Instruction Manual
Jenny Science
XENAX® Xvi EtherNet/IP® and Studio5000® Logix Designer

Version 2.0.6

Edition June 2020

EtherNet/IP™

XENAX® Ethernet servo controller with EtherNet/IP® Busmodul

General

This manual describes the connection of a XENAX® Xvi75V8 and XENAX® Xvi48V8 Servo controller to an Allen-Bradley PLC with Studio5000® Logix Designer V24 and the Jenny Science Add-On Profile (JSC_MC_AOP).

Contents

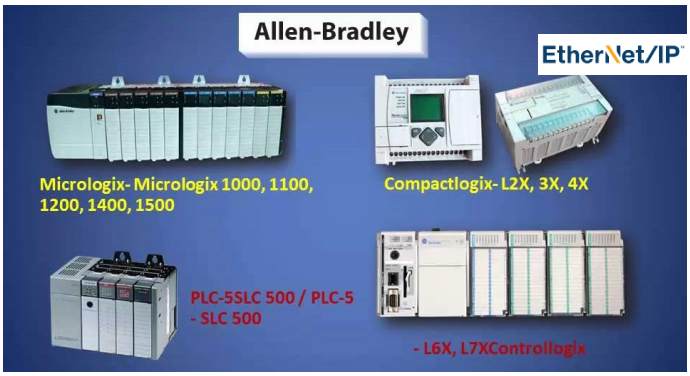
1 Development Environment **4**
1.1 Tools 4
1.2 Controller 5
1.3 Actuators 5
1.4 Additional Resources 6
1.5 Software requirements and basic settings 7
2 Example Projects for Studio 5000 **8**
2.1 Open Project 9
2.2 EDS-File Installation 9
2.3 IP Address Setup 10
2.4 Launch Example Project 11
2.4.1 SimpleTest 12
2.4.2 ForceLimit 13
2.4.3 Forceteq® 14
2.4.4 CyclicSyncMotion 16
2.4.5 Cyclic 2Axis 16
3 New Project in Studio 5000 **17**
3.1 Create Project 17
3.2 Library Installation 18
3.3 EDS-File Installation 18
3.4 IP Address Setup 19
3.5 Add XENAX® Module 20
3.6 Mc_Axis and Messages 22
3.7 Task Cycle Time 22
4 Cyclic Synchronous Motion **23**
4.1 Virtual Axis 23
4.2 Task configuration 27
4.2.1 Main Task 27
4.2.2 Motion Task 27
4.2.3 Enable time Synchronization 28
5 JS_MC_AOP Motion Library **29**
5.1 State Diagram 29
5.2 Required AOIs 31
5.2.1 Init 31
5.2.2 CyclicIn 31

5.2.3 CyclicOut 32
5.2.4 Power 32
5.2.5 Reference 33
5.2.6 Reset 34
5.2.7 HandleModeOfOperation 34
5.3 Optional Function Blocks 35
5.3.1 MoveCyclicPosition 35
5.3.2 MoveAbsolute 35
5.3.3 MoveRelative 36
5.3.4 JogVelocity 36
5.3.5 Halt 36
5.3.6 Stop 37
5.3.7 ForceCalibration 38
5.3.8 SetOA 38
5.3.9 ReadPSR 39
5.3.10 ReadStatus 39
5.3.11 ReadDigitalInput 40
5.3.12 WriteDigitalOutput 40
5.3.13 ReadActualPosition 40
5.3.14 ReadActualCurrent 41
5.3.15 ReadAxisError 41
5.3.16 AxisErrorCollector 42
5.4 Minimum and Maximum Values of Function Blocks 43
5.5 Error Numbers of JS_MC_AOP 43
5.6 Error Sources 45
5.7 Error Type 45
6 Appendix 46
6.1 Reference methods 46

1 Development Environment

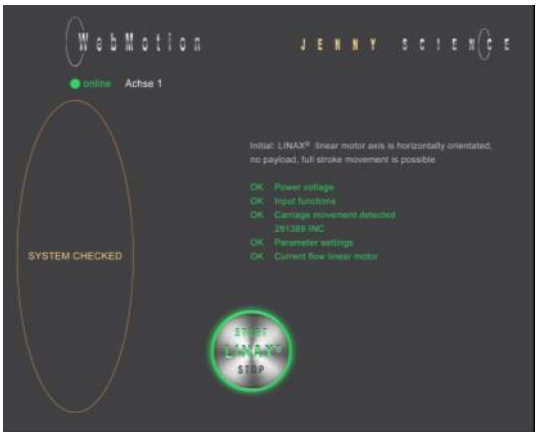
1.1 Tools

Allen-Bradley / Rockwell PLC
A programmable logic controller is used to control multiple axes over Ethernet/IP.



Studio5000® Logix Designer V24
In order to program an Allen-Bradley / Rockwell PLC, the engineering software Studio5000 Logix Designer is required.
All explanations in this instruction manual are based on Studio5000 Logix Designer V24.01.

WebMotion®
The proprietary graphical user interface from Jenny Science servo controllers is stored in the embedded web server of the XENAX® servo controller as a Java applet.
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 easily and immediately be operated.

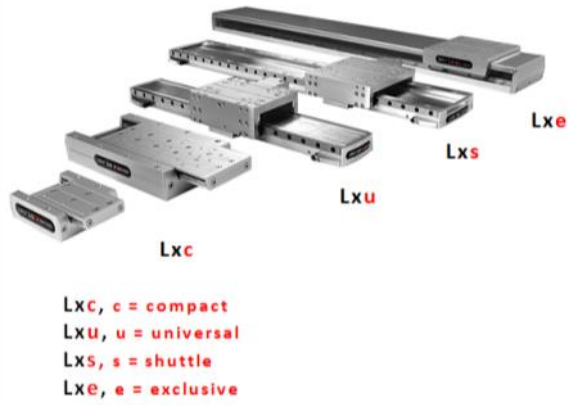


1.2 Controller

XENAX® servo controller
An optional EtherNet/IP bus module is available.
Each XENAX® can control one motor axis.

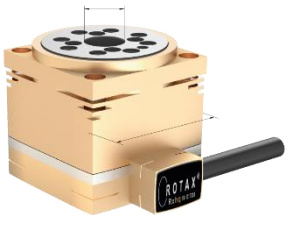
1.3 Actuators

LINAX® Linear motor axes
There are different series available. The LINAX® linear motor axes are highly modular and can be flexibly combined amongst each other.
The XENAX® servo controller identifies the connected LINAX® linear motor and configures the controller parameters automatically.



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 for connecting to the servo controller.
The XENAX® servo controller identifies the ELAX® linear motor slider and configures the controller parameters automatically.

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.
The XENAX® servo controller identifies the ROTAX® rotary axis and configures the controller parameters automatically.



Rxhq = high torque

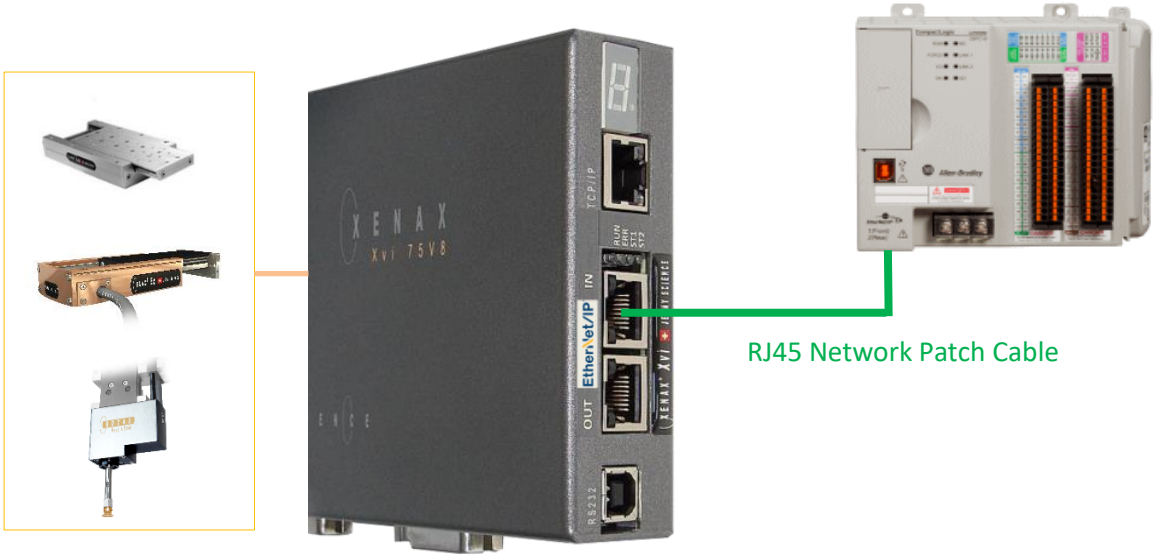
Rxvp = vacuum pressure

1.4 Additional Resources

The following resources are needed for the successful operation of the XENAX® servo controller with an EtherNet/IP bus module. Available to download from: http://www.jennyscience.ch/en/download/

Indication ***Filename***	**Description**
EtherNet/IP PDF.zip *EtherNet_IP_PDF.zip*	Manual of the EtherNet/IP bus module **Please make sure that the bus module is properly installed. For further instructions see the corresponding bus module manual.**
Version x.yz EtherNet/IP.zip *Version x.yz EtherNet_IP.zip*	Firmware EtherNet/IP bus module and the Electronic datasheets (EDS*) of the EtherNet/IP interface for XENAX® Xvi 75V8 and XENAX® Xvi 48V8 **Check the installed bus module firmware, see chapter 1.5 for recommended firmware version.** *: EDS - Electronic Data Sheet is a file format that describes the communication behaviour and the object dictionary entries of a device. This allows tools such as service tools, configuration tools, development tools, and others to handle the devices properly (Reference: https://de.wikipedia.org/wiki/CANopen).
PLCopenAOP Studio5000 AllenBradley Vx.y.zip *PLCopenAOP_Studio5000_Vx.y.zip*	Jenny Science Add-On profile for MotionControl (JSC_MC_AOP) as a library for Studio5000 and example projects to control a linear axis (including this manual)

Configuration to use the example projects:



Linear motor or ROTAX®

XENAX® servo controller Xvi75V8 / Xvi48V8

Allen-Bradley PLC

1.5 Software requirements and basic settings

Software requirements:

Subject	Remark
Studio5000® Logix Designer	Version 24.0 or higher
XENAX® firmware	V4.0 or higher
EtherNet/IP bus module firmware	V2.12 or higher

Basic settings in Studio 5000:

Please check that these basic settings properly done in your project to gain successful operation!

Subject	Remark
Module definition	Connection → ***Full Assembly*** (refer to green frame below)
Module RPI	Recommended: 4 ms (refer to blue frame below)
Task configuration including the axis program	Type: ***Periodic*** (refer to yellow frame below) Period: ***Higher or equal than the module RPI***

2 Example Projects for Studio 5000

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

There are three different example projects:

SimpleTest
Axis moves to two alternating positions.

ForceLimit
This program performs a force calibration first. Then the axis drives from position 0 to 44'000 with a limited Force.
As soon as an obstacle is detected, the motor drives back to position 0 with a higher speed.
If no obstacle is detected, the axis reaches position 44'000 and goes back to position 0.
Feel free to use your hand as an obstacle.

Forceteq
Extended version of ForceLimit. This demo showcases Force Monitoring with three sectors. It shows if an obstacle is detected on one of the sectors.

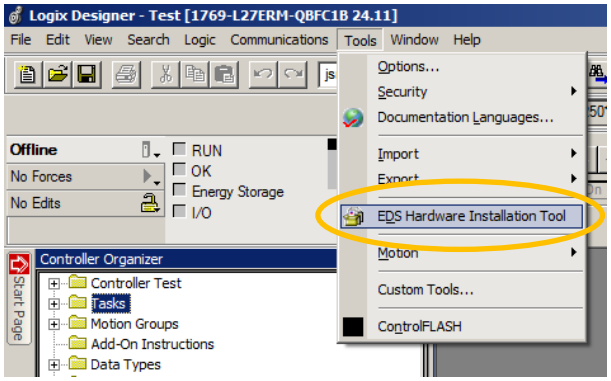
2.1 Open Project

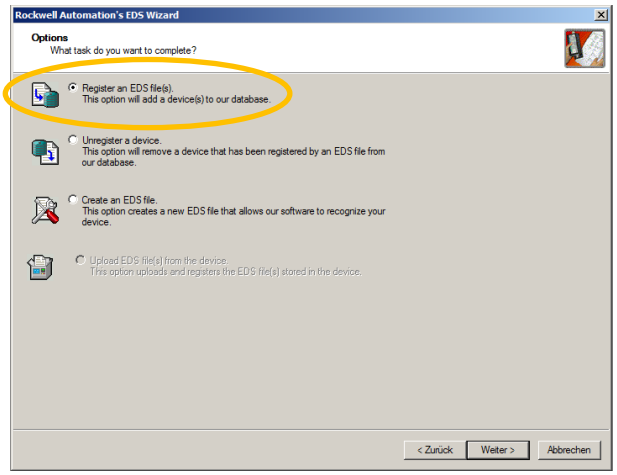
Start Studio 5000 and import the .L5K file from the demo folder.
Save the project to your project folder.

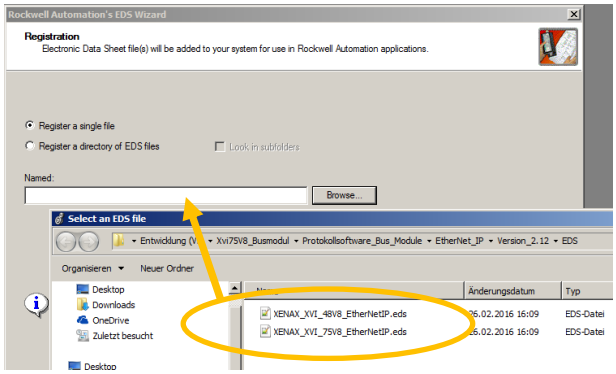
2.2 EDS-File Installation

Register the EDS file for your XENAX® controller with the EDS Wizard (EDS Hardware Installation Tool) in Studio5000 Logix Designer.

Download the latest EDS file from your webpage www.jennyscience.ch under XENAX® Servocontroller→Firmware Bus Module.



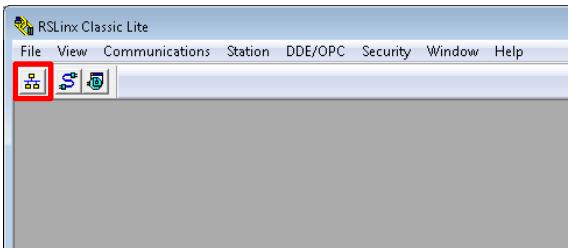




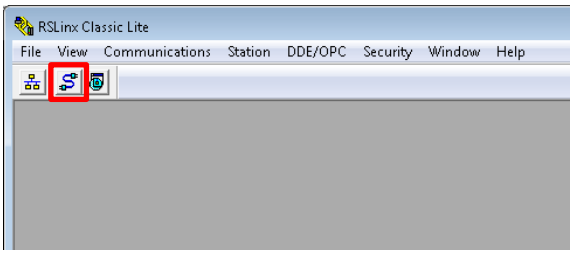
2.3 IP Address Setup

The IP address of the XENAX® Busmodule is set to DHCP by default. RSLinx is tool, that gets installed with Studio 5000. Use RSLinx to set a fixed IP address.

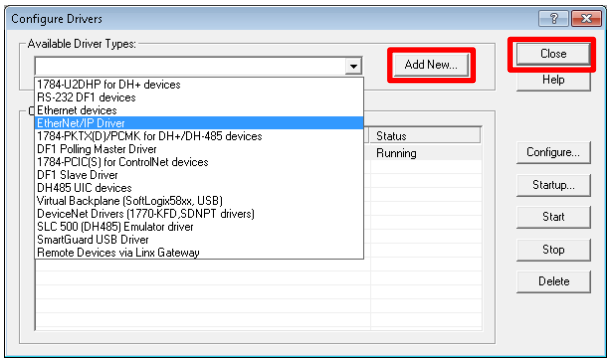
Start RSLinx and Open RsWho.



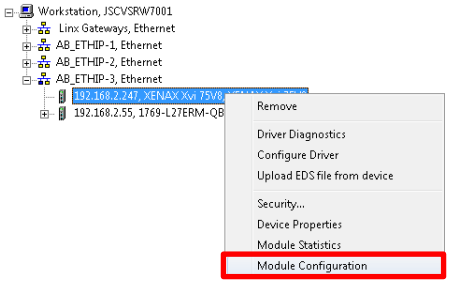
Configure a new Driver if there is none.



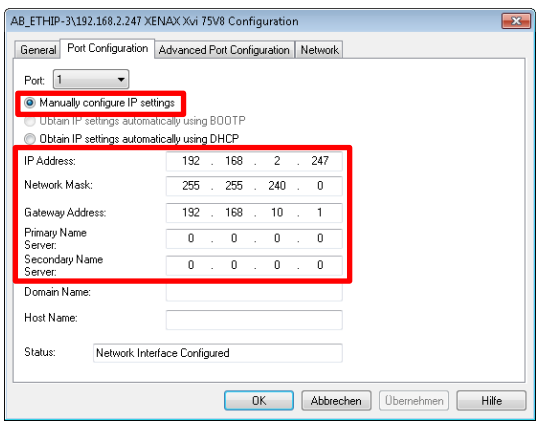
Select the EtherNet/IP driver, click on add New, follow the instructions and click close at the end.



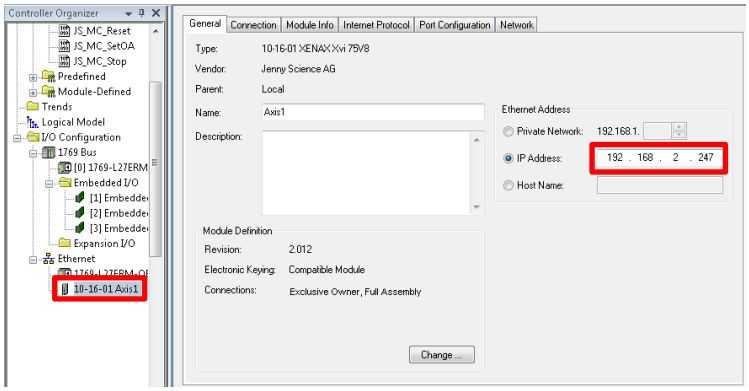
Open Module Configuration of each XENAX® Servocontroller.



Switch from DHCP to manually and enter the IP settings manually.

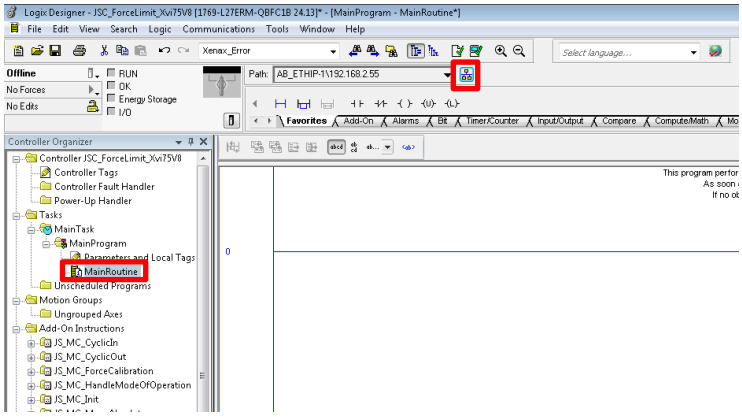


Enter the same IP address in Studio 5000 in the Axis settings.

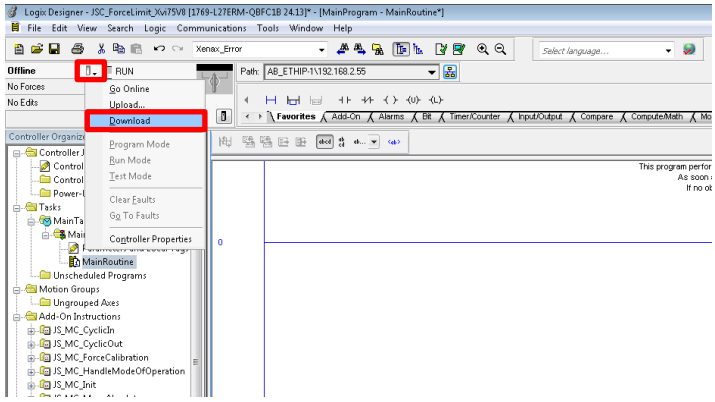


2.4 Launch Example Project

Choose the target PLC and open the MainRoutine.



The program can be downloaded to the PLC.



2.4.1 SimpleTest

Fist demo project moves an axis between 0 and 44000 increments. The demo project contains an example AOI for a Jenny Science Axis called JsAxisSimple. This AOI demonstrates the usage of the Jenny Science Motion Control (JS_MC_AOP) library. The source code JsAxisSimple AOI is open and shall be adjusted for user specific needs. The interface of JsAxisSimple AOI is designed to be as intuitive as possible.

JsAxisSimple			
Name	Type	Usage	Function
PowerEnable	BOOL	In	Swiches the output stage on and off.
TargetPosition	DINT	In	After power up, the motor will drive to this position. Modify this value to drive to a new position.
TargetReached	BOOL	Out	Wait for this output when you have enabled the output stage or after modifying Target Position.
Velocity	DINT	In	Maximal Velocity [increments/s] used to drive to target position.
Acceleration	DINT	In	Maximal Acceleration [increments/s^2] used to drive to target position.
Scurve	DINT	Out	Jerk or change in acceleration in %, 0% = rough drive, 100% = smooth drive, default: 20%.
RefOnPowerUp	BOOL	In	If true, Axis is referenced automatically after enabling the output stage,
ReferenceMode	DINT	In	Select reference mode: 0: motor type specific default, see 6.1 for more details
AutoResetOnError	BOOL	In	Only recommended for demos. Disables error handling and resets the axis whenever an error occurs.
AcknowledgeError	BOOL	In	Acknowledges a pending error. Acknowledging an Axis error calls MC_Reset in the background which clears the error on the XENAX®.
AcknowledgeDone	BOOL	Out	Wait for this signal after an error is Acknowledged. Axis errors take longer because an axis error is executed. Other errors are done immediately.
ErrorPending	BOOL	Out	True, if an error is pending and waits to be acknowledge.
ErrorSource	DINT	Out	Source of pending error. See 5.6 for more details.
ErrorNumber	DINT	Out	Error number of pending error. See 5.5 for more details.
CommunicationOK	BOOL	Out	Cyclic communication with axis is ok.
AxisRef	Module	InOut	A reference to the Axis which should be controlled by the AOI.

The switch **hmi_StartDemo** is used to start and stop the demo application. It is set by default.

If the Axis does not start moving after the program start, there is an error pending. Error can be acknowledged with the switch **hmi_acknowledgeError**. This demo contains correct error handling where AutoResetOnError is set to 0 and each error must be acknowledged.

2.4.2 ForceLimit

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. The demo contains a JsAxisExtended AOI which is an extended version of the JsAxisSimple. JsAxisExtended contains all required features for force limitation and the possibility to stop an ongoing movement of the axis.

JsAxisExtended (only extended signals listed, contains also all signals from JsAxisSimple 2.4.1)			
Name	Type	Usage	Function
ForceCalibOnPowerUp	BOOL	In	If true, Force Calibration is performed after enabling the output stage.
FCStartPos	DINT	In	Start position of Force Calibration in [increments].
FCEndPos	BOOL	In	End position of Force Calibration in [increments]. Set start and end position to 0 to clear the calibrartion.
ForceLimitation	DINT	In	0: no Force Limitation, >0: Force limited to x * [10mA]
ForceLimitationReached	DINT	Out	Current defined with Force Limitation is reached.
Stop	DINT	In	Axis stops and will not move until Stop input is set to 0.
StopDone	BOOL	Out	Axis stopped due to Stop input.
DigitalInput	DINT	Out	Read digital inputs of the XENAX.
DigitalOutput	BOOL	In	Write digital outputs of the XENAX.

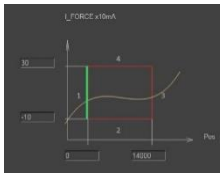
In this demo, the AutoResetOnError is set to 1. This means that the XENAX gets rested as soon as an error occurs. This is not recommended for real application since different error require usually deferent actions. See the first demo in chapter 2.4.1 for an example with correct error handling.

2.4.3 Forceteq®

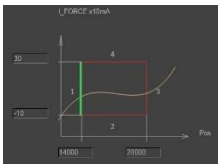
The Forceteq demo program showcases the main features of Forceteq® such as Force Calibration, Force Limitation and Force Monitoring.

This demo alternatingly calls JSC_MC_MoveAbsolute (Position 0 or 44000). During the move from position 0 to 44000, Force Monitoring with 3 sectors is activated and I_ForceLimit is set to 200mA. A sector is so configured that it has one entry and no exit to be valid. This way a sector is valid when the axis stopped moving in the sector. The sectors can be watched in Webmotion® under “move axis by Forceteq®” → “Diag I_Force”.

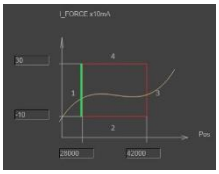
Configuration of the 3 sectors:



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



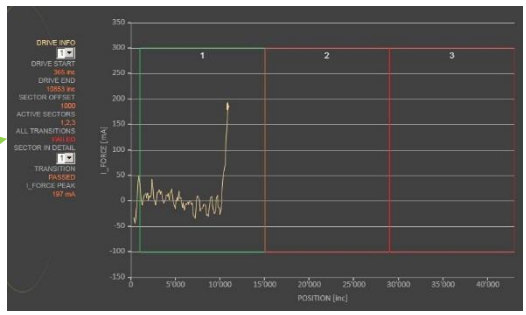
******* Sector 2 I_Force 2 **********
Sector IForce Start = 14000
Sector IForce End = 28000
IForce Low x10mA = -10
IForce High x10mA = 30
Sector Transit Config = 4096

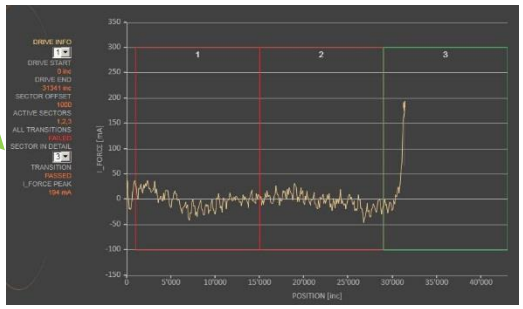


******* Sector 3 I_Force 3 **********
Sector IForce Start = 28000
Sector IForce End = 42000
IForce Low x10mA = -10
IForce High x10mA = 30
Sector Transit Config = 4096

If there is a touch detected in one of these sectors, the move is stopped (see the examples in Sector 1 and Sector 3 below) and a fast backward move to position 0 is started:

Fixed sector offset is set to the 1000





2.4.4 CyclicSyncMotion

In this example project, axis drives between 2 positions in cyclic synchronous position mode. Motion Axis Move (MAM) is used on a virtual axis. The virtual axis is then linked with the real Jenny Science axis.

2.4.5 Cyclic 2Axis

This example includes driving with a coordinate System. 2 axes are employed for a xy system. MCLM and MCCM function are used directly on the coordinate System for this demo application.

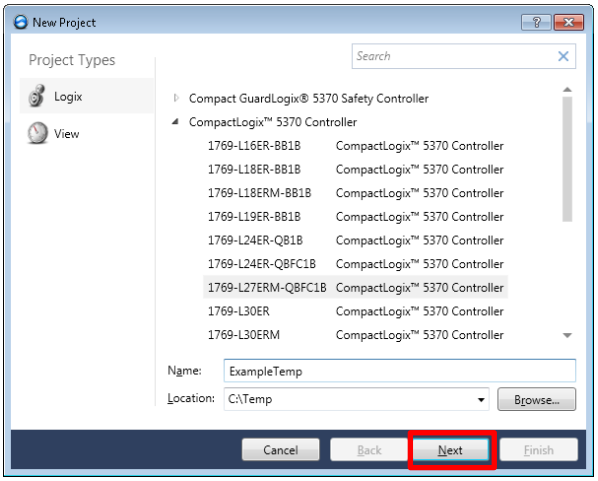
3 New Project in Studio 5000

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.

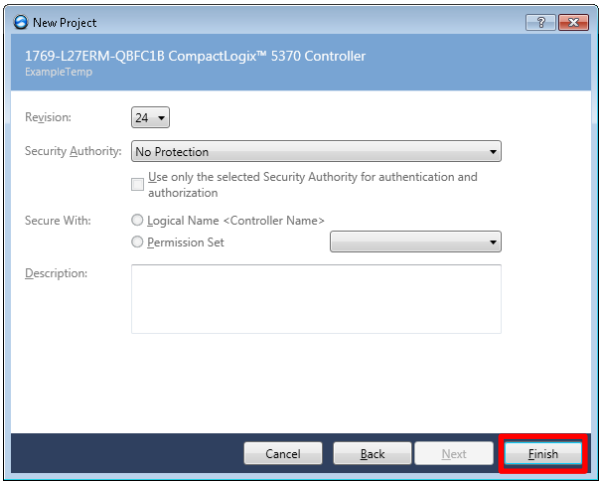
3.1 Create Project

Create a new project.

Choose your PLC and the name of the project.



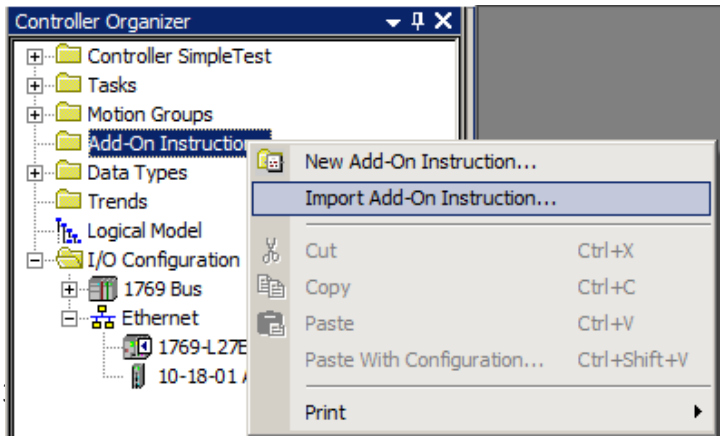
Select your preferred revision and click finish.



3.2 Library Installation

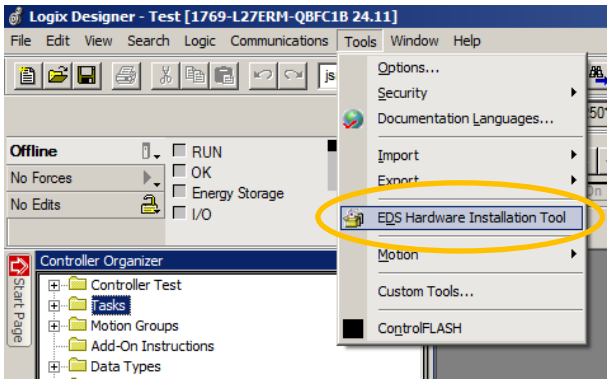
Add the Jenny Science PLCopen library to the project by importing Add-On Instructions.

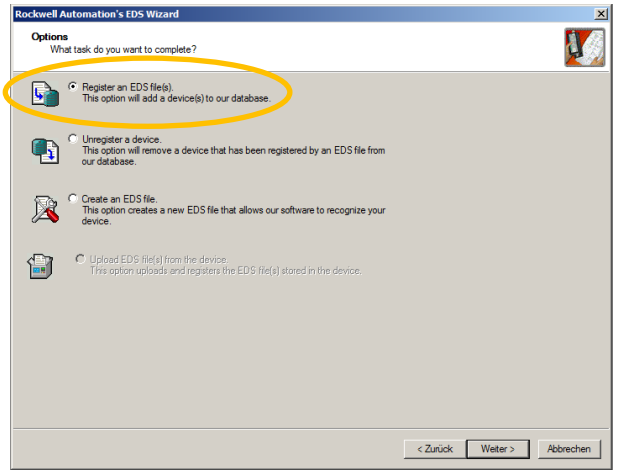
You can import all AOIs and delete the AOIs which are not needed.

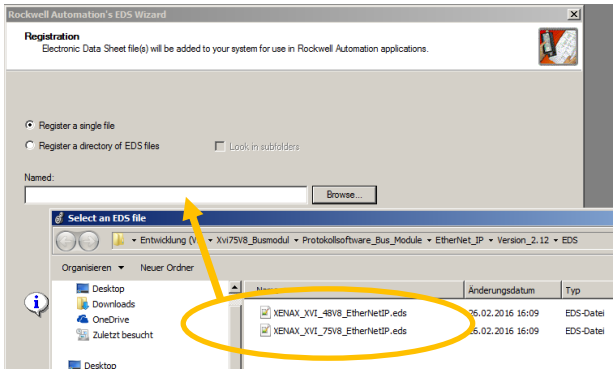


Register the EDS file for your XENAX® controller with the EDS Wizard (EDS Hardware Installation Tool) in Studio5000 Logix Designer.

Download the latest EDS file from your webpage www.jennyscience.ch under XENAX® Servocontroller→Firmware Bus Module.



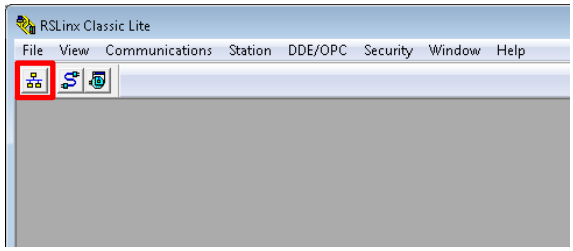




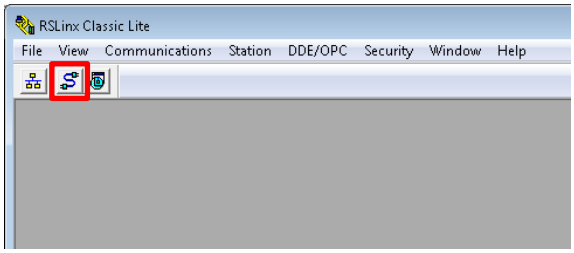
3.4 IP Address Setup

The IP adresse of the XENAX® Busmodule is set to DHCP by default. Use RSLinx to set a fixed IP address.

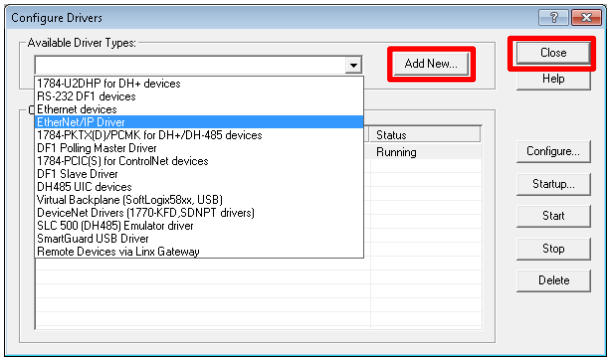
Start RXLinx and Open RsWho.



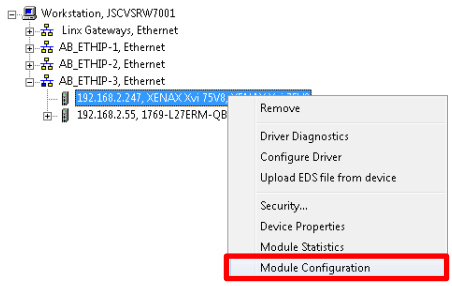
Configure a new Driver if there is none.



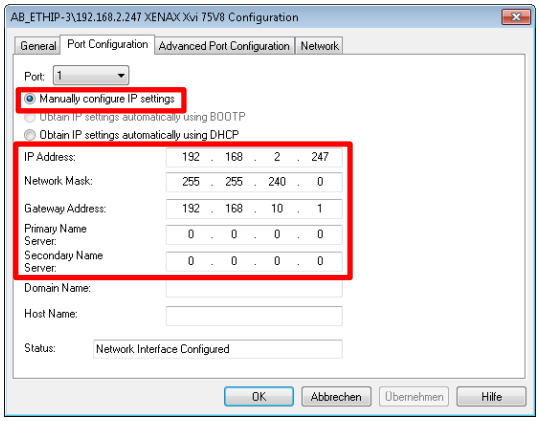
Select the EtherNet/IP driver, click on add New, follow the instructions and click close at the end.



Open Module Configuration of each XENAX® Servocontroller.

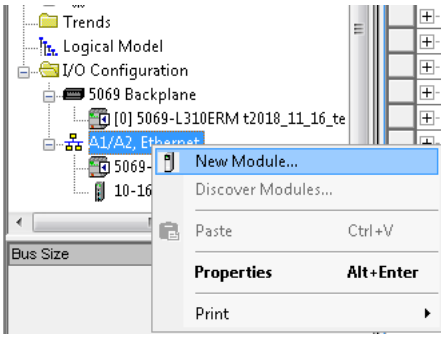


Switch from DHCP to Manually and configure the IP settings.

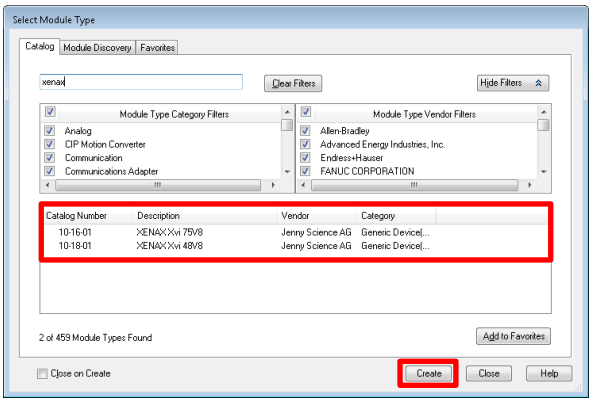


3.5 Add XENAX® Module

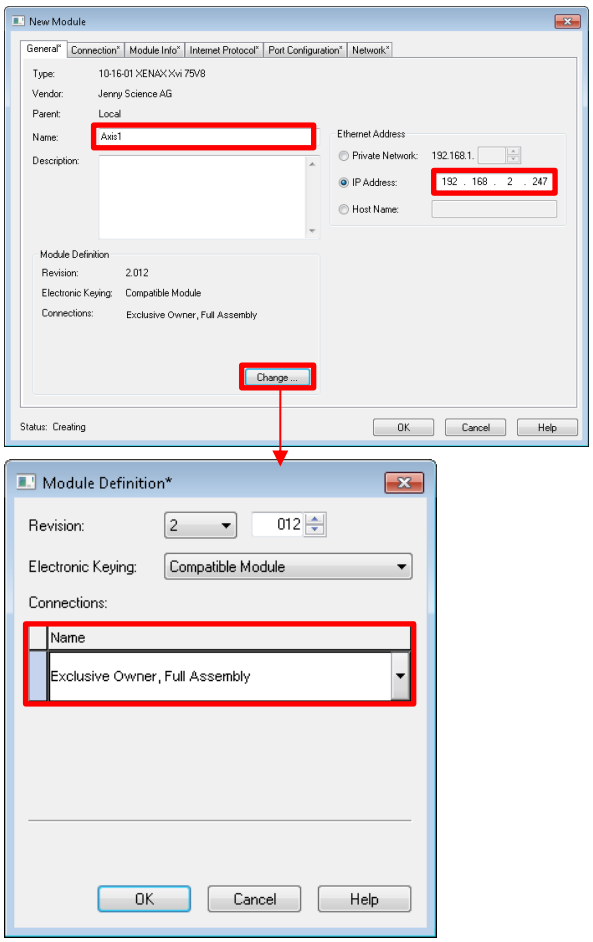
Select **New Module** in the I/O Configuration folder.



Select your XENAX® version and add it to the project.



Name the new XENAX® Module, enter the IP address which was set in RXLinx, and change the connection to “Exclusive Owner, Full Assembly”.

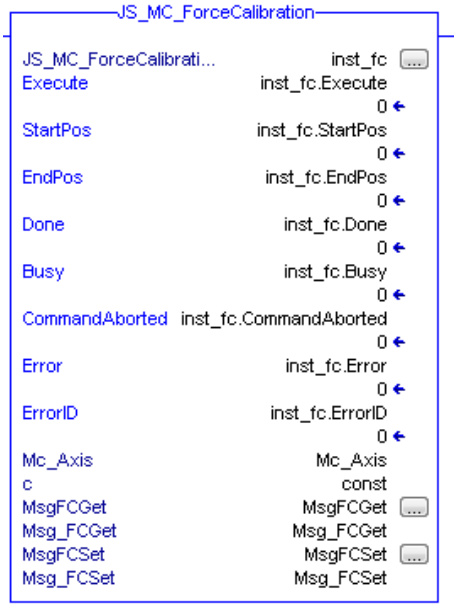


3.6 Mc_Axis and Messages

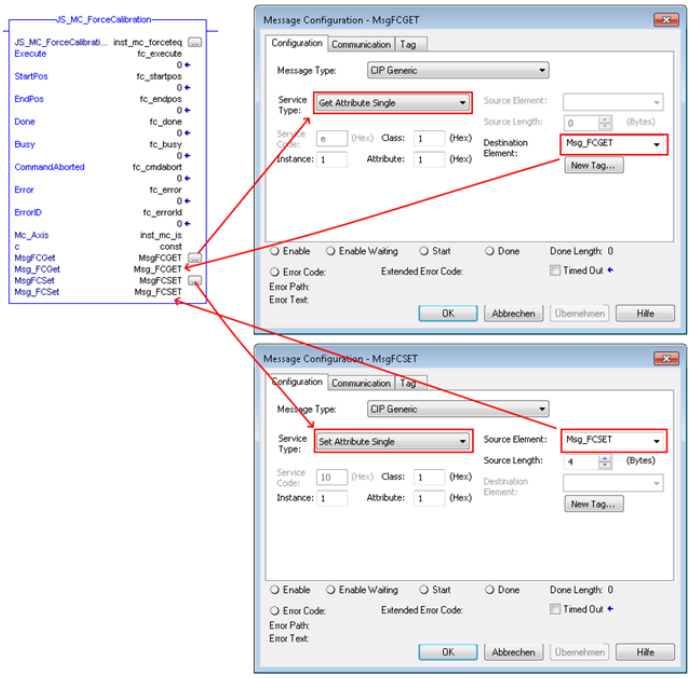
The Library is described in detail in chapter 5. However, there are some pitfalls which can be avoided by reading this section.

Every axis requires one instance of the **Mc_Axis** structure. All function blocks for the same axis share the same instance of the Mc_Axis structure. The input and output values of the Mc_Axis structure must be copied from/to the real Axis with a CPS block (see demo application).

Some function blocks require an instance of a **message**. A new instance is required for every block. The messages are mostly configured by the function block itself except for two settings.

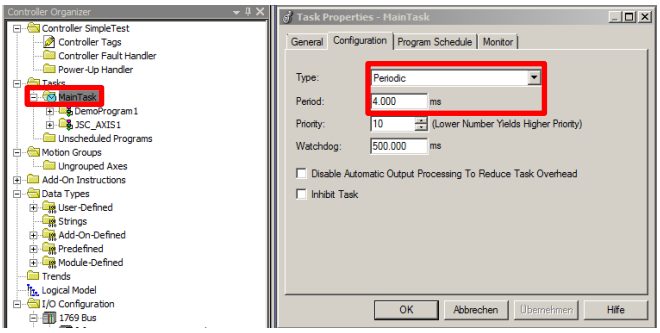


There are get and set messages. The correct service type for each message type and the linked source/destination element must be selected manually. Studio 5000 may ask you to configure other parameters than service type and Destination/Source element. However, those values will be overwritten at runtime.



3.7 Task Cycle Time

A periodic task cycle time of 4ms is recommended.



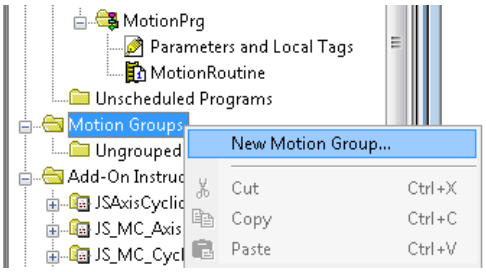
4 Cyclic Synchronous Motion

Cyclic synchronous motion enables driving with a virtual axis. A Jenny Science axis can be linked with a virtual axis. This operation mode requires a few more setup steps than described in the previous step. On the other hand, this driving mode allows to move multiple axes synchronized meaning multiple axes can drive a precalculated path. In addition to that, it is possible to use Coordinate Systems of Studio 5000 where the target position can be specified as [x,y] or [x,y,z] vector.

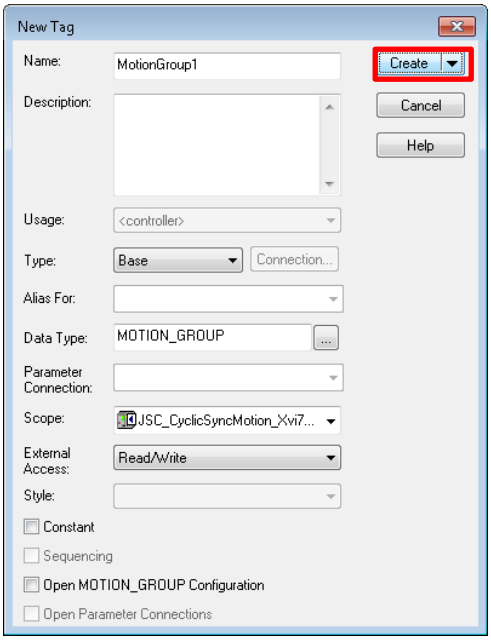
This chapter leads through the additional configuration steps for this mode which were not described in the previous chapter. There are also example project available.

4.1 Virtual Axis

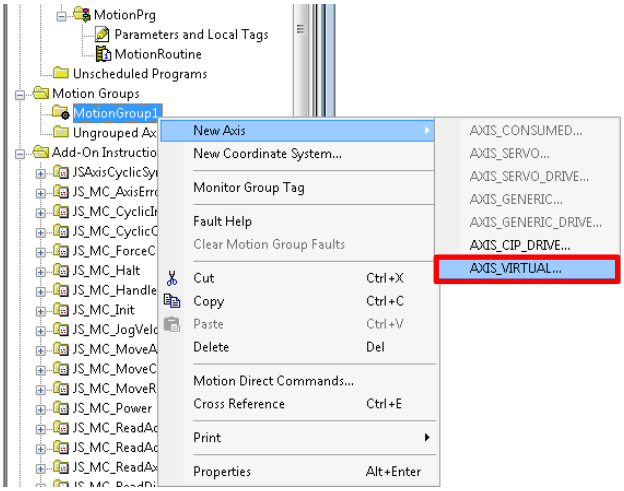
Add a Motion Group



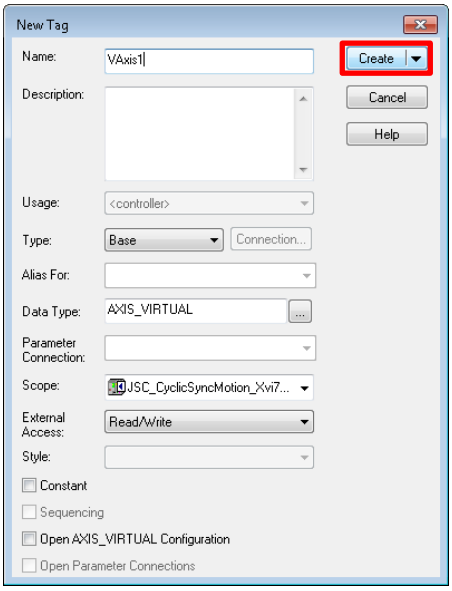
Name the motion group and create it



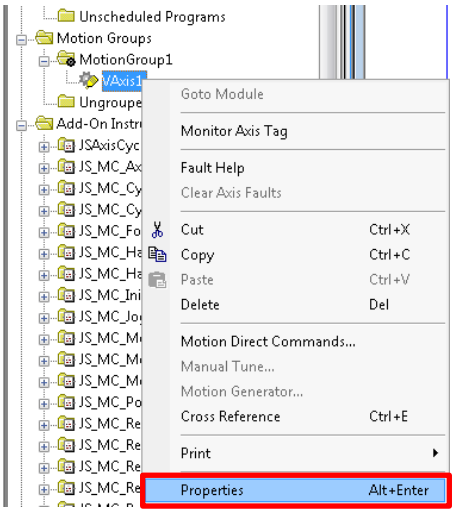
Add a virtual axis to the new motion group.



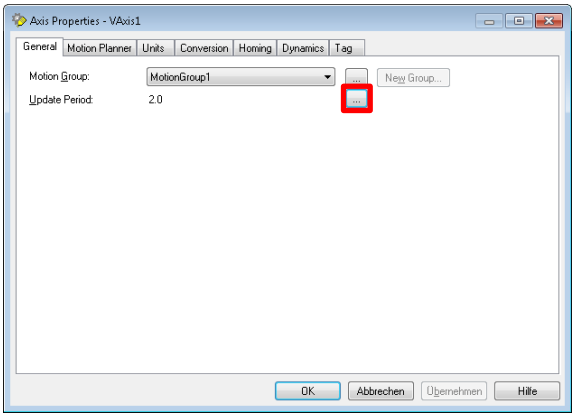
Name the virtual axis and press create.



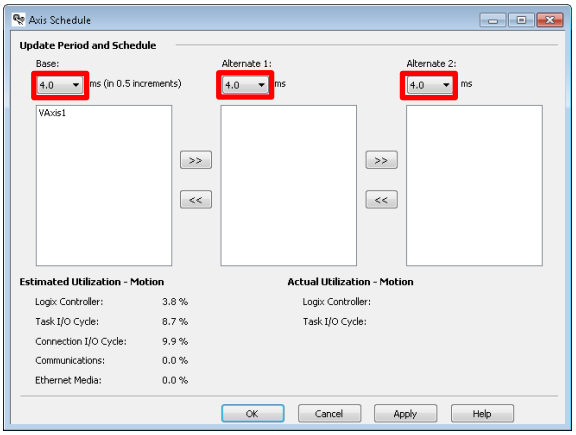
Open the properties of the virtual axis.



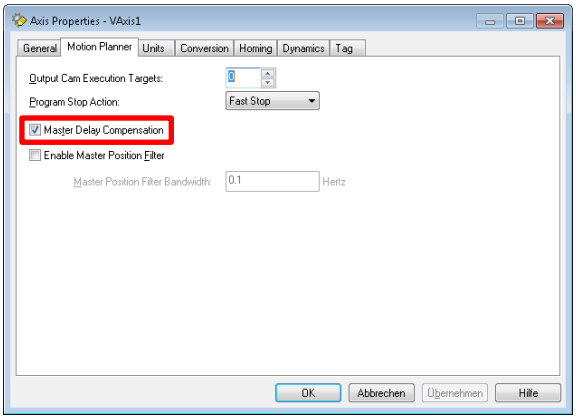
Open the update period settings.



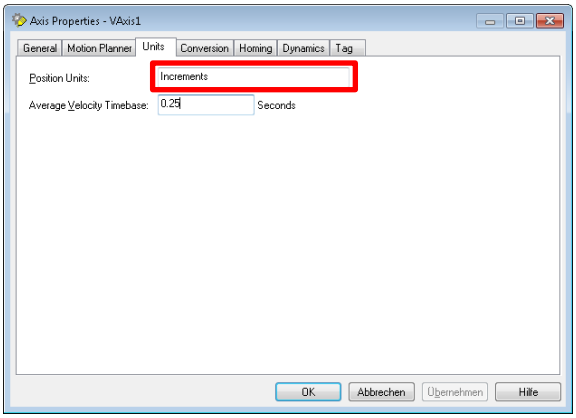
Set all periods to 4ms. Other update periods are currently not supported.



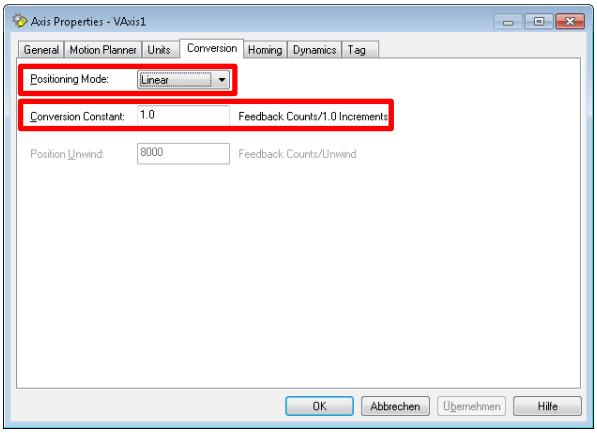
Make sure the master delay compensation is enabled.



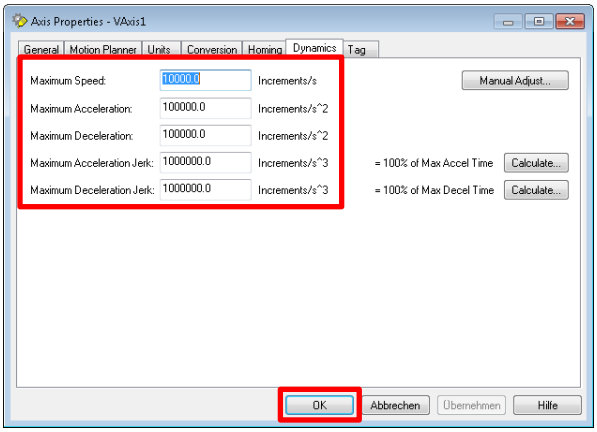
This example uses increments as position units.



Select a positioning mode according to your axis type and specify the ratio between encoder counts and increments. This is 1 in this case.



Set the maximum values according to your application and press ok.

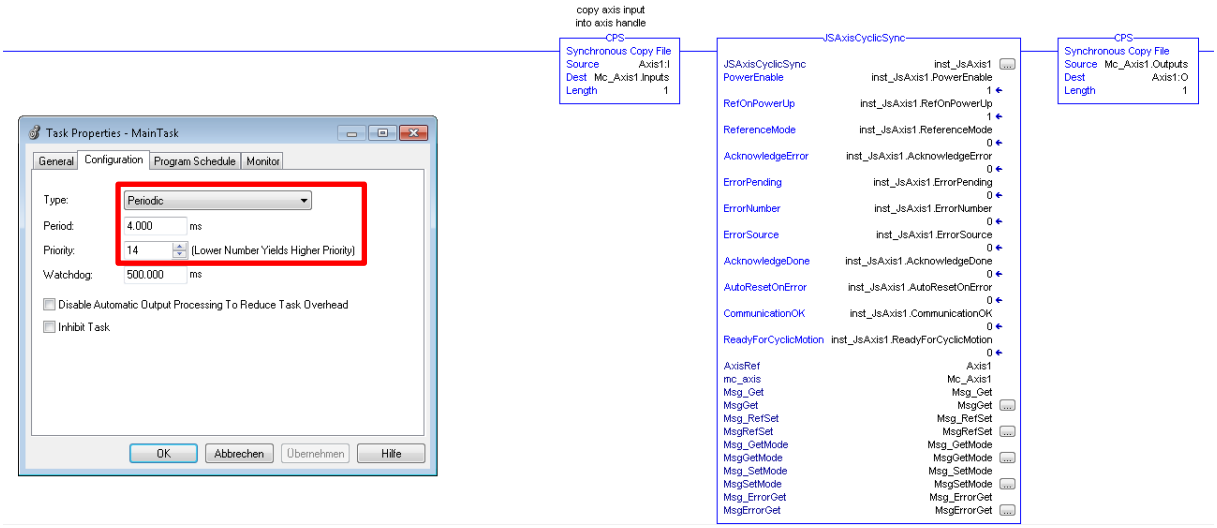


4.2 Task configuration

Two tasks are required for cyclic motion. The main Task is a periodic task with low priority. This periodic task is executed every 4ms or slower. The second task is an event task triggered by the Motion Group.

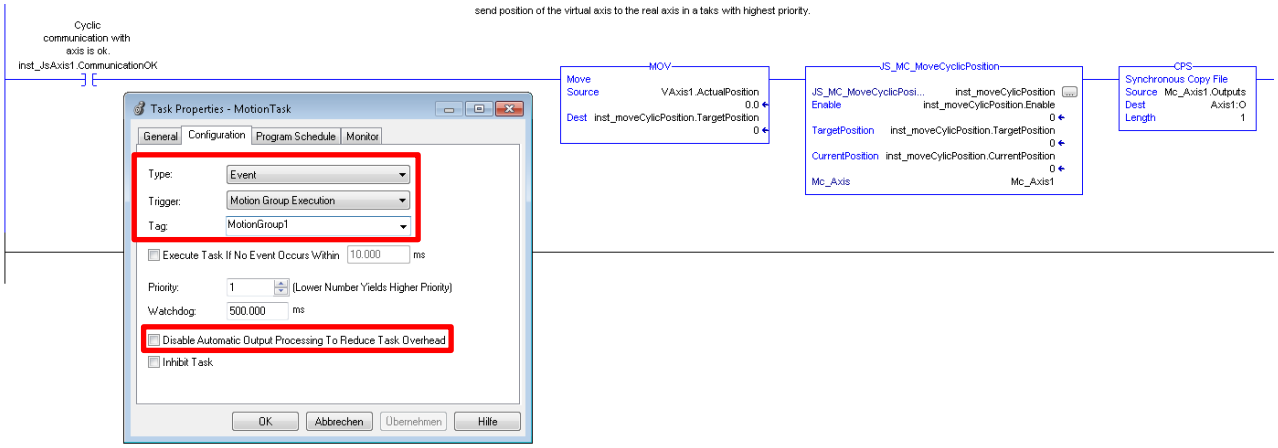
4.2.1 Main Task

The JSAxisCyclicSync is called in the Main Task. Main Task is a periodic tasks with low priority. It is possible to increase the period value to for example 20ms to reduce CPU consumption.



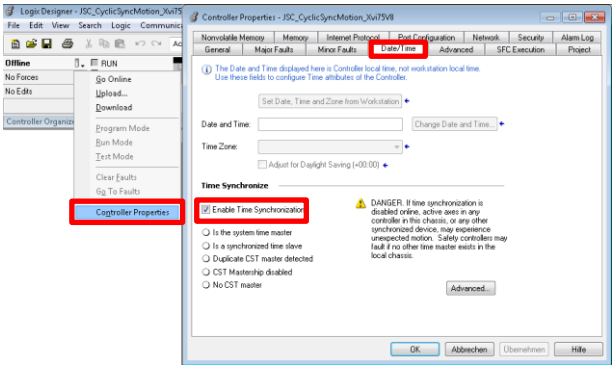
4.2.2 Motion Task

The second task is an event task triggered by the Motion Group. It must have highest priority. Make sure that the second checkbox is disabled. This task links the virtual axis with the real axis.



4.2.3 Enable time Synchronization

Although time synchronization is not used, some PLC families require it to be enabled in the Controller settings.



5 JS_MC_AOP Motion Library

Jenny Science provides a PLCopen library for Studio 5000. The PLCopen standard is easy to understand and includes basic movement functions as well as Jenny Science specific features. This library is called JS_MC_AOP and can be downloaded from www.jennysceince.ch. (See chapter 1.4 Additional Resources)

5.1 State Diagram

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

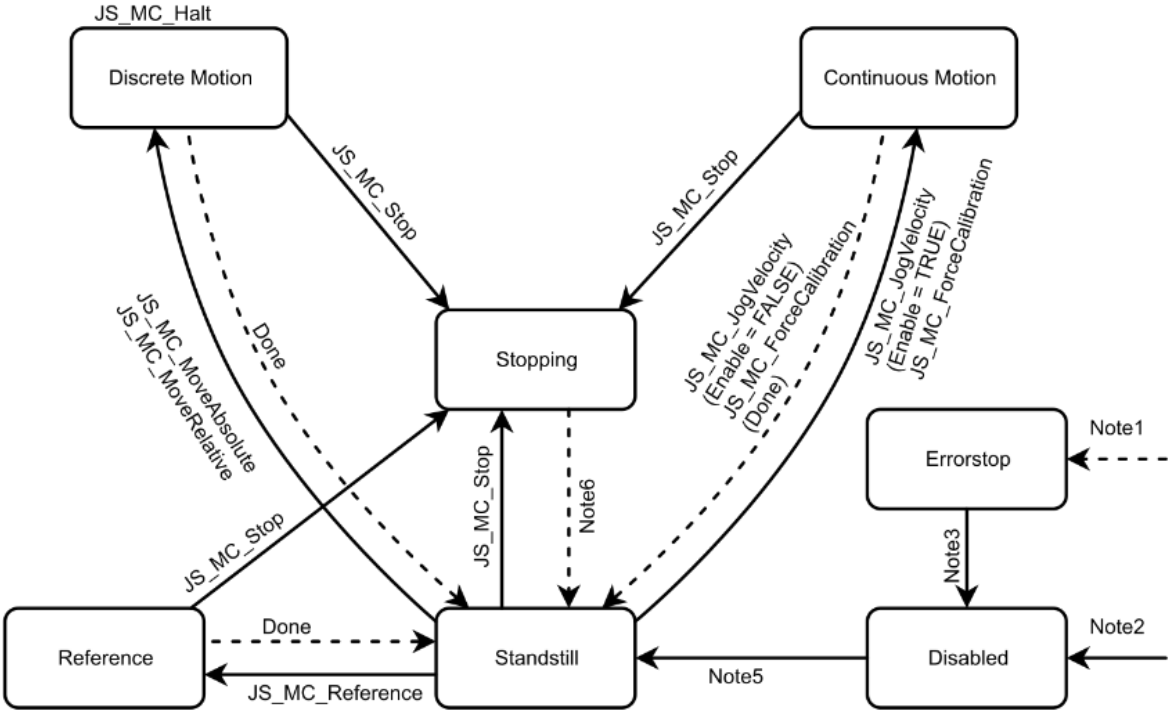
Each motion command is a transition that changes the state of the axis and, as a consequence, influences the method of calculation of the current movement.

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.



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

5.2 Required AOIs

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

5.2.1 Init

This function block must be called once at start up. It initializes the axis reference handle which is needed in all function blocks. The function block must be called before any other JS_MC_AOP block is called.

Signal Name	Direction	Description
Enable	In	The first positive edge initializes the library.
EnforceReferenceDrive	In	Linear axes only perform a reference drive if they are not referenced yet. If set, linear axes will always perform a reference drive. No effect on rotative motors.
OperationMode	In	1 = Profile position, drive with JS_MC_MoveAbsolute 8 = Cyclic Synchronous Motion, drive with JS_MC_MoveCyclicPosition and a virtual axis.
Valid	Out	Initialization finished successfully
Error	Out	Error during Initialization.
ErrorID	Out	Error number
Mc_Axis	In/Out	The axis reference handle.

5.2.2 CyclicIn

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

Signal Name	Direction	Type	Description
Enable	In	Bool	Enables cyclic communication.
Valid	Out	Bool	Cyclic communication is working.
Error	Out	Bool	Error, reset enable to clear error.
ErrorID	Out	DINT	Error number.
AxisRef	InOut	Module	Defines the axis which this function block operates on.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.
MsgGet	InOut	Message	Message to read motor type.
Msg_Result	InOut	SINT[10]	Message result.

5.2.3 CyclicOut

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

Signal Name	Direction	Type	Description
Mc_Axis	InOut	JS_MC_IS	Defines the axis which this function block operates on.
MsgGetMode	InOut	Message	Message to read mode of operation.
Msg_GetMode	InOut	DINT	Destination Element of the GetMode message.
MsgSetMode	InOut	Message	Message to write mode of operation.
Msg_SetMode	InOut	DINT	Source Element of the SetMode message.
MsgGet	InOut	Message	Message to read motor type.

5.2.4 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.

Signal Name	Direction	Type	Description
Enable	Input	Bool	Positive edge enables the drive power stage. Negative edge disables the drive power stage or clears function block error.
Status	Output	Bool	Effective status of the power stage.
Valid	Output	Bool	The power stage is in the requested state.
Error	Output	Bool	Error occurred within function block.
ErrorID	Output	DINT	Error number.
Mc_Axis	InOut	JS_MC_IS	Source Element of the SetMode message.

5.2.5 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.

Signal Name	Usage	Type	Description
Execute	Input	Bool	Start reference at rising edge.
ReferenceMode	Input	SINT	Reference modes: 0: motor default LINAX/ELAX: 2 ROTAX: 10 3-d Party: 10 1: REF_MARK_POS 2: REF_MARK_NEG 3: GANTRY_POS 4: GANTRY_NEG 5: GANTRY_POS_REV_MOT 6: GANTRY_NEG_REV_MOT 10: CW_CCW 11: CW_CW 12: CCW_CCW 13: CCW_CW 14: CW_SHORT 15: CCW_SHORT (For more details, see Appendix 6.1)
ZMarkSpeedRot	Input	DINT	Reference speed with Z-mark [increment/s] (only for rotative drives)
ReferenceSpeedRot	Input	DINT	Reference speed with HW switch [increment/s] (only for rotative drives)
Done	Output	Bool	Reference procedure has finished successfully.
Busy	Output	Bool	The function block is not finished.
CommandAborted	Output	Bool	Function block is aborted by another command.
Error	Output	Bool	Error occurred within function block.
ErrorID	Output	Bool	Error number.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.
MsgRefSet	InOut	MESSAGE	Message to configure reference settings.
Msg_RefSet	InOut	DINT	Source value for Message.

5.2.6 Reset

Resets the XENAX® servo controller. A reset brings the XENAX® servo controller from an error state back to normal operation.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Resets Axis Error on rising edge.
Done	Output	BOOL	Error is cleared.
Busy	Output	BOOL	The function block is not finished.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.2.7 HandleModeOfOperation

This is a library internal function block and must not be instantiated.

5.3 Optional Function Blocks

To save memory, optional function blocks can be removed from the project if they are not needed.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	Enables the Target Position input.
TargetPosition	Input	DINT	Virtual axis position. Use VAxis1.ActualPosition as input value.
CurrentPosition	Output	BOOL	The actual position of the real axis.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.1 MoveCyclicPosition

This AOI links a virtual axis with a Jenny Science Axis. All movements performed on the virtual axis will be reflected on the Jenny Science axis. Only use this AOI when operation mode input of Init AOI is set to 8 (Cyclic Synchronous Position mode).

5.3.2 MoveAbsolute

Drives to an absolute position. The drive is started with a positive edge at the execute input and is finished when done output gets set.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Start move at rising edge.
Position	Input	DINT	Target position [increment].
Velocity	Input	DINT	Maximum velocity [increment/s] (not necessarily reached).
Acceleration	Input	DINT	Maximum acceleration [increment/s2] (not necessarily reached).
Scurve	Input	DINT	S-curve parameter during the acceleration [%].
Done	Output	BOOL	Commanded position reached.
Busy	Output	BOOL	The function block is not finished.
CommandAborted	Output	BOOL	Function block is aborted by another command.
Error	Output	BOOL	Error occurred within function block.
ErrorID	Output	DINT	Error number.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.3 MoveRelative

Drives a defined relative distance from the current position. The drive is started with a positive edge at the execute input and is finished when done equals one.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Start move at rising edge.
Distance	Input	DINT	Target distance for the motion [increment]
Velocity	Input	DINT	Maximum velocity [increment/s] (not necessarily reached).
Acceleration	Input	DINT	Maximum acceleration [increment/s2] (not necessarily reached).
Scurve	Input	DINT	S-curve parameter during the acceleration [%].
Done	Output	BOOL	Commanded position reached.
Busy	Output	BOOL	The function block is not finished.
CommandAborted	Output	BOOL	Function block is aborted by another command.
Error	Output	BOOL	Error occurred within function block.
ErrorID	Output	DINT	Error number.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.4 JogVelocity

Drives with a constant speed. The speed can be changed during motion.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	Activates driving with a constant speed.
JogPositive	Input	BOOL	Executes a movement in the positive direction.
JogNegative	Input	BOOL	Executes a movement in the negative direction.
Velocity	Input	DINT	Value of maximum velocity [increment/s] Note: This value can also be changed while a movement is taking place.
Acceleration	Input	DINT	Value of maximum acceleration [increment/s2] Note: This value can also be changed while a movement is taking place (new value is used at the next velocity change).
Deceleration	Input	DINT	Value of maximum deceleration [increment/s2] Note: This value can also be changed while a movement is taking place (new value is used at the next velocity change)
Jogging	Output	BOOL	Movement being carried out.
Active	Output	BOOL	The function block is active, possible to execute movements.
Busy	Output	BOOL	The function block is not finished.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.5 Halt

Aborts any ongoing move absolute or move relative commands.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Aborts current MoveAbsolute or MoveRelative command on rising edge.
Deceleration	Input	DINT	Deceleration used to stop axis [increment/s2]. Range: 2'000- 1'000'000'000.
Done	Output	BOOL	Zero velocity is reached.
Busy	Output	BOOL	The function block is not finished.
CommandAborted	Output	BOOL	Function block is aborted by another command.
Error	Output	BOOL	Error occurred within function block.
ErrorID	Output	DINT	Error number.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.6 Stop

Stops any motion of the axis and switches to a stopping state. No drive commands are allowed in the sopping state. Reset the execute input to leave the stopping state.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Stops the axis from any movement on rising edge and blocks any further movement. The axis is blocked until execute is released.
Deceleration	Input	DINT	Deceleration used to stop axis [increment/s2]. Range: 2'000- 1'000'000'000.
Done	Output	BOOL	Zero velocity is reached.
Busy	Output	BOOL	The function block is not finished.
CommandAborted	Output	BOOL	Function block is aborted by another command.
Error	Output	BOOL	Error occurred within function block.
ErrorID	Output	DINT	Error number.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.7 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.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Starts a force calibration on rising edge.
StartPosition	Input	DINT	Start position for the force calibration [increment].
EndPosition	Input	DINT	End position for the force calibration [increment].
IterativeFcDisable	Input	BOOL	Set to 1 if the motor oscillates during the Force Calibration. This will clear old calibration data before a new calibration is started.
Done	Output	BOOL	Force calibration procedure has finished successfully.
Busy	Output	BOOL	The function block is not finished.
CommandAborted	Output	BOOL	Function block is aborted by another command.
Error	Output	BOOL	Error occurred within function block.
ErrorID	Output	BOOL	Error number.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.
MsgFCGet	InOut	Message	Get Message for force calibration.
Msg_FCGet	InOut	DINT	Destination element for Get Message.
MsgFCSet	InOut	Message	Set Message for force calibration.
Msg_FCSet	InOut	DINT	Source element for Set Message.

5.3.8 SetOA

SetOA is used to set optional parameters in the cyclic transmitted output assembly.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Writes the value on rising edge.
Index	Input	SINT	Select a variable from the output assembly. 5: LimitIForce 6: FollowingPositionErrorWindow 7: TargetPositionWindow
Value	Input	DINT	Value to set in the output assembly. Note: LimitIForce is set in 10mA steps. A value of 20 corresponds to 200mA. The other two parameters are set in increments.
Done	Output	BOOL	Value is set.
Error	Output	BOOL	Error occurred within function block.
ErrorID	Output	DINT	Error number.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.9 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.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	As long as "Enable" is TRUE, the actual Process Status Register of the XENAX controller is read continuously
Valid	Output	BOOL	A valid set of outputs is available at the function block.
ProcessStatusRegister	Output	JS_MC_IS_ProcStat	Process Status Register of the XENAX controller (For more details, see XENAX® documentation).
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.10 ReadStatus

Reads the current PLCopen state of the XENAX® servo controller. Only one state can be active at the time.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	As long as Enable is TRUE, the axis status is read continuously.
Valid	Output	BOOL	A valid set of outputs is available at the function block.
Disabled	Output	BOOL	JS_MC_Power has not powered the axis, or an error had been acknowledged by JS_MC_Reset and the axis has been turned off.
Standstill	Output	BOOL	JS_MC_Power powers the axis, but no motion command is active.
Reference	Output	BOOL	JS_MC_Reference has started referencing the axis.
DiscreteMotion	Output	BOOL	Axis is in motion due to one of the following function blocks: JS_MC_MoveAbsolute, JS_MC_MoveRelative JS_MC_Halt.
ContinuousMotion	Output	BOOL	Axis is in motion due to the following function block: JS_MC_JogVelocity, JS_MC_ForceCalibration.
Stopping	Output	BOOL	JS_MC_Stop is active.
ErrorStop	Output	BOOL	An error has occurred. Use JS_MC_Reset to acknowledge errors.
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.11 ReadDigitalInput

Reads digital inputs which are located in the XENAX® socket.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	As long as "Enable" is TRUE, the digital inputs are read continuously
Valid	Output	BOOL	A valid set of outputs is available at the function block.
DigitalInput	Output	DINT	The value of digital inputs (bit-coded)
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.12 WriteDigitalOutput

Writes digital outputs which are located in the XENAX® socket.

Signal Name	Direction	Type	Description
Execute	Input	BOOL	Writes the DigitalOutput value at the rising edge
DigitalOutput	Input	DINT	The value of digital outputs (bit-coded)
Done	Output	BOOL	Digital outputs are written
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.13 ReadActualPosition

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

Signal Name	Direction	Type	Description
Enable	Input	BOOL	As long as "Enable" is TRUE, the actual position is read continuously.
Valid	Output	BOOL	A valid set of outputs is available at the function block.
Position	Output	DINT	Actual position of the axis [increment].
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.14 ReadActualCurrent

Reads the motor current in mA.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	As long as "Enable" is TRUE, the actual current is read continuously.
Valid	Output	BOOL	A valid set of outputs is available at the function block
Current	Output	INT	Actual motor current [mA].
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.3.15 ReadAxisError

ReadAxisError records all errors and stores them in a list. Each error is then displayed and can be acknowledged individually. In case of an axis error, the error number of the axis is shown. A displayed function block error will provide the function block error number as well as the source block which produced the error.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	As long as Enable is TRUE, the function block can be used to read out axis and function block errors
AcknowledgeRecorded	Input	BOOL	Acknowledges the error record currently displayed at the output of this function block. Note: Only the displayed error record is acknowledged. The errors at the axis must be first acknowledged with the reset function block. Similarly, a function block error must be first acknowledged by resetting its enable or execute input.
Busy	Output	BOOL	The function block is not finished Note: This output is set after a positive edge at AcknowledgeRecorded until the new error is displayed.
ErrorRecordAvailable	Output	BOOL	Set if a recorded error is displayed.
RecordedErrorNumber	Output	DINT	Error Number (see chapter 5.5)
RecordedSource	Output	SINT	Indicates where the error happened (see chapter 5.6).
RecordedType	Output	SINT	1: Axis error 2: Axis warning 3: Function block error
AxisHasError	Output	BOOL	Pending axis error to display.
AxisHasWarning	Output	BOOL	Pending axis warning to display.
FunctionBlockErrorCount	Output	DINT	Number of pending function block errors to display.

Error	Output	BOOL	Error occurred within this function block (JS_MC_ReadAxisError).
ErrorID	Output	DINT	Error number of this function block (JS_MC_ReadAxisError).
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.
MsgErrorGet	InOut	Message	Get Message to read out axis error number.
Msg_ErrorGet	InOut	INT	Destination variable for get message.

5.3.16 AxisErrorCollector

The AxisErrorCollector is a lightweight version of the ReadAxisError. This block shows only that an error happened, but not exactly which one and where.

Signal Name	Direction	Type	Description
Enable	Input	BOOL	As long as "Enable" is TRUE, the function block determines error information.
Valid	Output	BOOL	A valid set of outputs is available at the function block.
AxisError	Output	BOOL	The axis is in error state.
AxisWarning	Output	BOOL	The axis is in warning state.
Errorstop	Output	BOOL	The current PCLopen State is "Errorstop".
FunctionBlockError	Output	BOOL	A Jenny Science function block is in error state
Mc_Axis	InOut	JS_MC_IS	The axis reference handle.

5.4 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 %

5.5 Error Numbers of JS_MC_AOP

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 Check in software if CPS block in front of JS_MC_INIT copies the axis input into the correct axis structure. (see demo application)
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	For library use, ~~at least~~ XENAX firmware V3.64D and powerlink bus module firmware V2.0 or higher are required
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 ~~so~~ 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

5.6 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

5.7 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 warning
3	Function block error

6 Appendix

6.1 Reference methods

LINAX® linear motor axes	
1	REFERENCE, start direction positive
2	REFERENCE, start direction negative
3	REFERENCE, gantry system, direction positive, linear motor axes same measurement system orientation
4	REFERENCE, gantry system, direction negative, linear motor axes same measurement system orientation
5	REFERENCE, gantry system, direction positive, linear motors axes contrary measurement system orientation
6	REFERENCE, gantry system, direction negative, linear motors axes contrary measurement system orientation
ROTAX® and third-party motors	
10	REFERENCE, start clockwise -> external reference Sensor(*), continue counter clockwise -> Z-mark
11	REFERENCE, start clockwise -> external reference Sensor(*), continue clockwise -> Z-mark
12	REFERENCE, start counter clockwise -> external reference Sensor(*), continue counter clockwise -> Z-mark
13	REFERENCE, start, counter clockwise -> external reference Sensor(*), continue clockwise -> Z-mark

(*) If there is no external sensor, then set input
“ReferenceSpeedRot” = 0

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