IO-LINK for MIS

Table of contents

1 History of document

Version	Author	Review status	Date
0.4 (draft/released)	HC	DRAFT	20250618
0.3 (draft/released)	HC	DRAFT	20250129
0.2 (draft/released)	HC	DRAFT	20241104
0.1 (draft/released)	HC	DRAFT	20241003

2 IO-LINK specification

Multiple IO-Link modes are supported. Mode66 is the default, “out of the box” -configuration.

This mode supports the JVL Function blocks for Siemens and Rockwell Add on instructions.

Mode66

ServoStepper	
Vendor ID	1374 (0x055e)
Vendor Name	JVL A/S
Vendor Text	JVL A/S
Vendor URL	http://www.jvl.dk
Device ID	26129 (0x006611)
DeviceFamily	MIS - ServoStep
Features	
Block Parameter	yes
Data Storage	yes
Profile Characteristic	0x4000 (Common Application Profile: Identification and Diagnosis)
Supported Access Locks	Parameter: no, Data Storage: no, Local Parameterization: no, Local User Interface: no
Communication	
IO-Link Revision	V1.1
Transmission Rate	230400 bit/s (COM3)
Minimum Cycle Time	5 ms
SIO Mode Supported	no
M-Sequence Capability	PREOPERATE = TYPE_1_V with 8 octets on-request data OPERATE = TYPE_2_V with 2 octets on-request data ISDU supported
Device Variant	**MIS-ServoStepper MODE66**
Description	MIS ServoStepper-Iolink
Product ID	MIS23x
Device Icon	
Device Symbol	
Connection Type	M12-5 connector
- pin 1	brown; L+
- pin 2	violet (purple); NC
- pin 3	(light) blue; L-
- pin 4	black; C/Q
- pin 5	grey (slate); NC

ModeFM

ServoStepper	
Vendor ID	1374 (0x055e)
Vendor Name	JVL A/S
Vendor Text	JVL A/S
Vendor URL	http://www.jvl.dk
Device ID	529 (0x000211)
DeviceFamily	MIS - ServoStep
Features	
Block Parameter	yes
Data Storage	yes
Profile Characteristic	0x4000 (Common Application Profile: Identification and Diagnosis)
Supported Access Locks	Parameter: no, Data Storage: no, Local Parameterization: no, Local User Interface: no
Communication	
IO-Link Revision	V1.1
Transmission Rate	230400 bit/s (COM3)
Minimum Cycle Time	1 ms
SIO Mode Supported	no
M-Sequence Capability	PREOPERATE = TYPE_1_2 with 2 octets on-request data OPERATE = TYPE_2_V with 2 octets on-request data ISDU supported
Device Variant	**MIS-ServoStepper MODEFM**
Description	MIS17/23-Iolink
Product ID	MIS23x
Device Icon	
Device Symbol	
Connection Type	M12-5 connector
- pin 1	brown; L+
- pin 2	violet (purple); NC
- pin 3	(light) blue; L-
- pin 4	black; C/Q
- pin 5	grey (slate); NC

Please observe

For system integration it is important to note that the IO-Link cycletime isn't necessarily the system cycletime, which can vary depending on the IO-Link master used, the packet interval of the bus communicating with the IO-Link master from the PLC and other parameters.

The features of IO-Link specification V 1.1.3 is supported

The JVL IO-LINK implementation is currently under development. Some standard IO-LINK functions might not be currently available.

IODD files and function blocks are available for download following this link:

https://www.jvl.dk/431/Fieldbus-PLC-Demo-Programs

3 Commissioning

3.1 Power and IO-Link connection

The MIS motors supports IO-Link Class A operation, meaning that an external power supply is needed and must be connected to according to the JVL specifications to the “PWR” inlet.

However due to the limited power consumption, the NEMA17 MIS17x -motors may be operated using IO-Link Class B connections, meaning that power is delivered through the IO-Link connector. A limited version is available only equipped with 2x M12 plugs.

This type is denoted MISxxx **L2** xxxxx, 1x8pin (IO1-4 + RS485) and 1x 5-pin for IO-Link (Class B).

The fully equipped version with the PWR connector is denoted MISxxx **L5** xxxxx.

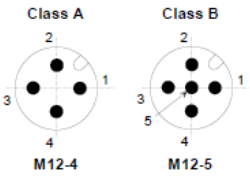
Please note that the supply voltage to the motor will be limited compared to the normal operating range of the supply voltage. This will influence the torque performance at higher velocity.

The max. current limit of the IO-Link master must also be considered.

Caution should be taken, since both the power and the IO-Link connector have the same mating.

The motor is protected internally against a cable intended for power going into the IO-LINK connector.

Common for all MIS motors is, that the IO-Link connection is established through **CN3 using an M12 connector.**



Pin	Signal	Designation	Remark
1	L+	Power supply (+)	See Table 7
2	I/Q a)	NC/DI(OSSDe)/DO/ AI/AO (port class A)	Option 1: NC (not connected) Option 2: DI (Master's view) Option 3: DO (Master's view) Option 4: Analog signal (I / U) d) Option 5: OSSDe (see [10])
	P24 b)	P24 (port class B)	Extra power supply for power Devices (port class B)
3	L-	Power supply (-)	See Table 7
4	C/Q c)	SIO(OSSDe)/SDCI	Standard I/O mode (DI/DO) or SDCI (see Table 6 for electrical characteristics of DO). See [10] for OSSDe definitions.
5	NC	NC (port class A)	Option 1: Shall not be connected on the Device side (port class A).
	N24	N24 (port class B)	Option 2: Reference to the extra power supply (port class B)

JVL has the following IO-Link cables available:

WI1000-M12M5TF5T.5N	500mm cable length
WI1000-M12M5TF5T01N	1000mm cable length
WI1000-M12M5TF5T03N	3000mm cable length

WI1000-M12M5TF5T20N	20000mm cable length

Other cable lengths are available upon request.

For further information about the PWR, CN2 and CN4, please find the official ”ServoStep” -manual, following this link:

https://www.jvl.dk/files/pdf-1/user%20manuals/lb0058gb.pdf

3.2 IODD file installation

For the commissioning of the motor it is necessary to install the necessary IODD files delivered from JVL. This file holds all the necessary data formats for the PLC to instantiate communication and present the data and format for the PLC programming environment.

The examples and screenshots used are taken from the Rockwell Logix5000 environment and from the Baumer “Sensor suite” however the procedures are the same for other vendors of IO-Link master devices.

3.2.1 Steps for importing the IODD files

2 different IODD files are supplied, one for each IO-Link mode.

Note that the IODD filename is formatted according to the IO-Link specifications and holds a name and a release date. The illustrated IODD filenames may not be the latest available.

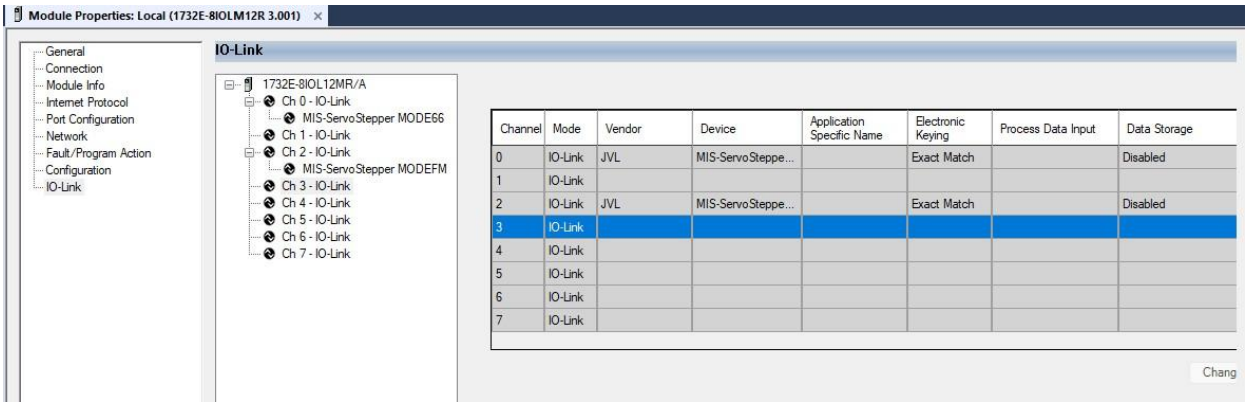
1. From the IODD import dialog, select the JVL IODD file for the requested mode:

The imported JVL devices are added to the list:

2. For the relevant channel, select which JVL device is connected. Note that by default, Mode66 is activated.

Observe that this process can vary depending on the IO-Link master vendor. For the Rockwell Armorblock each channel needs to be assigned to a specific IODD configuration.

Here Channel 0 and channel 2 is assigned.



3. Download the configuration to the PLC and make sure the motor is connected to the IO-Link master and power supply.

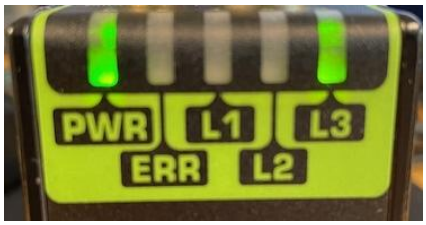
When the IO-Link connection has been established, the data exchange is running, both cyclically and the ISDU parameters can be handled.

3.3 Status indicator

The status indicators on the rear of the motor indicate the current status of the IO-Link connection.

When the motor is ready for the connection to be established, the L3-LED will blink.

When the IO-Link connections have been established, L3 will remain solid green.



4 Data structure

To satisfy the demand for both fast reacting IO-LINK with short cycletime and high dynamic control as well as a high degree of control over a lot of motion specific parameters, JVL has implemented 2 different IO-LINK data models.

One taking advantage of the integrated Motion Vector system “FastMac” which basically applies 3 motion settings in one command. The benefit by using this system is, that the bandwidth requirement is reduced significantly and the demand for data begin cyclically exchanged is drastically limited. Hence the cycletime can be optimized accordingly.

The disadvantage of using this system is that motion specific data such as actual position and velocity isn’t exchanged cyclically. The position, velocity and acceleration parameters must be configured in vectors.

The motor will move between these fixed points using a simple system. A few bits in a command register selects the vector to apply as well as starting the actual motion.

Only the most vital data is exchanged in 2 bytes each way.

The “FastMAC” motion system is ideal in gripper applications, where the motor always moves between 2 specific positions.

The second and more conventional method transfers 6 internal 32bit registers each way thus requiring higher bandwidth which again reduces the cycletime.

The benefit of using this method is that all motion related parameters are transferred in every cyclic data exchange.

The positioning and motion of the motor is more versatile as the exact positioning is controlled in every cycle.

The status feedback from the motor is also exchanged so that parameters such as actual velocity, position and applied torque can be read back during the motion.

The 2 different modes are referred to as:

ModeFM – 2 Octet of cyclic data.

Mode66 – 24 Octet of cyclic data, 6x motor registers each way.

4.1 ISDU parameters

The ISDU parameters are basically parameters that are accessible in an acyclic way (on request).

These parameters cover configurations like homing procedures and positions, I/O configuration etc.

Using the Datastorage functionality, the ISDU parameters declared in the IODD file can be saved automatically in the IO-Link master and a motor can be replaced in the application without the need for any external tool to be used.

The configuration of a new “out of the box” -motor is done automatically.

See the section DataStorage for further details.

The ISDU parameter selection, varies depending on the current mode.

5 IO-Link mode Mode66

The processdata of Mode66 transfers 6x 32bit motor registers in both the read and write -direction. This data format is very similar to the format used by the JVL Industrial Ethernet solutions and hence the support and integration into different PLC environment is more seamless through the well known JVL AOI's for Rockwell and function blocks for Siemens.

5.1 ISDU Parameters

The following parameters are accessible:

Process data | Identification | **Parameter** | Events | Favorites

NAME	VALUE	RANGE	DESCRIPTION
Homing Position	78 001	2 147 483 647 -2 147 483 647	
Homing Velocity	-333.66 rpm	3 000 -3 000	
Homing Torque	50.0 %	73.2779685 0.048851979	Torque thresshold for mechanical homing.
Homing sensor selection	0000 1000b	128 1	Homing sensor input selection IO1..8 bit[0..7]
Max. allowed follow err	0	2 147 483 647 -2 147 483 647	
Bottom Position limit	0	2 147 483 647 -2 147 483 647	Bottom Position limit or Modulus lower working range -position.
Top Position limit	0	2 147 483 647 -2 147 483 647	Top Position limit or Modulus top working range -position.
Error deceleration[RPM/s]	10 000	500 000 1	
IO Configuration	1111 1111b	65 535 0	IO1..8 Configuration [bit0..7]: 1=active high, [bit8..15]: 0=Input/Output]
Brake output selction	0000 0000b	255 0	Brake output selction IO1..8, use multiple for higher current.
Min. Torque applied	0.0 %	100.000001013 0	
Modulus operation setup	0	5 0	[0=disabled, 1=Singletun CW, 2=Singleturn CCW, 3=Multiturn CW, 4=Multiturn CCW]
Errorcode	0	2 147 483 647 0	
Warningcode	0	2 147 483 647 0	
P+ Supply voltage	24.6 V	5.6952E+7 0	
Temperature	35.0 °C	2 147 483.647 0	

5.1.1 Homing Position

[Units: Counts]

When a homing sequence is completed, the reference position is defined by this parameter.

For passive homing, where the current encoder position needs a new reference, this parameter is used.

5.1.2 Homing velocity

[Units: 1/100RPM]

The velocity used during the homing procedure. Note that the direction is determined from the sign.

5.1.3 Homing Torque

[Units: %]

The torque threshold when mechanical homing is performed.

5.1.4 Homing sensor input selection

Using a bitwise value, the input for the homing sensor is selected.

Exc: 0b1000 will select input 4 as homing input. Input 4 is the factory default value.

5.1.5 Max. allowed follow error

[Units: Counts]

Maximum allowed follow error before an error is flagged.

0 = Disabled (Default).

5.1.6 Min. allowed Position/Modulus size

[Units: Counts]

This parameter has 2 different functions depending on the current Modulus mode.

In case the modulus mode is disabled, the parameter defines the Min. position value within an allowed range. In case the motor moves beyond this position, an error is flagged and the motor is only allowed to move in the opposite direction.

In case the current modulus mode is enabled, this parameter indicates the min. Turntable size.

5.1.7 Max. allowed Position/Modulus size

[Units: Counts]

This parameter has 2 different functions depending on the current Modulus mode.

In case the modulus mode is disabled, the parameter defines the Max. position value within an allowed range. In case the motor moves beyond this position, an error is flagged and the motor is only allowed to move in the opposite direction.

In case the current modulus mode is enabled, this parameter indicates the max. Turntable size.

5.1.8 Error deceleration

[Units: RPM/S]

Deceleration used in case an error is flagged.

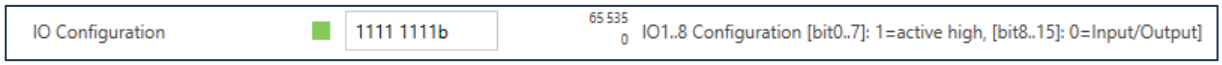
5.1.9 IO1..8 Configuration

[Units: Bitwise]

Bitwise configuration of the IO's available. These IO's can function both as input or output, depending on the configuration. The active level is also configured within this parameter.

The lower 8 bits are used to control the active level

Where 1 = active high 0 = active low



Bits 8..10 will control whether the IO is used for either Input or output.

Where 1 = Input, 0 = Output

Here IO1, IO3, IO5, IO7 is configured as outputs and IO2, IO4, IO6, IO8 is configured as inputs.

5.1.10 Brake Output selection

[Units: Bitwise]

To suit the need for higher current than one output is able to deliver, multiple outputs can be controlled in parallel to supply the needed current. The ISDU parameter "Brake output selection" is used to define which outputs to use.

Here IO1 + IO2 + IO3 is selected.

5.1.11 Min. Torque applied

[Units: %]

When the motor is equipped with an internal encoder, the motor can utilize the advanced closed loop current control. This parameter sets the absolute min. current (Torque) used at any time.

The closed loop current control will always limit the current to the lowest possible value in order to control the motion.

5.1.12 Modulus operation setup

For modulus operation (Turntable) this parameter is used to define the usage.

Value	Function
0	Disabled
1	Singleturn CW rotation
2	Singleturn CCW rotation
3	Shortest path
4	Multiturn CW rotation
5	Multiturn CCW rotation

Note!

The parameters:

Min. allowed Position/Modulus size *and* ***Max. allowed Position/Modulus size***
Defines the operating window for the modulus operation.

Please refer to the manual for further information about the Modulus (turntable) operation.

5.1.13 Error and other status information

The ISDU parameters also offers 3 additional ro -parameters to get the current status for error, warning, temperature and the actual Bus voltage.

These RO (Read Only) parameters are available for both the Mode66 and the ModeFM setting.

Errorcode	ro	0	
Warningcode	ro	0	
P+ Supply voltage	ro	20.6	V
Temperature	ro	32.2	°C

In case of an error the error code will display a value that will reveal the error(s).

The following bits are encoded in the errorcode:

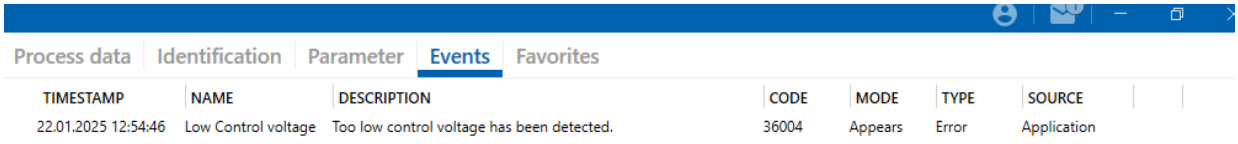
Bit 0: General error bit, always set with another bit.

Bit	Error description
0	General error bit, always set with another bit
1	Follow error
2	Output driver
3	Position limit
4	Low bus voltage (Default: set when bus voltage goes below 15V))
5	Over voltage
6	Temperature > 90°C
7	Internal, Self diagnostic detected an internal error
8	Absolute multiturn encoder lost position
9	Absolute multiturn encoder sensor counting error
10	Absolute multiturn encoder communication lost
11	SSI Encoder counting error
12	Closed-loop error
13	External memory error
14	Absolute singleturn encoder error
15	H4 Internal encoder error
16	Zero search timeout (sensor or mechanical torque threshold not detected within time)
17	CVI control voltage unstable
18	Motor driver overload
27	STO Alarm (Safe Torque Off)
29	STO

Warning codes are also available encoded in hexadecimal value as follows:

Bit	Error description
0	Positive position limit active (only motion in opposite direction possible)
1	Negative position limit active (only motion in opposite direction possible)
2	Positive limit has been active
3	Negative limit has been active
4	Low bus voltage
5	Reserved
6	Temperature > 80°C
7	SSI Encoder
8	Driver overload
9	STO active (Safe Torque Off)

In case of an error, an event is raised with a timestamp. All events are handled by the IO-Link master according to the IO-Link standard.



The event codes and descriptions are described in the IODD files.

P+ Supply voltage

The current measured bus voltage.

Note!

For MIS17x a class B supply from the IO-Link master to the motor can supply both the control and the bus voltage for the driver. The max. voltage in this case will be limited by the IO-Link master.

For all other motors a separate supply is needed for the motor and the P+ voltage will be limited by the specifications for the actual motor.

Temperature

The current internal measured temperature. A warning is issued if the temperature rises above **80°C**, the motor is faulted if the temperature exceeds **90°C.**

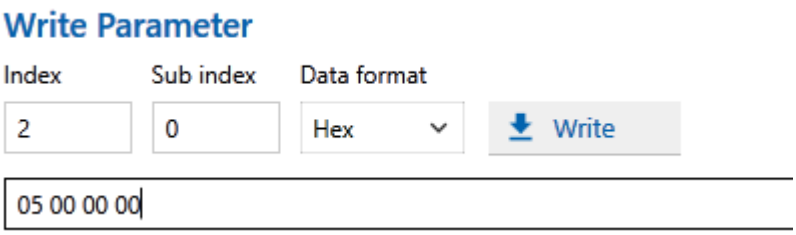
To save the parameters permanent in non-volatile memory, issue the command **127** into the **CMD** -byte of the processdata.

Datastorage is also supported, please find the “DataStorage” -section for details.

5.1.14 Save parameters in non-volatile memory

The parameters are saved in non-volatile memory by applying the systemcommand “ParamDownLoadstore” index=2, value=5:

or use send the systemcommand “ParamDownloadStore” Index = 2, value = 5.

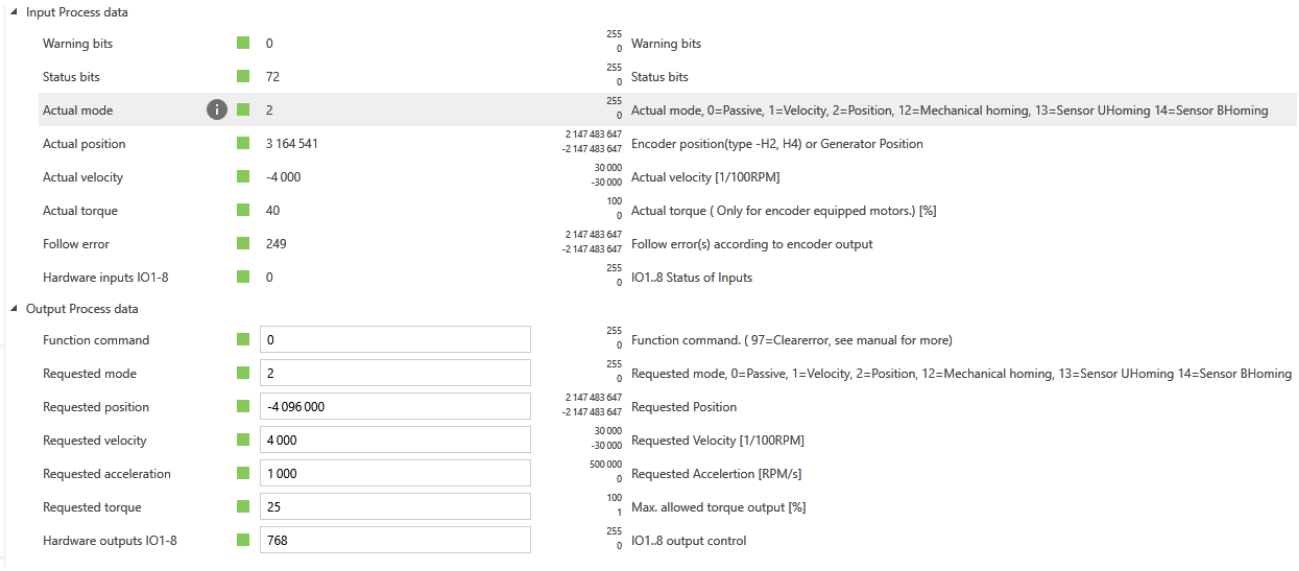


Note the order of bytes.

The connection is temporarily lost, but regained after a few seconds.

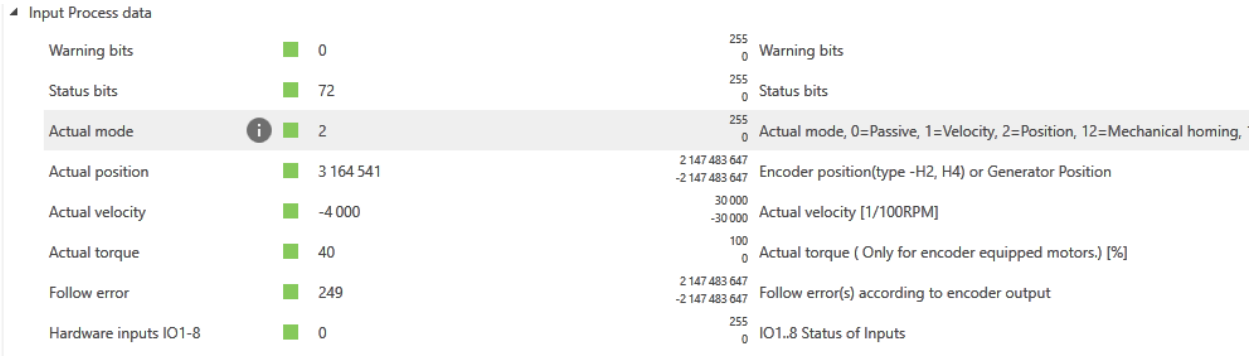
5.2 Processdata Mode66

The processdata format consists of 24bytes in both input and output.



Please note that function blocks for Siemens and AOI's for Rockwell has been developed, so that required knowledge about the different parameters is limited.

5.2.1 Input process data:



5.2.1.1.1 Warning bits

Units: Bitwise

Size: 1 octet, value range 0-255

Warning bits are distributed as follows:

Bit 0 : Positive Position Limit Active The Actual position has exceeded the Max. position limit configured.

Bit 1: Negative Position Limit Active The Actual position has exceeded the Min. position limit configured.

Bit 2: Low bus voltage A low bus voltage was detected.

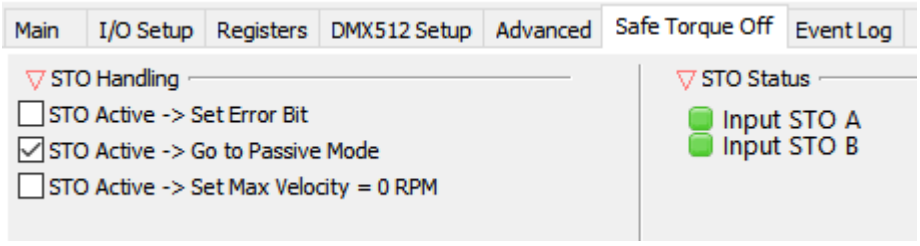
Bit 3: IO Driver overload, Overloaded output driver for IO1..8.

Bit 4: Temperature critical The temperature has reached a critical limit of **80°C.**

Bit 5: Overload, Motor driver overload.

Bit 6: STO Active, Safe Torque Off feature active. *Please Observe that the handling of the motor highly depends on the settings.*

From MacTalk, register 124, bit 29, 30, 31



7: Reserved

5.2.1.1.2 Status bits

Units: Bitwise

Size: 1 octet, value range 0-255

Status bits is distributed as follows:

0: Reserved

1: InPositon Actual position is within the configured position window

2: Reserved

3: HomingDone or Motor homed in lifetime.

4: Accelerating, Motor is accelerating

5: Decelerating, Motor is decelerating

6: At velocity, Motor has reached the requested velocity

7: Error An error has occurred for further details consult register 35 for details

5.2.1.1.3 Actual mode

Units: None

Size: 1 octet, value range 0-255

The motor supports different modes of operation, the most common used by function blocks and AOI are:

0: **Passive**, Motor is passive and shaft is torqueless.

1: **Velocity** (typically used for jogging purposes)

2: **Position**, The position of the motor shaft is controlled, with respect to the parameters Max. Velocity, Acceleration and torque.

12: **Mechanical homing**, The motor will move in a direction determined by the parameter “Homing velocity” up against a hard stop. The hard stop is determined based on the actual torque exceeding the settings in the parameter “Homing torque”.

13: **Sensor homing Unidirectional (UHoming)**, The motor will run in a direction with a velocity determined by the parameter “Homing velocity” until an active sensor input (default is Input 4) goes high. The motor will stop upon a rising edge detection and reference the position to the parameter “Homing position”.

14: **Sensor homing BiDirectional (BHoming)**, The motor will run in a direction with a velocity determined by the parameter “Homing velocity” until an active sensor input (default is Input 4) goes high, the direction is changed and the motor will slowly back away from the sensor until a falling edge is detected. Here the position will be referenced to the parameter “Homing position”.

For further information on homing, please refer to the main manual of the ServoStep motor.

Please notice that the JVL function blocks for Siemens or the AOI’s for Rockwell will handle homing procedures without requiring that the user has any knowledge about the different modes of the motor.

5.2.1.1.4 Actual position

Units: Counts

Size: 4 octet, value range -2147483647 – 2147483647

Note 1 revolution is 409600 counts.

The motor can either be equipped with an internal encoder that is able to keep track of the shaft position or without any encoder and thereby not ensuring a precise knowledge of the shaft position.

When the motor is equipped with an encoder, this parameter will show the encoder position. In cases where the motor isn’t equipped with an encoder, this parameter will display the position value of the stepper generator, which will **NOT** detect a stall of the shaft of loss of steps during the motion.

This parameter is signed 32bit.

5.2.1.1.5 Actual velocity

Units: 1/100 RPM

Size: 4 octet, value range -30000 – 30000

When the motor is equipped with an encoder, this parameter will hold the encoder velocity, otherwise it will hold the velocity value of the step generator.

5.2.2 Actual torque

Units: 2047 = 100% of requested torque setting.

Size: 4 octet, value range -10000 – 10000

Note: This value only holds a value if the motor is equipped with an internal encoder.

5.2.2.1.1 Follow error

Units: Counts

Size: 4 octet, value range -2147483647 – 2147483647

Note: This value only holds a value if the motor is equipped with an internal encoder.

Holds the deviation between the encoder position and the calculated projectory.

5.2.2.1.2 Hardware inputs IO1-8

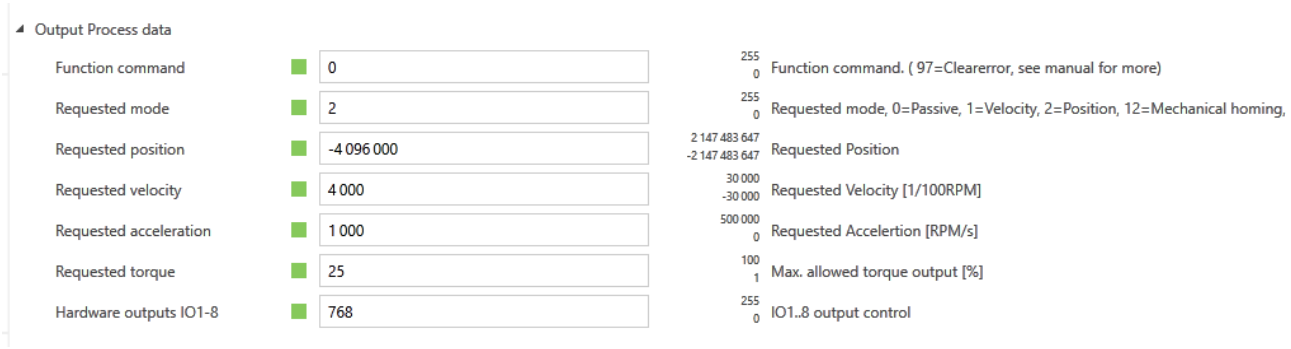
Units: Bitwise

Size: 4 octet, value range 0 – 255

Shows the status of the hardware inputs.

Bit 0-7: Input 1-8

5.2.3 Output process data:



5.2.3.1.1 Function command

Units: None

Size: 1 octet, value range 0 – 255

Activates a command to accomplish a non-motion related tasks.

97: Clear pending error.

124: Reset motor. Last saved parameters are loaded from non-volatile memory.

125: Preset the encoder to the value indicated in the “Homing position. See ISDU parameters”. Also called passive homing. Note the motor must be at standstill and in passive mode.

126: Set factory defaults. “Out of the box” -configuration.

127: Save current parameter and configuration into non-volatile memory.

Some commands will reset the motor, hence the communication is lost for a few seconds. An internal function will make sure that the command value isn’t executed again due to the cyclic behavior, until the command has been set to a different value.

5.2.3.1.2 Requested mode

Units: None

Size: 1 octet, value range 0-255

The motor supports different modes of operation, the most common used by function blocks and AOI are:

0: **Passive**, Motor is passive and shaft is torqueless.

1: **Velocity** (typically used for jogging purposes)

2: **Position**, The position of the motor shaft is controlled, with respect to the parameters Max. Velocity, Acceleration and torque.

12: **Mechanical homing**, The motor will move in a direction determined by the parameter “Homing velocity” up against a hard stop. The hard stop is determined based on the actual torque exceeding the settings in the parameter “Homing torque”.

13: **Sensor homing Unidirectional (UHoming)**, The motor will run in a direction with a velocity determined by the parameter “Homing velocity” until an active sensor input (default is Input 4) goes high. The motor will stop upon a rising edge detection and reference the position to the parameter “Homing position”.

14: **Sensor homing BiDirectional (BHoming)**, The motor will run in a direction with a velocity determined by the parameter “Homing velocity” until an active sensor input (default is Input 4) goes high, the direction is changed and the motor will slowly back away from the sensor until a falling edge is detected. Here the position will be referenced to the ISDU parameter “Homing position”.

See the section Homing Mode66 for further details.

Please notice that the JVL function blocks for Siemens or the AOI’s for Rockwell will handle homing procedures without requiring a deeper knowledge about the different operating modes.

5.2.3.1.3 Requested position

Units: Counts

Size: 4 octet, value range -2147483647 – 2147483647

Note 1 revolution is 409600 counts.

The motor can either be equipped with an internal encoder that is able to keep track of the shaft position or without any encoder and thereby not ensuring a precise knowledge of the shaft position.

5.2.3.1.4 Requested velocity

Units: 1/100 RPM

Size: 4 octet, value range -30000 – 30000

The max. velocity the motor is allowed to travel with during motion.

The direction is controlled in velocity mode by setting a negative number.

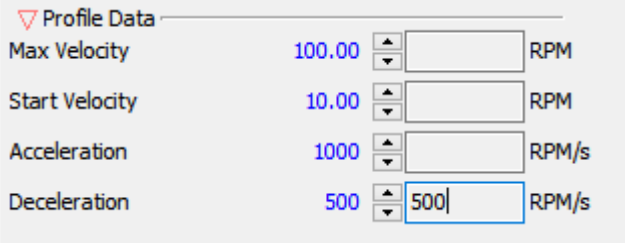
5.2.3.1.5 Requested Acceleration/Deceleration

Units: 10x RPM/s

Size: 4 octet, High 2 Octets sets Deceleration, Low 2 Octets sets Acceleration.

In order to maximize the data usage, both the deceleration and the acceleration is set.

These settings are available from MacTalk:



From the IO-Link interface the setting would be:

Acceleration = 100

Deceleration = 50

So, High 2 Octets = 0x32

And the lower 2 octets 0x64.

Combined the value is 0x00320064.

Note: The requested torque parameter might influence the acceleration ramp. A too low allowed max. torque will slow the acceleration.

5.2.3.1.6 Requested torque

Units: %

Size: 4 octet, value range 1 – 100

Max. allowed torque applied to the shaft.

For motors equipped with an encoder and closed loop enabled (default), the motor will always adjust the torque to the required amount.

For motors without encoder, this setting will be the torque applied to the shaft during motion.

5.2.3.1.7 Hardware outputs IO1-8

Units: Bitwise

Size: 4 octet, value range 0-255

Control of IO's when they are set for outputs.

Bit0-7 = Output 1-8

5.3 Homing Mode66

Basically 3 different homing methods are supported:

1. Homing using a sensor, connected to the motor
2. Homing using a mechanical hard stop
3. Passive homing only setting the encoder position reference.

Common for all the homing methods the position configured in the parameter “Homing position” is used as reference at the detection point.

The direction and velocity are controlled from the “Homing velocity” -parameter:

The direction is controlled by changing the sign. Positive value shaft rotation is CW, negative value shaft rotation is CCW.

To start homing the motor, set the requested mode to one of the following values:

12 = Mechanical homing

13 = Uni-directional sensor homing

14 = Bi directional sensor homing

The motor will immediately start the homing procedure. In case a sensor or hard stop isn’t detected within 60s a timeout error is flagged.

Common for the sensor homing methods, is that Input 4 is by default the input for the homing sensor. The homing sensor input is configured in the ISDU parameter “Homing sensor selection”. Any of IO1-8 can be selected.

Bit0 = Input1

Note, the input on which the sensor is connected must be configured to input.

When the homing procedure is completed, bit 3 goes high in the status octet of the procesdata.

Observe that when the homing procedure is completed, the motor returns to passive mode.

Note that passive homing is supported as well, but this is done by applying a command in the command octet.

Please see Function command for further details.

5.4 Position control

For position control, the requested mode must be set = 2.

The motor is energized, and the following motion specific parameters are used:

Requested position	Requested position	Counts
Requested Velocity	Requested shaft velocity	1/100 RPM
Requested acceleration	Requested acceleration	1/100 RPM/s
Requested Torque	Requested Max. torque	1-100%

Example 1:

The motor is commanded into “Position mode”, set processdata “requested mode” = 2.

The motor is commanded to the position 4096000 (10x revolutions) running 1200 RPM with 10000 in acceleration and 50% Max. torque.

Requested position	4096000	Counts
Requested Velocity	120000	1/100 RPM
Requested acceleration	10000	1/100 RPM/s
Requested Torque	50	1-100%

While the motor is in motion, we will observe from the process data “Status bits” Status bits, that

1. The Acceleration bit is set, during acceleration
2. The AtVelocity bit is set, when (if) the motor reaches the requested velocity, in case the acceleration doesn’t allow for the motor to ramp up to the requested velocity, this bit is never set.
3. The deceleration bit is set, while decelerating for the standstill.
4. The InPosition -bit goes high, when the motor is within the InPosition window of the requested target position.

5.5 Velocity mode

For position control, the “requested mode” must be set = 1.

The requested velocity is controlled in the Process data “Requested Velocity”.

The direction of rotation is controlled by changing the sign of the value.

A positive value sets the direction of rotation CW and a negative value sets the direction CCW.

Example 2

Running 1200RPM, acceleration = 10000, Max. torque is set to 50%.

Requested Velocity (V_SOLL)	120000	1/100 RPM
Requested acceleration	10000	1/100 RPM/s
Requested Torque	50	1-100%

Example 3

Running -1200 RPM, Acceleration = 10000, Max. Torque is set to 30%

Requested Velocity (V_SOLL)	-120000	1/100 RPM
Requested acceleration	10000	1/100 RPM/s
Requested Torque	30	1-100%

6 IO-Link mode ModeFM

The ModeFM has a very limited processdata size of only 2 octets with additional 2 octets of on demand data. This limits the data but optimizes the bandwidth.

Controlling the motor is done using commands and pre-configured motion vectors and settings.

Status of the motion is returned in one octet of the processdata.

Since the amount of data is very limited, the cycletime can be as low as 1ms. This mode is ideal for applications where the motion profile is simple yet demands a low latency.

6.1 ISDU Parameters

Due to the working nature of the ModeFM, more parameters are available through the ISDU parameter interface.

Up to 4 independent motion vectors are configured though the ISDU interface. The motion is triggered by using a separate command for each vector.

Different homing procedures are also configured through the ISDU interface and the homing process is triggered by a command.

For on “On the fly” -changes of a certain motion parameter, “Requested Velocity”, “Requested acceleration” and “Requested Torque” are always accessible through the ISDU parameters.

Available ISDU parameters covered by datastorage and available in the IO-Link master interface.

NAME	VALUE		RANGE	DESCRIPTION
Requested Velocity	100.00	rpm	3 000 -3 000	
Requested Acceleration	360 000		500 000 1	
Requested Deceleration	2 000		500 000 1	
Requested Torque	1	%	100 1	
Vector 1 Position	4 096 000		2 147 483 647 -2 147 483 647	
Vector 1 Velocity	99.00	rpm	3 000 -3 000	
Vector 1 Acceleration	1 000		500 000 1	
Vector 1 Torque	33	%	100 1	
Vector 2 Position	-4 096 000		2 147 483 647 -2 147 483 647	
Vector 2 Velocity	200.00	rpm	3 000 -3 000	
Vector 2 Acceleration	1 000		500 000 1	
Vector 2 Torque	33	%	100 1	
Vector 3 Position	0		2 147 483 647 -2 147 483 647	
Vector 3 Velocity	1 000.00	rpm	3 000 -3 000	
Vector 3 Acceleration	1 000		500 000 1	
Vector 3 Torque	33	%	100 1	
Vector 4 Position	0		2 147 483 647 -2 147 483 647	
Vector 4 Velocity	100.00	rpm	3 000 -3 000	
Vector 4 Acceleration	1 000		500 000 1	
Vector 4 Torque	33	%	100 1	
Homing method	3 328		3 584 0	Homing method, Mechanical=3073, Sensor Uni.Dir= 3328, Sensor Bi.Dir=3584
Homing Position	-100 000		2 147 483 647 -2 147 483 647	
Homing Velocity	-50.00	rpm	3 000 -3 000	
Homing Torque	50.0	%	73.2779685 0.048851979	Torque thresshold for mechanical homing.
Homing sensor selection	0000 1000b		128 1	Homing sensor input selection IO1..8 bit[0..7]
Max. allowed follow err	0		2 147 483 647 -2 147 483 647	
Bottom Position limit	0		2 147 483 647 -2 147 483 647	Bottom Position limit or Modulus lower working range -position.
Top Position limit	0		2 147 483 647 -2 147 483 647	Top Position limit or Modulus top working range -position.
Error deceleration [RPM/s]	10 000		500 000 1	
IO Configuration	0000 1000 1111 1111b		65 535 0	IO1..8 Configuration [bit0..7]: 1=active high, [bit8..15]: 0=Input/Output]
Brake output selction	0000 0000b		255 0	Brake output selction IO1..8, use multiple for higher current.
Min. Torque applied	0.0	%	100.000001013 0	
Modulus operation setup	0		5 0	[0=disabled, 1=Singletun CW, 2=Singleturn CCW, 3=Multiturn CW, 4=Multiturn CCW]
Errorcode	0		2 147 483 647 0	
Warningcode	0		2 147 483 647 0	
P+ Supply voltage	24.7	V	5.6952E+7 0	
Temperature	38.8	°C	2 147 483.647 0	

6.1.1 Requested velocity

[Units: 1/100RPM]

This parameter is always respected regardless of which mode the motor is currently utilizing. This parameter is also set using the motion vector.

6.1.2 Requested Acceleration

[Units: RPM/S]

This parameter is always respected regardless of which mode the motor is currently utilizing. This parameter is also set using the motion vector.

6.1.3 Requested Deceleration

[Units: RPM/S]

This parameter is always respected regardless of which mode the motor is currently utilizing.

Please note, that this parameter isn't included in the motion vectors so this setting is used across the activation of the motion vectors.

6.1.4 Requested Torque

[Units: %]

This parameter is always respected regardless of which mode the motor is currently utilizing.

6.1.5 Motion vectors 1-4 (Position, Velocity, Acceleration, Torque)

These parameters holds the motion specific data, that is activated using the commands explained in section Position control

6.1.6 Homing method

[Units: None]

4 different homing procedures can be used to reference the motor.

Mechanical, Sensor Uni-directional, Sensor based Bi-Directional and passive homing.

See section Homing ModeFM for further details.

6.1.7 Homing Position

[Units: Counts]

When a homing sequence is completed, the reference position is defined by this parameter.

For passive homing, where the current encoder position needs a new reference, this parameter is used.

6.1.8 Homing velocity

[Units: 1/100RPM]

The velocity used during the homing procedure. Note that the direction is determined from the sign.

6.1.9 Homing Torque

[Units: %]

The torque threshold when mechanical homing is performed.

6.1.10 Homing sensor input selection

Using a bitwise value, the input for the homing sensor is selected.

Exc: 0b1000 will select input 4 as homing input. Input 4 is the factory default value.

6.1.11 Max. allowed follow error

[Units: Counts]

Maximum allowed follow error before an error is flagged.

0 = Disabled (Default).

6.1.12 Bottom position limit

[Units: Counts]

This parameter has 2 different functions depending on the current Modulus mode.

In case the modulus mode is disabled, the parameter defines the Min. position value within an allowed range. In case the motor moves beyond this position, an error is flagged and the motor is only allowed to move in the opposite direction.

In case the current modulus mode is enabled, this parameter indicates the min. Turntable size.

6.1.13 Top position limit

[Units: Counts]

This parameter has 2 different functions depending on the current Modulus mode.

In case the modulus mode is disabled, the parameter defines the Max. position value within an allowed range. In case the motor moves beyond this position, an error is flagged, and the motor is only allowed to move in the opposite direction.

In case the current modulus mode is enabled, this parameter indicates the max. Turntable size.

6.1.14 Error deceleration

[Units: RPM/S]

Deceleration used in case an error is flagged.

6.1.15 IO1..8 Configuration

[Units: Bitwise]

Bitwise configuration of the IO's available. These IO's can function both as input or output, depending on the configuration. The active level is also configured within this parameter.

6.1.16 Brake Output selection

[Units: Bitwise]

In case the motor is equipped with an external brake, this brake can be controlled by the motor using up to 8 outputs. The parameter is configured by binary enabling the outputs.

The value 0b1111 will enable IO1-4 for brake operation. Remember that the IO1..4 Configuration needs to be configured for outputs as well.

6.1.17 Min. Torque applied

[Units: %]

When the motor is equipped with an internal encoder, the motor can utilize the advanced closed loop current control. This parameter sets the absolute min. current (Torque) used at any time.

The closed loop current control will always limit the current to the lowest possible value in order to control the motion.

6.1.18 Modulus operation setup

[Units: Value]

For modulus operation (Turntable) this parameter is used to define the usage.

Value	Function
0	Disabled
1	Singleturn CW rotation
2	Singleturn CCW rotation
3	Shortest path
4	Multiturn CW rotation
5	Multiturn CCW rotation

6.1.19 Error code

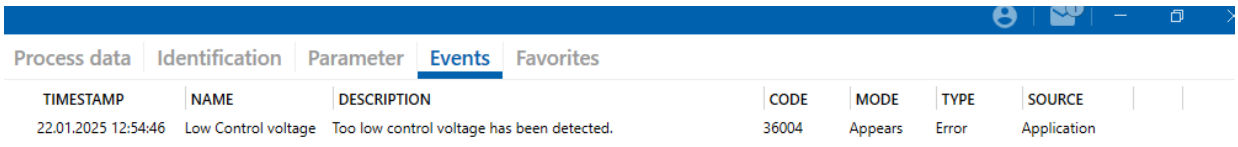
[Units: Bitwise]

The following bits are encoded in the error code:

Bit 0: General error bit, always set with another bit.

Bit	Error description
0	General error bit, always set with another bit
1	Follow error
2	Output driver
3	Position limit
4	Low bus voltage (Default: set when bus voltage goes below 15 V))
5	Over voltage
6	Temperature > 90°C
7	Internal, Self diagnostic detected an internal error
8	Absolute multiturn encoder lost position
9	Absolute multiturn encoder sensor counting error
10	Absolute multiturn encoder communication lost
11	SSI Encoder counting error
12	Closed-loop error
13	External memory error
14	Absolute singleturn encoder error
15	H4 Internal encoder error
16	Zero search timeout (sensor or mechanical torque threshold not detected within time)
17	CVI control voltage unstable
18	Motor driver overload
27	STO Alarm (Safe Torque Off)
29	STO

In case of an error, an event is raised with a timestamp. All events are handled by the IO-Link master according to the IO-Link standard.



The event codes and descriptions are described in the IODD files.

6.1.20 Warning code

[Units: Bitwise]

Warning codes are also available encoded in hexadecimal value as follows:

Bit	Error description
0	Positive position limit active (only motion in opposite direction possible)
1	Negative position limit active (only motion in opposite direction possible)
2	Positive limit has been active
3	Negative limit has been active
4	Low bus voltage
5	Reserved
6	Temperature > 80°C
7	SSI Encoder
8	Driver overload
9	STO active (Safe Torque Off)

6.1.21 P+ supply voltage

The current measured bus voltage.

[Units: V]

Note!

For MIS17x a class B supply from the IO-Link master to the motor can supply both the control and the bus voltage for the driver. The max. voltage in this case will be limited by the IO-Link master.

For all other motors a separate supply is needed for the motor and the P+ voltage will be limited by the specifications for the actual motor.

6.1.22 Temperature

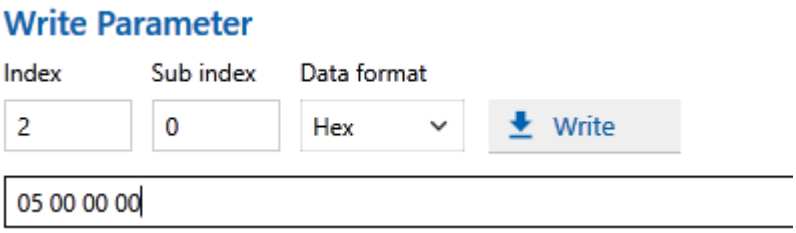
[Units: °C]

The current internal measured temperature. A warning is issued if the temperature rises above **80°C**, the motor is faulted if the temperature exceeds **90°C.**

6.1.23 Save parameters in non-volatile memory

The parameters are saved in non-volatile memory by applying the systemcommand “ParamDownLoadstore” index=2, value=5:

or use send the systemcommand “ParamDownloadStore” Index = 2, value = 5.



Note the order of bytes.

The connection is temporarily lost, but regained after a few seconds.

6.2 Processdata ModeFM

In the ModeFM the motor is controlled by issuing commands to activate and use up to 4 different motion vectors.

Using this method, the motor can be controlled for precise position control or for controlling the velocity of the shaft in basic velocity control mode.

Homing procedures are also controlled using commands and the homing configuration is controlled through ISDU parameters.

One of the Process data -octets is used as a command byte, where bit 7 is used as a trigger bit. Bit 0-6 is reserved for the command and the command is executed upon a rising edge on bit 7.

Output data:

Name	Description	Unit
Command	Execute command defined in bit[0..6] upon rising edge of bit 7	-
IO1-8 Output request	IO1-8 Hardware output request	Bitwise

A status octet from the process data is used for monitoring the process.

In the status word, monitoring the status for the motion can be monitored.

When the motion is completed, the **"InPosition" -bit 1** will go high indicating that the motor has reached the requested position.

Name	Description	Unit
Statusbits	Indicates current status of motion, cmd and error	-
IO1-8 Input status	IO1-8 Hardware inputs status	Bitwise

Statusbit bits[0..7] :

0: CMD in progress Bit is set high when a new cmd is initiated and remains high while the command is processed

1: In position or At velocity, Bit is set if mode = position and actual position is within the position window. In case the motor is running velocity mode, this bit is set when it has reached the requested velocity.

2: Energized, Bit=1 means that the motor is energized and ready.

3: Homing done, Motor has performed a homing procedure. Note that this bit is saved in non-volatile memory, so in case the motor is equipped with an absolute encoder, the homing procedure isn't required after a power cycle.

4: In Motion Bit is set while the shaft is in motion, that is velocity != 0.0 RPM

5: Optional, not used.

6: Warning is active, A warning is active one of the following warning incidents can set the Active warning bit:

- Low bus voltage, P+ busvoltage has been detected to be below 15V (default value)
- IO Driver overload, The output driver for IO1-4 has been overloaded.
- The temperature is detected to be > 85°C
- The Motor driver circuitry has been overloaded.
- STO (Safe Torque Off if equipped) -state has been detected.

7: Error present, In case of any errors, this bit is set. Note that the active mode of the motor will be forced to passive.

In case of either a the warning or the error bit goes high, further details can be found from the ISDU parameters “Errorcode” and the “Warningcode”.

Errorcode	ro	0	
Warningcode	ro	0	
P+ Supply voltage	ro	21.0	V
Temperature	ro	32.1	°C

6.3 Homing ModeFM

Basically 3 different homing methods are supported:

1. Homing using a sensor, connected to the motor
2. Homing using a mechanical endstop
3. Passive homing only setting the encoder reference.

The method is configured in the ISDU parameter “Homing method” using the following values:

0=Disabled, active homing is not possible, only passive homing.

3072 = Mechanical homing

3328 = Sensor Uni-Directional

3584 = Sensor Bi-Directional

Common for all the homing methods the position configured in the parameter “Homing position” is used as reference at the detection point.

The direction and velocity is controlled from the “Homing velocity” -parameter:

Note!

The Requested Acceleration parameter must be set to a value > 0

The direction is controlled by changing the sign. Positive value shaft rotation is CW, negative value shaft rotation is CCW.

To start the homing process, the command “Start homing” is configured in the CMD -octet of the cyclic data and initiated by a rising edge on bit 7.

Triggerbit and command bits[0..7] :

Bits 0..6: Value = 0x70 (112d) “Start homing”

Bit 7: Upon rising edge on this bit, the command configured in bits 0-6 is executed.

Note!

The motor must be in passive mode and at complete standstill before this command can be executed.

For passive homing the procedure is different and a separate command is used.

When the active homing procedure is completed, **bit3 "Homing done"** is set in the status octet, see section MODEFM.

For motors equipped with an absolute multiturn encoder, the homing state is saved in non-volatile memory, so bit3 in the status octet will indicate if the motor has been homed in its lifetime.

When a new homing procedure is executed, this bit goes low until the homing procedure is completed.

For passive homing without any motion, the ISDU parameter "Homing position" is used to preset the encoder position.

The command **125** sets the position reference when the triggerbit goes high.

6.3.1 Mechanical Homing

Homing Method = 3072

The motor detects an applied torque exceeding the threshold configured in parameter "Homing torque". The motor will stop immediately and set the position.

Note!

This homing method is only available in MIS motors equipped with an internal encoder.

The parameter "Homing torque threshold" will be used to detect the mechanical endstop position.

Homing Torque (for Mechanical homing)	rw	50.0	%

6.3.2 Sensor based homing Uni-Directional

Homing Method = 3328

The motor will home against a sensor mounted to one of the inputs on the motor. By default Input 4 is used for homing.

Using the Uni-directional homing method, the motor stops upon rising edge of the sensor signal and sets the position.

The parameter "Homing sensor input selection" determines which input that should be used for the homing sensor.

Homing sensor input selection IO1..4 [Bit 0..3]	rw	0b00000000000000000000000000001000	

The parameter is set bitwise. By default bit 3 is set, indicating Input 4 is used.

6.3.3 Sensor based homing Bi-Directional

Homing Method = 3584

The motor will home against a sensor mounted to one of the inputs on the motor. By default Input 4 is used for homing.

Using the Bi-directional homing method, the motor changes direction upon rising edge of the sensor signal and on the falling edge the motor will stop and set the position.

The parameter “Homing sensor input selection” determines which input that should be used for the homing sensor.

Homing sensor input selection IO1..4 [Bit 0..3]	rw	0b00000000000000000000000000001000

The parameter is set bitwise. By default bit 3 is set, indicating Input 4 is used.

6.4 Position control

Using the 4 motion vectors, the motors is able to move between the positions configured with the velocity and acceleration configured.

To energize the motor, set it into Positioning mode, issue the command **95d, 0x5F**. The motor will apply holding torque to the shaft.

From the ISDU parameters the vectors 1-4 can be configured:

Vector 1 Position	4 096 000		2 147 483 647 -2 147 483 647
Vector 1 Velocity	100.00	rpm	3 000 -3 000
Vector 1 Acceleration	1 000		500 000 1
Vector 1 Torque	20	%	100 1
Vector 2 Position	-4 096 000		2 147 483 647 -2 147 483 647
Vector 2 Velocity	250.00	rpm	3 000 -3 000
Vector 2 Acceleration	1 000		500 000 1
Vector 2 Torque	24	%	100 1
Vector 3 Position	0		2 147 483 647 -2 147 483 647
Vector 3 Velocity	10.00	rpm	3 000 -3 000
Vector 3 Acceleration	1 000		500 000 1
Vector 3 Torque	36	%	100 1
Vector 4 Position	15 000		2 147 483 647 -2 147 483 647
Vector 4 Velocity	900.00	rpm	3 000 -3 000
Vector 4 Acceleration	1 000		500 000 1
Vector 4 Torque	90	%	100 1

In the above example the following is configured:

Vector 1:

Position = 4096000 (10x motor revolutions)
Velocity = 100.00 RPM
Acceleration = 1000
Torque = 20%

Vector 2:

Position = -4096000
Velocity = 250.00 RPM (meaning it will not change the velocity, but use the current settins)
Acceleration = 1000
Torque = 24%

Vector 3:

Position = 0
Velocity = 10.00 RPM
Acceleration = 1000
Torque = 36%

Vector 4:

Position = 15 000
Velocity = 900.00 RPM
Acceleration = 1000
Torque = 90%

Each vector is activated using one of the vector executions commands.

Command	Description
95d, 0x5F	Motor is energized and Position mode is activated. Note 1
97d 0x61	Reset error (Note critical errors may only be reset upon powercycle)
108d, 0x6C	Execute position motion using vector 1
109d, 0x6D	Execute position motion using vector 2
110d, 0x6E	Execute position motion using vector 3
111d, 0x6F	Execute position motion using vector 4
112d, 0x70	Start homing

Upon execution of a command, the motor will start the motion using the parameters from the vector.

In case another command is executed prior to the motor has reached the destination, the new commanded vector will be prioritized and executed.

Note!

Prior a vector execution, the command 95d must be executed. This will set the motor into an active mode.

6.4.1 Example 1:

Activate Vector 1 and wait for the motor to reach the requested position.

Triggerbit and command bits[0..7] :

Bits 0..6: Value = 0x6C (108d) "Activate Vector 1"

Bit 7: Upon rising edge on this bit, the command configured in bits 0-6 is executed.

Triggerbit and command bits[0..7] = 0xEC (236 with bit7 = 1)

Monitor the status octet bit 1 for a ring edge indicating that the motor has reached the requested position.

The in position bit goes low as soon as the motor start the motion. The bit is set, when the motor reaches the requested position within the "In position window". By default this window is 20000 counts.
In case the requested position is within the window, due to the nature of the cycletime the In position bit may never be detected low. In this case assume the positioning as finished when the "CMD in progress "-bit goes low.

6.4.2 Example 2

Activate Vector 2 and await for the motor to reach the destination.

Output.Data[0] = 128 (bit 7) + 109 (activate vector 2) = 237

Triggerbit and command bits[0..7] :

Bits 0..6: Value = 0x6D (109d) "Activate Vector 2"

Bit 7: Upon rising edge on this bit, the command configured in bits 0-6 is executed.

Triggerbit and command bits[0..7] = 0xED (237 with bit7 = 1)

Monitor the status octet bit 1 for a ring edge indicating that the motor has reached the requested position.

6.5 Velocity mode

The motor can run in velocity mode which is useful for exc. jogging -purposes. The position information is updated, but the motor will move with the requested velocity until it is stopped.

The 4 different vector values can be used but only the velocity and the acceleration data are relevant.

To activate a velocity profile, one of the following commands is used:

Command	Description
97d 0x61	Reset error (Note critical errors may only be reset upon powercycle)
40d, 0x28	Execute velocity motion using velocity from vector 1
41d, 0x29	Execute velocity motion using velocity from vector 2
42d, 0x2A	Execute velocity motion using velocity from vector 3
43d, 0x2B	Execute velocity motion using velocity from vector 4

The motor is stopped by issuing the command **0x1** which will leave the motor in a passive state with no torque.

Note!

The deceleration is controlled from the ISDU parameter "Requested deceleration" and is not a part of the motion vectors.

6.6 Controlling HW outputs

In the cyclic processdata, the second octet is used for controlling the outputs. 4 outputs are available, configured as either outputs or inputs.

Bit 0..7 will control the output.

For configuration of the Inputs/outputs, the ISDU parameter "IO1..8 Configuration" is used:

IO1..4 Configuration [bit 0..3: active level, bit 8..10: Input/Output]	rw	0b00000000000000000000000011111111	

The lower 8 bits are used to control the active level (active high or active low).

Where 1 = active high 0 = active low

Bits 8..15 will se each IO for either input or output -control.

Where 1 = Input, 0 = Output

So lets assume we want IO2 and IO3 as outputs and IO3 and IO4 as inputs (Input 4 is used as homing sensor input by default) all configured as active high.

Then we would write the following value to the ISDU parameter:

IO1..4 Configuration [bit 0..3: active level, bit 8..10: Input/Output]	rw	0b00000000000000000000011011111111	

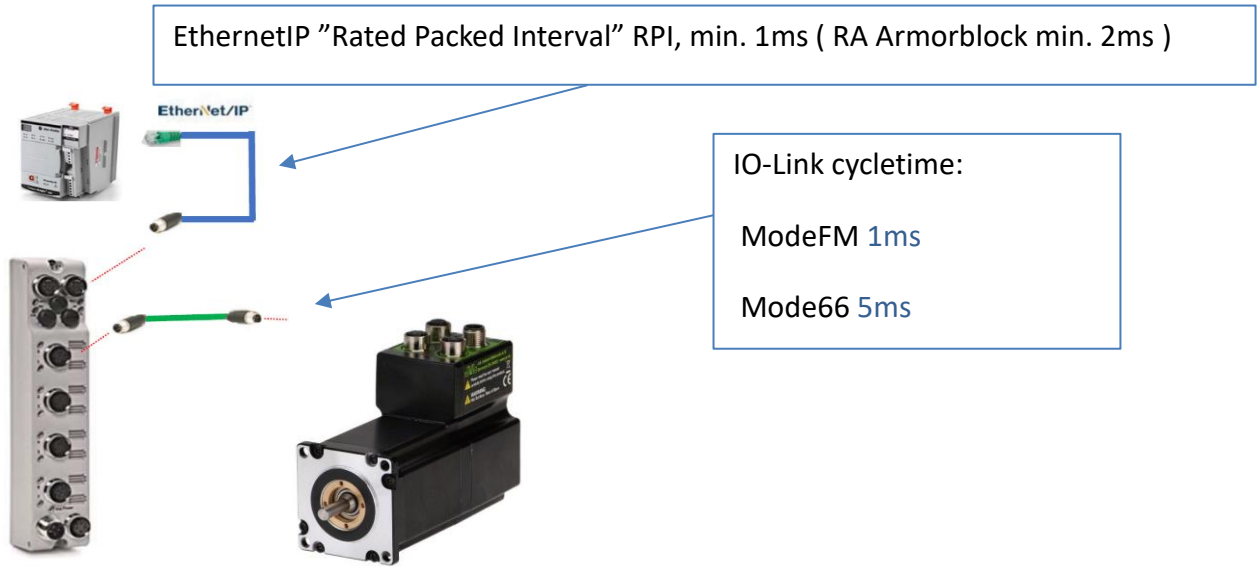
6.7 Detecting HW inputs

Since the IO's can be either input or outputs, setting an output the corresponding input will be activated.

Bit 0..7 indicates the status for the inputs of IO1..8

7 Cyclic timing Rockwell EthernetIP

Picture of scope missing



Real life measured **system cycletime** using the following equipment:

Compactlogix 5380

Armorblock IOL8 IO-Link master, running 2ms RPI

ModeFM: 8-10ms (IO-Link cycletime = 1ms)

Mode66: 25ms (IO-Link cycletime = 5ms)

8 Cyclic timing Siemens Profinet

Real life measured **system cycletime** using the following equipment:

Siemens 1511-1-PN

Balluff BNI PNT-507-005-Z040

ModeFM: 20ms (IO-Link cycletime = 1ms)

Mode66: 30ms (IO-Link cycletime = 5ms)

9 Data Storage

One of the great functions in IO-Link is the Data storage functionality.

Basically, the list of ISDU parameters described in the IODD -file, is saved into the IO-Link master. When a motor is connected that doesn't hold the same parameter configuration, the master will upload the stored values to the motor, all automatically.

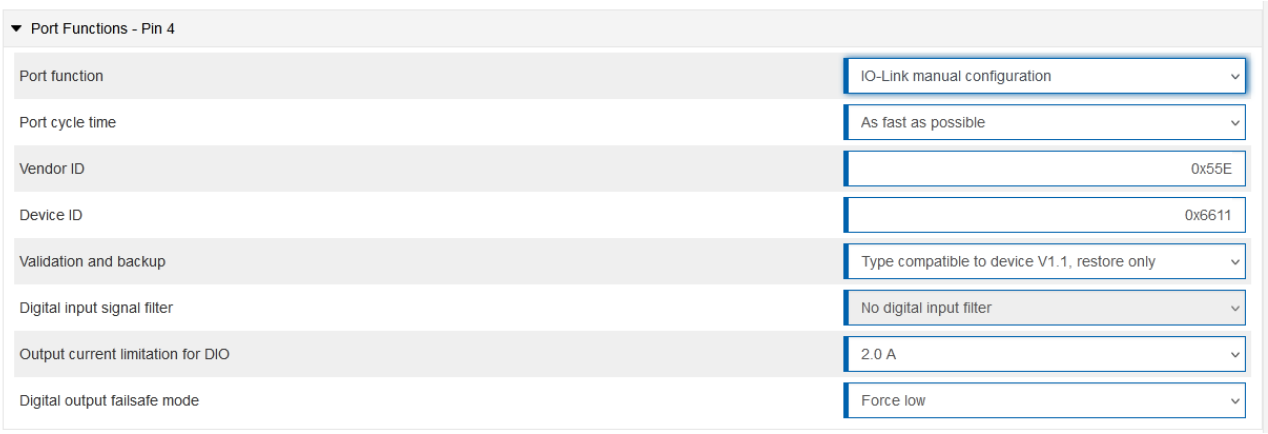
In a replacement situation, a new motor can be unboxed and mounted directly in the machine without any intervention from a highly skilled technical staff or the use of any external software tools and the motor will be configured with the settings stored in the IO-Link master.

When a new master is configured, a file containing the settings for the specific port can be loaded into the Master.

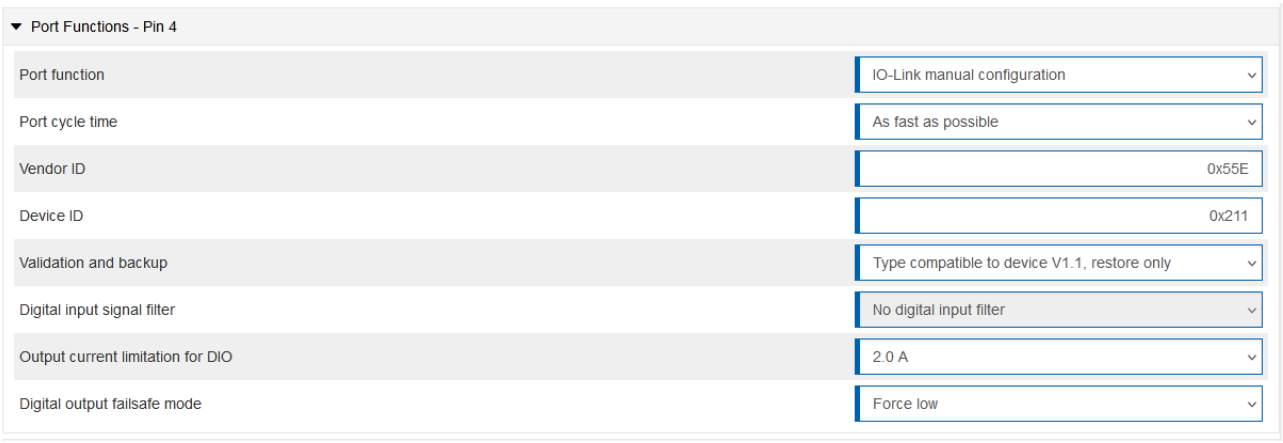
The concept of initial setup of an IO-Link master is beyond the scope of this manual.

To prepare an IO-Link master for Data storage, the port needs to be configured for "Manual configuration" and the Device ID of the motor and the Vendor ID for JVL A/S needs to be configured for the port on which the motor is connected.

For a motor configured to **Mode66** the settings are as follows:



ModeFM, Notice the Device ID.



Follow the instructions for the IO-Link master for the data storage to work properly.

10 Changing the IO-Link mode

The motor is delivered with the Mode66 as the default factory setting.

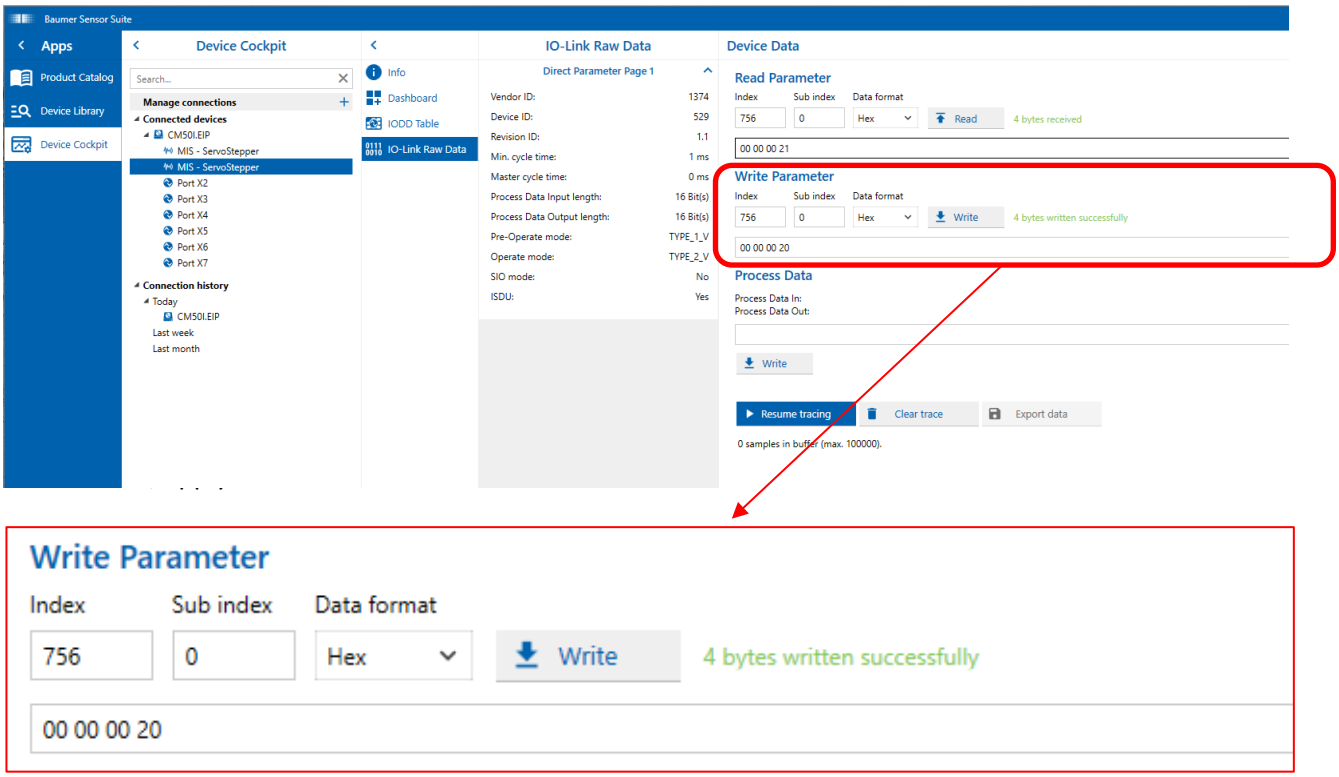
Note!

Please make sure that all IODD files provided from JVL A/S are installed prior to changing the IO-Link mode.

To change the current mode the **index 756, subindex 0** can be configured directly through the IO-Link interface or using MacTalk, by setting **register500**.

Using IO-Link:

Screenshot from “Baumer sensor suite”:



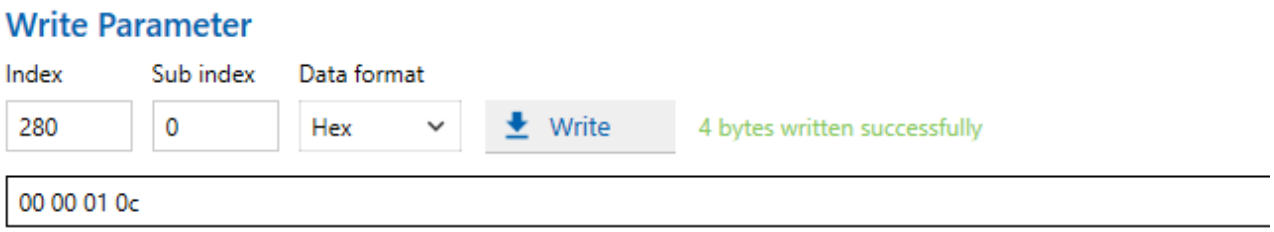
Note the value is 4 octets and the following values applies:

0x 00 00 00 00: Mode66, Default out of the box -mode.

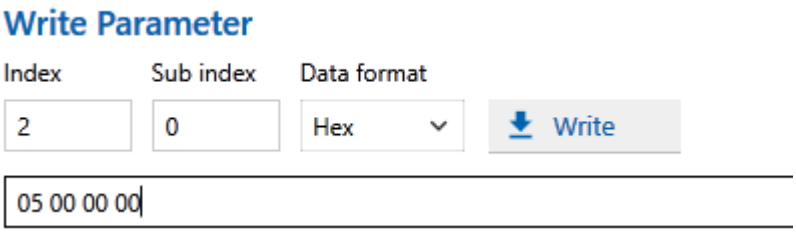
0x 00 00 00 20: ModeFM

Finally, to save the setting permanently in non-volatile memory, apply the

value 268 (Hexadecimal 0x010c) to **index 280, subindex 0**:



or use send the systemcommand “ParamDownloadStore” Index = 2, value = 5.



Note the order of bytes.

NOTE!

The connection will temporarily be lost due to the reset of the motor but regained after a few seconds.

When changing the mode, the DeviceID of the motor will change, so in case the IO-Link is configured to “Manual mode” with either restore or backup/restore, the port will no longer connect to the motor and will require the port to be assigned to the matching IODD file according to the mode configured.

IODD information for the different modes:

Mode66:

IODD Info

http://www.jvl.dk

Vendor ID:	1374
Vendor name:	JVL A/S
Vendor text:	JVL A/S
Device ID:	26129
Device name:	ServoStepper
Device family:	MIS - ServoStep
IODD version:	V1.0
Release date:	09.01.2025
IO-Link revision:	1.1

ModeFM:

IODD Info

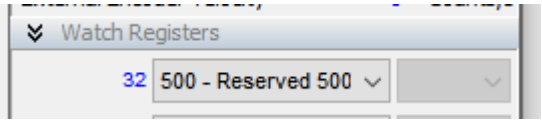
http://www.jvl.dk

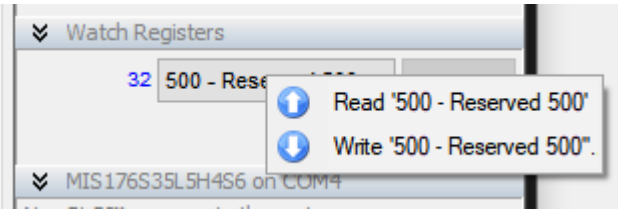
Vendor ID:	1374
Vendor name:	JVL A/S
Vendor text:	JVL A/S
Device ID:	529
Device name:	ServoStepper
Device family:	MIS - ServoStep
IODD version:	V1.0
Release date:	15.01.2025
IO-Link revision:	1.1

Mode change Using MacTalk:

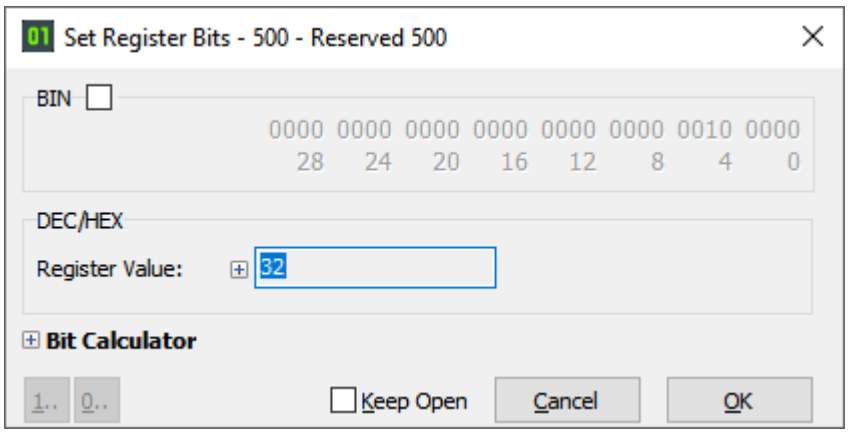
Find Register 500 in the “Watch” -list.

Right click and select ”Write ‘500 – Reserved 500”



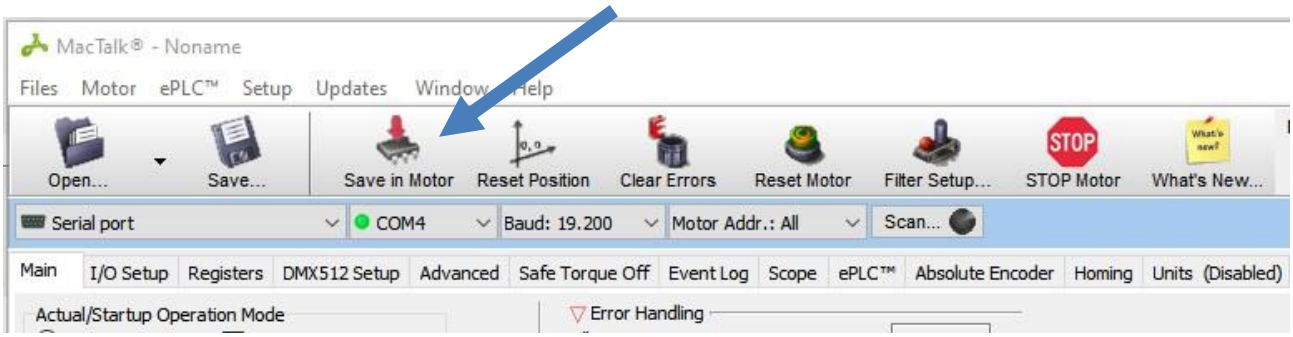


Enter the value “32” and press ok:



Now the mode has been configured, but the parameters need to be saved in non-volatile memory and is used upon reset.

Press the “Save in motor” -button to save the selection permanently in the non-volatile memory.



All the internal motor registers can be accessed through the ISDU parameters.

The ISDU index has an offset of 0x100 (256d) so accessing register 10 (actual position) requires the ISDU index to be set to 266.

A complete list of registers can be found in the MIS “ServoStep” - manual:

https://www.jvl.dk/files/pdf-1/user%20manuals/lb0058gb.pdf