Date: 2024-10-03	Author: HC		Name: User manual
Version: 0.2	Reviwer: -	Approver: -	IO-Link for MIS User manual Preliminary

IO-LINK for MIS

Table of contents

1 History of document 1
2 IOLINK specification 2
3 Data structure 3
3.1 ISDU parameters 4
4 Commissioning 13
4.1 Power and IO-Link connection 13
4.2 Installation of the IODD file 14
4.3 Status indicator 17
5 Controlling a motor using ModeFM 18
5.1 Homing 20
5.2 Position control 22
5.3 Velocity mode 24
5.4 Modulus mode 24
5.5 Controlling HW outputs 24
5.6 Detecting HW inputs 25
6 Controlling a motor Mode66 26
6.2 Position control 29
6.3 Velocity 30
6.4 Output control 31
6.5 Input control 31

1 History of document

Version	Author	Review status	Date
0.1 (draft/released)	HC	DRAFT	20241003

2 IOLINK specification

2 modes of operation are supported **Mode66** and **ModeFM**. The difference is the way of operation, payload size of the processdata and the cycletime.

ModeFM *(name may change)*

ISDU support	True
Processdata payload size	2 Octets
Min. Cycletime	1ms
PREOPERATE = TYPE_1_2, On Request size	2 Octets
OPERATE = TYPE_2_v, On Request size	2 Octets
Profile characteristic	0x4000(Common profile: Identification and Diagnosis)
SIO Mode support	FALSE
Transmission rate	COM3 (230400 bits/s)
Data storage	True
IO-Link Revision	V1.1

Mode66

ISDU support	True
Processdata payload size	24 Octets
Min. Cycletime	5ms
PREOPERATE = TYPE_1_2, On Request size	2 Octets
OPERATE = TYPE_2_v, On Request size	24 Octets
Profile characteristic	0x4000(Common profile: Identification and Diagnosis)
SIO Mode support	FALSE
Transmission rate	COM3 (230400 bits/s)
Data storage	True
IO-Link Revision	V1.1

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 are supported

The JVL IOLINK implementation is currently under development. Some standard IOLINK functions might not be currently available.

3 Data structure

To satisfy the demand for both fast reacting IOLINK 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 IOLINK data models.

One taking advantage of the integrated Motion Vector system “FastMac” which basically applies 3 motion settings in one command. The benefit 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 prior to the motion. A few bits in a status 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 ideel 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 – 24Octet of cyclic data, 6x registers each way.

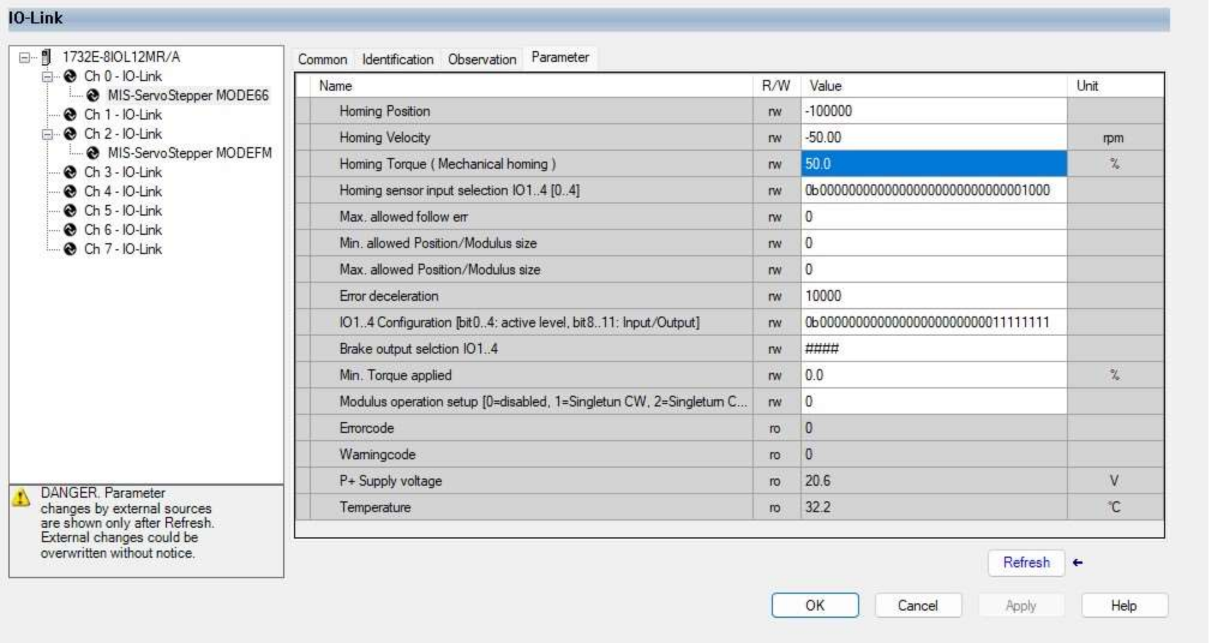
3.1 ISDU parameters

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

The parameters available are different from ModeFM to Mode66 due to the different way of usage between the modes.

3.1.1 ISDU Mode66

The following parameters are accessible:



DANGER. Parameter changes by external sources are shown only after Refresh. External changes could be overwritten without notice.

Name	R/W	Value	Unit
Homing Position	rw	-100000	
Homing Velocity	rw	-50.00	rpm
Homing Torque (Mechanical homing)	rw	50.0	%
Homing sensor input selection IO1..4 [0..4]	rw	0b00000000000000000000000000001000	
Max. allowed follow err	rw	0	
Min. allowed Position/Modulus size	rw	0	
Max. allowed Position/Modulus size	rw	0	
Error deceleration	rw	10000	
IO1..4 Configuration [bit0..4: active level, bit8..11: Input/Output]	rw	0b00000000000000000000000011111111	
Brake output selction IO1..4	rw	####	
Min. Torque applied	rw	0.0	%
Modulus operation setup [0=disabled, 1=Singletun CW, 2=Singletum C...	rw	0	
Errorcode	ro	0	
Warningcode	ro	0	
P+ Supply voltage	ro	20.6	V
Temperature	ro	32.2	°C

Parameter name	Description
Homing Position	Referenced position after homing procedure is completed
Homing Velocity	Velocity used during homing procedure
Homing Torque	Threshold on which the mechanical homing is completed
Homing sensor input selection	Input for sensor type homing (Default is Input4)
Max. allowed follow err.	Max. allowed follor error before fault is flagged. 0=disabled
Min. allowed Position/Modulus size	Used for either position window or modulus operation
Max. allowed Position/Modulus size	Used for either position window or modulus operation
Error deceleration	Deceleration used in case of a fault has been detected
IO1..4 Configuration	Configuration of IO1..4 as input or output individually
Brake Output selection	Bitwise value selecting which output(s) to use as brake -output(s)
Min. Torque applied	Closed loop current control, Min. Torque applied in an active mode.
Modulus operation setup	Configuration of modulus operation

3.1.2 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.

3.1.3 Homing velocity

[Units: RPM]

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

3.1.4 Homing Torque

[Units: %]

The torque threshold when mechanical homing is performed.

3.1.5 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.

3.1.6 Max. allowed follow error

[Units: Counts]

Maximum allowed follow error before an error is flagged.

0 = Disabled (Default).

3.1.7 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.

3.1.8 Max. allowed Position/Modulus size

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.

3.1.9 Error deceleration

[Units: RPM/S]

Deceleration used in case an error is flagged.

3.1.10 IO1..4 Configuration

Bitwise configuration of the 4x 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.

3.1.11 Brake Output selection

In case the motor is equipped with an external brake, this brake can be controlled by the motor using up to 4 IO's. 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.

3.1.12 Min. Torque applied

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.

3.1.13 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.

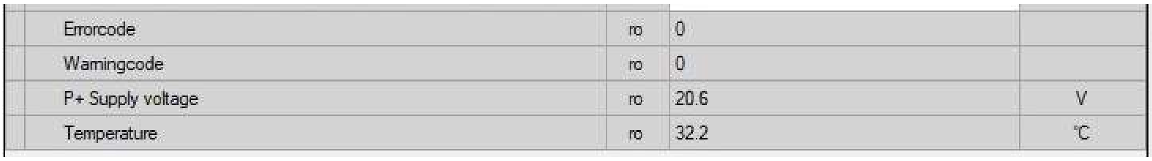
Please refer to the manual for further information about the Modulus (turntable) operation.

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

3.1.14 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 is encoded in the errorcode:

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

Bit	Error decription
0	Generel error bit, always set with another bit
1	Follow error
2	Output driver
3	Position limit
4	Low bus voltage (Default: set when busvoltage 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

Warningcodes are also available encoded in hexadecimal value as follows:

Bit	Error decription
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)

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 rises above 90°C.

3.1.15 ISDU ModeFM

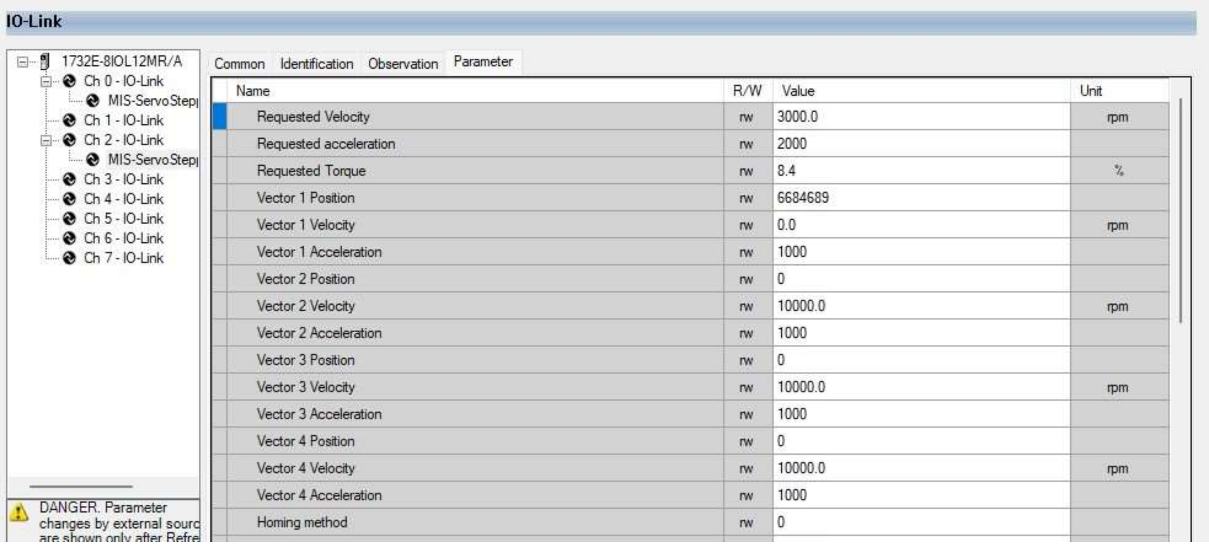
Due to the working nature of the ModeFM, more parameters are available through the ISDU parameter interface.

Up to 4 independent motion vectors can be set and triggered through the cyclic interface.

However the parameters “Requested Velocity”, “Requested acceleration” and “Requested Torque” are always accessible through the ISDU parameters to be changed “on the fly” through an acyclic interface.

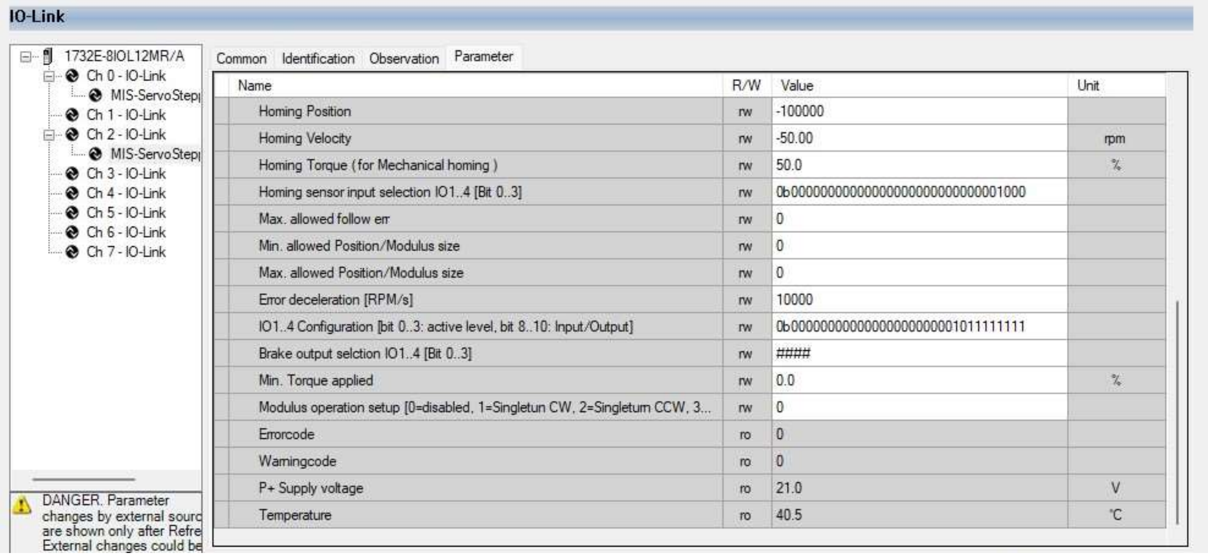
The parameter “Homing method” is also specific for the ModeFM.

The rest of the parameters are identical to the parameters used in Mode66 and the description can found in the section for the Mode66 -parameters.



Name	R/W	Value	Unit
Requested Velocity	rw	3000.0	rpm
Requested acceleration	rw	2000	
Requested Torque	rw	8.4	%
Vector 1 Position	rw	6684689	
Vector 1 Velocity	rw	0.0	rpm
Vector 1 Acceleration	rw	1000	
Vector 2 Position	rw	0	
Vector 2 Velocity	rw	10000.0	rpm
Vector 2 Acceleration	rw	1000	
Vector 3 Position	rw	0	
Vector 3 Velocity	rw	10000.0	rpm
Vector 3 Acceleration	rw	1000	
Vector 4 Position	rw	0	
Vector 4 Velocity	rw	10000.0	rpm
Vector 4 Acceleration	rw	1000	
Homing method	rw	0	

DANGER. Parameter changes by external sourc are shown only after Refre



Name	R/W	Value	Unit
Homing Position	rw	-100000	
Homing Velocity	rw	-50.00	rpm
Homing Torque (for Mechanical homing)	rw	50.0	%
Homing sensor input selection IO1..4 [Bit 0..3]	rw	0b00000000000000000000000000001000	
Max. allowed follow err	rw	0	
Min. allowed Position/Modulus size	rw	0	
Max. allowed Position/Modulus size	rw	0	
Error deceleration [RPM/s]	rw	10000	
IO1..4 Configuration [bit 0..3: active level, bit 8..10: Input/Output]	rw	0b00000000000000000000001011111111	
Brake output selction IO1..4 [Bit 0..3]	rw	####	
Min. Torque applied	rw	0.0	%
Modulus operation setup [0=disabled, 1=Singletun CW, 2=Singleturn CCW, 3...	rw	0	
Errorcode	ro	0	
Warningcode	ro	0	
P+ Supply voltage	ro	21.0	V
Temperature	ro	40.5	°C

DANGER. Parameter changes by external sourc are shown only after Refre External changes could be

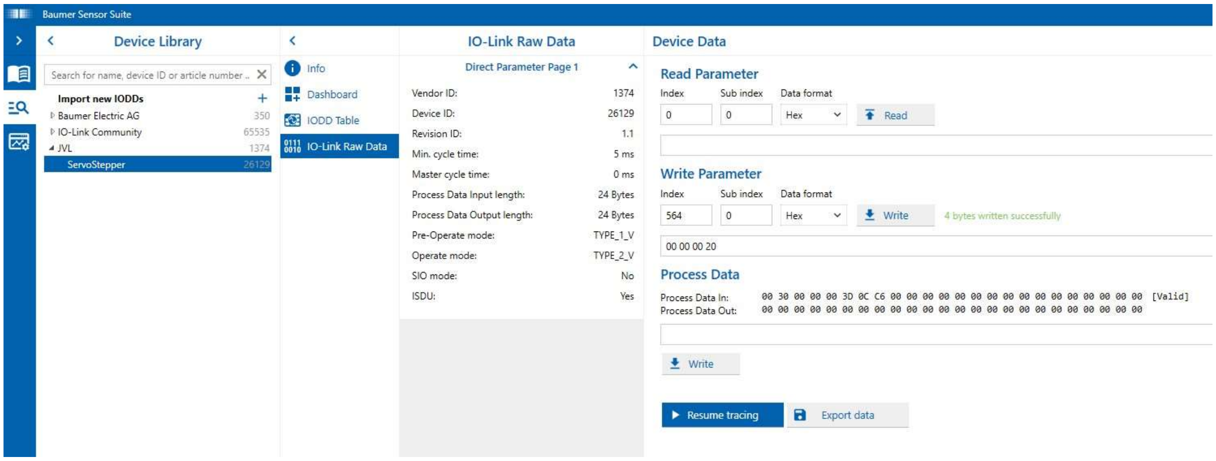
3.1.16 Changing the IO-Link mode

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

To change the current mode the parameter 564 can be configured directly through the IO-Link interface or using MacTalk, by setting the register500.

Using IO-Link:

Screenshot from “Baumer sensor suite”:



Search for name, device ID or article number ..

Import new IODDs
- Baumer Electric AG 350
- IO-Link Community 65535
- JVL 1374
  - ServoStepper 26129

IO-Link Raw Data – Direct Parameter Page 1

Vendor ID:	1374
Device ID:	26129
Revision ID:	1.1
Min. cycle time:	5 ms
Master cycle time:	0 ms
Process Data Input length:	24 Bytes
Process Data Output length:	24 Bytes
Pre-Operate mode:	TYPE_1_V
Operate mode:	TYPE_2_V
SIO mode:	No
ISDU:	Yes

Read Parameter

Index	Sub index	Data format
0	0	Hex

Write Parameter

Index	Sub index	Data format
564	0	Hex

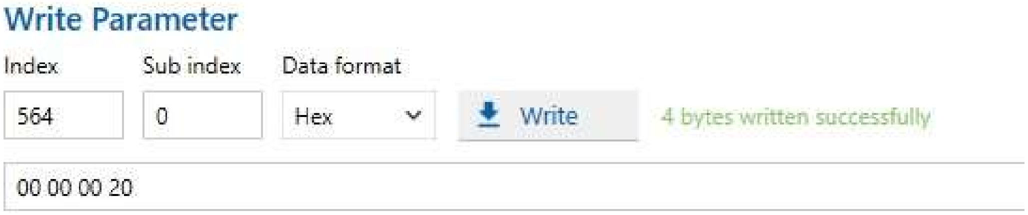
4 bytes written successfully

00 00 00 20

Process Data

```
Process Data In:   00 30 00 00 00 3D 0C C6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  [Valid]
Process Data Out:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Directly writing to parameter 564 with the value 0x20 which is the ModeFM configuration:



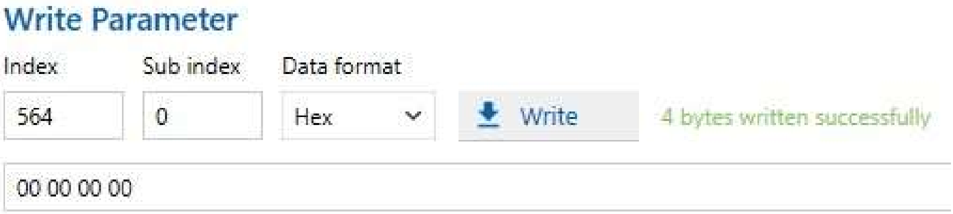
Write Parameter

Index	Sub index	Data format
564	0	Hex

4 bytes written successfully

00 00 00 20

For Mode66, apply the value 0x0:



Write Parameter

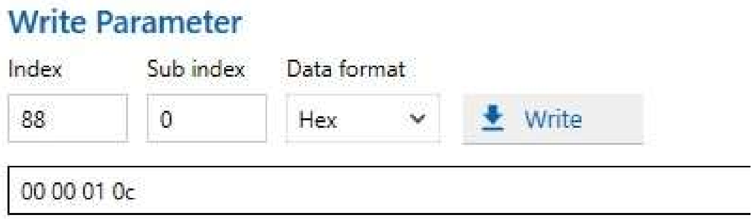
Index	Sub index	Data format
564	0	Hex

4 bytes written successfully

00 00 00 00

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

value 268 (Hexadecimal 0x010c) to index = 88:



Write Parameter

Index: 88

Sub index: 0

Data format: Hex

00 00 01 0c

NOTE!

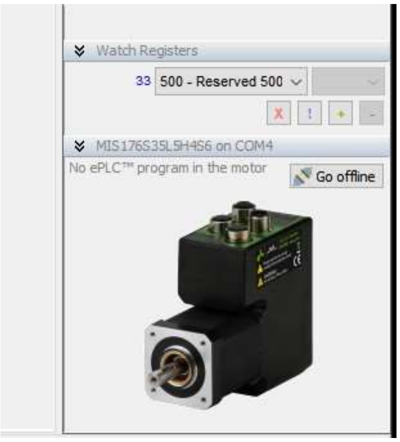
The connection will temporarily be lost due to the reset of the motor but regained after a few seconds.

Device disconnected

Device MIS176S35L5H4S6 disconnected

Using MacTalk:

Find Register 500 in the "Watch" -list.



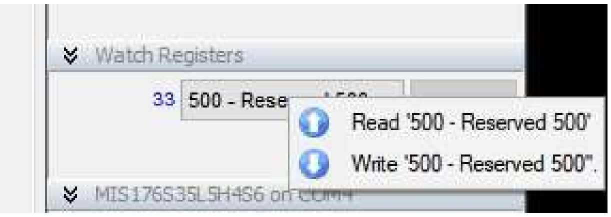
Watch Registers

33 500 - Reserved 500

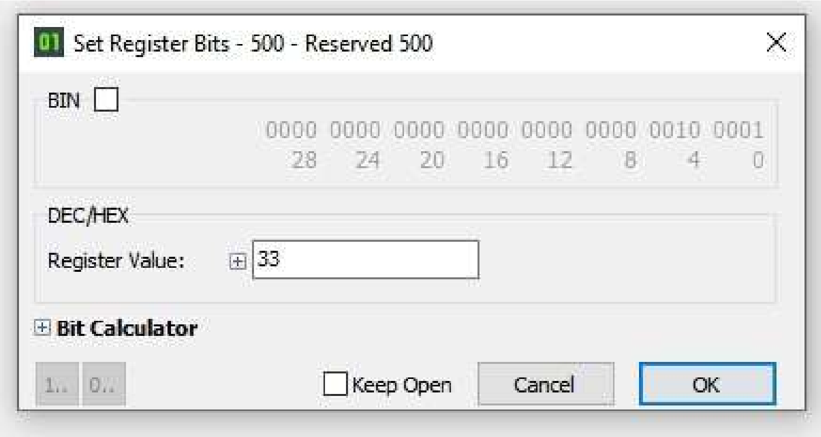
MIS176S35L5H4S6 on COM4

No ePLC™ program in the motor

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



Enter the value “33” and press ok:


Set Register Bits - 500 - Reserved 500

BIN □

0000 0000 0000 0000 0000 0000 0010 0001
28 24 20 16 12 8 4 0

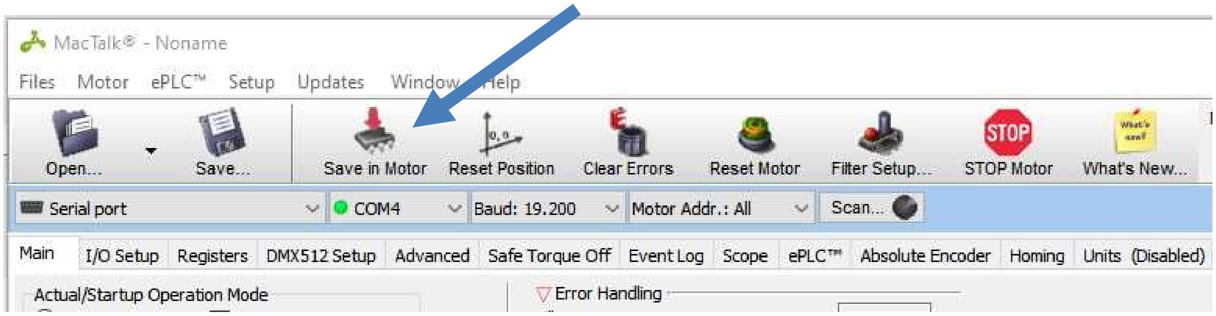
DEC/HEX

Register Value: 33

Bit Calculator

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

Press the “Save in motor” -button.



Serial port | COM4 | Baud: 19.200 | Motor Addr.: All

The motor saves the parameters and performs a reset. The IO-Link connection is lost a few seconds and eventually restored.

4 Commissioning

4.1 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.

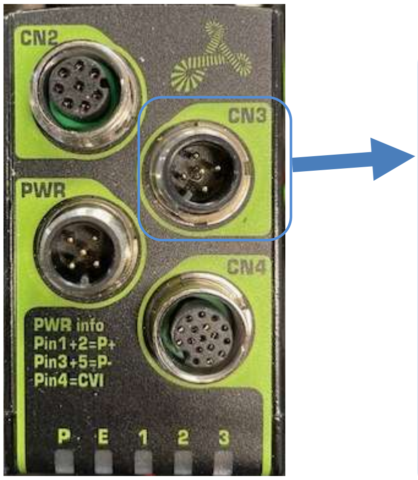
This type is denoted MISxxx **L2** xxxxx, 1x8pin (IO1-4 + RS485) and 1x 5-pin for IO-Link (Class B).

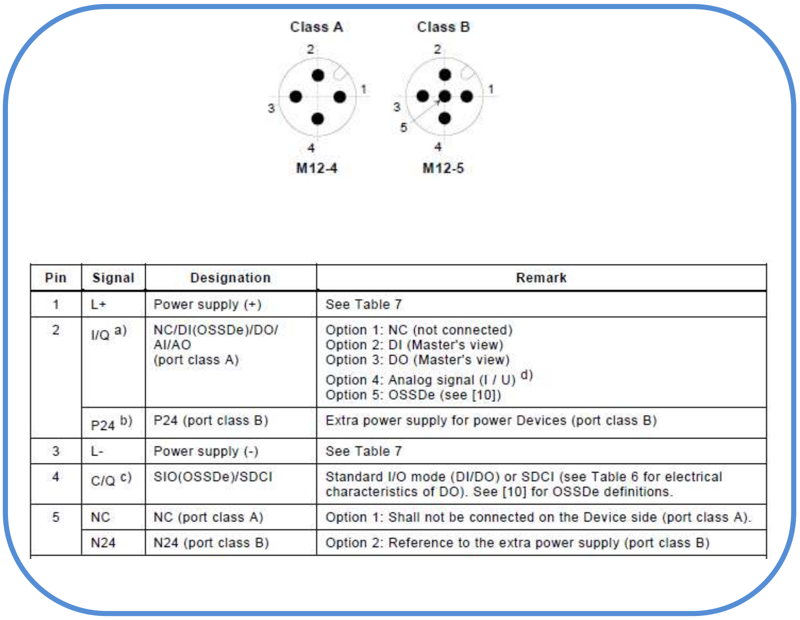
The fully equipped version with the PWR connector is denoted MISxxx **L5** xxxxx.

Caution should be taken, since both the power and the IO-Link connector has 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 done through **CN3 using a 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.

4.2 Installation of the IODD file

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

The examples used are taken from the Rockwell Logix environment and from the Baumer “Sensor suite” however the procedures are the same for other vendors.

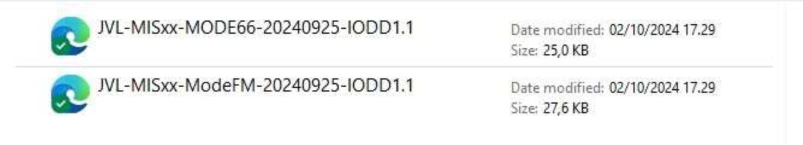
The Rockwell Armorblock IOLINK master and the Baumer EthernetIP EIP50 is used.

There might be differences in how the IODD files are imported and handled between the different IO-Link master vendors.

Steps for importing the IODD file and adding a JVL motor to the IOLINK master.

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

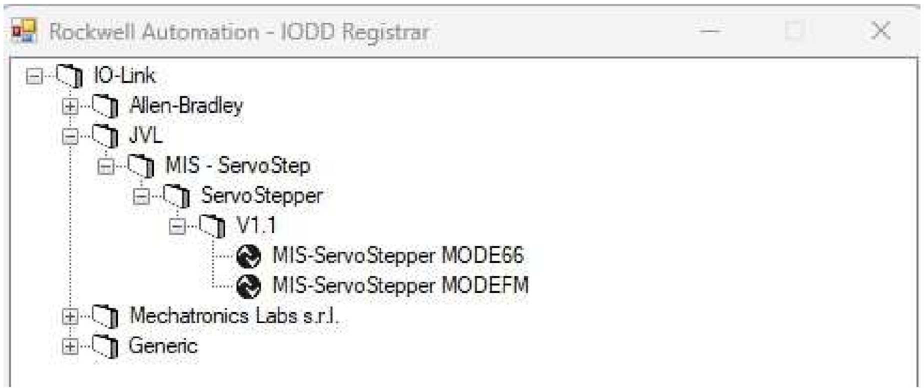
1. From the IODD import dialog, select the JVL IODD file for the requested mode:



JVL-MISxx-MODE66-20240925-IODD1.1 Date modified: 02/10/2024 17.29 Size: 25,0 KB

JVL-MISxx-ModeFM-20240925-IODD1.1 Date modified: 02/10/2024 17.29 Size: 27,6 KB

2. Select which channel on which the motor is connected, and assign the channel to the IODD configuration.

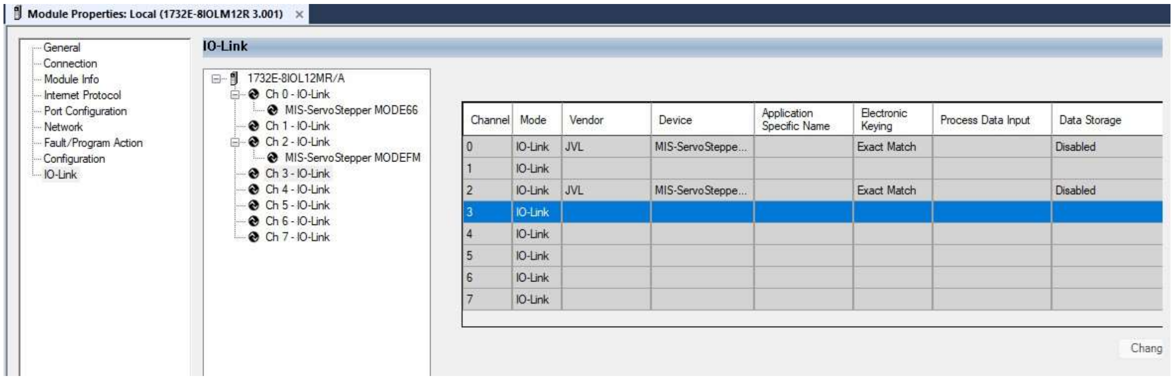


Rockwell Automation - IODD Registrar

```
IO-Link
  Allen-Bradley
  JVL
    MIS - ServoStep
      ServoStepper
        V1.1
          MIS-ServoStepper MODE66
          MIS-ServoStepper MODEFM
  Mechatronics Labs s.r.l.
  Generic
```

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.



Module Properties: Local (1732E-8IOLM12R 3.001)

IO-Link

```
1732E-8IOL12MR/A
  Ch 0 - IO-Link
    MIS-ServoStepper MODE66
  Ch 1 - IO-Link
  Ch 2 - IO-Link
    MIS-ServoStepper MODEFM
  Ch 3 - IO-Link
  Ch 4 - IO-Link
  Ch 5 - IO-Link
  Ch 6 - IO-Link
  Ch 7 - IO-Link
```

Channel	Mode	Vendor	Device	Application Specific Name	Electronic Keying	Process Data Input	Data Storage
0	IO-Link	JVL	MIS-ServoSteppe...		Exact Match		Disabled
1	IO-Link						
2	IO-Link	JVL	MIS-ServoSteppe...		Exact Match		Disabled
3	IO-Link						
4	IO-Link						
5	IO-Link						
6	IO-Link						
7	IO-Link						

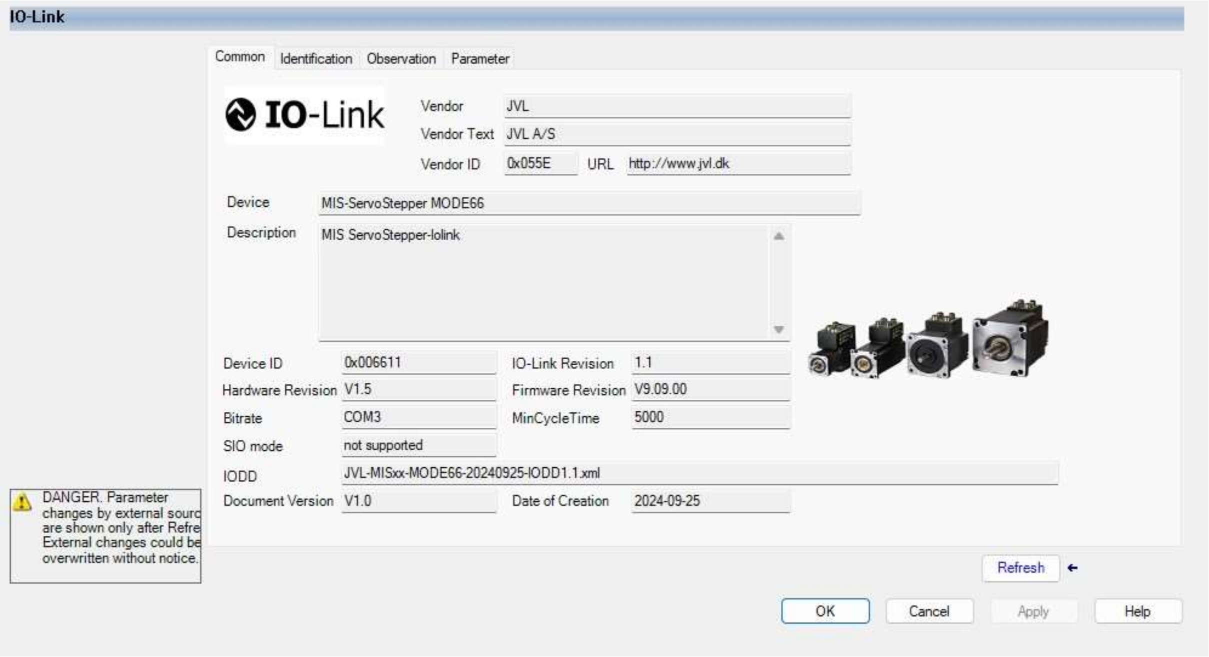
3. Download the configuration to the PLC and make sure the motor is connected to the IO-Link master and power supply.

In case the configuration doesn't match the setting, follow the instructions Changing the IO-Link mode.

The motor is configured for Mode66 from the factory, only the IODD file for Mode66 will work with this mode.

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

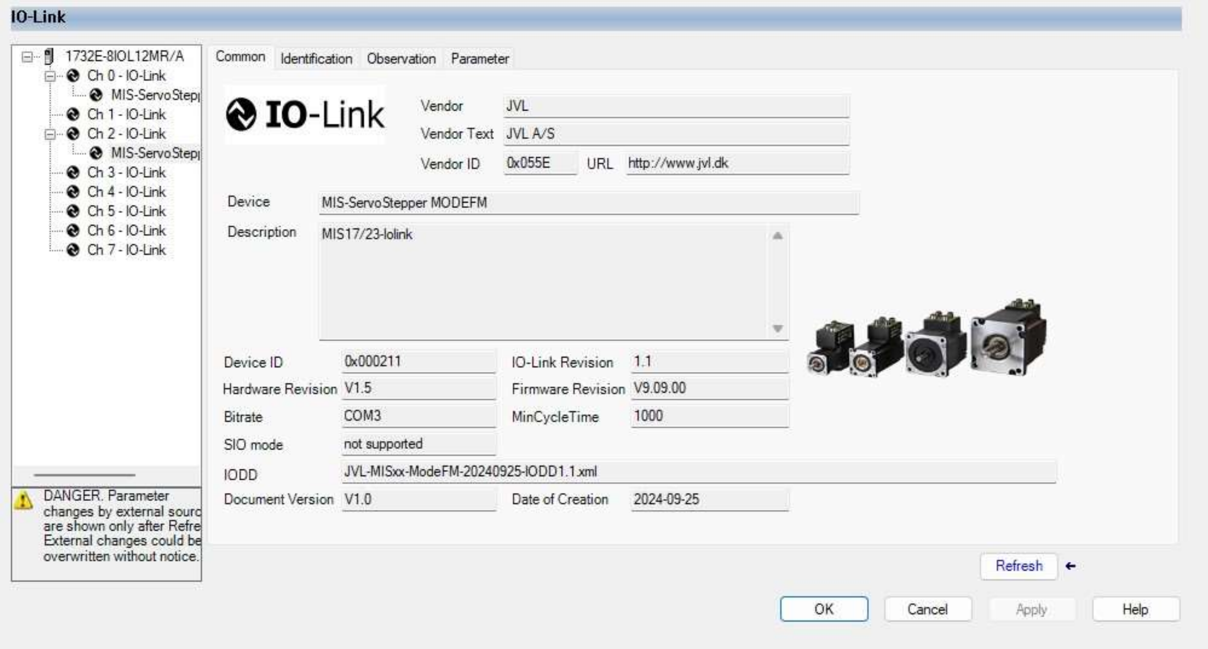
Mode66:



Vendor JVL
Vendor Text JVL A/S
Vendor ID 0x055E URL http://www.jvl.dk
Device MIS-ServoStepper MODE66
Description MIS ServoStepper-Iolink
Device ID 0x006611 IO-Link Revision 1.1
Hardware Revision V1.5 Firmware Revision V9.09.00
Bitrate COM3 MinCycleTime 5000
SIO mode not supported
IODD JVL-MISxx-MODE66-20240925-IODD1.1.xml
Document Version V1.0 Date of Creation 2024-09-25

DANGER. Parameter changes by external sourc are shown only after Refre External changes could be overwritten without notice.

ModeFM:



DANGER. Parameter changes by external sourc are shown only after Refre External changes could be overwritten without notice.

Vendor: JVL
Vendor Text: JVL A/S
Vendor ID: 0x055E URL: http://www.jvl.dk
Device: MIS-ServoStepper MODEFM
Description: MIS17/23-Iolink
Device ID: 0x000211 IO-Link Revision: 1.1
Hardware Revision: V1.5 Firmware Revision: V9.09.00
Bitrate: COM3 MinCycleTime: 1000
SIO mode: not supported
IODD: JVL-MISxx-ModeFM-20240925-IODD1.1.xml
Document Version: V1.0 Date of Creation: 2024-09-25

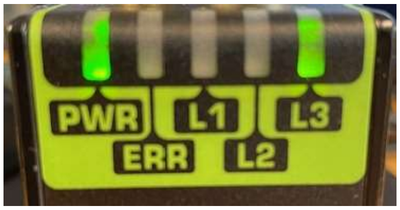
4.3 Status indicator

The status indicators on the rear of the motor indicates 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 has been established, L3 will remain solid green.

In case the connection is interrupted, L3 will begin blinking again, until the connection has been re-established.



5 Controlling a motor using 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	**-**
IO1-4 Output request	IO1-4 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	**-**
IO1-4 Input status	IO1-4 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 isnt required after a power cycle.

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

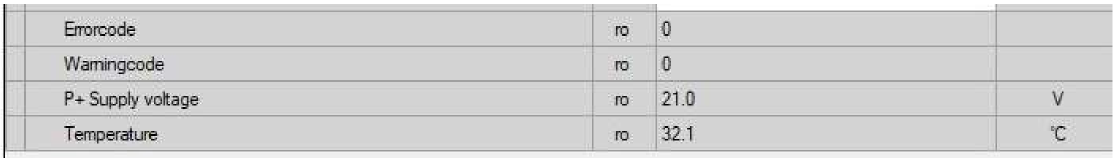
5: Optional, not used.

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”.



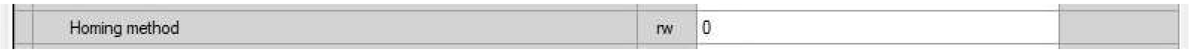
Errorcode	ro	0	
Warningcode	ro	0	
P+ Supply voltage	ro	21.0	V
Temperature	ro	32.1	℃

5.1 Homing

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:



Homing method	rw	0	

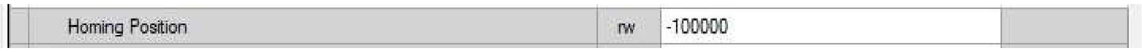
0=Disabled, active homing is not possible, only passive homing.

3072 = Mechanical homing

3328 = Sensor Uni-Directional

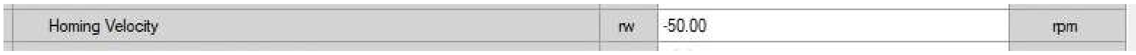
3584 = Sensor Bi-Directional

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



Homing Position	rw	-100000	

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



Homing Velocity	rw	-50.00	rpm

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 -octed 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.

5.1.1 *Mechanical Homing*

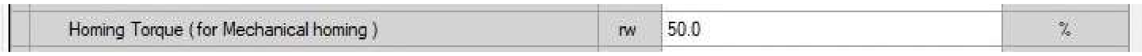
Homing Method = 3072

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 %

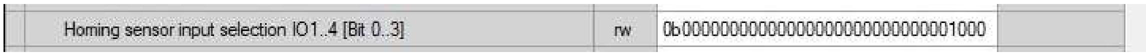
5.1.2 *Sensor based homing Uni-Directional*

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.

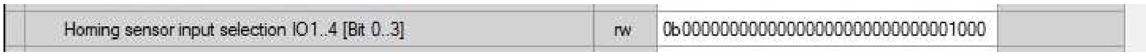
5.1.3 *Sensor based homing Bi-Directional*

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

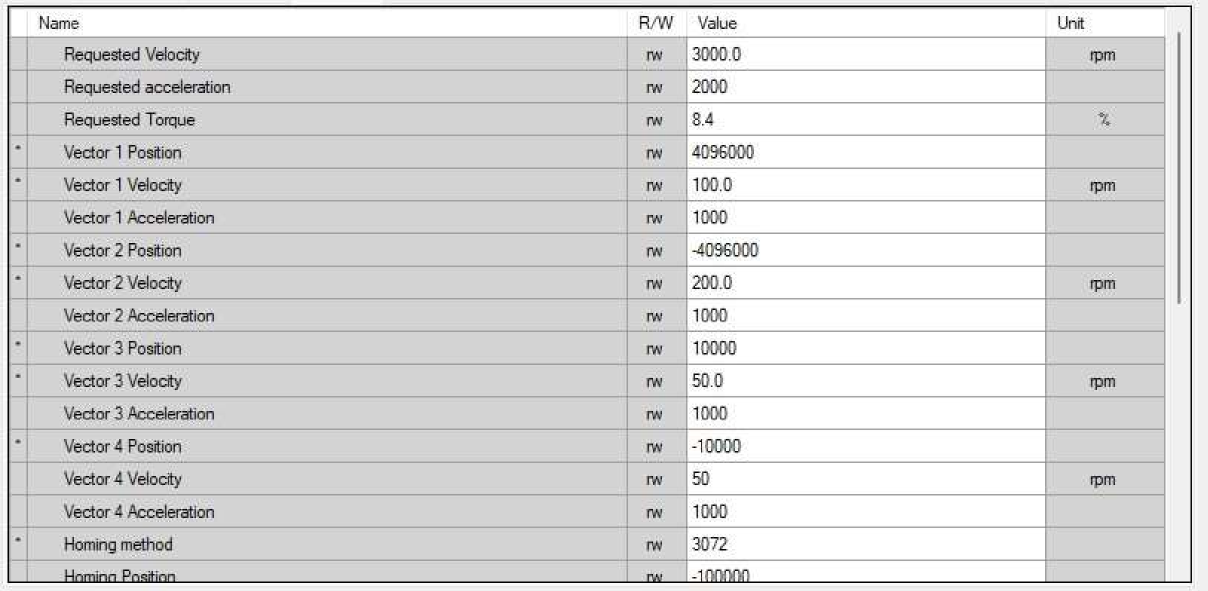
The parameter is set bitwise. By default bit 3 is set, indicating Input 4 is used.

5.2 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 can be configured:



	Name	R/W	Value	Unit
	Requested Velocity	rw	3000.0	rpm
	Requested acceleration	rw	2000	
	Requested Torque	rw	8.4	%
*	Vector 1 Position	rw	4096000	
*	Vector 1 Velocity	rw	100.0	rpm
	Vector 1 Acceleration	rw	1000	
*	Vector 2 Position	rw	-4096000	
*	Vector 2 Velocity	rw	200.0	rpm
	Vector 2 Acceleration	rw	1000	
*	Vector 3 Position	rw	10000	
*	Vector 3 Velocity	rw	50.0	rpm
	Vector 3 Acceleration	rw	1000	
*	Vector 4 Position	rw	-10000	
	Vector 4 Velocity	rw	50	rpm
	Vector 4 Acceleration	rw	1000	
*	Homing method	rw	3072	
	Homing Position	rw	-100000	

In the above example the following is configured:

Vector 1:

Position = 4096000 (10x motor revolutions)
Velocity = 100.00 RPM (meaning it will not change the velocity, but use the current setting)
Acceleration = 1000

Vector 2:

Position = -4096000
Velocity = 200.00 RPM (meaning it will not change the velocity, but use the current settins)
Acceleration = 1000

Vector 3:

Position = -10000
Velocity = 50.00 RPM
Acceleration = 1000

Vector 4:

Position = 10000
Velocity = 50.00 RPM
Acceleration = 1000

So Regardless of which position the motor is in issuing one of the following commands, will set the motor into an active motor and start the motion with the parameters configured in the parameters.

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

Note 1

This command must be applied prior to sending commands 108d-111d.

5.2.1 Example 1:

Activate Vector 1 and await for the motor to reach the destination.

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.

5.2.2 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.

5.3 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 run with an 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 vector 1 (Velocity 1 and Acceleration 1)
41d, 0x29	Execute velocity motion using vector 2 (Velocity 2 and Acceleration 2)
42d, 0x2A	Execute velocity motion using vector 3 (Velocity 3 and Acceleration 3)
43d, 0x2B	Execute velocity motion using vector 4 (Velocity 4 and Acceleration 4)

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

5.4 Modulus mode

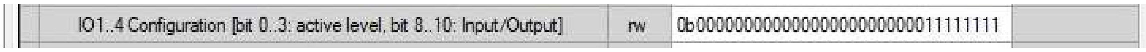
To be updated…

5.5 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..3 will control the output.

For configuration of the Inputs/outputs, the ISDU parameter “IO1..4 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), only bit0..3 are used.

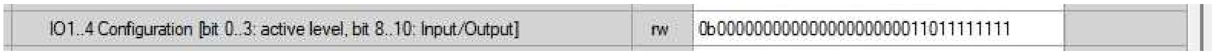
Where 1 = active high 0 = active low

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

Where 1 = Input, 0 = Output

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

5.6 Detecting HW inputs

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

Bit 0..3 indicates the status for the inputs of IO1..4

5.6.1 Cyclic timing

To be updated….

6 Controlling a motor Mode66

Opposed to the ModeFM, the Mode66 contains more data in the cyclic processdata exchange.

Process data is divided into 6x 32bits (4x bytes) -datablocks.

The cyclic exchanged process data is organized as follows:

Input process data:

Name	**Description**	**Unit**
Warning + Status + Actual mode (Note 1)	Statusbits and actual operation mode	-
Actual position (P_IST)	Position of generator output/ Encoder position	Counts
Actual Velocity (V_IST)	Shaft velocity	1/100 RPM
Actual Torque (Note 2)	Applied torque in closed loop configuration	2047=100%
Follow error	Position error	Counts
IO1-4 Input status (Note 3)	IO1-4 Hardware inputs status	Bitwise

Note 1:

The **32bit datablock** is divided into the following bytes:

Byte 3	Byte 2	Byte 1	Byte 0
Reserved	Warningbits[15..8]	Statusbits[7..0]	Actual mode

Actual mode 8-bit value: **0=Passive, 1=velocity, 2=Position**

Statusbit bits[0..7] :

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

Warningbits bits[8..15]:

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

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

10: Low bus voltage A low bus voltage is detected

11: IO Driver overload, Overloaded output driver for IO1..4.

12: Temperature critical The temperature has reached a critical limit of **80°C**

13: Overload, Motor driver overload.

14: STO Active, Safe Torque Off feature active

15: Reserved

Note 2:

This value will only be updated for motors equipped with the H2 or H4 -encode option. Due to the closed loop current control functionality.

Note 3

The **32bit datablock** is divided into the following bytes:

Reserved	Reserved	Reserved	IO1-4 Input status

Output process data:

Name	Description	Unit
CMD + Requested mode (Note 1)	Requested operation mode + Command	**-**
Requested position (P_SOLL)	Requested position	Counts
Requested Velocity (V_SOLL)	Requested shaft velocity	1/100 RPM
Requested acceleration	Requested acceleration	1/100 RPM/s
Requested Torque	Requested Max. torque	1533=100%
IO1-4 Output request (Note 2)	IO1-4 Hardware output	Bitwise

Note 1:

The **32bit datablock** is divided into the following bytes:

Reserved	Reserved	CMD	Requested mode

Requested mode 8-bits: **0=Passive, 1=velocity, 2=Position, 12= mechanical homing, 13= sensor unidirectional homing, 14 = Sensor Bi directional homing.**

Note 2

The **32bit datablock** is divided into the following bytes:

Reserved	Reserved	Reserved	IO1-4 Output request

CMD 8 bit -value:

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. Note! This function will eventually be replaced by the IOLINK data storage functionality.

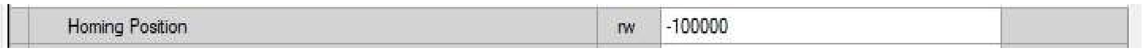
Some commands will reset the motor, hence the communication is lost for a few seconds. An internal function will make sure that the command value isn’t executed again due to the cyclic behavior, until the command has been set to a different value.

6.1.1 Homing

Basically 3 different homing methods are supported:

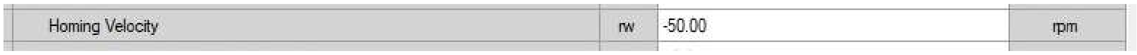
1. Homing using a sensor, connected to the motor
2. Homing using a mechanical endstop
3. Passive homing only setting the encoder reference.

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



Homing Position	rw	-100000	

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



Homing Velocity	rw	-50.00	rpm

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

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

12 = Mechanical homing

13 = Uni-directional sensor homing

14 = Bi directional sensor homing

Common for the sensor homing methods, is that Input 4 is by default the input for the homing sensor.

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

Note that passive homing is supported as well, but this is done by applying a command in the command octet. Please see the command section for further details.

6.2 Position control

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

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

Requested position (P_SOLL)	Requested position	Counts
Requested Velocity (V_SOLL)	Requested shaft velocity	1/100 RPM
Requested acceleration	Requested acceleration	1/100 RPM/s
Requested Torque	Requested Max. torque	1533=100%

Example 1:

The motor is commanded into “Position mode”.

The processdata word 0 with therequested data in Octet 0 is set = 2

CMD + Requested mode (Note 1)	Requested operation mode + Command	**-**

The **32bit datablock** is divided into the following bytes:

Reserved	Reserved	CMD	Requested mode

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 (P_SOLL)	4096000	Counts
Requested Velocity (V_SOLL)	120000	1/100 RPM

Requested acceleration	10000	1/100 RPM/s
Requested Torque	766	1533=100%

While the motor is in motion we will observe from the statusbits, 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.

6.3 Velocity

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

The requested velocity is controlled in the Procesdata "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	766	1533=100%

Example 3

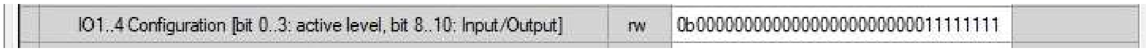
Running -1200 RPM, Acceleration = 10000, Max. Torque is set to 30%

Requested Velocity (V_SOLL)	-120000	1/100 RPM
Requested acceleration	10000	1/100 RPM/s
Requested Torque	460	1533=100%

6.4 Output control

Bit 0..3 will control the output.

For configuration of the Inputs/outputs, the ISDU parameter “IO1..4 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), only bit0..3 are used.

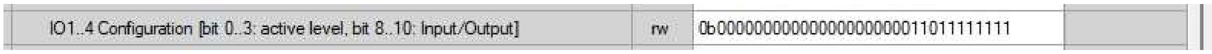
Where 1 = active high 0 = active low

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

Where 1 = Input, 0 = Output

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

6.5 Input control

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

Bit 0..3 indicates the status for the inputs of IO1..4