are not tested)
XENAX® firmware	4.10 or later
EtherCAT bus module firmware	2.21 or later

2 PLCopen Library (JsMcLib)

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

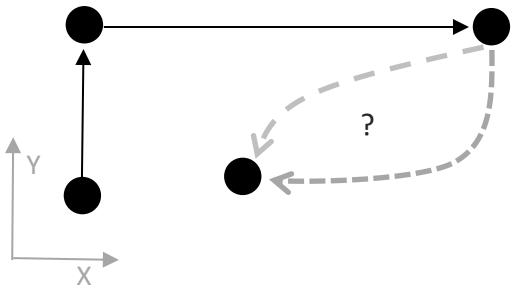
2.1 Drive Modes: point to point or interpolated

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**.

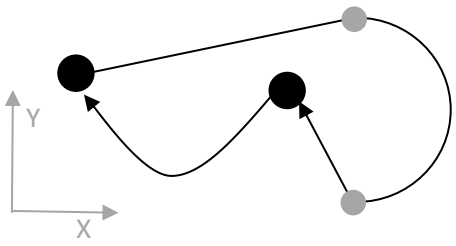
XY-Axis Profile Position



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 Rexroth 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.

2.2 State Diagram

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.

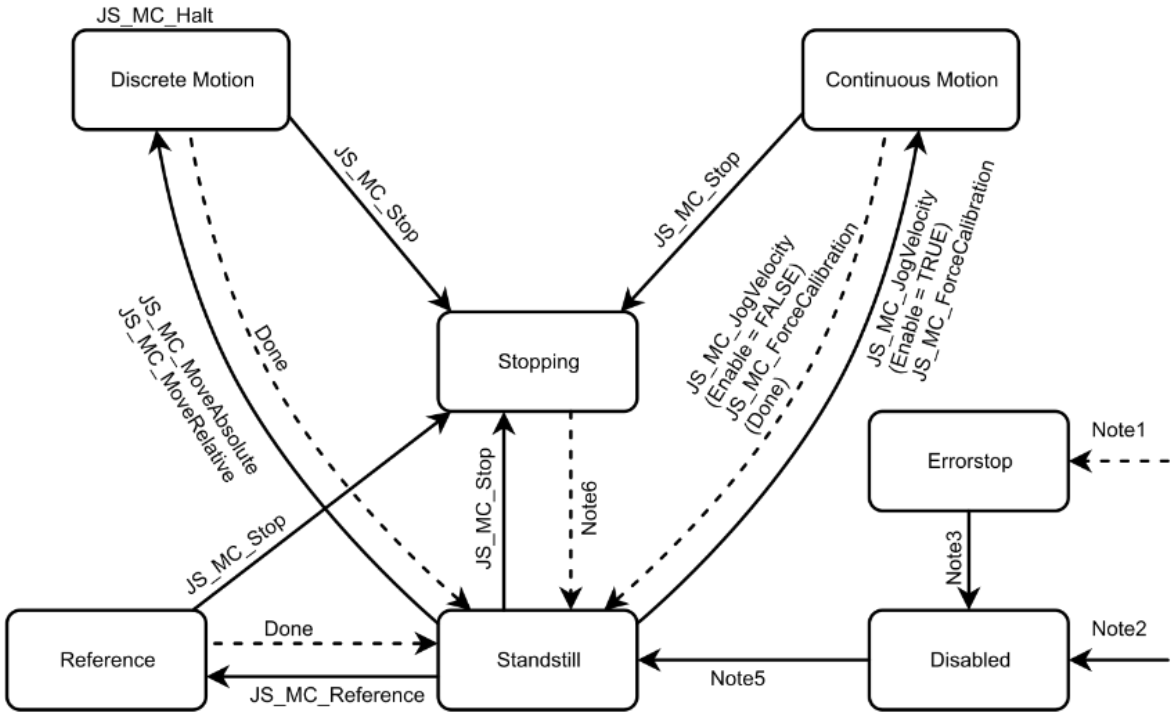
All function blocks which do not appear in the state diagram, do not affect the state of the axis.

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.

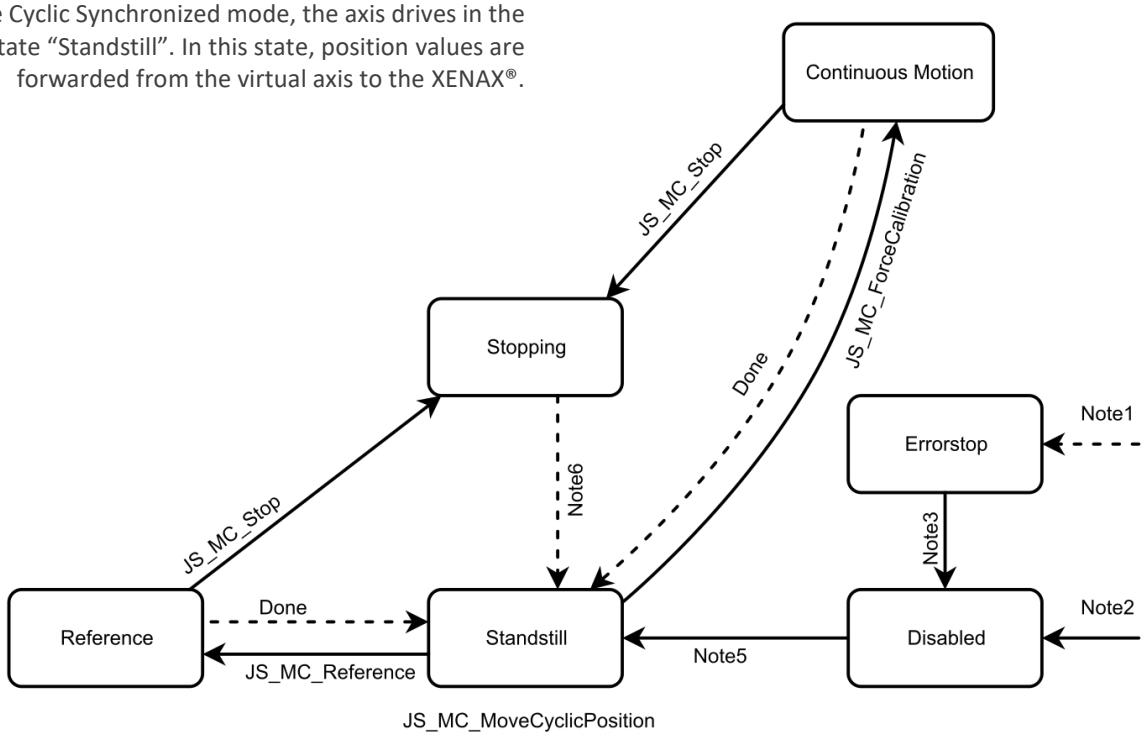
Important:
In the states “**Stopping**”, “**ErrorStop**”, ”**Disabled**” and “**Reference**” no motion blocks can be called. In standstill condition, an axis must always be referenced before starting a movement.

2.2.1 Profile Position Mode



2.2.2 Cyclic Synchronous Position Mode

In the Cyclic Synchronized mode, the axis drives in the state “Standstill”. In this state, position values are forwarded from the virtual axis to the XENAX®.



Note 1:
From any state. An error in the axis occurred.
Note 2:
From any state. JS_MC_Power.Enable = FALSE and there is no error in the axis.
Note 3:
JS_MC_Reset AND JS_MC_Power.Status = FALSE.
Note 5:
JS_MC_Power.Enable = TRUE AND JS_MC_Power.Status = TRUE
Note 6:
JS_MC_Stop.Done = TRUE AND JS_MC_Stop.Execute = FALSE

2.3 Function blocks

The functionality of the JsMcLib is implemented in various small function blocks. In this subchapter, all those function blocks are described. Demo programs in the subsequent chapters will show the function blocks in action.

If Indraworks has a similar function block, the JsMcLib function block is implemented based on the original block. Note that the Units can be a bit confusing. The Units are always in increments. Velocity is in increments per minute while acceleration and deceleration are in increments per second^2.

Function Block Input Name	**Unit**
Position	Increments
Velocity	Increments/**min**
Acceleration	Increments/**s^2**
Deceleration	Increments/**s^2**

2.3.1 INIT

This function block must be called once at start up. It provides a reference to the axis which is needed in all other JsMcLib function blocks. The function block must be called before any other JsMCLib block is called.

JS_MC_INIT	
Input	Output
AdrIoMap pDevice Node OperationMode	Axis

2.3.2 CyclicIn

Has to be called at the start of the periodically called program. This block reads the data from the fieldbus.

JS_MC_CyclicIn	
Input	Output
Enable Axis	Valid Error ErrorID

2.3.3 CyclicOut

Has to be called as the last JsMcLib block in the periodically called program. This block writes values to the fieldbus.
Important: All other JsMcLib blocks must be called between CyclicIn and CyclicOut.

JS_MC_CyclicOut	
Input	Output
Axis	

2.3.4 Power

The enable input of the power blocks switches the power stage on and off. The power stage is turned on when the Status and Valid output is set.

JS_MC_Power	
Input	Output
Enable	Status
Axis	Valid
	Error
	ErrorID

2.3.5 Reference

With linear motors, a reference drive must be executed before any other movement can be performed. During a reference drive, the motor moves in one direction. The direction can be specified with the RefrenceMode input.

Rotary motors can be referenced, but they do not need to be referenced. However, some functions of the XENAX® servo controller require a referenced motor. Motors with ABZ encoders can be referenced with a Z-Mark in the Motor. ZMarkSpeedRot defines the speed during such a reference drive and the ReferenceMode defines the direction. All rotary motors can be optionally referenced with a limit switch. The speed during a limit switch reference drive is defined by the input ReferenceSpeedRot.

JS_MC_Reference	
Input	Output
Execute	Done
ReferenceMode	Busy
ZMarkSpeedRot	CommandAborted
ReferenceSpeedRot	Error
Axis	ErrorID

2.3.6 MoveCyclicPosition

In cyclic synchronous position mode, all move commands like MoveAbsolute are perfomred with Indraworks function blocks and executed on the Nc-axis. This function block connects the Nc-axis with the real axis. MoveCyclicPosition is only required in cyclic synchronous position mode.

JS_MC_MoveCyclicPosition	
Input	Output
Position	Valid
Enable	CommandAborted
Axis	Error
	ErrorID

2.3.7 Stop

Stops the current motion an switches to a Stopping state. No movement command can be executed in this state. The JsMCLib remains in a Stopping state until the execute input is reset.

Set DisableRamp for slow drives with limited force towards a touching position. This option makes sure that not more force than allowed is applied to the touching object by immediately stopping the axis. Only necessary in cyclic synchronous position mode

JS_MC_Stop	
Input	Output
Execute Deceleration DisableRamp Axis	Done Busy CommandAborted Error ErrorID

2.3.8 Reset

Resets the XENAX® servo controller. This is needed when the servo controller is in the Error state.

JS_MC_Reset	
Input	Output
Execute Axis	Done Busy Error ErrorID

2.3.9 ReadStatus

Reads the current state of the XENAX® servo controller.

JS_MC_ReadStatus	
Input	Output
Axis Enable	Valid Error ErrorID Errorstop Disabled Stopping Standstill DiscreteMotion Reference ContinuousMotion

2.3.10 ReadPSR

Reads the Process Status Register (PSR). This registers contains various information about the XENAX® servo controller. The PSR shows for example if the servo controller is referenced.

JS_MC_ReadPSR	
Input	Output
Axis Enable	Valid Error ErrorID ProcessStatusRegister

2.3.11 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.
If the motor oscillates during the Force Calibration, set IterativeFcDisable. This will clear old calibration data before a new calibration is started.

JS_MC_ForceCalibration	
Input	Output
Execute	Done
StartPosition	Busy
EndPosition	CommandAborted
IterativeFcDisable	Error
Axis	ErrorID

2.3.12 ReadDigitalInput

Reads digital inputs which are located in the XENAX® socket. DigitalInputs PDO must be mapped-

JS_MC_ReadDigitalInput	
Input	Output
Enable	Valid
Axis	Error
	ErrorID
	DigitalInput

2.3.13 WriteDigitalOutput

Writes digital outputs which are located in the XENAX® socket. PhysicalOutputs PDO must be mapped.

JS_MC_WriteDigitalOutput	
Input	Output
DigitalOuput	Done
Execute	Error
Axis	ErrorID

2.3.14 ReadActualPosition

Reads the position of the XENAX® servo controller in increments.

JS_MC_ReadActualPosition	
Input	Output
Enable	Position
Axis	Valid
	Error
	ErrorID

2.3.15 ReadActualCurrent

Reads the motor current in mA.
I_ForceActual PDO must be mapped.

JS_MC_ReadActualCurrent	
Input	Output
Enable	Current
Axis	Error
	ErrorID
	Valid

2.3.16 SetPDO

This block is used to limit the motor current (LimitIForce). The index input must be set to 5 and the Value must be set to $x * 10mA$. A value of 7 would set the maximal current to 70mA. Limit I_force PDO must be mapped.

JS_MC_SetPDO	
Input	Output
Axis	Done
Execute	Busy
Index	Error
Value	ErrorID

2.3.17 WriteParameter

Use this block to write CANopen parameters. The available CANopen parameters are described in the Document Xvi_CANopen which can be downloaded from our Website under XENAX® servo controller -> Manual Bus Module-> CANOPEN.zip. Most of the parameters from WebMotion® can be changed in a PLC program this way.

DataObject selects the parameter group and is a number between 1000h and 8000h. A parameter in a parameter group can be selected with SubID. Note that all direct commands have a SubID of 0. Data Length is the number of bytes of the value. The new Value of the parameter must be set too.

JS_MC_WriteParameter	
Input	Output
Execute	Done
DataObject	Busy
SubID	Error
DataLength	ErrorID
Value	
Axis	

2.3.18 ReadParameter

The CANopen parameters can also be read. Addressing a parameter works the same way as with WriteParameter. A value can be read as soon as the Done output is set.

JS_MC_ReadParameter	
Input	Output
Axis	Value
Enable	Done
DataObject	Busy
SubID	Error
DataLength	ErrorID

2.3.19 ReadAxisError

JS_MC_ReadAxisError	
Input	Output
Enable	Valid
Acknowledge	Busy
DataAddress	Error
DataLength	ErrorID
DataObjectName	ErrorRecordAvailable
Axis	ErrorRecord
	FunctionBlockErrorCount
	AxisErrorCount
	AxisWarningCount

All errors in an Axis or in a JsMcLib function are collected by the ReadAxisError block. Each collected error must be acknowledged before further drive commands are possible.

In case of an Error, the ErrorRecordAvailable gets set. An Error Code and the source block of the error can be found in the ErrorRecord. A list of error codes can be found at the end of this chapter.
In addition to that, it is possible to get an error message as a string. A string with a minimal length of 50 must be instantiated. The address and the actual length of the string must be written to the DataAddress and DataLength input. DataObjectName sould be set to 'JsMcEtxEn'.

ReadAxisError collects 3 different types of messages: axis errors, axis warnings and JsMcLib errors. The number of each type is counted by AxisErrorCount, AxisWarningCount and FunctionBlockErrorCount. Every message must be acknowledged with a positive edge at the Acknowledge input. Note that axis errors must be cleared first with the RESET block. In case of a JsMcLib error, the enable or execute input of the block in which the error occurred must be reset before it can be acknowledged. The error type and the error source can be found in the ErrorRecord output.

2.4 Additional blocks for profile position

Profile position mode works without an Nc-Axis. The JsMcLib contains special move commands which work without an Nc-Axis.

2.4.1 MoveAbsolute

Drives to an absolute position. The drive is started with a positive edge at the execute input and is finished when done equals one.
(Only for profile position mode. In cyclic synchronous position mode use MC_MoveAbsolute from Indraworks)

JS_MC_MoveAbsolute	
Input	Output
Axis	Done
Execute	Busy
Position	CommandAborted
Velocity	Error
Acceleration	ErrorID
Scurve	

2.4.2 MoveRelative

Drives a defined relative distance. The drive is started with a positive edge at the execute input and is finished when done equals one.
(Only for profile position mode. In cyclic synchronous position mode use MC_MoveRelative from Indraworks)

JS_MC_MoveRelative	
Input	Output
Execute	Done
Distance	Busy
Velocity	CommandAborted
Acceleration	Error
Scurve	ErrorID
Axis	

2.4.3 JogVelocity

Drives with a constant speed in positive or negative direction.
(Only for profile position mode. In cyclic synchronous position mode use MC_JogVelocity from Indraworks)

JS_MC_JogVelocity	
Input	Output
Enable	Active
Velocity	Busy
Acceleration	CommandAborted
Deceleration	Error
JogPositive	ErrorID
JogNegative	Jogging
Axis	

2.4.1 Halt

Stops and ongoing MoveAbsolute or MoveRelative command and switches to Stanstill state.

JS_MC_Halt	
Input	Output
Execute	Done
Deceleration	Busy
Axis	CommandAborted
	Error
	ErrorID

2.5 Minimum and Maximum Values of Function Blocks

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

name	datatype	min	max
Velocity linear	UDINT	10 inc/s	9000000 inc/s
Velocity rotative	UDINT	10 inc/s	100000000 inc/s
Deceleration	UDINT	2000 inc/s^2	1000000000 inc/s^2
Acceleration	UDINT	2000 inc/s^2	1000000000 inc/s^2
S-curve	UDINT	1 %	100 %

2.6 Error numbers

The following ErrorIDs can be generated by the JsMcLib function blocks. Lower numbers than 5000 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	controller is not ready to switch on	Check controller for errors
50002	jsmcERR_DRIVE_SWITCHED_OFF	controller is switched off	Don't call function block when 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 to 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 power link connection to the Servo Controller
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 to 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...250000 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...100000 inc/s)
50022	jsmcERR_VELOCITY_NOT_IN_RANGE	Velocity is out of range	Use velocity in allowed range (10...9000000 inc/s for linear motor, 10...100000000 inc/s for rotative motor)
50023	jsmcERR_ACCE_TO_LARGE	Acceleration is to large	Use smaller acceleration (smaller than 1000000000 inc/s^2)
50024	jsmcERR_CYCLE_TIME_FAILURE	Cycle time setting failure	Use correct cycle time setting (powerlink bus cycle time >= 400us and software task cycle time >= powerlink bus cycle time)
50025	jsmcERR_DECE_TO_SMALL	Deceleration is to small	Use larger deceleration (>=2000 inc/s)

50026	jsmcERR_DECE_TO_LARGE	Deceleration is to large	Use smaller deceleration (smaller than 1000000000 inc/s^2)
50027	jsmcERR_FW_VERS_FAILURE	Firmware version failure	Use at least XENAX firmware V3.64D and powerlink bus module firmware V2.0
50028	jsmcERR_PDO_MAPPING_CHK_FAILURE	Failure during PDO mapping check	Error in AsIOPVInfo() function block of AsIO library
50029	jsmcERR_PDO_MAPPING_MISSING	Necessary PDO mapping missing	Check, if all necessary PDOs are mapped in I/O Mapping
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	
50034	jsmcERR_INDEX_NOTVALID	Index not valid	
50035	jsmcERR_VALUE_OUTOFRANGE	Value not in range	
50036	jsmcERR_FC_INPUTS_NOTVALID	Force calibration inputs not valid	
50037	jsmcERR_FC_NO_LINEAR	Force calibration only with linear motors	
50038	jsmcERR_FC_REF_ERROR	Force calibration: Error during reference	
50039	jsmcERR_FC_MOTION_ERROR	Force calibration: Error during motion	
50040	jsmcERR_UNKNOWN_MOTORTYPE	Unknown motor type	

2.7 Error sources

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

ErrorSource	Error srouce
1	Axis error or warning
2	CyclicIn
3	Power
4	Reference
5	MoveAbsolute
6	MoveRelative
7	MoveCyclicPosition
8	Stop
9	Halt
10	AxisErrorCollector
11	ReadAxisError
12	ReadParameter
13	WriteParameter
14	JogVelocity
15	ReadActualCurrent
16	ReadDigitalInput
17	ReadDigitalOutput
18	WriteDigitalOuput
19	SetPDO
20	ForceCalibration

2.8 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	Funktionsblock im Fehler
1	Axis error
2	Axis warnung
3	Function block error

3 Example Projects in IndraWorks

This chapter describes how to put a Jenny Science axis into operation. Example projects are used for this purpose

There are three different examples:

Simple Demo
Axis moves to two alternating positions.

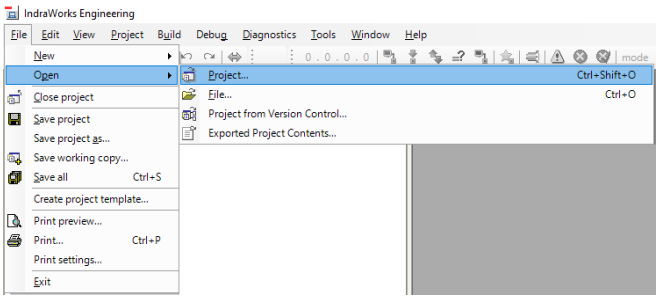
Force Limit
Axis drives forward with a limited Force. If the limited force is reached due to an obstacle in the forward path, the axis stops and moves back quickly.

Forceteq
This is an extended version of the Force Limit example 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.

All examples are available in profile position without a virtual (NC) Axis or in cyclic synchronous position mode with a virtual (NC) Axis.

3.1 Open Project

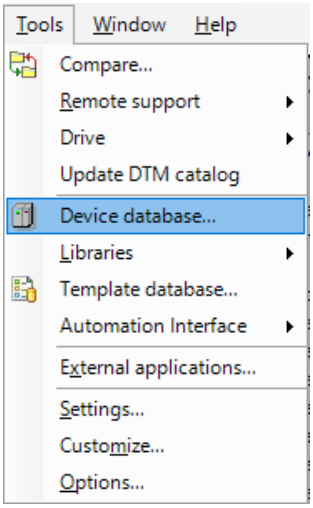
Start TwinCAT3, select “Open Solution from Archive”, choose the demo project and save it to your project folder. It is recommended to start with the “Single Axis” example project



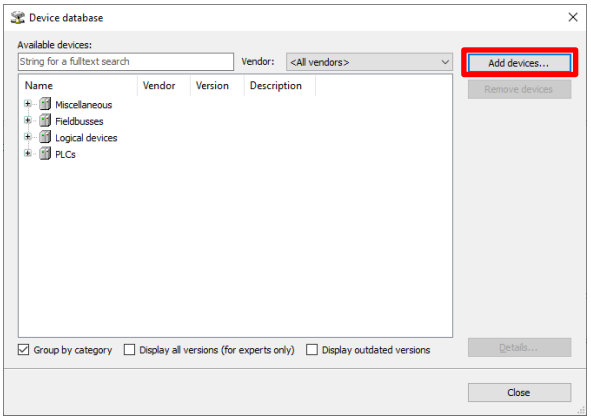
3.2 ESI XML Installation

The EtherCAT Slave Information XML for the XENAX can be installed over Tool -> Device database

The required XML file can be downloaded from www.jennyscience.ch under “XENAX Servocontroller->Firmware Bus Module ->EtherCAT”.

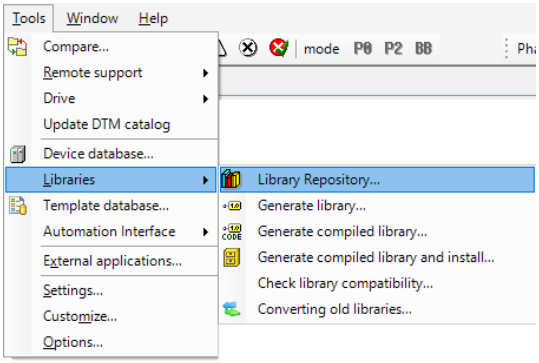


Click add device and select the XML file in the ESI folder. There are other files in the same folder but **only the XML** file is needed.

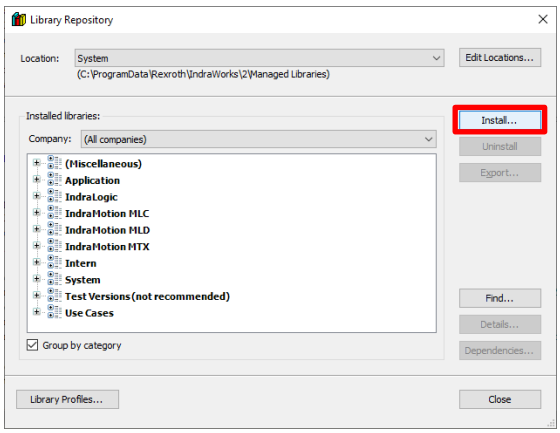


3.3 Library Installation

To install the JS_MC_Lib, open the “Library repository” over Tools->Libraries->Library Repository

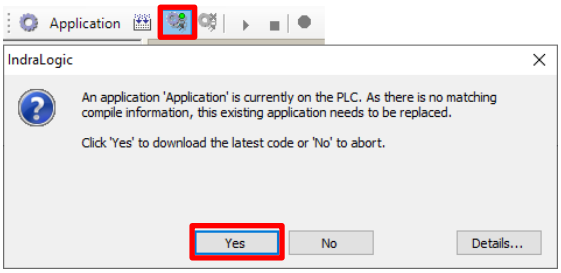


Install the library which is provided in the same download as this documentation.



3.4 Download Programm

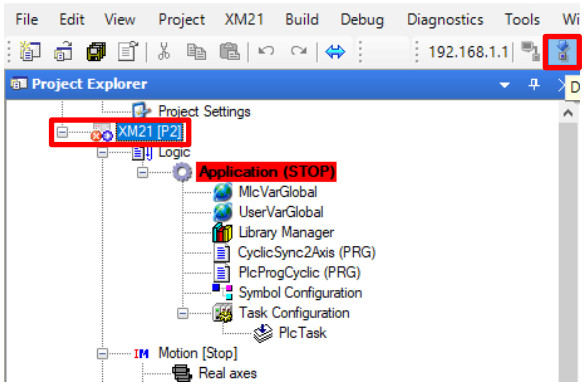
Go online and download the PLC program.



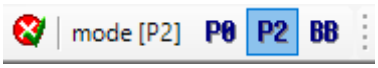
3.5 Configure Online Parameters (Cyclic Sync only)

The cyclic synchronous position mode uses virtual axes. The configuration for those axes can not be downloaded and must be done by hand.

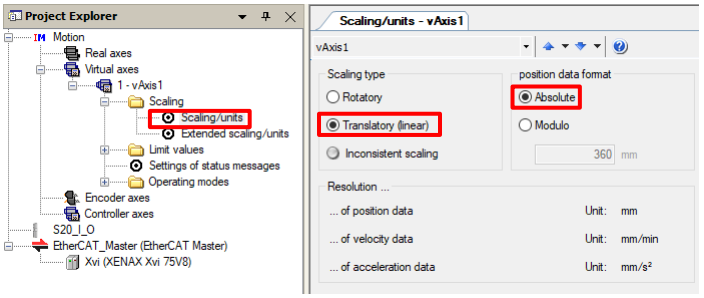
Select the PLC and click on “Download motion configuration”.



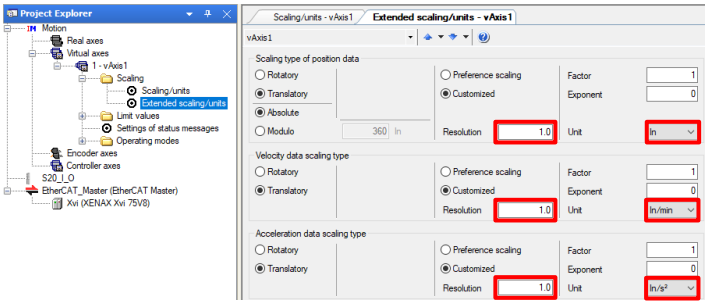
Switch to parametrization mode P2.



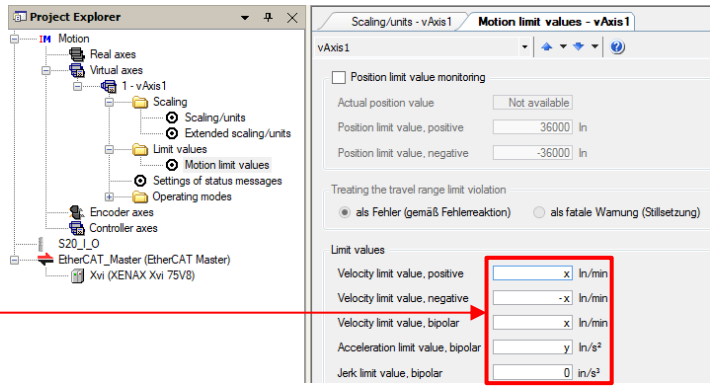
Set the scaling type to Translatory and the motor is rotative. In most cases, the position data format should be set to Absolute.



Switch all units to Increments (In) and set the resolution to 1.



Replace x and y with the values in the table below. Jerk can be set according to your application or left at 0.



Axis Type	Speed **x**	Acceleration **y**
LINAX®,ELAX® 1µm/inc	270000000	1000000000
LINAX®,ELAX®100nm/inc	5400000	1000000000
ROTAX® Rxvp	96000000	1000000000
ROTAX® Rxhq	144000000	1000000000

3.6 Launch Demo Program

Press play to start the demo program.

The following subsection will describe 3 different demo applications in more details.

3.6.1 Simple Demo

This demo initialises and powers on the motor. After that, the demo programs calls MC_MoveAbsolute with an alternating target position of 0 and 44’000 increments.

3.6.2 Force Limit

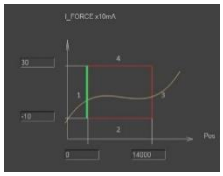
This project demonstrates the force limitation part of Forceteq®. The axis drives forward with a limited force. If an obstacle is in the forward path, the force limit will be reached and the axis moves back quickly to the starting position.

After power up, a Force Calibration is performed. During a Force Calibration, the motor drives slowly from start- to end position and records friction and cogging. Those forces are then automatically compensated by the XENAX®. Such a calibration makes the force limitation much more accurate.

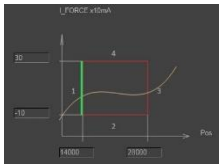
3.6.3 Forceteq®

The Forceteq® demo expands the “Force Limit” demo program with the features of Force Monitoring. With Force Monitoring, it is possible to define force-position sectors. After a drive is finished, each sector is either in a successful or failed state. Each edge of a sector can be defined as entry or exit edge or both. A sector is successful if the curve entered through an entry edge and left through an exit edge. The demo program defines 3 such sectors without an exit edge. This means that a sector is successful if the curve does not leave the sector. This happens in the demo program if an obstacle was detected in that sector.

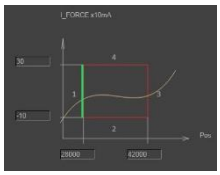
Definition of the 3 sectors:



******* Sector I_Force 1 **********
Sector IForce Start = 0
Sector IForce End = 14000
IForce Low x10mA = -10
IForce High x10mA = 30
Sector Transit Config = 4096



******* Sector I_Force 2 **********
Sector IForce Start = 14000
Sector IForce End = 28000
IForce Low x10mA = -10
IForce High x10mA = 30
Sector Transit Config = 4096

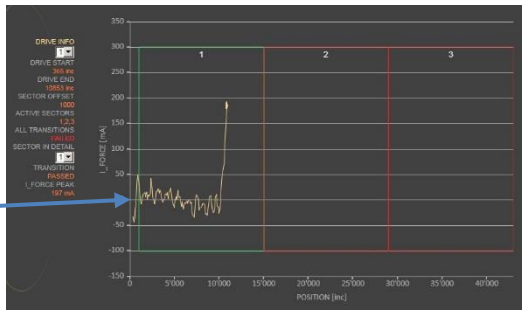


******* Sector I_Force 3 **********
Sector IForce Start = 28000
Sector IForce End = 42000
IForce Low x10mA = -10
IForce High x10mA = 30
Sector Transit Config = 4096

The variables of the main program indicate in which sector the obstacle was detected.

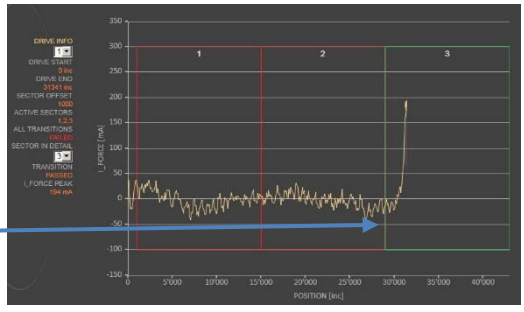
XM21.Application.PlcProg

Expression	Type	Value
State	INT	2022
bInitDoneBB	BOOL	TRUE
bForceCalibDone	BOOL	TRUE
StartPosition	DINT	0
EndPosition	DINT	44000
IForce_Test	UDINT	20
FT_State	INT	99
WaitTime	INT	0
TouchSector1	BOOL	TRUE
TouchSector2	BOOL	FALSE
TouchSector3	BOOL	FALSE
NoTouch	BOOL	FALSE
TestError	BOOL	FALSE



XM21.Application.PlcProg

Expression	Type	Value
State	INT	2022
bInitDoneBB	BOOL	TRUE
bForceCalibDone	BOOL	TRUE
StartPosition	DINT	0
EndPosition	DINT	44000
IForce_Test	UDINT	20
FT_State	INT	99
WaitTime	INT	0
TouchSector1	BOOL	FALSE
TouchSector2	BOOL	FALSE
TouchSector3	BOOL	TRUE
NoTouch	BOOL	FALSE
TestError	BOOL	FALSE
_IoMap	JS_MC_IoMap	

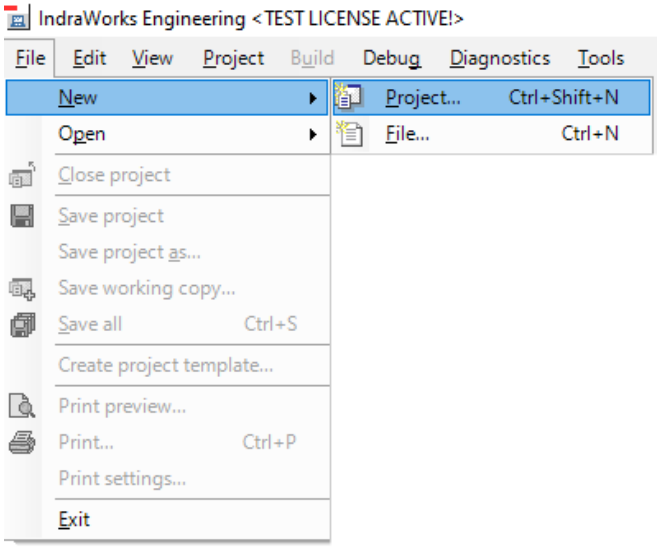


4 New Project in Indraworks

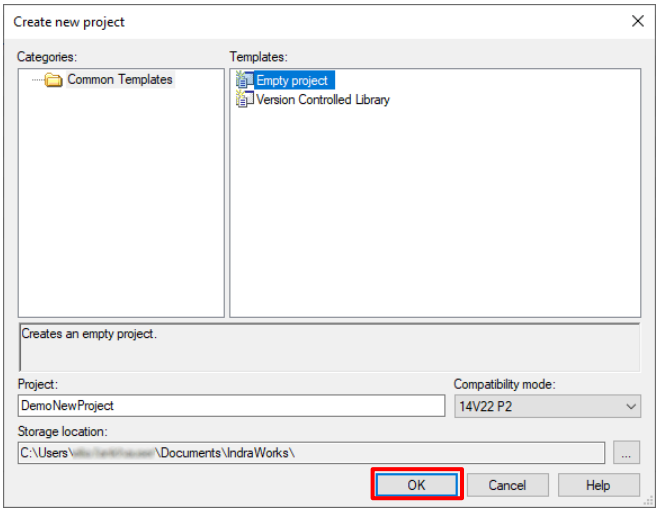
This chapter describes how to put a Jenny Science Axis into operation without a demo project. It is possible to create a new project or to add a Jenny Science axis into an existing project.

4.1 Create Project

File→New→Project…

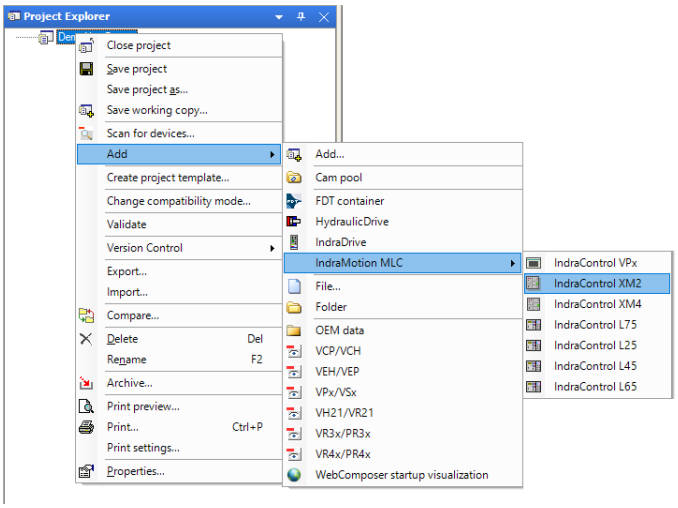


Select Empty project and click ok.

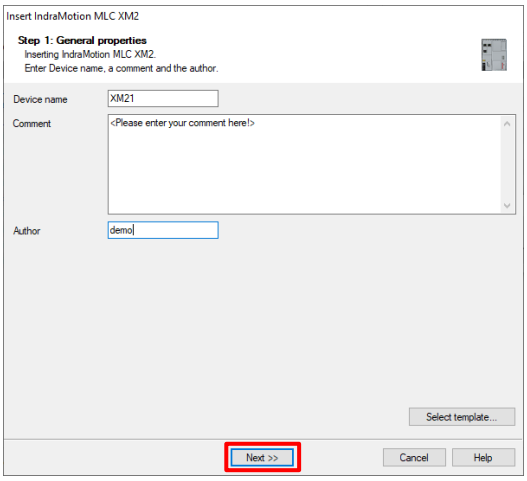


4.2 Add PLC

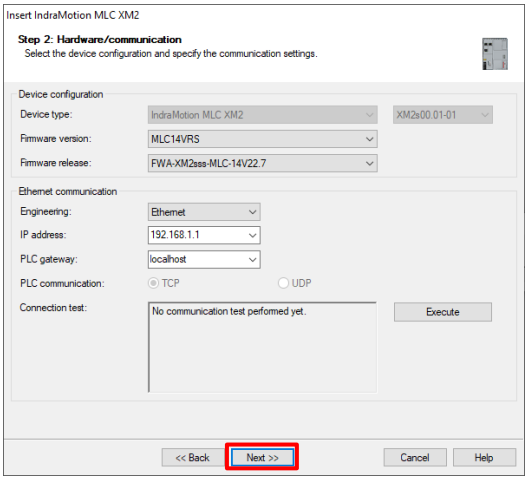
Add your PLC to the Project.



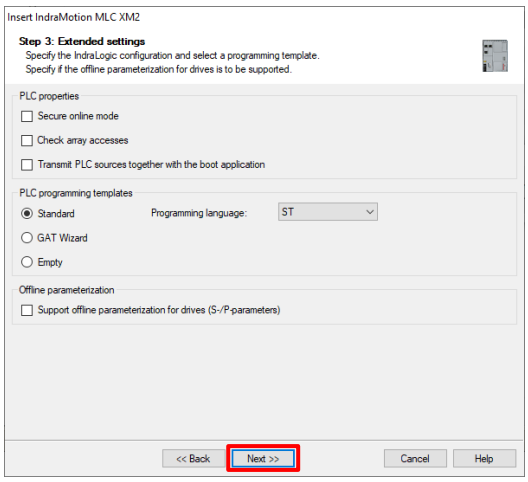
Give the PLC a name and press next.



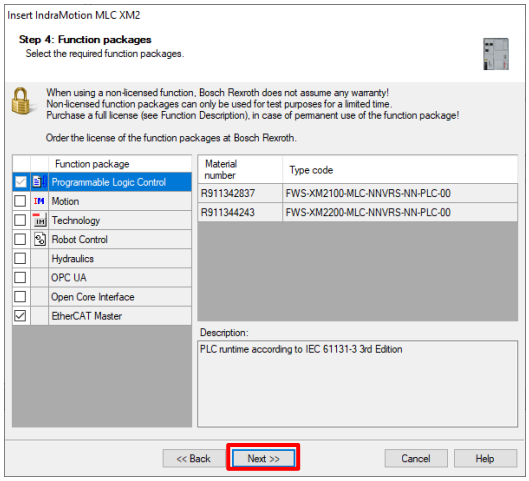
Enter the IP address of the PLC.



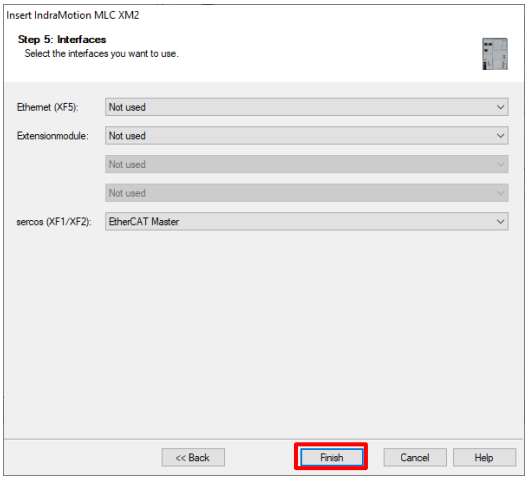
Press next.



Enable Programmable logic and EtherCAT Master.

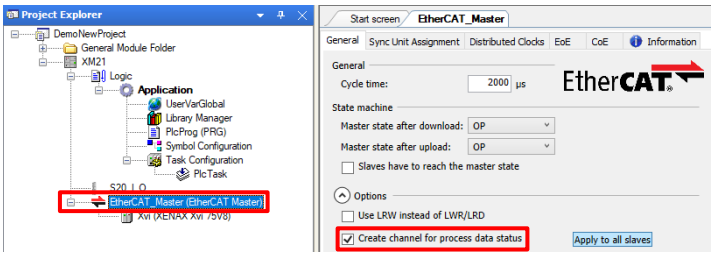


Press finish.



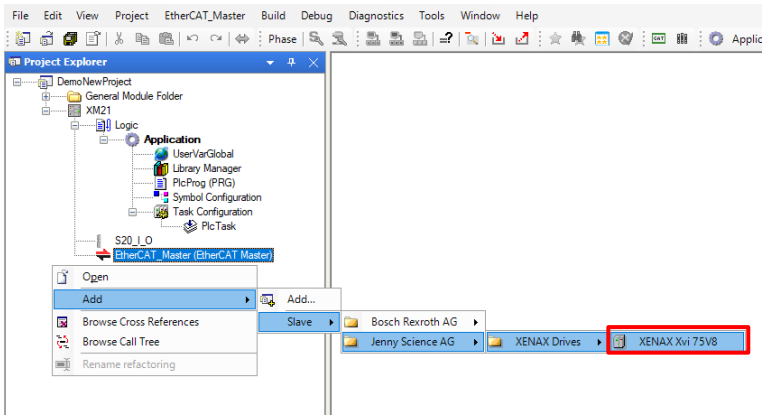
4.3 Config EtherCAT Master

Enable process data status PDO.



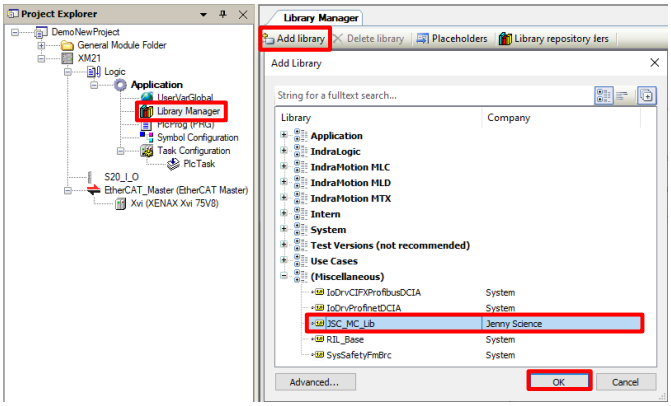
4.4 Add XENAX

Add the XENAX® Xvi 75V8 or Xvi 48V8. The XENAX® can only be selected if it was installed as described in section 3.2 ESI XML Installation.



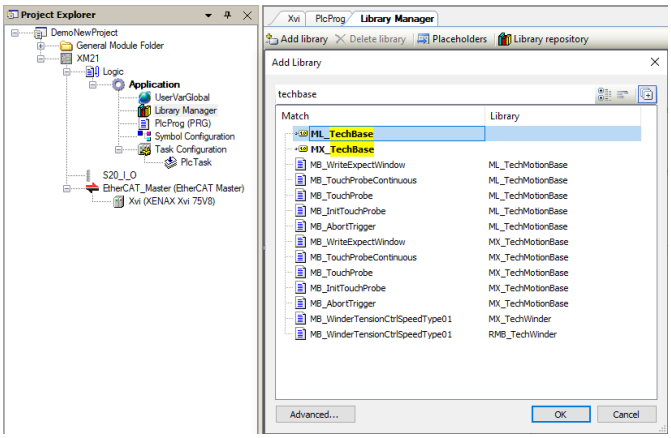
4.5 Add Libraries

Add the installed JSC_MC_Lib.
Installing the library is described in 3.3 Library Installation.



Also add the following libraries to the project:

ML_TechBase
SysTypes
SysTask

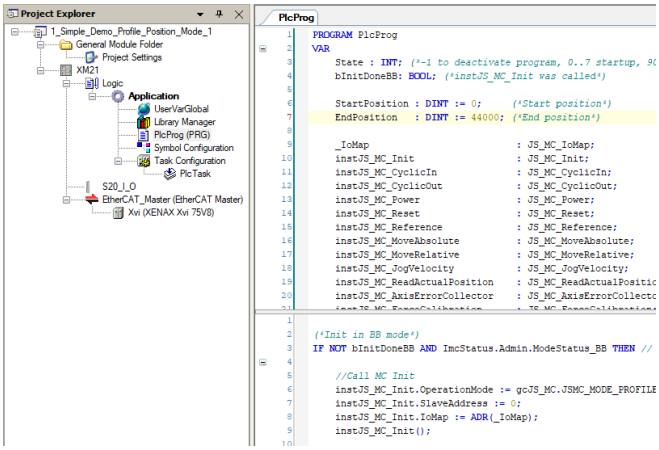


4.6 Profile Position

This section is only required for profile position mode.

4.6.1 Copy Code

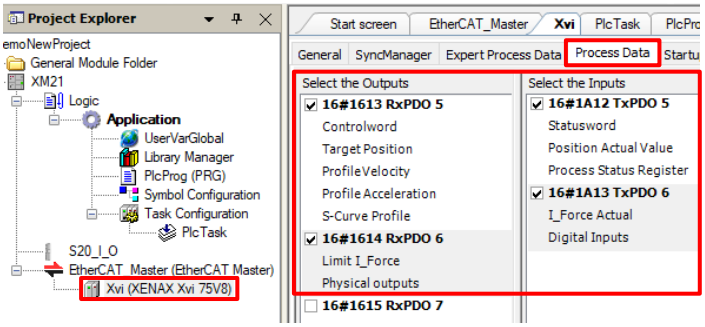
Copy the code from 1_Simple_Demo_Profile_Position into the current project.



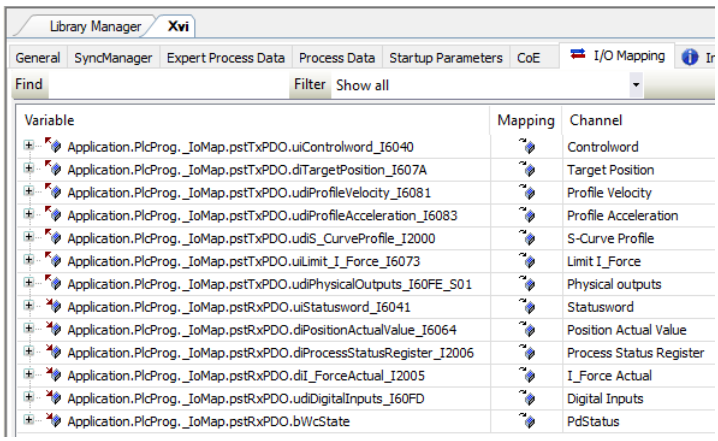
4.6.2 PDO Mapping

Only enable the two required PDOs. There are two additional PDOs which can be enabled for additional features.

Rx	Tx
required PDOs	
PDO 5	PDO 5
Optional PDOs	
PDO 6	PDO 6



Map all PDO channels to a variable in the _IOMap structure.



4.6.3 Launch Project

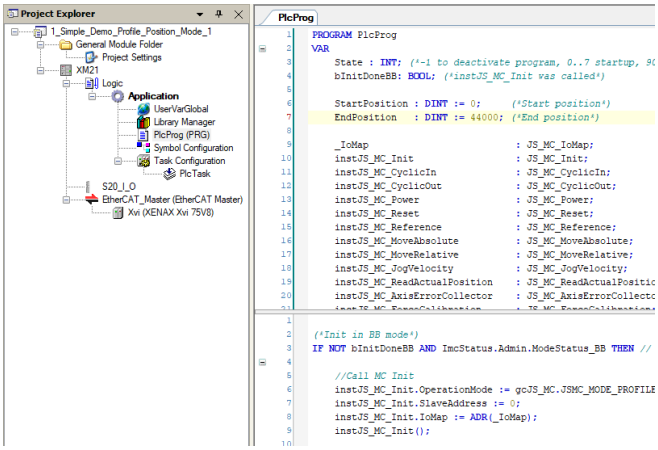
Compile and make sure there are no errors. Go online and press play.

4.7 Cyclic Synchronous Position Mode

This section is only required for cyclic synchronous position mode.

4.7.1 Copy Code

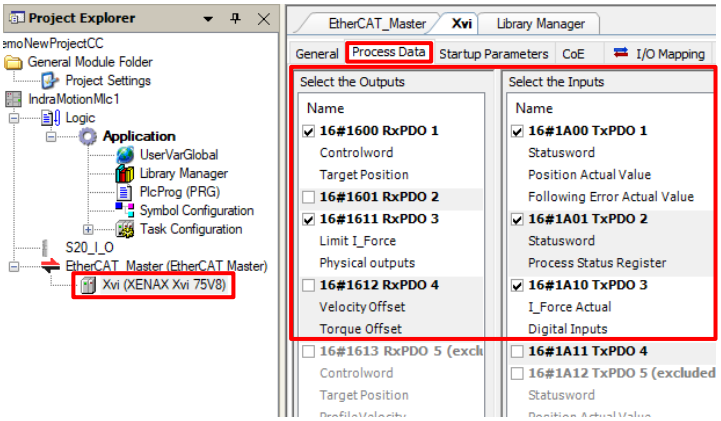
Copy the code from 1_Simple_Demo_Cyclic_Sync into the current project.



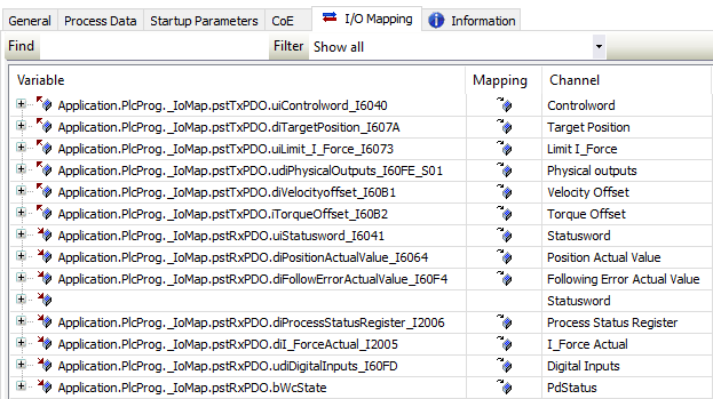
4.7.2 PDO Mapping

Only enable the two required PDOs. There are two additional PDOs which can be enabled for additional features. PDO4 transmits velocity and acceleration of the virtual nc-axis. This improves accuracy for high dynamic profiles.

Rx	Tx
required PDOs	
PDO 1	PDO 1,2
optional PDOs	
PDO 3	PDO 3
optional PDO for high dynamic profiles	
PDO 4	

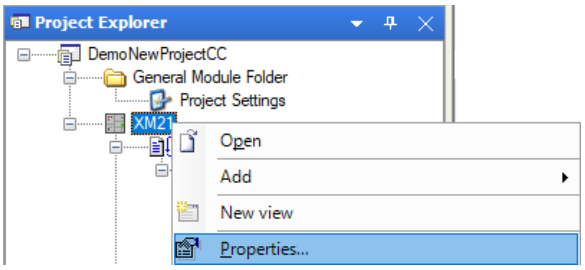


Map all PDO channels to a variable in the _IOMap structure. There are two Statusword channels. Only one of them needs to be mapped.

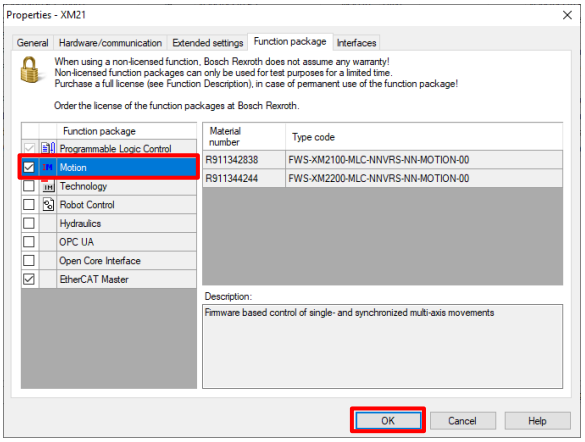


4.7.3 Add Virtual Axis

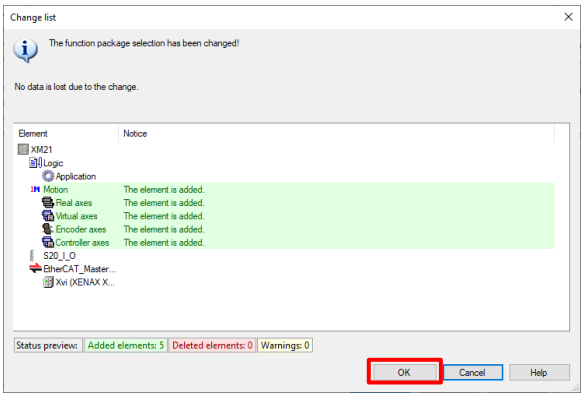
The motion function package can be added in the properties of the PLC.



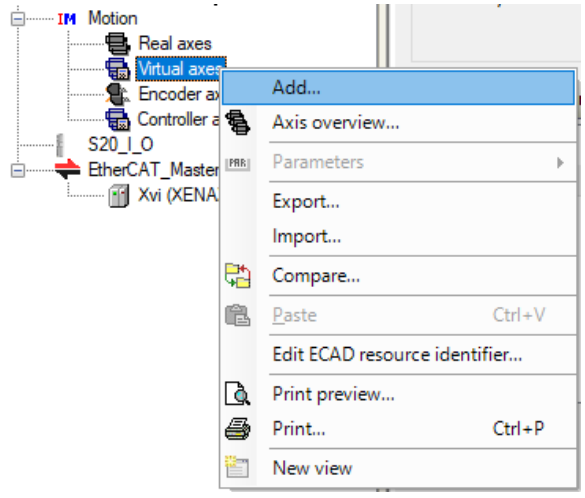
Enable the function package Motion und press OK.



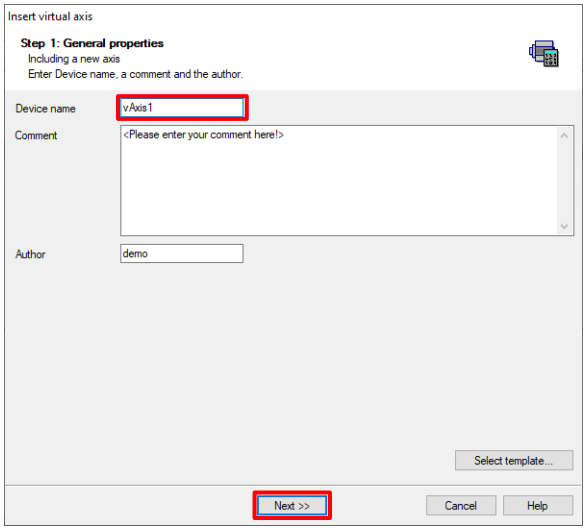
Press Ok.



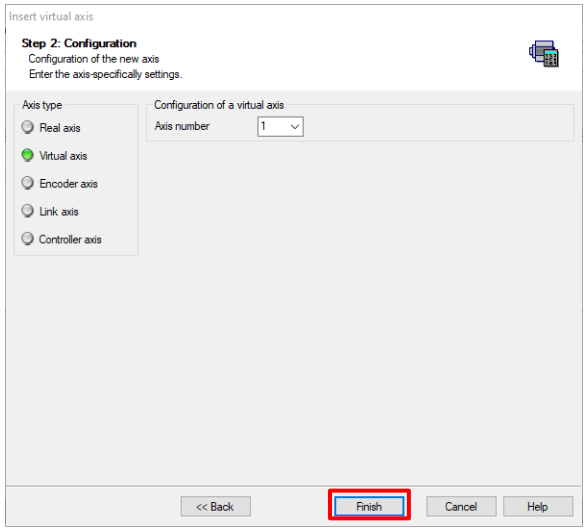
Now, a virtual axis can be added.



Name the axis and press Next.
The name will be used later to reference the axis.



Index number of the axis can be selected.

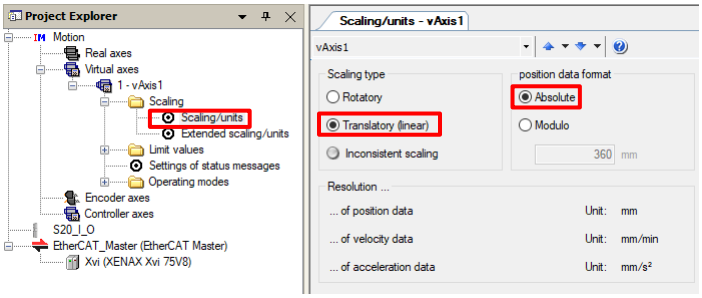


JS_MC_CyclicIn requires the reference to the virtual axis.

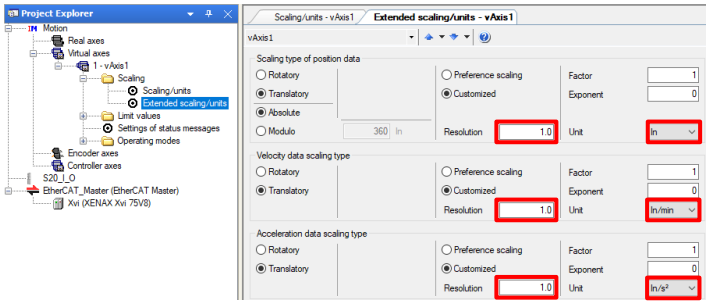
```
29  IF bInitDoneBB AND ImcStatus.Admin.ModeStatus_BB THEN
30      //Call CyclicIn
31      instJS_MC_CyclicIn(pstAxis := instJS_MC_Init.pstAxis, VirtualAxis := MlcVarGlobal.vAxis1);
32
33      IF instJS_MC_CyclicIn.Error AND State <> -1 THEN (* Communication Error occured? *)
34          state := 90;
35      END_IF
36
37      CASE State OF
```

4.7.4 Configure Virtual Axis

Set the scaling type to Translatory and the motor is rotative. In most cases, the position data format should be set to Absolute.

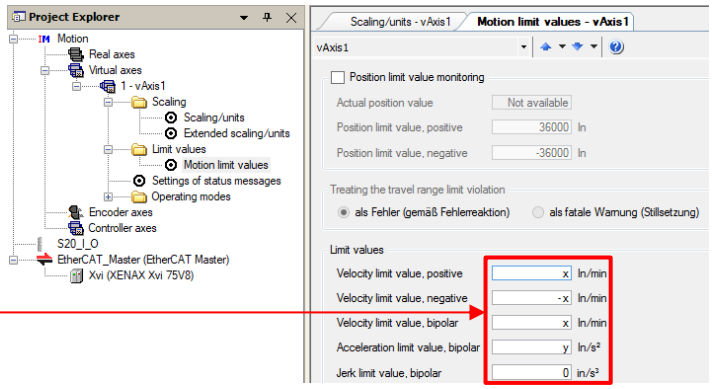


Switch all units to Increments (In) and set the resolution to 1.



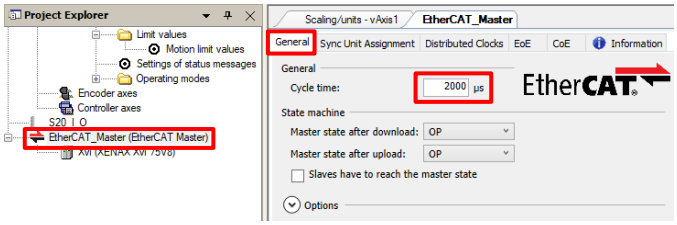
Replace x and y with the values in the table below. Jerk can be set according to your application or left at 0.

Axis Type	Speed **x**	Acceleration **y**
LINAX®,ELAX® 1µm/inc	270000000	1000000000
LINAX®,ELAX®100nm/inc	5400000	1000000000
ROTAX® Rxvp	96000000	1000000000
ROTAX® Rxhq	144000000	1000000000

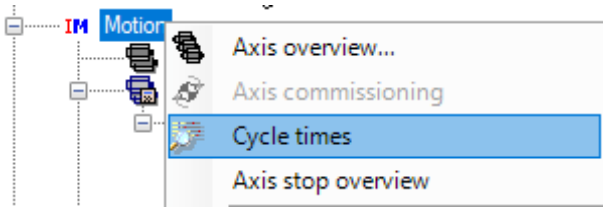


4.7.5 Configure Cycle Time

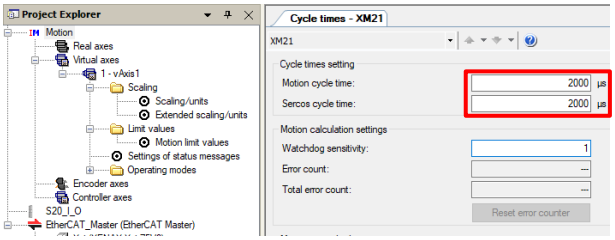
The bus cycle time defines how often the position of the virtual axis is transferred from the PLC to the XEANX®. The valid range for the cycle time depends on the PLC type.



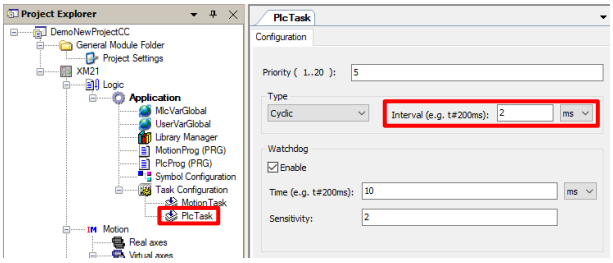
The motion cycle time can be set under Motion→ cycle times



Set the Motion cycle time and sercos cycle time to the same value as the EtherCAT Bus cycle time.



Also the Task, where the JS_MC_Lib functions blocks are called, must be called with the same cycle time.

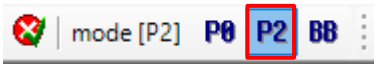


4.7.6 Online Configuration

Download the motion configuration.

Go Online

Switch to parametrization mode P2.



Some configurations are not downloaded to the PLC. They can only be changed online. Because of this, the steps in chapter **4.7.4** Configure Virtual Axis and **4.7.5** Configure Cycle Time must be **done again while online**.

4.7.1 Launch Project

Put the PLC in run mode. If the axis does not move, power the PLC and the XENAX® off and on again.

Jenny Science AG
Sandblatte 7a
CH-6026 Rain

Tel +41 (0) 41 455 44 55
Fax +41 (0) 41 455 44 50

www.jennyscience.ch
info@jennyscience.ch