

# **Integrated Step Motors, QuickStep,**

**MIS/L17x, MIS/L23x (Generation 2),  
MIS/L34x and MIS/L43x**

**Including Step Motor Controller  
SMC66, SMC85**

## **User Manual**



**JVL**  
intelligent motors

## **JVL A/S**

# Important User Information



## Warning



The MIS and SMC series of products are used to control electrical and mechanical components of motion control systems. You should test your motion system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel.

Please contact your nearest JVL representative for technical assistance. Your nearest contact can be found on our web site [www.jvl.dk](http://www.jvl.dk)

Copyright 1998-2021, JVL A/S. All rights reserved.

This user manual must not be reproduced in any form without prior written permission of JVL A/S.

JVL A/S reserves the right to make changes to information contained in this manual without prior notice.

Furthermore JVL A/S assumes no liability for printing errors or other omissions or discrepancies in this user manual.

*MacTalk and MotoWare are registered trademarks*

JVL A/S  
Bregnerødvej 127  
DK-3460 Birkerød  
Denmark  
Tlf. +45 45 82 44 40  
Fax. +45 45 82 55 50  
e-mail: [jvl@jvl.dk](mailto:jvl@jvl.dk)  
Internet: <http://www.jvl.dk>

# Contents

---

<b>I</b>	<b>Introduction .....</b>	<b>7</b>
1.1	Feature overview .....	8
1.2	General description .....	11
1.3	Quick start guide .....	13
<b>2</b>	<b>Hardware .....</b>	<b>15</b>
2.1	Power Supply .....	16
2.2	User Inputs .....	25
2.3	Analogue Inputs .....	29
2.4	User Outputs .....	35
2.5	Serial interfaces overview .....	37
2.6	RS485 Interface .....	38
2.7	EMC considerations .....	39
2.8	How to connect a MIS motor .....	41
2.9	LED indicators basic motor .....	57
2.10	LED indicators using CANopen .....	58
2.11	LED indicators using Ethernet .....	59
<b>3</b>	<b>Functional Safety .....</b>	<b>61</b>
3.1	STO - Safe Torque Off .....	62
3.2	Safety .....	64
3.3	Installation .....	67
3.4	Commissioning .....	70
3.5	Specifications and Certifications .....	81
<b>4</b>	<b>Using MacTalk .....</b>	<b>83</b>
4.1	Using the MacTalk software .....	84
4.2	How to update MacTalk .....	92
4.3	How to update the motor firmware .....	93
4.4	How to update the encoder FW .....	94
4.5	How to get SW/HW motor info .....	95
<b>5</b>	<b>Description of functions .....</b>	<b>97</b>
5.1	Setting up the motor current .....	98
5.2	Auto Correction .....	100
5.3	Closed loop operation .....	104
5.4	Absolute position back-up .....	114
5.5	Multifunction I/O setup .....	118
5.6	Dedicated outputs .....	121
5.7	SSI encoder/sensor interface .....	123
5.8	Absolute Multi-turn Encoder .....	128
5.9	Position Limits .....	134
5.10	Under voltage Handling .....	139
5.11	Electro Mechanical brake .....	142
5.12	Turn Table Mode .....	145
<b>6</b>	<b>Modes .....</b>	<b>151</b>
6.1	Passive Mode .....	152
6.2	Velocity Mode .....	153
6.3	Positioning Mode .....	154
6.4	Gear Mode .....	155

6.5	Zero search modes .....	163
<b>7</b>	<b>Error Handling .....</b>	<b>171</b>
7.1	Setup error limits .....	172
7.2	Error messages .....	173
<b>8</b>	<b>Registers .....</b>	<b>183</b>
8.1	Introduction to registers .....	184
8.2	Internal registers .....	185
<b>9</b>	<b>Building Sequential Programs .....</b>	<b>243</b>
9.1	Getting started with programming .....	244
9.2	Programming Main window .....	245
9.3	Programming menu .....	246
9.4	How to build a program .....	247
9.5	General programming hints .....	250
9.6	Command toolbox description .....	251
9.7	Graphic programming command reference .....	252
9.8	Command timing .....	273
9.9	More about program timing .....	274
<b>10</b>	<b>Ethernet protocols (optional) .....</b>	<b>275</b>
<b>11</b>	<b>CANopen (optional) .....</b>	<b>277</b>
11.1	General info about CANopen .....	278
11.2	Connect and setup of the CAN bus .....	283
11.3	Using CAN-explorer .....	285
11.4	Objects in the DS301 standard .....	290
11.5	Objects in the DSP-402 standard .....	304
11.6	Running MIS as NC axis in TwinCAT .....	321
11.7	LSS Layer setting service .....	344
11.8	J1939 CAN protocol .....	347
11.9	More details of CANopen Theory .....	352
<b>12</b>	<b>Modbus interface .....</b>	<b>363</b>
12.1	Modbus Interface .....	364
12.2	Modbus Configuration .....	365
12.3	Reading and Writing Registers .....	367
12.4	Read Holding Registers (0x03) .....	368
12.5	Write Multiple Registers (0x10) .....	369
12.6	PDO Registers .....	370
12.7	Read/write PDO register command .....	373
<b>13</b>	<b>DMX512 Protocol and interface .....</b>	<b>375</b>
13.1	DMX512 Description .....	376
<b>14</b>	<b>Stand alone electronics .....</b>	<b>381</b>
14.1	Step motor controllers (SMCxx) .....	382
<b>15</b>	<b>Non-intelligent products .....</b>	<b>385</b>
<b>16</b>	<b>Technical Data .....</b>	<b>387</b>
16.1	MIS17x Technical Data .....	388
16.2	MIS23x Technical Data .....	389
16.3	MIS34x Technical Data .....	390
16.4	MIS43x Technical Data .....	391
16.5	Torque Curves .....	392
16.6	Physical Dimensions .....	401
16.7	Life time .....	405
16.8	Trouble-shooting guide .....	406
<b>17</b>	<b>Accessories .....</b>	<b>407</b>

17.1	Cables .....	408
17.2	Power Supplies .....	409
17.3	Brakes and shaft reinforcement .....	410
17.4	Gear and brake mounting instruction .....	411
<b>18</b>	<b>Appendix .....</b>	<b>413</b>
18.1	Motor Connections .....	414
18.2	Serial communication .....	416
18.3	Ordering information .....	421
18.4	Encoder calibration .....	422
<b>19</b>	<b>Declarations .....</b>	<b>424</b>
19.1	CE Declaration of Conformity .....	425
19.2	Vibrationtest certificates .....	427
19.3	UL certificate - MIS34x .....	429
19.4	TÜV certificate .....	431





This user manual describes the set-up and usage of the following products:

Complete motors with build-in controller or driver

- Types **MIS171**, **MIS172** and **MIS176** (NEMA17 sizes)
- Types **MIS231**, **MIS232** and **MIS234** (NEMA23 sizes)
- Types **MIS340**, **MIS341** and **MIS342** (NEMA34 sizes)
- Types **MIS43x** (NEMA43 sizes) - only limited supported in this manual.

**Important:**

Please notice that only the latest generation of MIS17x and MIS23x is supported by this user manual. The extension after MIS23x must be: S, Q, T or R.

Stand-alone electronics without motor

- Types **SMC66** and **SMC85** controller PCB with intelligence (fully programmable)

All the Quickstep motors are available as a fully programmable product with a wide range of features also covering a simple pulse and direction interface or Ethernet options.

Examples of motors and stand alone electronics.



## 1.1

## Feature overview



The compact MIS motors are designed for easy installation and high performance use. A large number of features are available and also multiple options available as listed below.

- Serial RS485 or 5V serial position controller.
- Build-in mini PLC with graphic programming.
- Option for CANbus, CANopen DS-301. Fully ISO 11898-2:2016 compliant/(DSP-402 in development).
- Options for EthernetIP, Profinet, Powerlink, ModbusTCP, Sercos-III and EtherCAT.
- A dual supply facility is available so that position and parameters are maintained at emergency stop.
- Electronic Gear mode.
- MACmotor protocol so MAC servomotors and MIS stepper motors can be connected on the same RS485 bus.
- Command for easy PLC/PC setup and communication.
- Power supply 7-72 VDC.
- Extremely high torque vs speed - up to 3000 RPM with good performance.
- Fixed 409600 steps per revolution
- Built-in 32Bit  $\mu$ processor with 8 In/Out that can be configured as inputs, PNP outputs or analogue inputs. 5V serial and RS485 interface for set up and programming.
- MODBUS interface.
- 9.6kbit/sec. to 1Mb/sec. communication.

Benefits when using the MIS motors:

- De-central intelligence.
- Simple installation. No cables between motor and controller/driver.
- EMC safe. Switching noise remains within motor.
- Compact. Does not take space in the control cabinet.
- Low-cost alternative traditional systems where motor and controller is separated.
- Option: Closed loop feature by means of magnetic encoder with resolution of up to 4096 pulses/rev. (H2 or H4 option)
- Option: Absolute multi turn encoder for keeping the position permanent also during power down. (H3 or H4).
- Vibration tested at 4G in 3 axis and shock tested at 15G in 3 axis according to IEC60068.
- Interface possibilities:
- From PC/PLC with serial commands via 5V serial or RS485.
- Pulse/direction input. Encoder output.
- CANopen.
- 8 I/O, 7-30 VDC that can be configured as Inputs, Outputs or analogue inputs.
- Wireless options: WiFi, Bluetooth and Zigbee.

# 1.1

## Feature overview

---

The MIS motors are also available with alternative options.

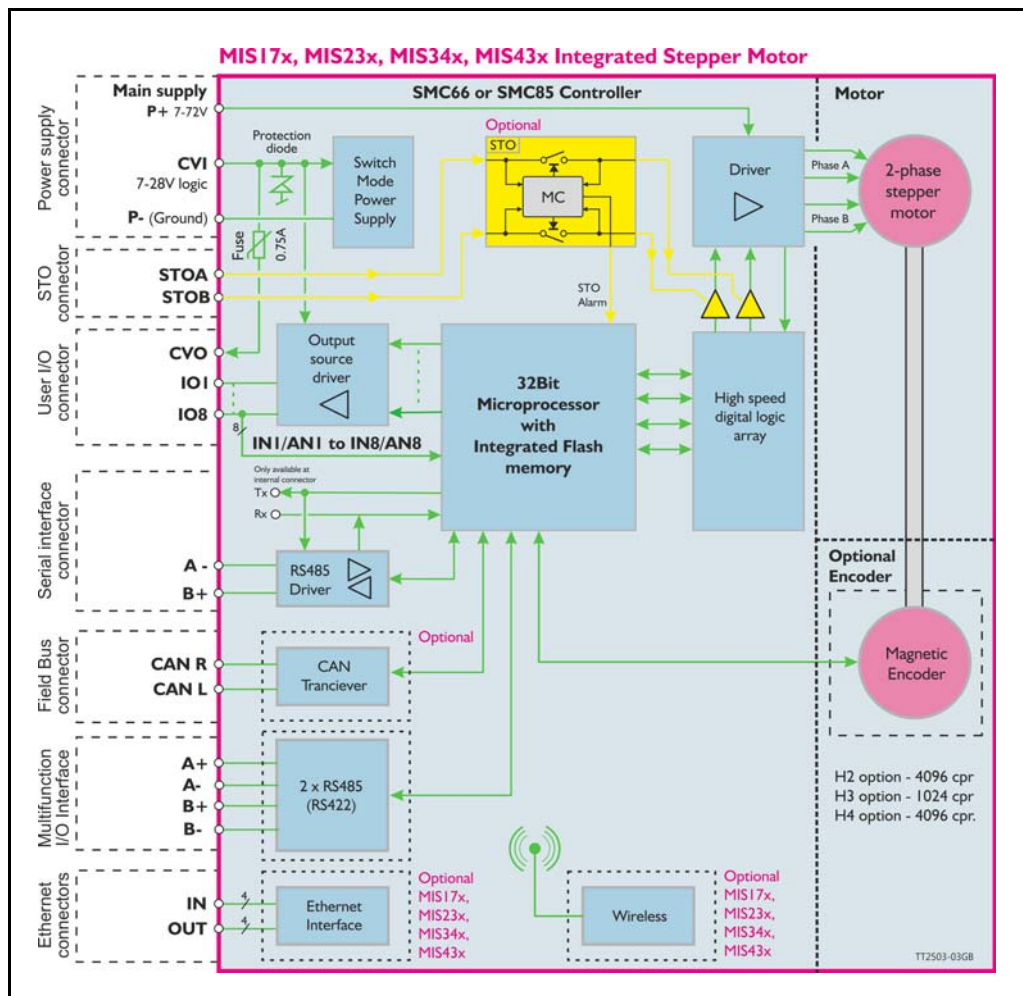
- Protection class IP42 and higher.
- Hollow shaft.
- Integrated ball screw or spindle for linear movement.
- Custom made design for special applications



# 1.1

## Feature overview

### 1.1.1 Block diagram, Positioning/Speed Control

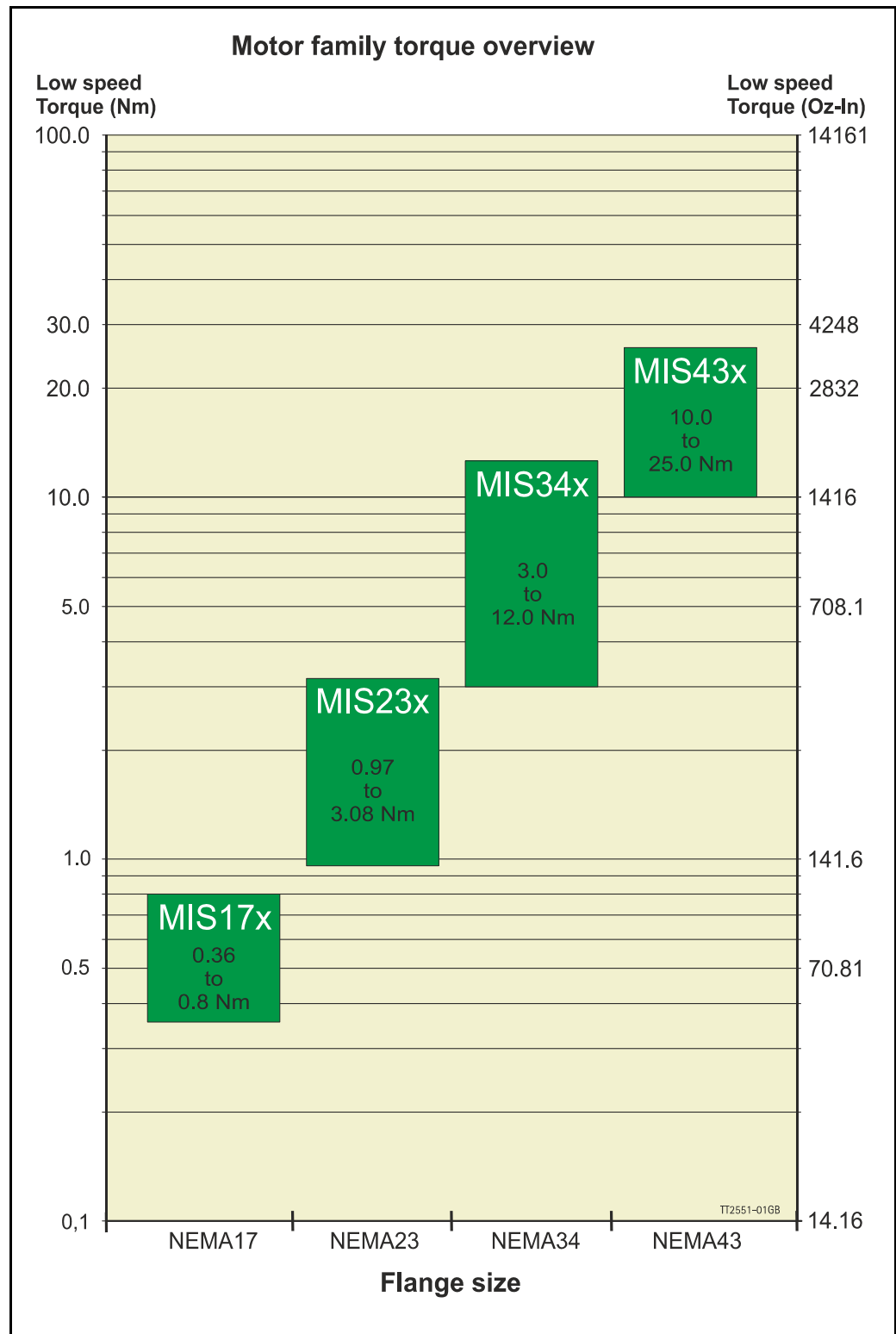


## 1.2

## General description

The QuickStep motors are currently available in 4 different family sizes which have a NEMA 17, 23, 34 and 43 flange.

Under each flange size additional versions exist with different motor torques.



## 1.2 General description

Below are more detailed specification of the individual motors in each motor size.

### 1.2.1 MIS17x - NEMA17 flange

Motor type:	MIS171S/Q	MIS173S/Q	MIS176S/Q		
Holding torque Nm [Oz-In]	0.36 [50.9]	0.56 [79.3]	0.80 [113.3]		
Inertia kgcm <sup>2</sup> [Oz-In-Sec <sup>2</sup> ]	0.020 / [0.000283]	0.054 / [0.000793]	0.102 / [0.00144]		
Weight - kg [lb]	0.54 [1.19]	0.68 [1.5]	0.9 [1.99]		
Max. axial shaft force N	-	-	-		
Max. radial shaft force (N) (applied 20 mm from flange)	-	-	-		

### 1.2.2 MIS23x - NEMA23 flange

Motor type:	MIS231S/Q	MIS231T/R	MIS232S/Q	MIS232T/R	MIS234S/Q
Holding torque Nm [Oz-In]	0.97 [137.4]	1.16 [164.3]	1.97 [279.0]	2.53 [358.3]	3.08 [436.2]
Inertia kgcm <sup>2</sup> [Oz-In-Sec <sup>2</sup> ]	0.3 [0.00423]	0.3 [0.00423]	0.48 [0.00677]	0.48 [0.00677]	0.65 [0.0092]
Weight - kg [lb]	1.1 [2.43]	1.1 [2.43]	1.4 [3.09]	1.4 [3.09]	2.0 [4.41]
Max. axial shaft force N	15	15	15	15	15
Max. radial shaft force (N) (applied 20 mm from flange)	75	75	75	75	75

### 1.2.3 MIS34x - NEMA34 flange

Motor type:	MIS340	MIS341	MIS342	MIS343
Holding torque - Nm [Oz-In]	2.0 [282]	4 [575]	6.0 [849]	8.0 [1151]
Low speed torque - 1 RPM - Nm [Oz-In]	3.0 [424]	6.1 [863]	9.0 [1274]	12.0 [1727]
Inertia - kgcm <sup>2</sup> [Oz-In-Sec <sup>2</sup> ]	1.4 [0.0198]	2.7 [0.0381]	4.0 [0.0564]	4.0 [0.0564]
Weight - kg [lb]	2.05 [4.52]	3.13 [6.9]	4.2 [9.26]	4.5 [10.71]
Max. axial shaft force	115N			
Max. radial shaft force	180N applied 12.5mm from shaft end			

### 1.2.4 MIS43x - NEMA43 flange

Motor type:	MIS430	MIS432	
Holding torque - Nm [Oz-In]	6.7 [944]	16.7 [2360]	
Low speed torque - 1 RPM Nm [Oz-In]	10.0 [1416]	25.0 [3540]	
Inertia - kgcm <sup>2</sup> [Oz-In-Sec <sup>2</sup> ]	5.5 [0.0779]	16.2 [0.229]	
Weight - kg [lb]	5.5 [12.13]	12.2 [26.9]	
Max. axial shaft force	80 N		
Max. radial shaft force	At 5mm max 640N, At 10mm max 425 N, At 15mm max 320N, At 20mm max 240N		

Concerning info for the **MILxxx (linear)** motors please consult your nearest JVL representative.

### 1.2.5 Basic modes/functions in the QuickStep motor

The QuickStep motor offers the following functions:

Mode	Description
Passive	The motor will be in a completely passive state but communication is active and internal registers can be setup. Motor shaft can be turned by hand.
Velocity	The motor velocity can be controlled using MacTalk software or by setting register 5 (V_SOLL) using serial or program commands.
Position	The motor position can be controlled using MacTalk or by setting register 3 (P_SOLL) using serial or program commands.
Gear	The motor position and velocity can be controlled by pulse and direction or encoder signals at the inputs "IN1" and "IN2". The gear ratio can be set to a large ratio by using register 14 (GEAR1) and register 15 (GEAR2).
CSP Mode	Cyclic Synchronous Position mode (Ethernet only)

## 1.3

## Quick start guide

---

Please follow this link:

<https://www.jvl.dk/files/downloads-1/pr/lb0062.pdf>



The following pages explains how the I/O, Power supply, Interface etc. can be connected and used.

## 2.1

## Power Supply

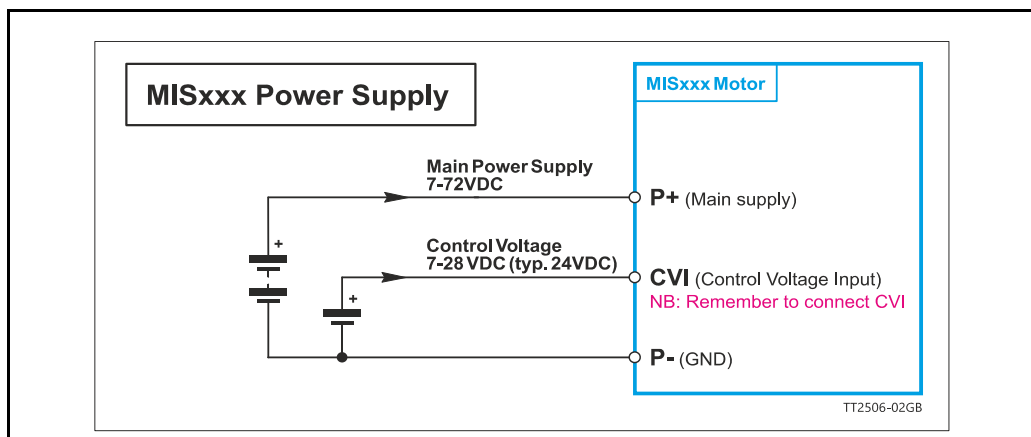
### 2.1.1 General Aspects of Power Supply

Powering of the motor is relatively simple. The supply input of the MIS motor family is equal for all family members. The supply consists of 2 inputs:

**CVI** The control voltage input is supplying all the internal control circuits including the microprocessor, encoder and user I/O circuitry. The voltage needed must be in the range 7-30 VDC which also support battery driven applications. Supply current is below 200 mA (voltage dependant and no user outputs activated).

**P+** The main power for driving the motor is supplied on this terminal and must be in the range 7-72 VDC. The voltage at this terminal will also influence torque at higher velocities. A voltage of 72 VDC will give much higher torque than using for example 24 VDC. The supply current can get as high as 6 ARMS. See also [Torque Curves](#), page 392 which shows the relation between supply voltage and the torque.

If the motor needs to be stopped it can be done by removing P+ but keeping CVI connected. This will keep the complete motor alive including I/O's and encoder circuit (if present) except that the motor driver and motor is power less and will not rotate. Supplying both terminals from the same power supply can be done, but then 30VDC is the absolute maximum voltage allowed (= maximum limit of CVI).



**NB:** for actual connections, see drawing [How to connect a MIS motor](#), page 41

### 2.1.2 Supplying only the control core



The main supply at the terminal P+ can be removed while keeping CVI supplied. This will keep all internal functions alive including all communication channels such as Ethernet (Option), position counters etc. When P+ later is re-applied the motor is fully operational and can continue the movement and operation it was doing just before the P+ supply was removed.

**Important:** CVI and P+ are not in any way connected together internally. There is no diode or resistor from CVI to P+ or from P+ to CVI. With 24VDC present at CVI and P+ disconnected, due to a minor leakage current, the voltage at the P+ will increase to typical 0.95V. This voltage increase at P+ is far from enough to make the motor able to turn - but if you need full safety please use the optional STO function which is available for all MIS and MIL motors. See also [STO - Safe Torque Off](#), page 62.

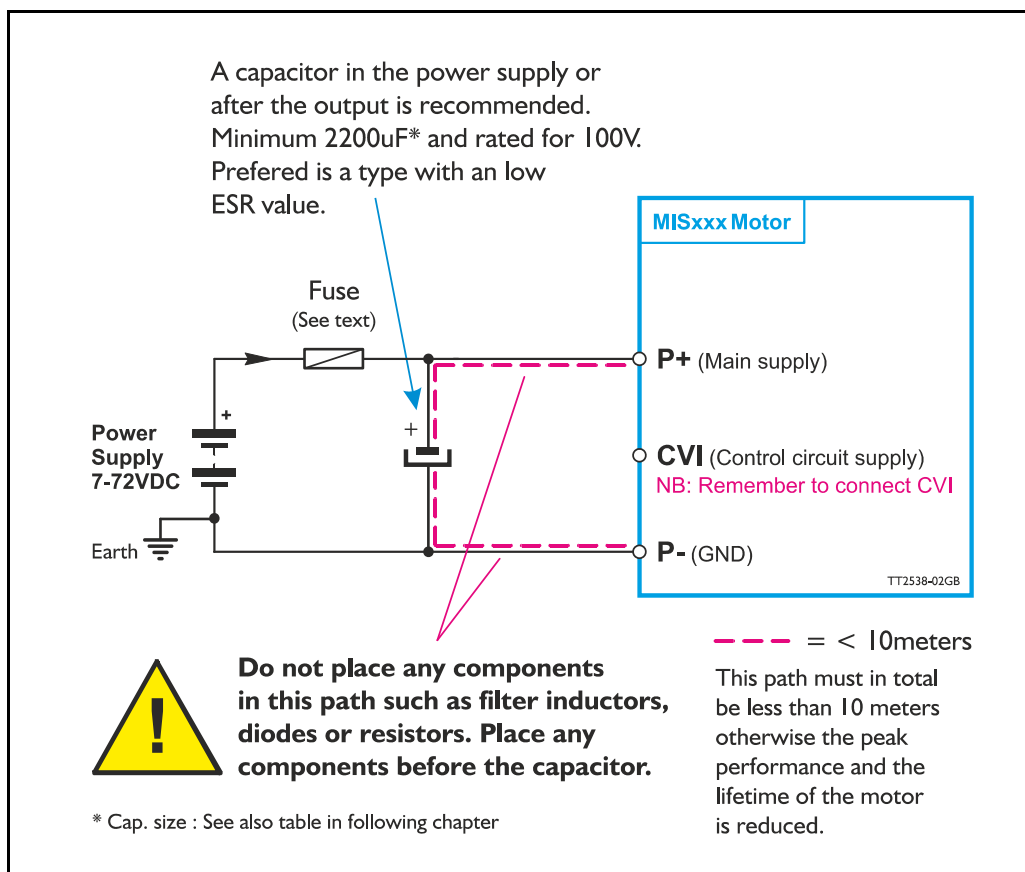
## 2.1

## Power Supply

### 2.1.3 Power Supply - Requirements and Precautions

For optimum performance and lifetime of the MIS motor, it is recommended that a capacitance of minimum  $2200\mu\text{F}$  is connected to the power supply that supply the P+ terminal. It should be mounted as close as possible to the motor.

Also, it is recommended that minimum  $0.75\text{ mm}^2$  / AWG22 cable or  $2 \times 0.5\text{ mm}^2$  is used to connect the power supply to the motor. If the supply voltage drops below 7V, the internal reset circuitry will reset the driver and an under voltage error is generated. Provision should therefore be made to ensure that the supply voltage is always maintained at a minimum of 7V, even in the event of a mains voltage drop.



#### CVI supply Precautions.

The CVI supply is not critical since the supply current is quite small ( $<200\text{ mA}$ ). Only make sure that the voltage stay at 24 VDC nominal and do not exceed 30 VDC. A CVI voltage down to 7 VDC is also possible but a software setup is needed to allow this. See also [Setup position backup using MacTalk](#), page 114

**Warning:** A supply voltage at CVI or P+ higher than 100VDC will cause permanent damages. A voltage over 30V at the CVI will activate a protection circuit which shuts down the supply input. In this case CVI need to be disconnected to reset the protection state.

#### Fuse dimensioning :

See [Dimensioning power supply and pre-fuse - MIS23x/MIL23x motors](#), page 21 or [Dimensioning power supply and pre-fuse - MIS34x/MIL34x motors](#), page 22

## 2.1

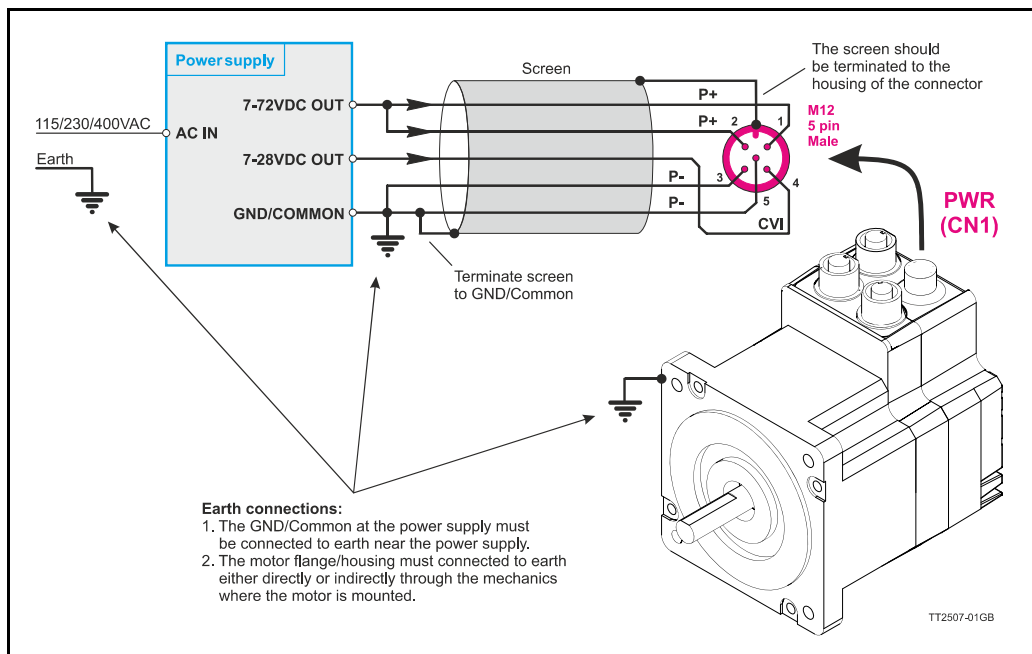
## Power Supply

### 2.1.4 Power Supply Grounding and Earthing

It is mandatory that the motor flange is connected to earth. Also it is mandatory that the earth is connected to GND/Common at a central point near the power supply.

The P- (GND/Common) is internally connected to the motor chassis/housing. Also the outside body at all M12 connectors is connected to the chassis/housing and thereby to the P- (GND/Common).

The illustration below shows how to make a good power and earth connection of the MIS motor.



## 2.1

## Power Supply

### 2.1.5 Control Voltage (CVI terminal)

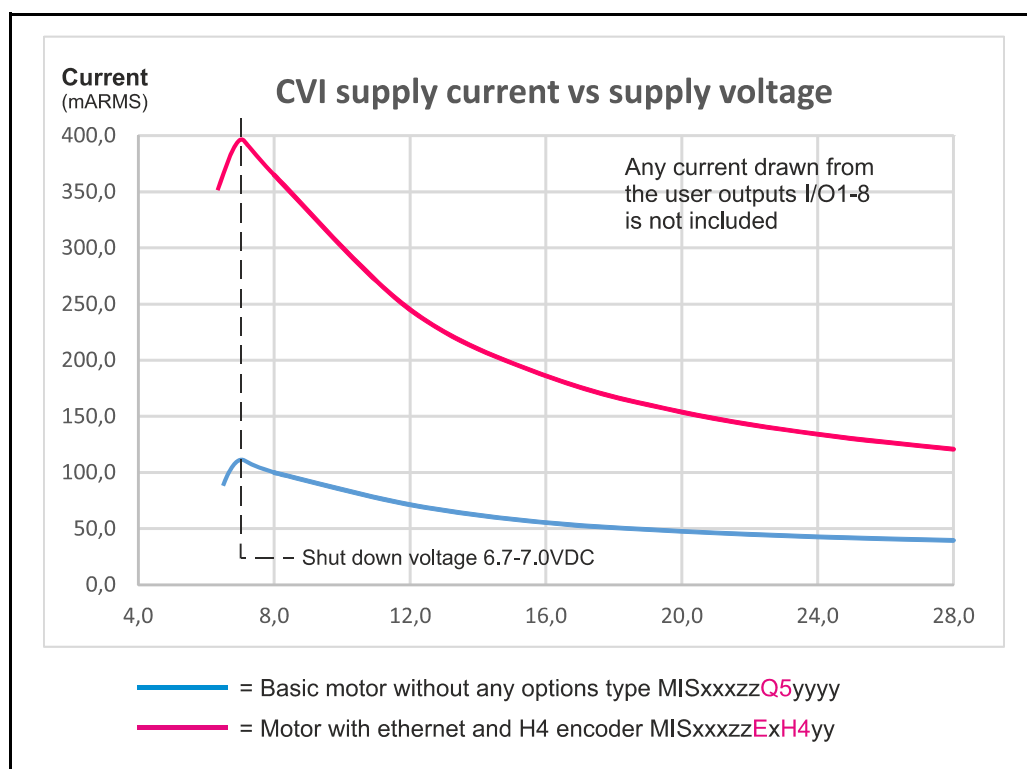
The control voltage should be in the range 7-30VDC and is used to supply the microprocessor circuit, internal functions in general and the user output driver (O1-8).

To ensure that position and parameters are maintained after an emergency stop, the control voltage should be maintained under a stop situation where the P+ (main power) is disconnected.

**Warning:** a voltage at the CVI terminal higher than 30VDC can damage the controller or cause malfunction. A protection circuit will protect from damages. If this protection circuit get triggered the CVI power need to be cycled in order to reset the protection.

A CVI voltage down to 7 VDC is also possible but a software setup is needed to allow this. See also [Setup position backup using MacTalk](#), page 114

The figure below shows the typical relation between supply current and supply voltage at the CVI input. As shown the current is very dependant at which options is installed in the motor. Worst case is if the motor is equipped with Ethernet and H4 encoder and best case is the basic motor such without Ethernet and encoder.



**Please notice that the user I/O's are supplied from the CVI terminal. The curves shown above do NOT include any load current at the user outputs. Add the load current(s) to the current shown above. If the motor is equipped with an internal electromechanical brake this must also be added to the current consumption.**

## 2.1

## Power Supply

### 2.1.6 Dimensioning power supply and pre-fuse - MIS17x motors

Note that this manual only covers MIS17x generation 2 motors.

The power supply must be dimensioned according to the actual motor size.

The size of the pre-fuse also depends on the actual model of the MIS motor.

Use the following table to select the power supply size and fuse ratings.

Supply voltage	MIS171 (All types)		MIS173 (All types)		MIS176 (All types)	
	Supply rating	Fuse size	Supply rating	Fuse size	Supply rating	Fuse size
-						
12VDC	35W *	T4A	40W *	T4A	40W *	T4A
24VDC	70W *	T3.15A	70W *	T4A	70W *	T4A
48VDC	90W *	T3.15A	125W *	T4A	150W *	T4A
72VDC	110W *	T2A	165W *	T3.15A	220W *	T4A
Recommended power supplies	<b>PSU24-075</b> (24V/75W) <b>PSU24-240</b> (24V/240W) <b>PSU48-240</b> (48V/240W) <b>PSU80-4</b> (80V/400W) **		<b>PSU24-240</b> (24V/240W) <b>PSU48-240</b> (48V/240W) <b>PSU80-4</b> (80V/400W) **		<b>PSU24-240</b> (24V/240W) <b>PSU48-240</b> (48V/240W) <b>PSU80-4</b> (80V/400W) **	

\* = Worst case scenario at higher velocities. Less may also do. Consult the torque/power curves in the appendix to get more precise info.

\*\* = Not recommended for new designs.

See also the appendix [Power Supplies, page 409](#) which shows the standard power supplies that JVL offers.

## 2.1

## Power Supply

### 2.1.7 Dimensioning power supply and pre-fuse - MIS23x/MIL23x motors

Note that this manual only covers MIS23x generation 2 motors.

The power supply must be dimensioned according to the actual motor size.

The size of the pre-fuse also depends on the actual model of the MIS motor.

Use the following table to select the power supply size and fuse ratings.

Supply voltage	MIS231S /MIL231		MIS232S /MIL232		MIS234S /MIL234	
-	Supply rating	Fuse size	Supply rating	Fuse size	Supply rating	Fuse size
12VDC	80W *	T4A	70W *	T6.3A	60W *	T5A
24VDC	130W *	T6.3A	110W *	T6.3A	100W *	T5A
48VDC	240W *	T6.3A	240W *	T6.3A	200W *	T5A
72VDC	370W *	T6.3A	350W *	T6.3A	300W *	T5A
Recommended power supplies	PSU24-075 ** PSU48-240 ** PSU72-1000 ** PSU80-4 **		PSU24-240 ** PSU48-240 ** PSU72-1000 ** PSU80-4 **		PSU24-240 ** PSU48-240 ** PSU72-1000 ** PSU80-4 **	
Supply voltage	MIS231T (High torque)		MIS232T (High torque)			
-	Supply rating	Fuse size	Supply rating	Fuse size		
12VDC	110W *	T6.3A	100W *	T6.3A		
24VDC	160W *	T8A	150W *	T8A		
48VDC	300W *	T8A	320W *	T8A		
72VDC	390W *	T8A	450W *	T8A		
Recommended power supplies	PSU24-075 PSU48-240 PSU72-1000 PSU80-4 **		PSU24-240 PSU48-240 PSU72-1000 PSU80-4 **			

\* = Worst case scenario at higher velocities. Less may also do. Consult the torque/power curves in the appendix to get more precise info.

\*\* = Not recommended for new designs.

See also the appendix [Power Supplies, page 409](#) which shows the standard power supplies that JVL offers.

## 2.1

## Power Supply

### 2.1.8 Dimensioning power supply and pre-fuse - MIS34x/MIL34x motors

The power supply must be dimensioned according to the actual motor size.

The size of the pre-fuse also depends on the actual model of the MIS motor.

Use the following table to select the power supply size and fuse ratings.

Supply voltage	MIS340/ MIL340		MIS341/ MIL341		MIS342/ MIL342		MIS343	
-	Supply rating	Fuse size	Supply rating	Fuse size	Supply rating	Fuse size	Supply rating	Fuse size
24VDC	120W *	T6.3A	125W *	T6.3A	130W *	T6.3A	140W *	T6.3A
48VDC	240W *	T6.3A	250W *	T6.3A	260W *	T10A	240W *	T10A
72VDC	450W *	T6.3A	460W *	T6.3A	470W *	T10A	400W *	T10A
Recommended power supply	PSU24-240 PSU48-240 PSU72-1000-10 PSU80-4 **		PSU24-240 PSU48-240 PSU72-1000-10 PSU80-4 **		PSU24-240 PSU48-240 PSU72-1000-10 PSU80-4 **		PSU24-240 PSU48-240 PSU72-1000-10 PSU80-4 **	

\* = Worst case scenario at higher velocities. Less may also do. Consult the torque/power curves in the appendix to get more precise info.

\*\* = Not recommended for new designs.

See also the appendix [Power Supplies, page 409](#) which shows the standard power supplies that JVL offers.

### 2.1.9 Dimensioning power supply and pre-fuse - MIS43x motors

The power supply must be dimensioned according to the actual motor size.

The size of the pre-fuse also depends on the actual model of the MIS motor.

Use the following table to select the power supply size and fuse ratings.

Supply voltage	MIS430		MIS431		MIS432		
-	Supply rating	Fuse size	Supply rating	Fuse size	Supply rating	Fuse size	
24VDC	110W *	T6.3A	100W *	T6.3A	100W *	T6.3A	
48VDC	210W *	T6.3A	200W *	T6.3A	200W *	T6.3A	
72VDC	300W *	T6.3A	305W *	T6.3A	330W *	T6.3A	
Recommended power supply	PSU24-240 PSU48-240 PSU72-1000-10 PSU80-4 **		PSU24-240 PSU48-240 PSU72-1000-10 PSU80-4 **		PSU24-240 PSU48-240 PSU72-1000-10 PSU80-4 **		

\* = Worst case scenario at higher velocities. Less may also do. Consult the torque/power curves in the appendix to get more precise info.

\*\* = Not recommended for new designs.

See also the appendix [Power Supplies, page 409](#) which shows the standard power supplies that JVL offers.

## 2.1

## Power Supply

---

### 2.1.10 UL applications

If the motor is used in an application which is to be UL approved following must be followed.

1. Supply voltage (P+): Do not use a supply voltage higher than 60VDC
2. Place a UL approved fuse in the P+ wire.
3. Make sure the motor is properly connected to a solid protective earth.
4. The surrounding ambient temperature must maximum be 40°C.

Notice that only the **MIS34x** family is UL recognized under UL file no: **E254947**

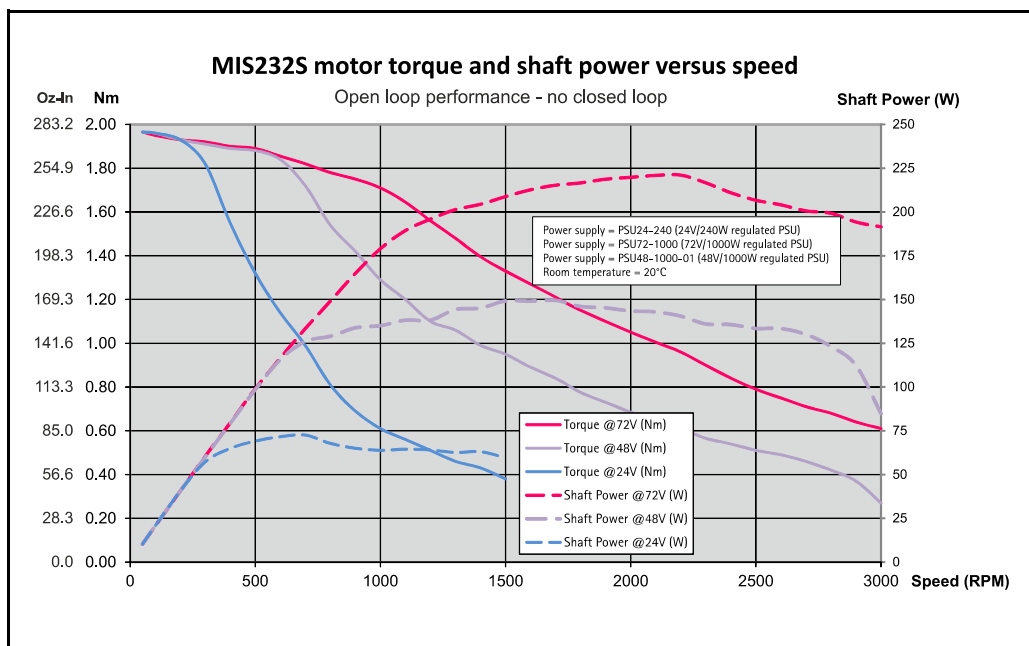
## 2.1 Power Supply

### 2.1.11 Select Your Power Supply

We recommend the use the highest possible voltage to supply the motor.  
In general the motor torque from a MIS/MIL is not affected by the supply voltage at speeds below 100RPM but at higher velocities the torque will be very influenced by the supply voltage (P+ terminal).  
Additionally, higher voltage gives better current and filter regulation and thereby better performance. If there is a tendency for motor resonance, a lower supply voltage can be a solution to the problem.

Below an example showing the torque curve for the MIS232S

As seen the torque and power stay high up to much higher speeds when using 72VDC supply compared with 24 or 48VDC.



## 2.2

## User Inputs

### 2.2.1 User Inputs

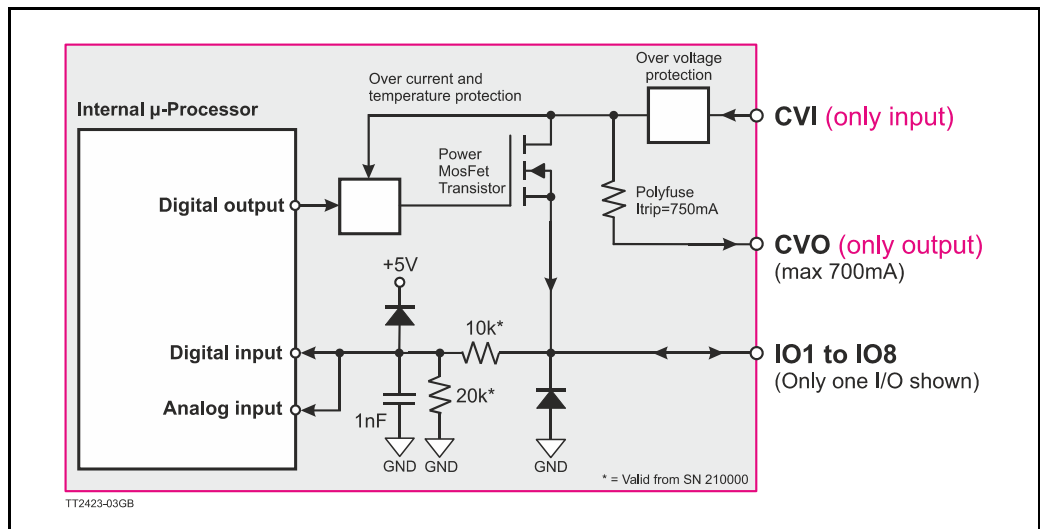
The MIS motors has 8 inputs/outputs (IO's) that each can be set individually to input, output or analogue input 0-5 VDC via MacTalk or software commands.

This makes it for example possible to have 4 inputs, 3 outputs and one analogue input.



**Please notice:** The number of available I/O terminals available may differ depending at which motor type and connector configuration you are using. Please consult the chapter [Connector overview for the MIS motors](#), page 41

**Input/output functional diagram:**

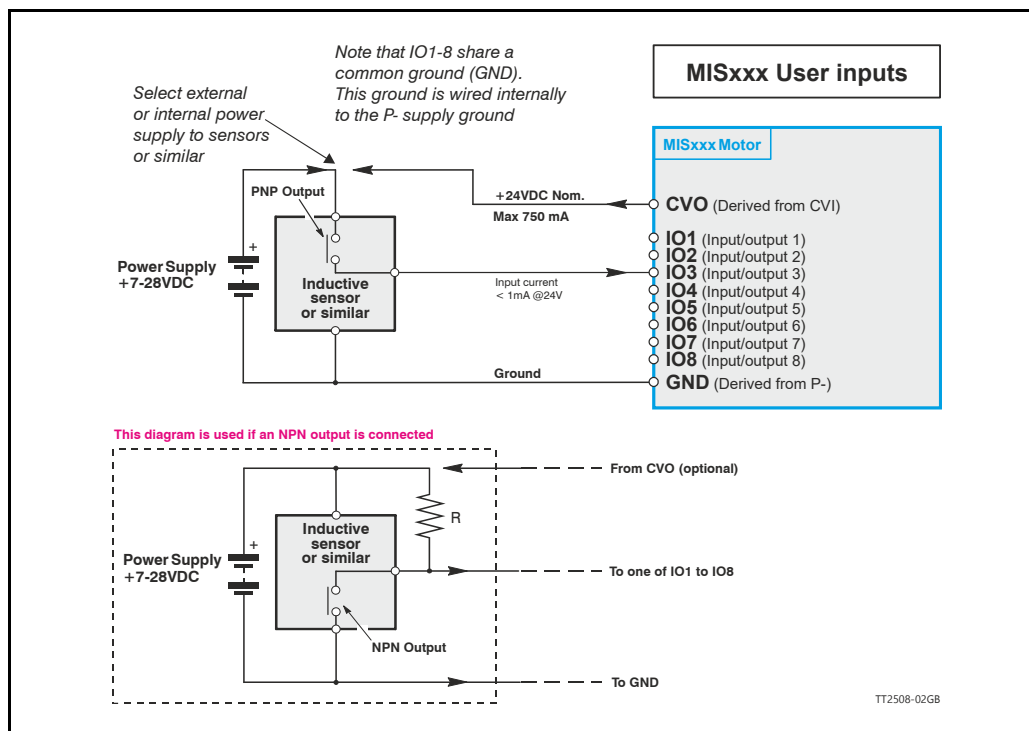


### 2.2.2 General Input features

- Inputs are TTL to 30 VDC compliant. Trigger levels LTL= 1.2 V and UTL 2.8 V.
- Over-current protection and thermal shut-down.
- 27.3 kOhm input resistance (SN>210000)
- No galvanic isolation but very robust against noise and spikes/surges.
- Zero search input can be selected to any input 1 to 8.
- Digital filter can be enabled for each input selectable from 0 to 100 ms. If disabled (default), the response time is 100  $\mu$ s.
- Limit switch inputs

## 2.2

## User Inputs



### 2.2.3

#### General

The Controller is equipped with a total of 8 digital inputs. Each input can be used for a variety of purposes depending on the actual application. Each of the inputs can be detected from the actual program that has been downloaded to the Controller or via serial commands.

The Inputs are not optically isolated from other Controller circuitry. All of the Inputs have a common ground terminal, denoted **GND**. Each Input can operate with voltages in the range 5 (TTL) to 30VDC. Note that the Inputs should normally be connected to a PNP output since a positive current must be applied for an input to be activated.

Note that CVO (control voltage output) is internally connected to the CVI supply terminal in the PWR connector. This provides the facility that local sensors can be supplied directly from the controller. CVO is internally fused to a maximum allowable current of 750 mA

### 2.2.4

#### Connection of NPN Output

If an Input is connected to an NPN output, a Pull-Up resistor must be connected between the Input and the + supply. See the illustration above.

The value of the resistance used depends on the supply voltage. The following resistances are recommended:

Supply Voltage	Recommended Resistance R
5-12VDC	1kOhm / 0.25W
12-18VDC	2.2kOhm / 0.25W
18-24VDC	3.3kOhm / 0.25W
24-28VDC	4.7kOhm / 0.25W

## 2.2

## User Inputs

### 2.2.5 Digital inputs - Usage.

All of the eight I/O signals can be used as digital inputs. The sampled and possibly filtered value of each input is stored in the Input's register (reg. 18). Unlike the analogue inputs, there is only one value for each digital input, so it must be configured to be either unfiltered or filtered.

Unfiltered (high-speed) digital inputs are sampled every 100  $\mu$ S (micro-seconds).

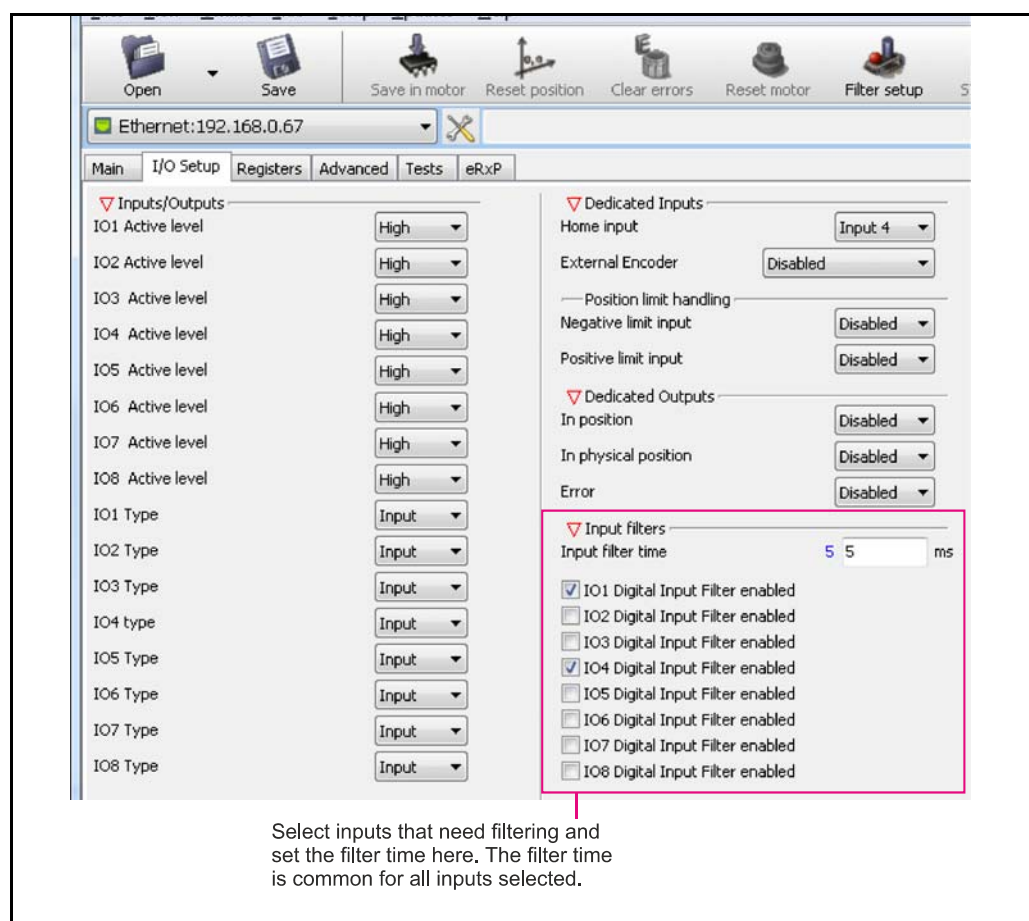
Filtered digital inputs are sampled every milli-second, and the filter value can be set in the range 1 to 100 mS, so the filtered input must be sampled to have the same logical value for that number of samples in a row. Once an input has changed state after passing the filtering, it will again take the same number of samples of the opposite logical level to change it back. For example, if the filter is set to 5 mS and the start value is 0 (zero), the input will remain at zero until three samples in succession have been read as 1 (one). If the signal immediately drops down to 0 again, it will take three samples of zero in succession before the register bit gets set to zero.

Note that enabling filtering of the digital inputs does load the micro-controller, so if filtering of the digital inputs is not needed, ALL the inputs can be selected as high-speed to optimise the available resources from the micro controller.

### 2.2.6 Digital input filter setup with MacTalk:

By default, the digital input filters are disabled and therefore the inputs are sampled every 100  $\mu$ s.

If "IOx Digital Filter enabled" is set, the specific input will use the digital filter according to the "Input filter time". The remaining digital inputs will still be updating every 100  $\mu$ s.



Select inputs that need filtering and set the filter time here. The filter time is common for all inputs selected.

## 2.2

## User Inputs

---

### 2.2.7 Digital input filter setup without MacTalk:

If MacTalk is not used for setting up parameters/registers related to the digital filters it must be done as follows.

The motor contains a number of registers which can be accessed from various protocols depending at which options the motor has.

Protocols available are for example Ethernet (EthernetIP, ProfiNet etc.) and CANopen, Modbus or the MacTalk protocol.

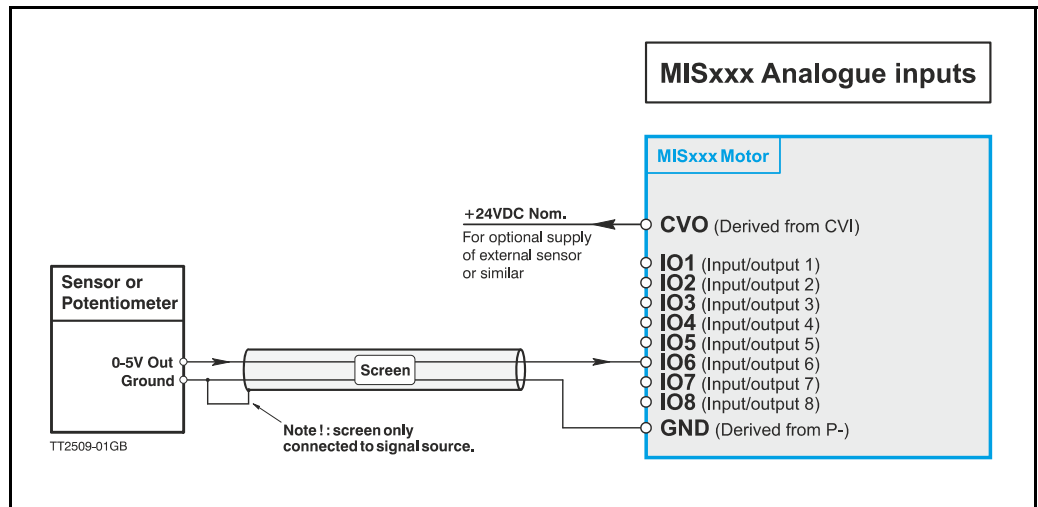
Each field in MacTalk described earlier in this chapter is accessing a register in the motor. The registers that are relevant for digital filters setup are:

**R135** INPUT\_FILTER\_MASK      This register controls filtering of each of the eight I/O pins that are used as digital inputs. If the bit corresponding to the input number is set in this register, the filter will be enabled.  
See also: [Input\\_Filter\\_Mask](#), page 223

**R136** INPUT\_FILTER\_CNT      The filtering of all of the eight digital inputs is controlled by the value in this register together with register 135. The input must be sampled at the same value for the specified number of milliseconds in this register to be accepted as the new filtered usable value. See also [Digital inputs - Usage.](#), page 27  
See also: [Input\\_Filter\\_Cnt](#), page 223

## 2.3

## Analogue Inputs



### 2.3.1

#### General

The 0-5V Analogue Inputs are used for example when the Controller is operated as a stand-alone unit. In this kind of application it can be an advantage to use a potentiometer, joystick or other device for adjusting speed, position, acceleration, etc.

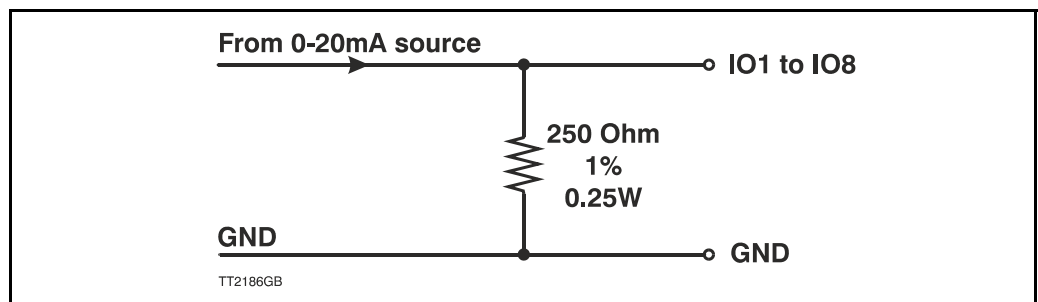
In these modes of operation, the motor is controlled to produce a velocity or position, etc., which is determined by, and proportional to, the voltage applied to the Analogue Input.

The Analogue Inputs share a common internal supply with the GND and P- terminal and are not optically isolated from all other inputs and outputs. The Analogue Inputs are protected against voltage overload up to 30V peak and have a built-in filter which removes input signal noise. See [Analogue input filters](#), page 30.

Always use shielded cable to connect the source used to control an Analogue Input since the motor, etc., can easily interfere with the analogue signal and cause instability.

The Controller is equipped with 8 analogue-to-digital converters (ADC) which convert the detected analogue signal level. The ADCs have a resolution of 12bit.

In order to use the Analogue Inputs as 0-20 mA inputs, a 250  $\Omega$ , 1% resistor must be connected between IO 1-8 and GND.



**Please notice:** The number of available I/O terminals available may differ depending at which motor type and connector configuration you are using. Please consult the chapter [Connector overview for the MIS motors](#), page 41

## 2.3 Analogue Inputs

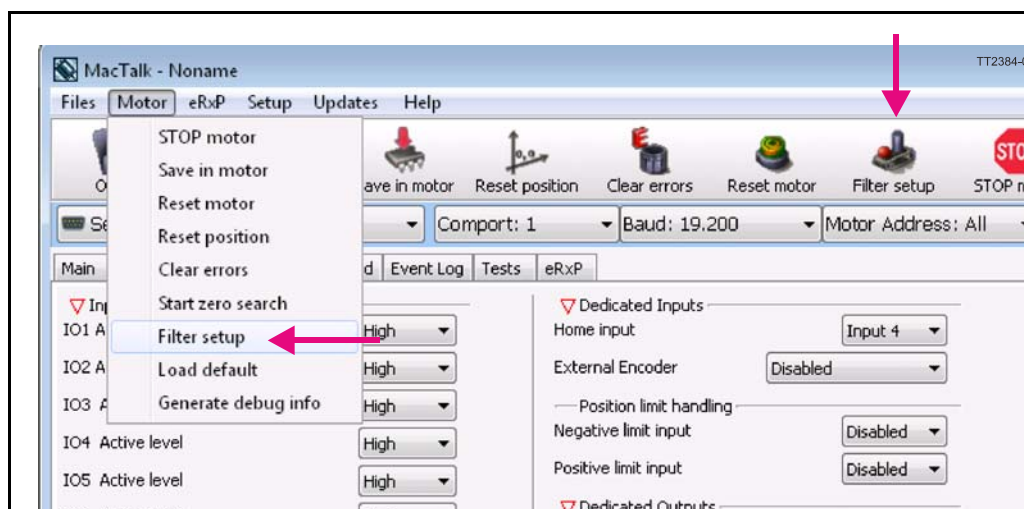
### 2.3.2 Analogue input filters

An analogue signal is not always fully stable and may fluctuate a bit. Also general noise and sudden spikes from other equipment can be a problem.

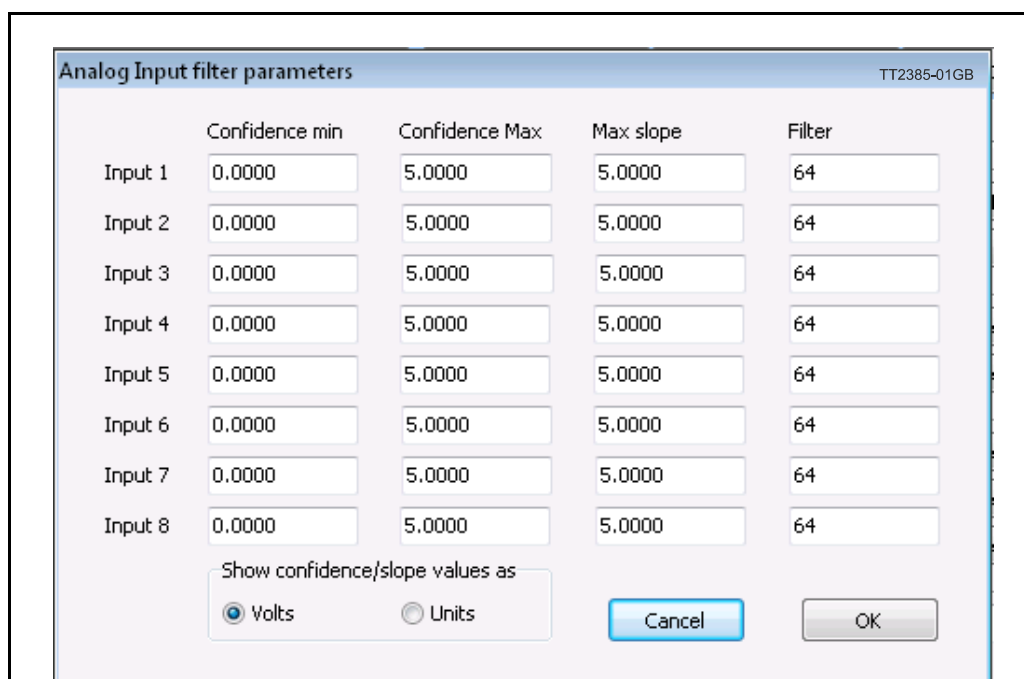
To help filtering an analogue input signal the MIS have an advanced input filter function. This can be setup as follows.

### 2.3.3 Analog filter setup with MacTalk:

It is strongly recommended to setup the analog input filtering using MacTalk. The setup dialog is found in the menu Motor -> Filter setup, or the “Filter setup” button on the toolbar.



Write the settings for each Input and click 'OK'. The parameters can afterwards be “Saved in motor”.



## 2.3

## Analogue Inputs

---

### 2.3.4 Analog filter setup without MacTalk:

If MacTalk is not used for setting up parameters/registers related to the analog filters it must be done as follows.

The motor contains a number of registers which can be accessed from various protocols depending at which options the motor has.

Protocols available are for example Ethernet (EthernetIP, ProfiNet etc.) and CANopen, Modbus or the MacTalk protocol.

Each field in MacTalk described earlier in this chapter is accessing a register in the motor.

The registers that are relevant for analog filters setup are:

#### **R100 Afzup\_WriteBits**

When changing values for the analogue input filter parameters, this register is used in combination with registers 102-106. First, all of the registers 102-106 must be loaded with the values to be used for one or more analogue input filters. Then the lower eight bits in this register are set to select which inputs the parameters in registers 102-106 should control. The firmware will detect this and copy the parameter values from registers 102-106 to internal storage. Once this has been completed, the firmware sets bit 15 in this register to show that registers 102-106 are free to receive new values for programming the remaining inputs with other filter parameters. To use the same filtering for all analogue inputs, this register can be loaded with 255 (hex FF).

#### **R101 Afzup\_ReadIndex**

This register makes it possible to read back the analogue input filter parameters for 1 analogue input at a time. To select a new input, write a value of 1 to 8 to this register and wait for bit 15 to be set high. When bit 15 has been set by the firmware, the registers 102-106 have been loaded with the filter parameters currently used by that analogue input.

These registers acts as setup registers for the analogue filters, but also as the read-out of the actual settings. The setup and read-out procedures are described above.

<b>R102 Afzup_ConfMin</b>	Confidence minimum.
<b>R103 Afzup_ConfMax</b>	Confidence maximum.
<b>R104 Afzup_MaxSlope</b>	Max slope.
<b>R105 Afzup_Filter</b>	Filter.

The result of the filtered analog inputs can be read in the following registers. They are not visible in MacTalk, but can be used in an RXP program or monitored through other interfaces (Ethernet, CANopen, Modbus etc.)

#### **R81-88 Analog\_Filtered**

The voltage on inputs 1 to 8 after being filtered in firmware. See the [Afzup\\_Filter](#), page 215 for filter parameters. 5.00V is equal to a value of 4095.

#### **R89-96 Analog Input**

The unfiltered voltage on inputs 1 to 8. 5.00V is equal to a value of 4095.

## 2.3

## Analogue Inputs

---

### 2.3.5 Detailed description of the analog filter function

The MIS motors have 8 general-purpose I/Os, that can be used as both digital inputs, digital outputs and analogue inputs. When an I/O is configured to be an input, it simultaneously has both a digital value (high or low) and an analogue value in the range 0.00 to 5.00 Volts. Input voltages higher than 5.0 Volts will be internally limited and read as 5.00 Volts.

The inputs use a resolution of 12 bits, which means that in the raw motor units a value of 5.00 Volts reads out as the value 4095.

This gives a resolution of  $5.00/4095 = 1.221$  mV per count.

The eight values from the analogue inputs are maintained by the MIS firmware in the registers 89...96 as raw, unfiltered values with the fastest possible update frequency, and additionally in the registers 81...88 as filtered values. The firmware does not use any of the values for dedicated functions. It is always up to the program in the motor to read and use the values.

The analogue filtered values are typically used to suppress general noise or to define how quickly the input value is allowed to change, or in some cases to limit the input voltage range. A typical example is an analogue input that is connected to a manually controlled potentiometer, so an operator can regulate the speed of the machine by turning a knob. In many environments, this setup is subject to noise, which could make the motor run unevenly, and cause too sharp accelerations or decelerations when the knob is turned.

The filter functions supported in the MIS firmware always use three different steps.

#### Confidence check

First the raw input value is compared to two Confidence limits: Confidence Min and Confidence Max. If the new value is either smaller than the Confidence Min limit or larger than the Confidence Max limit, it is simply discarded (not used at all), and the value in its associated register is unchanged. This is done to eliminate noise spikes. Confidence limits can only be used if not all of the measurement range is used. Values of 0 for Confidence Min and 4095 for Confidence Max will effectively disable the confidence limits.

#### Slope limitation

After a new sample has passed the Confidence limit checks, its value is compared with the last filtered value in its associated register. If the difference between the old and the new value is larger than the Max Slope Limit, the new value is modified to be exactly the old value plus or minus the Max Slope Limit. This limits the speed of change on the signal. Since the samples come at fixed intervals of 10 mS, it is easy to determine the number of Volts per millisecond. A value of 4095 will effectively disable slope limitation.

#### Filtering

After a new sample has both passed the confidence limits checks and has been validated with respect to the slope limitation, it is combined with the last filtered value by taking a part of the new sample and a part of the old filtered value, adding them together and writing the result back to the final destination register – one of the registers 81...88. For instance a filter value of 14 would take 14/64 of the new sample plus 50/64 of the old value. A filter of 64 would simply copy the new sample to the rule, thus disabling the filtering. This completes the filtering of the analogue inputs.

## 2.3

## Analogue Inputs

---

### Confidence alarms

If either of the Confidence Min or Confidence Max limits is used, it may be possible that no new samples are accepted, which means that the filtered value will never change even though there is a change in the input voltage. For instance, if the Confidence Min limit is set to 2.0 V, and the actual input voltage is 1.50 V, the filtered value may continue to read out 0.00 V (or the last value it had before exceeding the confidence limits).

To help troubleshooting in cases like this, each input has a status bit that is set if at least half of the new samples during the last second lie outside either confidence limit. It is not possible to see which of the confidence limits is violated. The status bits are updated once per second.

### Slope alarms

If the Max Slope limit is used (by setting its value lower than 4095), it may be possible that many samples have their value limited. This is not necessarily an error in itself, but can be a sign of a fault causing a noisy signal, or it can be a sign that the Max Slope limit is set too low, which can have implications if the analogue voltage is used to control the motor speed, torque, etc.

To help troubleshooting in cases like this, each input has a status bit that is set if at least half of the new samples during the last second were limited by the Max Slope setting. The status bits are updated once per second.

### Example of analogue input filter operation:

Note that even though the examples use units rather than Volts, decimal values are used, since the motor uses a much higher resolution internally to store the units.

Also note that as long as the slope limitation is in effect, the result will keep a constant slope even when using a filter. When the slope limitation is no longer in effect, the filter will cause the value to approach the final result more slowly as it approaches the result.

Confidence Min = 0, Confidence Max = 500, Max Slope = 10, Filter = 8, Old filtered value = 0.

Sample 1 = 100      Confidence OK, slope limit to 0 + 10 = 10,  
result =  $100 \cdot (8/64) + 0 \cdot (56/64) = 1.25$  units.

Sample 2 = 100      Confidence OK, slope limit to 1.25 + 10 = 11.25,  
result =  $11.25 \cdot (8/64) + 1.25 \cdot (56/64) = 2.5$  units.

Sample 3 = 100      Confidence OK, slope limit to 2.5 + 10 = 12.5,  
result =  $12.5 \cdot (8/64) + 2.5 \cdot (56/64) = 3.75$  units.

Sample 4 = 800      Confidence error, keep old value, result = **3.75** units.

...and so on until the result gets  $\sim$  **95.0** units...

Sample 78 = 100      Confidence OK, no slope limitation needed,  
result =  $100 \cdot (8/64) + 95 \cdot (56/64) = 95.625$  units.

Sample 79 = 100      Confidence OK, no slope limitation needed,  
result =  $100 \cdot (8/64) + 95.625 \cdot (56/64) \sim 96.171875$  units.

Sample 80 = 100      Confidence OK, no slope limitation needed,  
result =  $100 \cdot (8/64) + 96.171875 \cdot (56/64) \sim 96.65$  units.

Sample 81 = 100      Confidence OK, no slope limitation needed,  
result =  $100 \cdot (8/64) + 96.65 \cdot (56/64) \sim 97.07$  units.

## 2.3

## Analogue Inputs

---

Sample 82 = 100    Confidence OK, no slope limitation needed,  
result =  $100 \cdot (8/64) + 97.07 \cdot (56/64) \sim =$  **97.44** units.

Sample 83 = 100    Confidence OK, no slope limitation needed,  
result =  $100 \cdot (8/64) + 97.44 \cdot (56/64) \sim =$  **97.76** units.

... The following samples produce the following results ending up with the input value (100.0).

98.04, 98.28, 98.49, 98.68, 98.85, 99.00, 99.12, 99.23, 99.33, 99.41, 99.48, 99.55, 99.60,  
99.65, 99.70, 99.74, 99.77, 99.80, 99.82, 99.84, 99.86, 99.88, 99.90, 99.91, 99.92, 99.93,  
99.94, 99.95, 99.95, 99.96, 99.96, 99.97, 99.97, 99.98, 99.98, 99.98, 99.98, 99.99, 99.99,  
99.99, ..... 100.0

## 2.4

## User Outputs

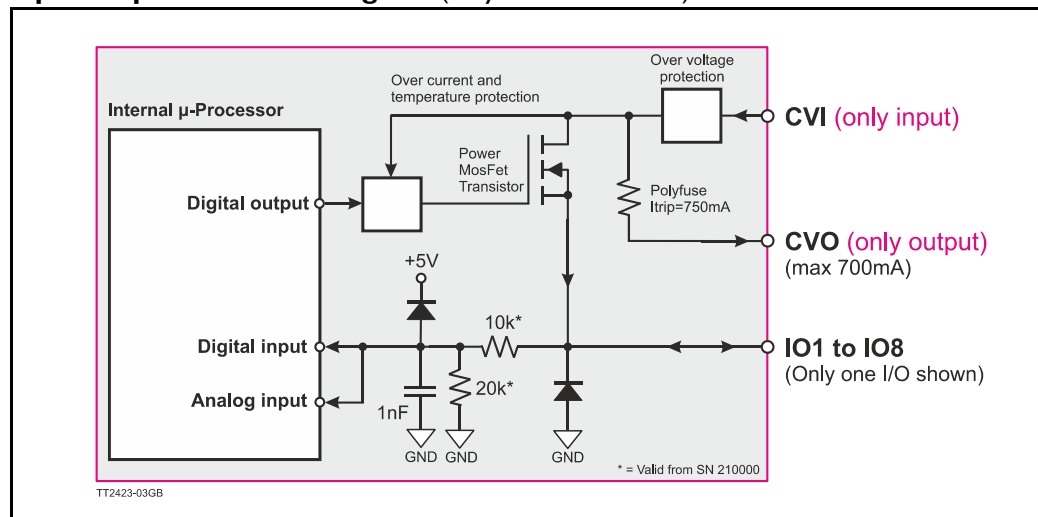
### 2.4.1 User outputs

The MIS motors has 8 inputs/outputs (IO's) that each can be set individually to input, output or analogue input 0-5V via MacTalk or software commands. This means that it for example is possible to have 4 inputs, 3 outputs and one analogue input.



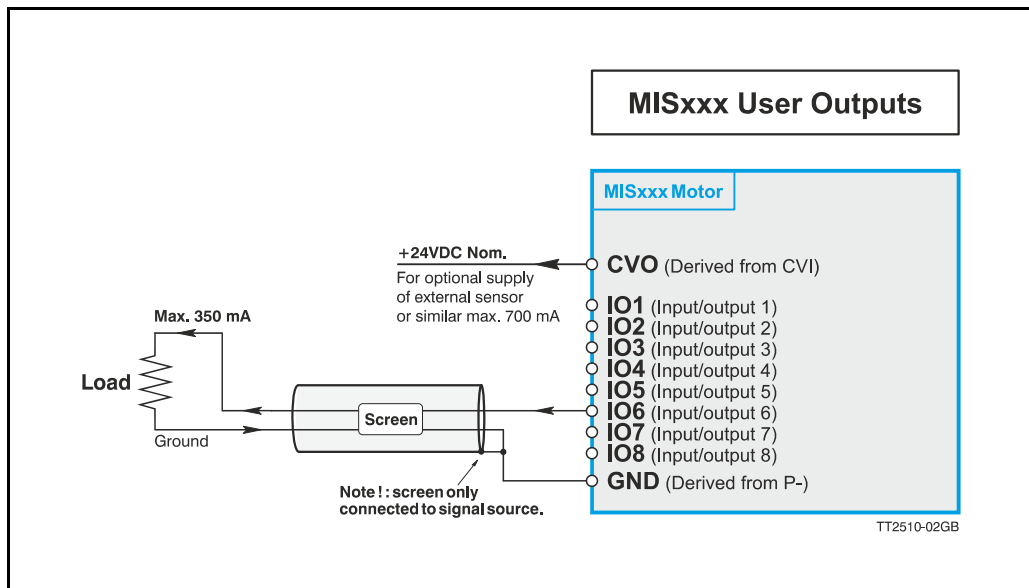
**Please notice:** The number of available I/O terminals available may differ depending at which motor type and connector configuration you are using. Please consult the chapter [Connector overview for the MIS motors](#), page 41

**Input/output functional diagram (only one I/O shown)**



- The Outputs are Source outputs and 7-30VDC compliant
- No galvanic isolation
- Short-circuit to ground protected that shuts down all outputs and sets Error bit in software.
- Optional “In Position” and “Error” signals can be selected to be on any outputs 1 to 8
- Optional Encoder outputs
- 350 mA output current per channel even with all channels fully loaded at the same time.
- Internal ground clamp diodes to protect when inductive load is driven.

## User Outputs



### 2.4.2 General

The Controller is equipped with a total of 8 digital outputs. Each output can be used for a variety of purposes depending on the Controller's basic mode of operation. The Outputs are not galvanically isolated from other circuitry in the motor.

The output circuitry is powered from the control voltage supply terminal CVI.

See also *Control Voltage (CVI terminal)*, page 19.

The output circuitry operates with voltages in the range 7-30 VDC.

Each output can supply a continuous current up to 350 mA (max).

The outputs are all source drivers, i.e. if a given output is activated, contact is made between the control voltage (CVI) and the respective output terminal. See above illustration.

### 2.4.3 Overload of User Outputs

All of the outputs are short-circuit protected, which means that the program and the motor is stopped and the output is automatically disconnected in the event of a short circuit. The output will first function normally again when the short-circuit has been removed.

**Note:** Do not connect a voltage greater than 30 VDC to the CVI terminal as the output circuitry may be seriously damaged and the unit will require factory repair.

If one or more outputs are short circuited, MacTalk will show Error “Output Driver” and Bit 2 will be set in Err\_Bits.

See also [Err\\_Bits](#), page 208.

## 2.5 Serial interfaces overview

---

### 2.5.1 Serial interfaces

The Controller has 2 serial interfaces:

- RS485 (Dual channel A and B) balanced for up to 32 units in multi-axis applications and Modbus communication. (Standard)
- CANbus - CANopen DS-301. Fully ISO 11898-2:2016 compliant
- CANbus - CANopen DSP-402 is in development but not available now.

CANbus and RS485 can be used at the same time.



**Please notice:** The number of available I/O terminals available may differ depending at which motor type and connector configuration you are using. Please consult the chapter [Connector overview for the MIS motors](#), page 41

## 2.6

## RS485 Interface

### 2.6.1 RS485 - General description when using a QuickStep motor

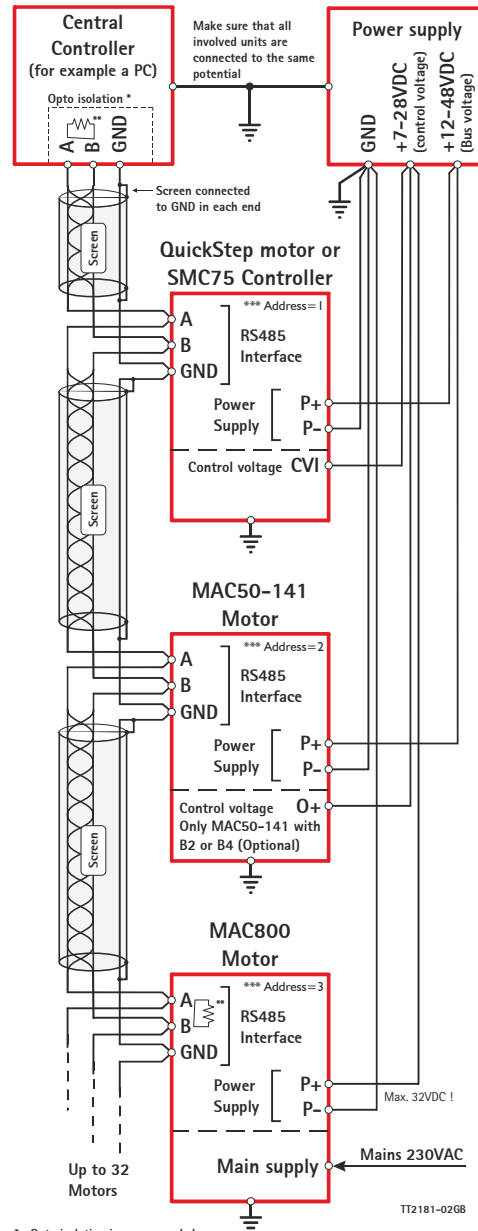
The RS485 interface offers more noise immune communication compared to a USB or RS232 interface. Up to 32 motors can be connected to the same interface bus.

When connecting the RS485 interface to a central controller, the following rules must be followed:

- 1 Use twisted pair cable.
- 2 Use shielded cable.
- 3 Make sure that the GND is also connected.
- 4 Ensure that all units have a proper connection to safety ground (earth) in order to refer to the same potential.
- 5 The last unit in each end of the network must be terminated with a 120 Ohm resistor between A and B.
- 6 Ensure that the supply lines are made individually in order to reduce the voltage drop between the motors.
- 7 Central Controller RS485 interface:  
If available, it is strongly recommended a type with optical isolation is used.

The default configuration:  
Data bits = 8  
Baud rate = 19200  
Stop bit = 1  
Parity = None

RS485 network with 1 x QuickStep, 1 x MAC140 and 1 x MAC800 mounted with MAC00-B1, B2 or B4 modules.



\* Opto isolation is recommended.

\*\* The last unit in each end of the line must be terminated. The MAC00-B1, B2 and B4 contain this feature. See the individual module descriptions. The QuickStep motor does not have a resistor built-in, the resistor has to be mounted externally, for instance in the M12 connector.

\*\*\* Each unit connected must be setup with an address via The MacTalk program. If only one unit is connected no address is needed.

## 2.7

## EMC considerations

---

### 2.7.1 EMC considerations

The MIS family of motors eliminates the traditional problems with noise from long motor cables that emit noise and feedback cables that are sensitive to noise from external sources.

However, it is still necessary to be aware of noise problems with communications cables and the 8 general-purpose inputs and outputs.

Whenever a digital signal changes level quickly, a noise spike is generated, and is transferred to the other wires in the same cable, and to a lesser degree to wires in other cables located close to the cable with the switching signal. A typical example is when a digital output from the MIS motor changes from low to high to drive a relay. If this digital output signal is transmitted in a multi-wire cable together with the RS-485 signals, there is a high risk that the RS-485 signal will be affected to the extent that the communication will fail, and require software retries.

If communication is used during operation, and operation includes either digital input signals or digital output signals, some precautions must be taken to avoid noise problems. The following sections describe a number of measures which can be taken to solve noise problems. In most installations, no special measures will be required, but if noise problems are experienced – and/or must be avoided – it is highly recommended the instructions below are followed.

### 2.7.2 Use short cables

The shorter a cable is, the less noise problems it will induce. Be sure to keep the cables as short as possible. Instead of curling up the cables, cut them off at the minimum required length.

### 2.7.3 Use separate cables

Avoid running digital signals in the same multi-wire cables as RS-485 communication signals.

On some models of the MIS motors, the same connector contains both RS-485 signals and I/O signals – typically the I/Os 1-4.

In many applications, far from all inputs and outputs are used. If only up to four I/Os are required, consider using only I/Os 5-8 which are typically available via another connector on the motor.

### 2.7.4 Use filters

If more than 4 I/Os are needed, consider using I/Os 1-4 for inputs and I/Os 5-8 for outputs. It is normally possible to install a hardware filter on the digital input signals before they enter the cable. With such a (good) filter, noise on the RS-485 signals will not be a problem.

It is also possible to use filters on the outputs, but it is more difficult. It can be done by using short cables from the motor to the filters, and then using longer cables from the filters to the output targets. It may be easier to use a short cable from the motor to a splitter box, and then split the I/Os in one cable and the RS-485 signals in another cable.

### 2.7.5 Use termination (resistors) on the RS-485 signals

RS-485 is typically used to connect a single master PC or PLC to one or more motors in a chain. Both ends of the chain must have a 120 Ohms termination resistor connected between the A- and B+ signals. There is typically a terminating resistor in the master PC or PLC, but there is no termination inside the motors. Therefore an external resistor must be connected at the end of the cable out of the last motor in the chain. If the last motor has no connection cable, a connector with a resistor soldered between the A- and B+ pins should be used.

## 2.7

## EMC considerations

---

As an alternative, a connector with a short cable can be used with the resistor soldered between the two wires carrying A- and B+.

Use individually shielded cables.

In some installations, it will be necessary to have RS-485 signals in the same multi-wire cables as fast-switching digital signals. In addition to keeping cable lengths to a minimum and using termination resistors, high-quality cables, where each wire is shielded from the other wires in the cable, should be used. This is typically done using a metal foil wrapped around each wire. These types of cables are more expensive, but the overall cost and noise immunity requirements may justify the solution instead of splitting cables.

### 2.7.6 Use simple shielding

Using cables with only a single shield shared by all the signal wires will also improve noise problems to some degree, but will not guarantee completely stable operation for mixed signal cables. If a cable carries only RS-485 or only digital I/O, this simple and inexpensive form of shielding is recommended.

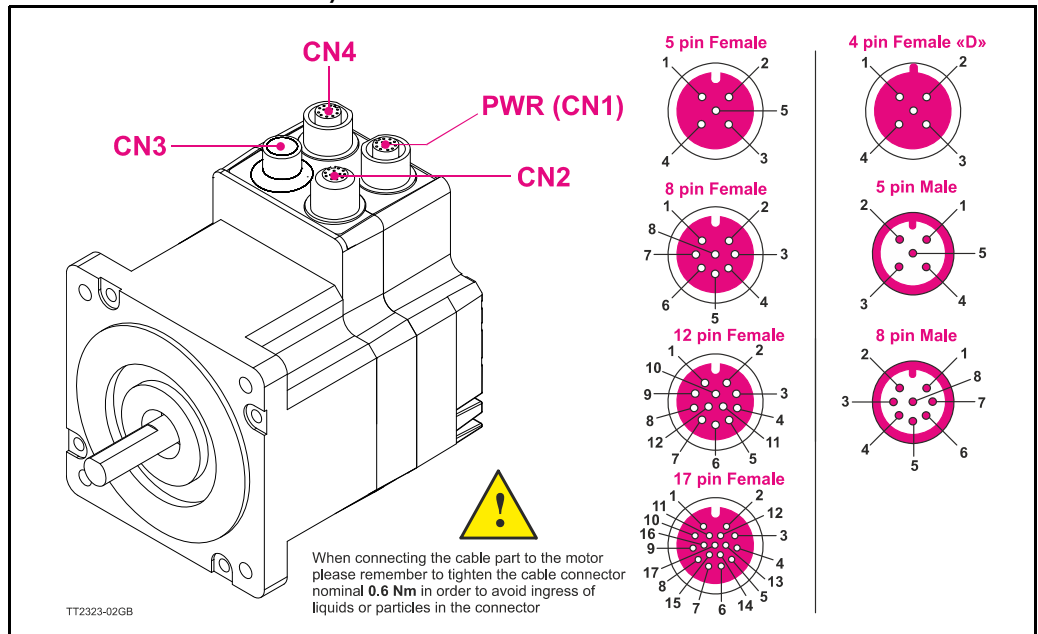
## 2.8

## How to connect a MIS motor

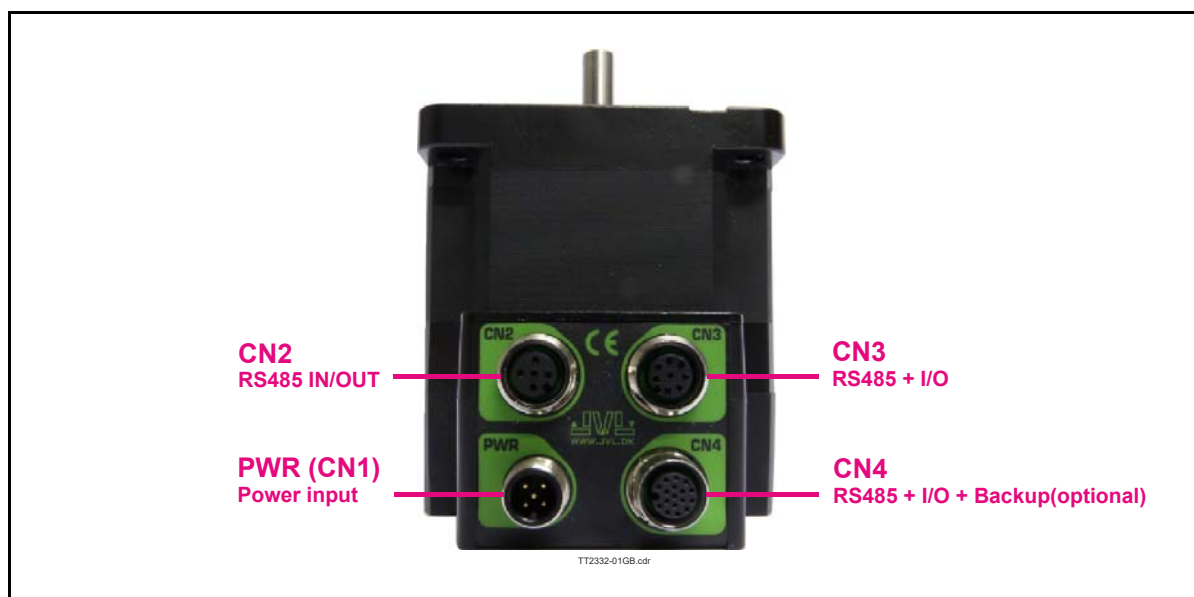
### 2.8.1 Connector overview for the MIS motors

QUICKSTEP Connector Overview	Power Male 5 Pin	IO1-8, RS485, MFIO Female 17 Pin	RS485 Female 5 Pin	RS485 + IO1-4 Female 8 Pin	RS485 + IO1-4 Female 8 Pin	CANopen Female 5 Pin	SSI Encoder Male 8 Pin	Profibus Male 5 Pin	Ethernet Female 4 Pin
Connector ID	PWR (CN1)	CN4	CN2	CN3	CN2	CN2 & CN3	CN3	CN2 & CN3	CN2 & CN3
MISxxxnyyQ5zz85 (8IOA) Preferred type	x	x	x	x					
MISxxxnyyP6zz85 (CANopen)	x	x				x			
MISxxxnyyQ9zz85 (SSI input)	x	x			x		x		
MISxxxnyyExzz85 (Ethernet)	x	x							x
MISxxxnyyFBzz85 (Bluetooth)	x	x		x	x				
MISxxxnyyFPzz85 (Profibus)	x	x						x	
M12 Pin1	P+ (12-72VDC)	IO1	B0- (RS485)	IO1	IO1	CAN_SHLD	IO5 Zero Set	5VDC	TX0_P
M12 Pin2	P+ (12-72VDC)	GND	A0+ (RS485)	IO2	IO2	Unused	IO6 CNTDIR	A-	RX0_P
M12 Pin3	P- (GND)	IO2	B0- (RS485)	IO3	IO3	CAN_GND	A1+ (Clock+)	DGND	TX0_N
M12 Pin4	CVI (12-30VDC)	IO3	A0+ (RS485)	GND	GND	CAN_H	GND	B+	RX0_N
M12 Pin5	P- (GND)	B1- (RS422)	GND	B0- (RS485)	B0- (RS485)	CAN_L	B1- (Data In-)	SHIELD	-
M12 Pin6	-	IO4	-	A0+ (RS485)	A0+ (RS485)	-	B1+ (Data In+)	-	-
M12 Pin7	-	A1- (RS422)	-	IO4	IO4	-	A1- (Clock -)	-	-
M12 Pin8	-	B1+ (RS422)	-	CVO (out)	CVO (out)	-	CVO (out)	-	-
M12 Pin9	-	CVO (out)	-	-	-	-	-	-	-
M12 Pin10	-	A1+ (RS422)	-	-	-	-	-	-	-
M12 Pin11	-	IO5	-	-	-	-	-	-	-
M12 Pin12	-	IO6	-	-	-	-	-	-	-
M12 Pin13	-	IO7	-	-	-	-	-	-	-
M12 Pin14	-	IO8	-	-	-	-	-	-	-
M12 Pin15	-	B0- (RS485)	-	-	-	-	-	-	-
M12 Pin16	-	GND	-	-	-	-	-	-	-
M12 Pin17	-	A0+ (RS485)	-	-	-	-	-	-	-
M12 Connector solder terminals	WI1008-M12F5SS1	(not available)	WI1008-M12M5SS1	WI1008-M12M8SS1	WI1008-M12M8SS1	WI1008-M12M5SS1	WI1008-M12F8SS1	WI1028-M12F5SS1	(not available)
M12 Cables 5m	WI1000-M12F5T05N	WI1009-M12M17T05N	WI1005-M12M8V M5V03N	WI1000-M12M8T05N	WI1000-M12M8T05N	WI1006-M12F5 TM5T05N	WI1000-M12F8T05N	WI1026-M12-F5S0R	WI1046-M12M4S05R

Connector layout - The shown motor is a MIS34x motor but the connector locations are the same at other MIS family members with radial standard connectors.



## 2.8 How to connect a MIS motor



### 2.8.2 MISxxxxxx**Q5**xxxx connector description.

The MIS motors offers robust M12 connectors which makes it ideal for automation applications. The M12 connectors offer solid mechanical protection and are easy operate. Following scheme gives the relevant information about each connector and the pins, wire colours and a short description of the signals available.

The connector layout:

“PWR” (CN1) - Power input. M12 - 5pin male connector				
Signal name	Description	Pin no.	JVL Cable WI1000- M12F5TxxN	Isolation group
P+	Main supply +7-72VDC. Connect with pin 2 *	1	Brown	1
P+	Main supply +7-72VDC. Connect with pin 1 *	2	White	1
P-	Main supply ground. Connect with pin 5 *	3	Blue	1
CVI	Control and user output supply +7-30VDC. <b>DO NOT connect &gt;30V to this terminal!</b>	4	Black	1
P-	Main supply ground. Connect with pin 3 *	5	Grey	1
* Note: P+ and P- are each available at 2 terminals. Make sure that both terminals are connected in order to split the supply current in 2 terminals and thereby avoid an overload of the connector.				



**Please notice:** When connecting the a cable to one of the M12 connectors it must be tightened with 0.6 Nm to prevent ingress of fluids and/or particles in the connector which can cause malfunction.

(Continued next page)

## 2.8

## How to connect a MIS motor

“CN2” - RS485 IN/OUT. M12 - 5pin female connector.				
Signal name	Description	Pin no.	JVL Cable WI1000-M12 M5TxxN	Isolation group (See note)
RS485: B0-	RS485 interface. Leave open if unused	1	Brown	1
RS485: A0+	RS485 interface. Leave open if unused	2	White	1
RS485: B0-	RS485 interface. Leave open if unused	3	Blue	1
RS485: A0+	RS485 interface. Leave open if unused	4	Black	1
GND	Ground intended to be used together with the other signals in this connector	5	Grey	1
“CN3” - RS485 + I/O connector - M12 - 8pin female connector.				
Signal name	Description	Pin no.	JVL Cable WI1000-M12 M8TxxN	Isolation group (See note)
IO1	I/O channel 1. Can be used as input or output	1	White	1
IO2	I/O channel 2. Can be used as input or output	2	Brown	1
IO3	I/O channel 3. Can be used as input or output	3	Green	1
GND	Ground intended to be used together with the other signals in this connector	4	Yellow	1
RS485: B0-	RS485 interface. Leave open if unused	5	Grey	1
RS485: A0+	RS485 interface. Leave open if unused	6	Pink	1
IO4	I/O channel 4. Can be used as input or output	7	Blue	1
CVO	Supply output. Connected internally to the <b>CVI</b> terminal in the PWR connector. Max 700 mA. <b>DO NOT connect &gt;30V to this terminal!</b> USB interface. Supply input 5VDC nominal	8	Red	1
“CN4” - RS485 + I/O + Backup (option) connector - M12 - 17pin female connector				
Signal name	Description	Pin no.	JVL Cable WI1009M12 M17TxxN	Isolation group (see note)
IO1	I/O channel 1. Can be used as input or output	1	Brown	1
GND	Ground intended to be used together with the other signals in this connector	2	Blue	1
IO2	I/O channel 2. Can be used as input or output	3	White	1
IO3	I/O channel 3. Can be used as input or output	4	Green	1
RS422: B1-	RS422 I/O terminal B-	5	Pink	1
IO4	I/O channel 4. Can be used as input or output	6	Yellow	1
RS422: A1-	RS422 I/O terminal A-	7	Black	1
RS422: B1+	RS422 I/O terminal B+	8	Grey	1
CVO	Supply output. Connected internally to the <b>CVI</b> terminal in the PWR connector. <b>DO NOT connect &gt;30V to this terminal!</b>	9	Red	1
RS422: A1+	RS422 I/O terminal A+	10	Violet	1
IO5	I/O channel 5. Can be used as input or output	11	Grey/pink	1
IO6	I/O channel 6. Can be used as input or output	12	Red/blue	1
IO7	I/O channel 7. Can be used as input or output	13	White/Green	1
IO8	I/O channel 8. Can be used as input or output	14	Brown/Green	1
RS485: B0-	RS485 interface. Leave open if unused	15	White/Yellow	1
EXTBACKUP	Only for motors with the -H3 or -H4 option (abs. multi-turn encoder). This terminal can be connected to an external supply. <b>Connect to ground (GND) if not used.</b>	16	Yellow/brown	1
RS485: A0+	RS485 interface. Leave open if unused	17	White/grey	1
* Note: Isolation group indicate which terminals/circuits that a galvanic connected to each other. In other words group 1, 2, 3 and 4 are all fully independently isolated from each other. Group 1 correspond to the housing of the motor which may also be connected to earth via the DC or AC input supply.				

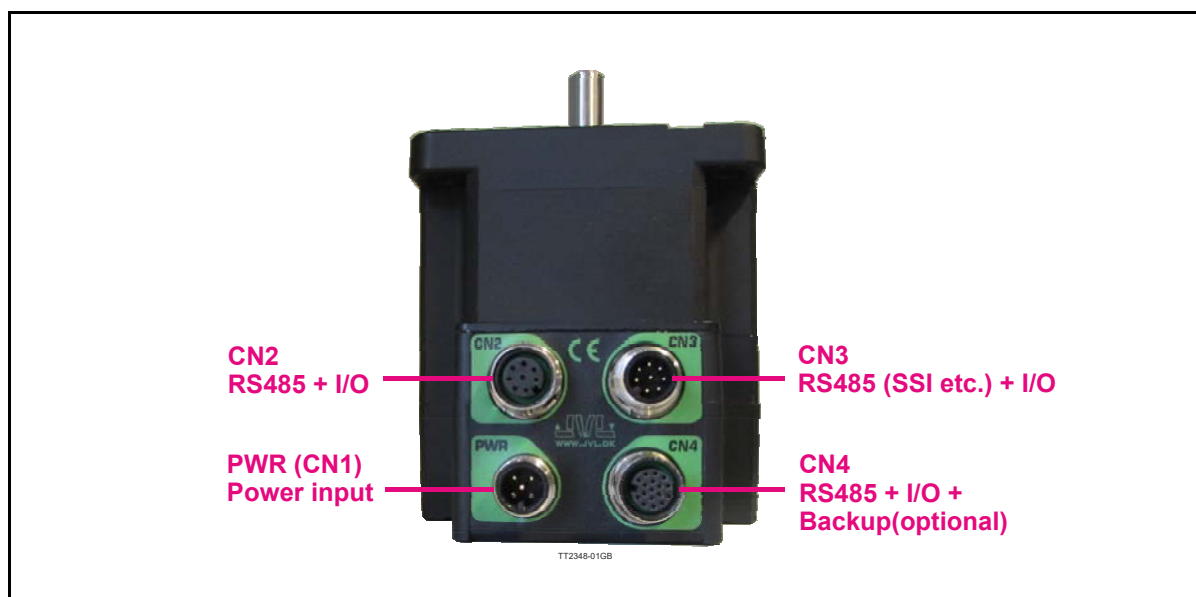
## 2.8 How to connect a MIS motor

### 2.8.3 Cables for the MISxxxxxx**Q5**xxxx

The following cables equipped with M12 connector can be supplied by JVL.

MISxxx Connectors				Description	JVL Order no.	Picture
"PWR" 5pin Male	"CN2" 5pin Female	"CN3" 8pin Female	"CN4" 17pin Female			
	X			RS485 Interface cable. Connects directly from the MIS motor to a RS485 comport. Length: 5m (197 inch)	RS485-M12-1-5-5	
		X		RS485 Interface cable. Connects directly from the MIS motor to a RS485 comport. Length: 5m (197 inch)	RS485-M12-1-5-8	
Not relevant. The RS485-USB-ATC-820 connect to CN2 through cable type RS485-M12-1-5-8				USB to RS485 Converter. To be used if no RS485 COM port is available.	RS485-USB-ATC-820	
	X			Cable (Ø5.5mm) with M12 <b>male</b> 5-pin connector loose wire ends 0.35mm <sup>2</sup> (22AWG) and foil screen. Length: 5m (197 inch)	WI1000-M12M5T05N	
	X			Same as above but 20m (787 inch)	WI1000-M12M5T20N	
		X		Cable with M12 <b>male</b> 8-pin connector loose wire ends 0.35mm <sup>2</sup> (22AWG) and screen.	WI1000-M12M8T05N	
		X		Same as above but 20m (787 inch)	WI1000-M12M8T20N	
			X	Cable with M12 <b>male</b> 17-pin connector loose wire ends 0.22mm <sup>2</sup> (24AWG) and screen. Length: 5m (197 inch)	WI1009-M12M17T05N	
			X	Same as above but 20m (787 inch)	WI1009-M12M17T20N	
			X	Junction box for splitting the 17 pin I/O connector into 4 independent connectors. Include also 9 LED's for monitoring the I/O status and communication. Cable length: 0,5m (20 inch)	PA0190	
<b>Protection caps. Optional if connector is not used to protect from dust / liquids.</b>						
	X	X	X	IP67 protection cap for M12 <b>female</b> connector.	WI1000-M12FCAP1	
X				IP67 protection cap for M12 <b>male</b> connector.	WI1000-M12MCAP1	

## 2.8 How to connect a MIS motor



### 2.8.4 MISxxxxxx**Q9**xxxx connector description.

The MIS motors offers robust M12 connectors which makes it ideal for automation applications. The M12 connectors offer solid mechanical protection and are easy to operate.

The following scheme gives the relevant information about each connector and the pins, wire colours and a short description of the signals available.

The connector layout:

“PWR” (CN1) - Power input. M12 - 5pin male connector				
Signal name	Description	Pin no.	JVL Cable WI1000- M12F5TxN	Isolation group
P+	Main supply +7-72VDC. Connect with pin 2 *	1	Brown	1
P+	Main supply +7-72VDC. Connect with pin 1 *	2	White	1
P-	Main supply ground. Connect with pin 5 *	3	Blue	1
CVI	Control and user output supply +7-30VDC. <b>DO NOT connect &gt;30V to this terminal!</b>	4	Black	1
P-	Main supply ground. Connect with pin 3 *	5	Grey	1

\* Note: P+ and P- are each available at 2 terminals. Make sure that both terminals are connected in order to split the supply current in 2 terminals and thereby avoid an overload of the connector.



**Please notice:** When connecting the a cable to one of the M12 connectors it must be tightened with 0.6 Nm to prevent ingress of fluids and/or particles in the connector which can cause malfunction.

(Continued next page)

## 2.8 How to connect a MIS motor

<b>“CN2” - RS485 + I/O connector - M12 - 8pin female connector.</b>				
Signal name	Description	Pin no.	JVL Cable WI1000-M12 M8TxxN	Isolation group (See note)
IO1	I/O channel 1. Can be used as input or output	1	White	1
IO2	I/O channel 2. Can be used as input or output	2	Brown	1
IO3	I/O channel 3. Can be used as input or output	3	Green	1
GND	Ground intended to be used together with the other signals in this connector	4	Yellow	1
RS485: B0-	RS485 interface. Leave open if unused	5	Grey	1
RS485: A0+	RS485 interface. Leave open if unused	6	Pink	1
IO4	I/O channel 4. Can be used as input or output	7	Blue	1
CVO	Supply output. Connected internally to the <b>CVI</b> terminal in the PWR connector. Max 700 mA	8	Red	1
<b>“CN3” - RS485 (SSI etc.) + I/O. M12 - 8pin Male connector.</b>				
Signal name	Description	Pin no.	JVL Cable WI1000-M12 F8TxxN	Isolation group (See note)
IO5	Used for zero set. Leave open if unused	1	White	1
IO6	Counting direction. Leave open if unused	2	Brown	1
RS422: A1+	SSI Clock +. Leave open if unused	3	Green	1
GND	Signal ground. Leave open if unused	4	Yellow	1
RS422: B1-	SSI Data in -. Leave open if unused	5	Grey	1
RS422: B1+	SSI Data in +. Leave open if unused	6	Pink	1
RS422: A1-	SSI Clock -. Leave open if unused	7	Blue	1
CVO	Supply output. Connected internally to the <b>CVI</b> terminal in the PWR connector. Max 700 mA	8	Red	1
<b>“CN4” - RS485 + I/O + Backup(option) connector - M12 - 17pin female connector</b>				
Signal name	Description	Pin no.	JVL Cable WI1009M12 M17TxxN	Isolation group (see note)
IO1	I/O channel 1. Can be used as input or output	1	Brown	1
GND	Ground intended to be used together with the other signals in this connector	2	Blue	1
IO2	I/O channel 2. Can be used as input or output	3	White	1
IO3	I/O channel 3. Can be used as input or output	4	Green	1
RS422: B1-	RS422 I/O terminal B-	5	Pink	1
IO4	I/O channel 4. Can be used as input or output	6	Yellow	1
RS422: A1-	RS422 I/O terminal A-	7	Black	1
RS422: B1+	RS422 I/O terminal B+	8	Grey	1
CVO	Supply output. Connected internally to the <b>CVI</b> terminal in the PWR connector. Max 700 mA	9	Red	1
RS422: A1+	RS422 I/O terminal A+	10	Violet	1
IO5	I/O channel 5. Can be used as input or output	11	Grey/pink	1
IO6	I/O channel 6. Can be used as input or output	12	Red/blue	1
IO7	I/O channel 7. Can be used as input or output	13	White/Green	1
IO8	I/O channel 8. Can be used as input or output	14	Brown/Green	1
RS485: B0-	RS485 interface. Leave open if unused	15	White/Yellow	1
EXTBACKUP	Only for motors with the -H3 or -H4 option (abs. multiturn encoder). This terminal can be connected to an external supply. <b>Connect to ground (GND) if not used.</b>	16	Yellow/brown	1
RS485: A0+	RS485 interface. Leave open if unused	17	White/grey	1
* Note: Isolation group indicate which terminals/circuits that a galvanic connected to each other. In other words group 1, 2, 3 and 4 are all fully independently isolated from each other. Group 1 correspond to the housing of the motor which may also be connected to earth via the DC or AC input supply.				

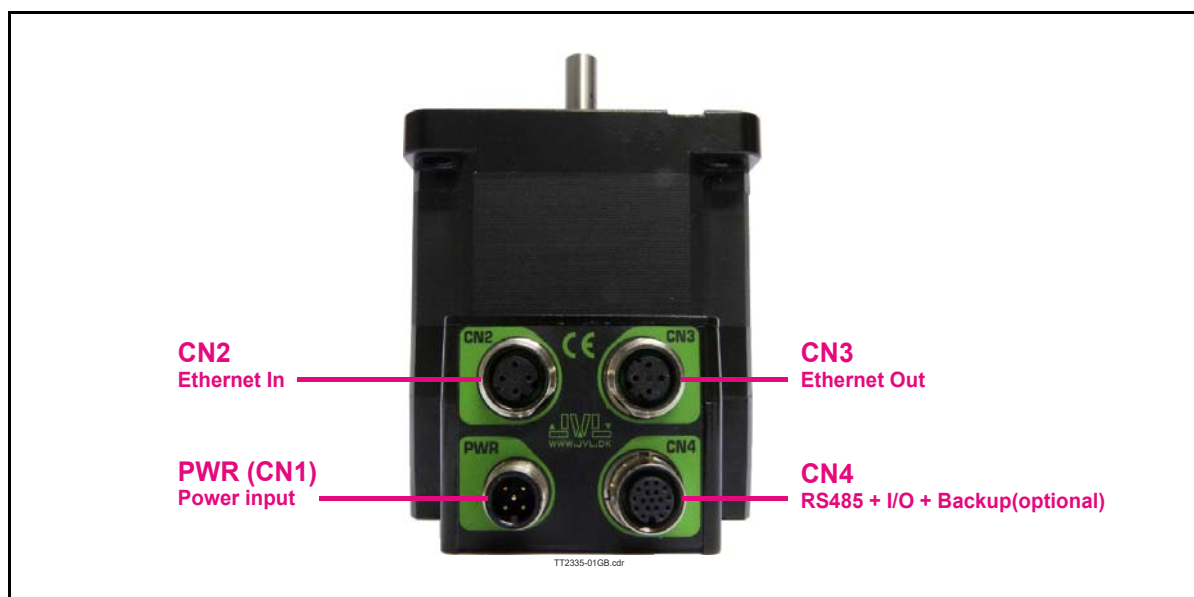
## 2.8 How to connect a MIS motor

### 2.8.5 Cables for the MISxxxxxx**Q9**xxxx

The following cables equipped with M12 connector can be supplied by JVL.

MISxxx Connectors				Description	JVL Order no.	Picture
"PWR" 5pin Male	"CN2" 8pin Female	"CN3" 8pin Male	"CN4" 17pin Female			
X				Cable (Ø5.5mm) with M12 <b>female</b> 5-pin connector loose wire ends 0.35mm <sup>2</sup> (22AWG) and foil screen. Length: 5m (197 inch)	WI1000-M12F5T05N	
	X			RS485 Interface cable. Connects directly from the MIS motor to a RS485 comport. Length: 5m (197 inch)	RS485-M12-1-5-8	
Not relevant. The RS485-USB-ATC-820 connect to CN2 through cable type RS485-M12-1-5-8				USB to RS485 Converter. To be used if no RS485 COM port is available. Use also RS485-M12-1-5-8	RS485-USB-ATC-820	
	X			Cable (Ø5.5mm) with M12 <b>male</b> 8-pin connector loose wire ends 0.35mm <sup>2</sup> (22AWG) and foil screen. Length: 5m (197 inch)	WI1000-M12M8T05N	
	X			Same as above but 20m (787 inch)	WI1000-M12M8T20N	
		X		Cable with M12 <b>female</b> 8-pin connector loose wire ends 0.22mm <sup>2</sup> (24AWG) and screen. Length: 5m (197 inch)	WI1000-M12F8T05N	
		X		Same as above but 20m (787 inch)	WI1000-M12F8T20N	
			X	Cable with M12 <b>male</b> 17-pin connector loose wire ends 0.22mm <sup>2</sup> (24AWG) and screen. Length: 5m (197 inch)	WI1009-M12M17T05N	
			X	Same as above but 20m (787 inch)	WI1009-M12M17T20N	
			X	Junction box for splitting the 17 pin I/O connector into 4 independent connectors. Include also 9 LED's for monitoring the I/O status and communication. Cable length: 0,5m (20 inch)	PA0190	
<b>Protection caps. Optional if connector is not used to protect from dust / liquids.</b>						
	X		X	IP67 protection cap for M12 <b>female</b> connector.	WI1000-M12FCAP1	
X		X		IP67 protection cap for M12 <b>male</b> connector.	WI1000-M12MCAP1	

## 2.8 How to connect a MIS motor



### 2.8.6 MISxxxxxxExxxxx connector description.

Hardware wise all the MIS motors with the Ethernet option are equal and offer the connectivity shown in the table below.

The following Ethernet protocols are supported in this moment:

MISxxxxxxEPxxxx : ProfiNet / MISxxxxxxEIxxxx : EtherNet/IP /  
 MISxxxxxxECxxxx : EtherCAT / MISxxxxxxELxxxx : Ethernet POWERLINK  
 MISxxxxxxEMxxxx : Modbus TCP / MISxxxxxxESxxxx : Sercos III

The MIS motors offers robust M12 connectors which makes it ideal for automation applications. The M12 connectors offer solid mechanical protection and are easy operate. Following scheme gives the relevant information about each connector and the pins, wire colours and a short description of the signals available.

The connector layout:

“PWR” (CN1) - Power input. M12 - 5pin male connector				
Signal name	Description	Pin no.	JVL Cable WI1000-M12F5TxxN	Isolation group
P+	Main supply +7-72VDC. Connect with pin 2 *	1	Brown	1
P+	Main supply +7-72VDC. Connect with pin 1 *	2	White	1
P-	Main supply ground. Connect with pin 5 *	3	Blue	1
CVI	Control and user output supply +7-30VDC. <b>DO NOT connect &gt;30V to this terminal !</b>	4	Black	1
P-	Main supply ground. Connect with pin 3 *	5	Grey	1
* Note: P+ and P- are each available at 2 terminals. Make sure that both terminals are connected in order to split the supply current in 2 terminals and thereby avoid an overload of the connector.				



**Please notice:** When connecting the a cable to one of the M12 connectors it must be tightened with 0.6 Nm to prevent ingress of fluids and/or particles in the connector which can cause malfunction.

(Continued next page)

## 2.8 How to connect a MIS motor

<b>“CN2” - Ethernet In port connector - M12 - 4pin female connector “D” coded</b>				
Signal name	Description	Pin no.	JVL Cable WI1046-M12M4S05R	Isolation group (See note)
Tx0_P	Ethernet Transmit channel 0 - positive terminal	1	-	2
Rx0_P	Ethernet Receive channel 0 - positive terminal	2	-	2
Tx0_N	Ethernet Transmit channel 0 - negative terminal	3	-	2
Rx0_N	Ethernet Receive channel 0 - negative terminal	4	-	2
<b>“CN3” - Ethernet Out port connector. M12 - 4 pin female connector “D” coded</b>				
Signal name	Description	Pin no.	JVL Cable WI1046-M12M4S05R	Isolation group (see note)
Tx1_P	Ethernet Transmit channel 1 - positive terminal	1	-	3
Rx1_P	Ethernet Receive channel 1 - positive terminal	2	-	3
Tx1_N	Ethernet Transmit channel 1 - negative terminal	3	-	3
Rx1_N	Ethernet Receive channel 1 - negative terminal	4	-	3
<b>“CN4” - RS485 + I/O + Backup (option) connector - M12 - 17pin female connector</b>				
Signal name	Description	Pin no.	JVL Cable WI1009M12 M17TxxN	Isolation group (see note)
IO1	I/O channel 1. Can be used as input or output	1	Brown	1
GND	Ground intended to be used together with the other signals in this connector	2	Blue	1
IO2	I/O channel 2. Can be used as input or output	3	White	1
IO3	I/O channel 3. Can be used as input or output	4	Green	1
RS422: B1-	RS422 I/O terminal B-	5	Pink	1
IO4	I/O channel 4. Can be used as input or output	6	Yellow	1
RS422: A1-	RS422 I/O terminal A-	7	Black	1
RS422: B1+	RS422 I/O terminal B+	8	Grey	1
CVO	Supply output. Connected internally to the CVI terminal in the PWR connector. Max 700 mA	9	Red	1
RS422: A1+	RS422 I/O terminal A+	10	Violet	1
IO5	I/O channel 5. Can be used as input or output	11	Grey/Pink	1
IO6	I/O channel 6. Can be used as input or output	12	Red/Blue	1
IO7	I/O channel 7. Can be used as input or output	13	White/Green	1
IO8	I/O channel 8. Can be used as input or output	14	Brown/Green	1
RS485: B0-	RS485 interface. Leave open if unused	15	White/Yellow	1
EXTBACKUP	Only for motors with the -H3 or -H4 option (abs. multiturn encoder). This terminal can be connected to an external supply. <b>Connect to ground (GND) if not used.</b>	16	Yellow/Brown	1
RS485: A0+	RS485 interface. Leave open if unused	17	White/Grey	1
* Note: Isolation group indicate which terminals/circuits that a galvanic connected to each other. In other words group 1, 2, 3 and 4 are all fully independently isolated from each other. Group 1 correspond to the housing of the motor which may also be connected to earth via the DC or AC input supply.				

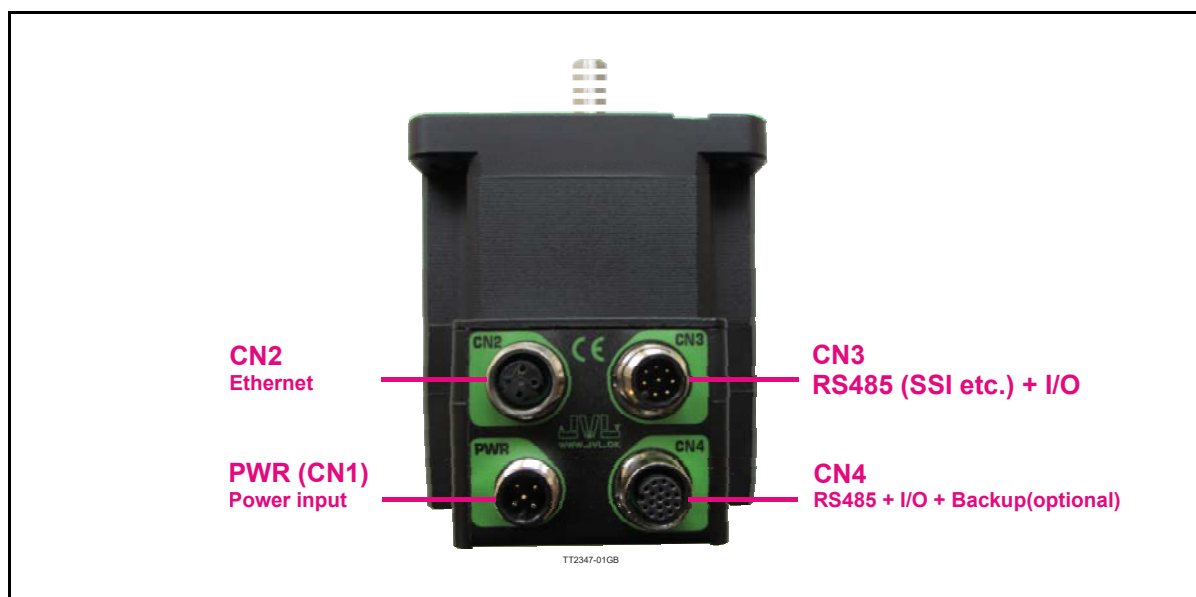
## 2.8 How to connect a MIS motor

### 2.8.7 Cables for the MISxxxxxxExxxxx

The following cables equipped with M12 connector can be supplied by JVL.

MIS34x Connectors				Description	JVL Order no.	Picture
“PWR” 5pin Male	“CN2” 5pin Female	“CN3” 8pin Female	“CN4” 17pin Female			
			X	RS485 Interface cable. Connects directly from The MIS motor to a RS485 comport. Length: 5m (197 inch)	RS485-M12-1-5-17S	
X				Cable (Ø5.5mm) with M12 male 5-pin connector loose wire ends 0.35mm² (22AWG) and foil screen. Length: 5m (197 inch)	WI1000-M12F5T05N	
Not relevant. The RS485-USB-ATC-820 connect to CN2 through cable type RS485-M12-1-5-8				USB to RS485 Converter. To be used if no RS485 COM port is available.	RS485-USB-ATC-820	
	X	X		Cable (Ø5.5mm) with M12 male D-coded 4-pin connector loose wire ends 0.35mm² (22AWG) and foil screen. Length: 5m (197 inch)	WI1046-M12M4S05R	
	X	X		Same as above but 15m (590 inch)	WI1046-M12M4S15R	
	X	X		Cable (Ø5.5mm) with M12 male D-coded 4-pin connector and RJ45 connector. Length: 5m (197 inch)	WI1046-M12M4S05-NRJ45	
This cable only exist in 5m length						
			X	Cable with M12 male 17-pin connector loose wire ends 0.22mm² (24AWG) and screen. Length: 5m (197 inch)	WI1009-M12M17T05N	
			X	Same as above but 20m (787 inch)	WI1009-M12M17T20N	
Protection caps. Optional if connector is not used to protect from dust / liquids.						
	X	X	X	IP67 protection cap for M12 female connector.	WI1000-M12FCAP1	
X				IP67 protection cap for M12 male connector.	WI1000-M12MCAP1	

## 2.8 How to connect a MIS motor



### 2.8.8 MISxxxxxxE(2-7)xxxx connector description.

Hardware wise all the MIS motors with the Ethernet option are equal and offer the connectivity shown in the table below.

The following Ethernet protocols are supported in this moment:

MISxxxxxxE2xxxx : EtherCAT  
 MISxxxxxxE3xxxx : EtherNet/IP  
 MISxxxxxxE4xxxx : Ethernet POWERLINK  
 MISxxxxxxE5xxxx : Modbus TCP  
 MISxxxxxxE6xxxx : ProfiNet  
 MISxxxxxxE7xxxx : Sercos III

The MIS motors offers robust M12 connectors which makes it ideal for automation applications. The M12 connectors offer solid mechanical protection and are easy operate. Following scheme gives the relevant information about each connector and the pins, wire colours and a short description of the signals available.

The connector layout:

“PWR” (CN1) - Power input. M12 - 5pin male connector				
Signal name	Description	Pin no.	JVL Cable WI1000-M12F5TxN	Isolation group
P+	Main supply +7-72VDC. Connect with pin 2 *	1	Brown	1
P+	Main supply +7-72VDC. Connect with pin 1 *	2	White	1
P-	Main supply ground. Connect with pin 5 *	3	Blue	1
CVI	Control and user output supply +7-30VDC. <b>DO NOT connect &gt;30V to this terminal !</b>	4	Black	1
P-	Main supply ground. Connect with pin 3 *	5	Grey	1

\* Note: P+ and P- are each available at 2 terminals. Make sure that both terminals are connected in order to split the supply current in 2 terminals and thereby avoid an overload of the connector.



**Please notice:** When connecting the a cable to one of the M12 connectors it must be tightened with 0.6 Nm to prevent ingress of fluids and/or particles in the connector which can cause malfunction.

(Continued next page)

## 2.8 How to connect a MIS motor

<b>“CN2” - Ethernet In port connector - M12 - 4pin female connector “D” coded</b>				
Signal name	Description	Pin no.	JVL Cable W11046-M12M4S05R	Isolation group (See note)
Tx0_P	Ethernet Transmit channel 0 - positive terminal	1	-	2
Rx0_P	Ethernet Receive channel 0 - positive terminal	2	-	2
Tx0_N	Ethernet Transmit channel 0 - negative terminal	3	-	2
Rx0_N	Ethernet Receive channel 0 - negative terminal	4	-	2
<b>“CN3” - RS485 (SSI etc.) + I/O. M12 - 8pin Male connector.</b>				
Signal name	Description	Pin no.	JVL Cable W11000-M12 F8TxxN	Isolation group (See note)
IO5	Used for zero set. Leave open if unused	1	White	1
IO6	Counting direction. Leave open if unused	2	Brown	1
RS422: A1+	SSI Clock +. Leave open if unused	3	Green	1
GND	Signal ground. Leave open if unused	4	Yellow	1
RS422: B1-	SSI Data in -. Leave open if unused	5	Grey	1
RS422: B1+	SSI Data in +. Leave open if unused	6	Pink	1
RS422: A1-	SSI Clock -. Leave open if unused	7	Blue	1
CVO	Supply output. Connected internally to the CVI terminal in the PWR connector. Max 700 mA	8	Red	1
<b>“CN4” - RS485 + I/O + Backup (option) connector - M12 - 17pin female connector</b>				
Signal name	Description	Pin no.	JVL Cable W11009M12 M17TxxN	Isolation group (see note)
IO1	I/O channel 1. Can be used as input or output	1	Brown	1
GND	Ground intended to be used together with the other signals in this connector	2	Blue	1
IO2	I/O channel 2. Can be used as input or output	3	White	1
IO3	I/O channel 3. Can be used as input or output	4	Green	1
RS422: B1-	RS422 I/O terminal B-	5	Pink	1
IO4	I/O channel 4. Can be used as input or output	6	Yellow	1
RS422: A1-	RS422 I/O terminal A-	7	Black	1
RS422: B1+	RS422 I/O terminal B+	8	Grey	1
CVO	Supply output. Connected internally to the CVI terminal in the PWR connector. Max 700 mA	9	Red	1
RS422: A1+	RS422 I/O terminal A+	10	Violet	1
IO5	I/O channel 5. Can be used as input or output	11	Grey/Pink	1
IO6	I/O channel 6. Can be used as input or output	12	Red/Blue	1
IO7	I/O channel 7. Can be used as input or output	13	White/Green	1
IO8	I/O channel 8. Can be used as input or output	14	Brown/Green	1
RS485: B0-	RS485 interface. Leave open if unused	15	White/Yellow	1
EXTBACKUP	Only for motors with the -H3 or -H4 option (abs. multi-turn encoder). This terminal can be connected to an external supply. <b>Connect to ground (GND) if not used.</b>	16	Yellow/Brown	1
RS485: A0+	RS485 interface. Leave open if unused	17	White/Grey	1
* Note: Isolation group indicate which terminals/circuits that a galvanic connected to each other. In other words group 1, 2, 3 and 4 are all fully independently isolated from each other. Group 1 correspond to the housing of the motor which may also be connected to earth via the DC or AC input supply.				

## 2.8 How to connect a MIS motor

### 2.8.9 Cables for the MISxxxxxxE6xxxx

The following cables equipped with M12 connector can be supplied by JVL.

MIS34x Connectors				Description	JVL Order no.	Picture
“PWR” 5pin Male	“CN2” 5pin Female	“CN3” 8pin Female	“CN4” 17pin Female			
			X	RS485 Interface cable. Connects directly from The MIS motor to a RS485 comport. Length: 5m (197 inch)	RS485-M12-1-5-17S	
X				Cable (Ø5.5mm) with M12 male 5-pin connector loose wire ends 0.35mm² (22AWG) and foil screen. Length: 5m (197 inch)	WI1000-M12F5T05N	
Not relevant. The RS485-USB-ATC-820 connect to CN2 through cable type RS485-M12-1-5-8				USB to RS485 Converter. To be used if no RS485 COM port is available.	RS485-USB-ATC-820	
	X	X		Cable (Ø5.5mm) with M12 male D-coded 4-pin connector loose wire ends 0.35mm² (22AWG) and foil screen. Length: 5m (197 inch)	WI1046-M12M4S05R	
	X	X		Same as above but 15m (590 inch)	WI1046-M12M4S15R	
	X	X		Cable (Ø5.5mm) with M12 male D-coded 4-pin connector and RJ45 connector. Length: 5m (197 inch)	WI1046-M12M4S05-NRJ45	
This cable only exist in 5m length						
			X	Cable with M12 male 17-pin connector loose wire ends 0.22mm² (24AWG) and screen. Length: 5m (197 inch)	WI1009-M12M17T05N	
			X	Same as above but 20m (787 inch)	WI1009-M12M17T20N	
Protection caps. Optional if connector is not used to protect from dust / liquids.						
	X	X	X	IP67 protection cap for M12 female connector.	WI1000-M12FCAP1	
X				IP67 protection cap for M12 male connector.	WI1000-M12MCAP1	

## 2.8 How to connect a MIS motor

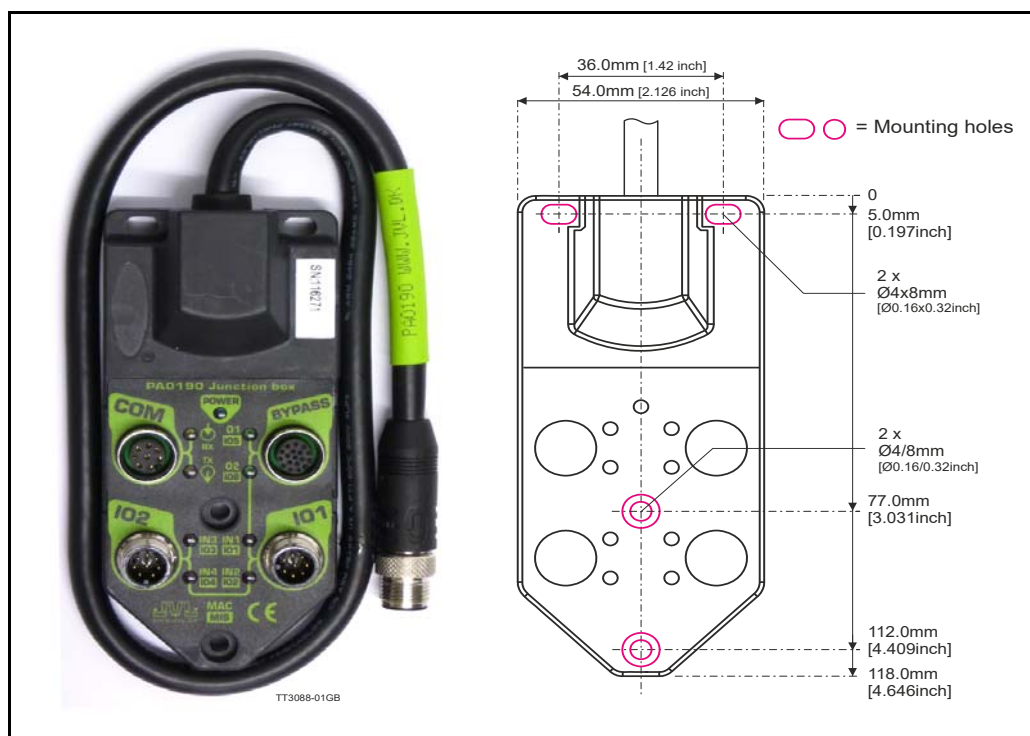
### 2.8.10 Drawing and description of PA0190

Junction box that splits the signals in the MIS motors “CN4” 17 pin I/O connector into 4 individual connectors giving an easy and more flexible installation.

#### Usage hints:

The LED's will only work with a MIS motor where the OUT+ and IO- is supplied from the CN4 connector. See also the I/O description for the module.

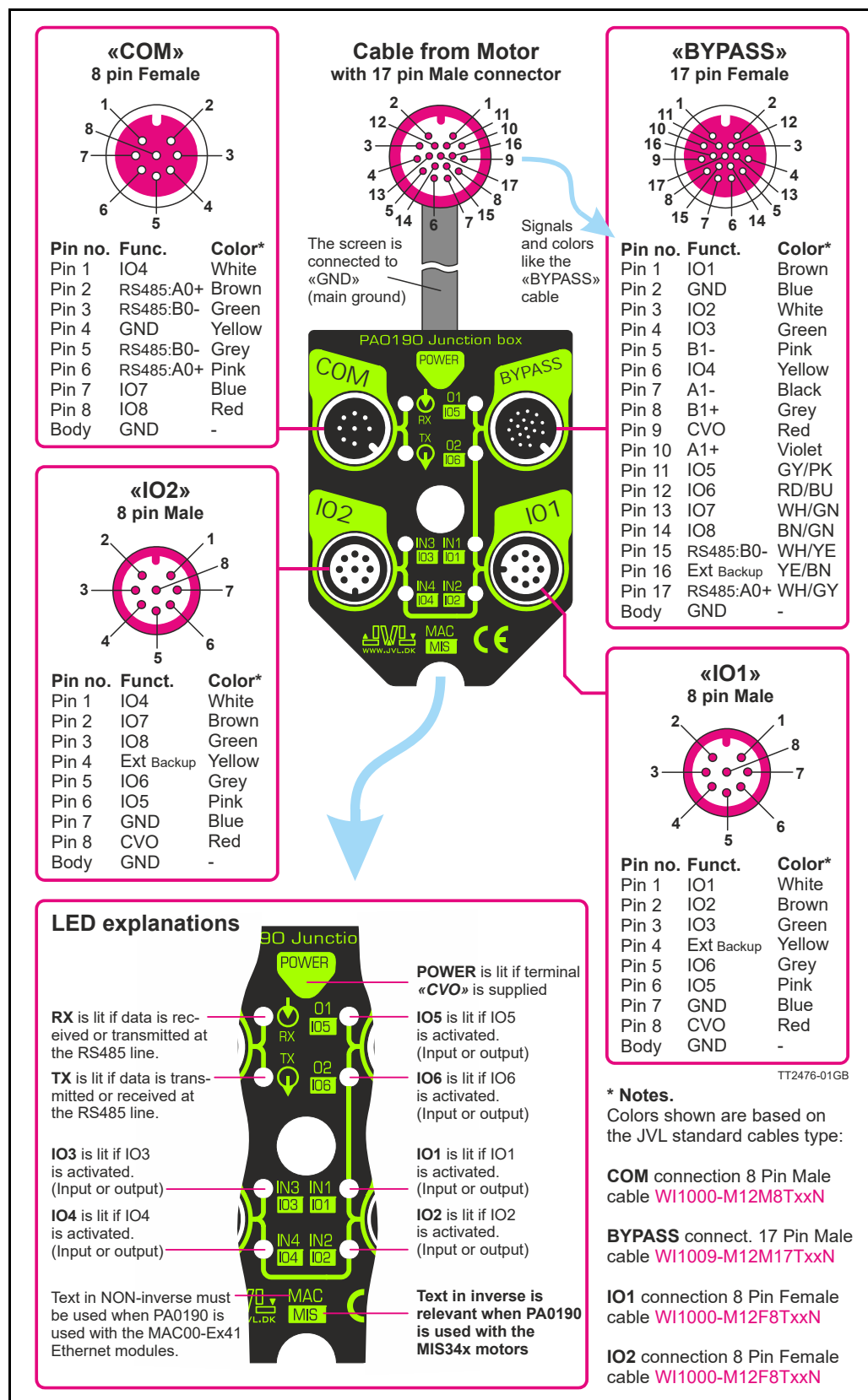
If a cable is connected to the “BYPASS” then the Communication pins and GND must be properly connected to valid signals (pins 2, 15, 17). AND “COM” must not be used. In other words use EITHER the “BYPASS” OR the “COM” connector. Not both.



**Please notice:** When connecting the a cable to one of the M12 connectors it must be tightened with 0.6 Nm to prevent ingress of fluids and/or particles in the connector which can cause malfunction.

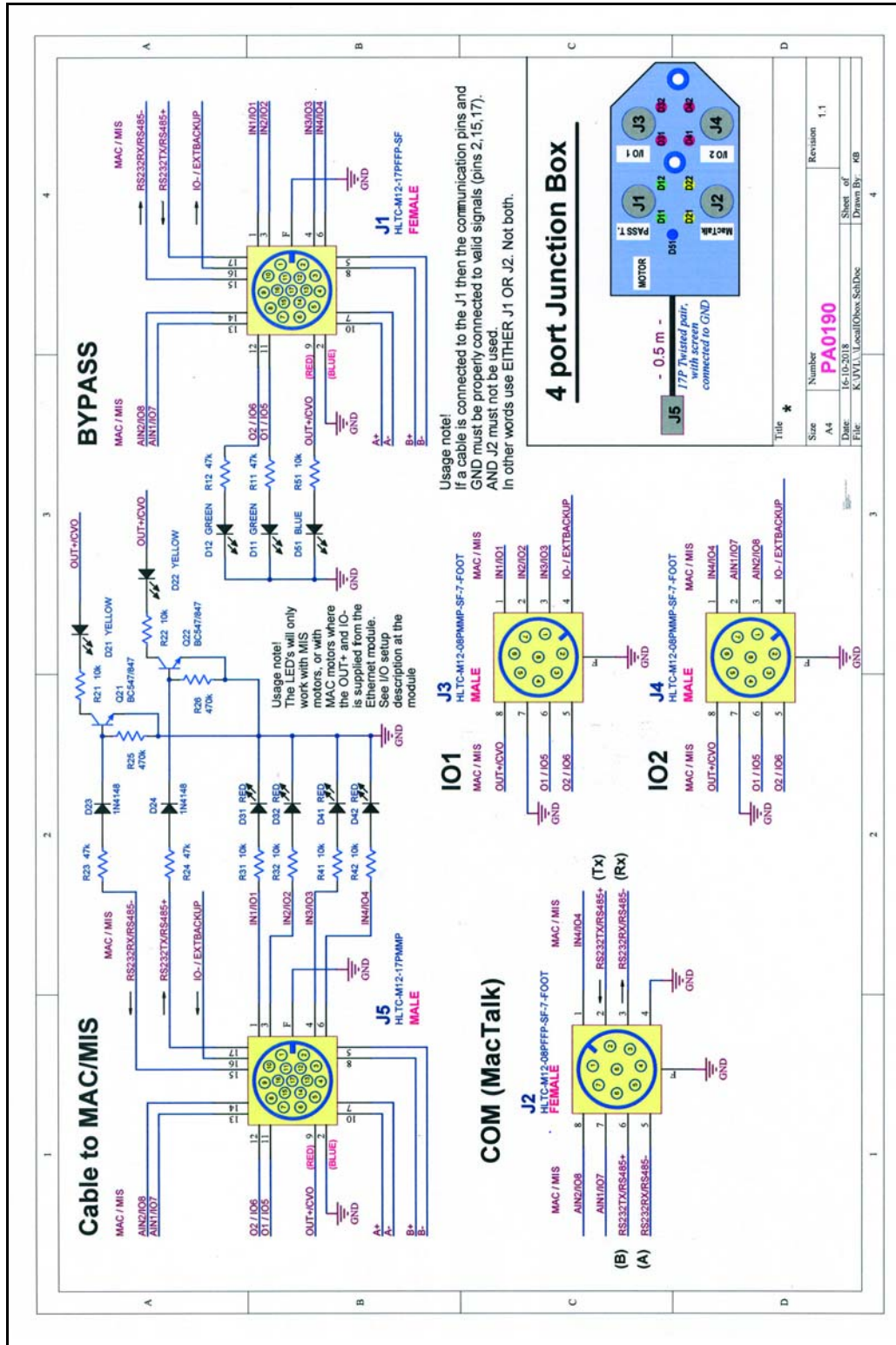
## 2.8 How to connect a MIS motor

Terminal and LED description of the PA0190 Junction box.



## How to connect a MIS motor

Diagram of the internal details in the PA0190 Junction Box.

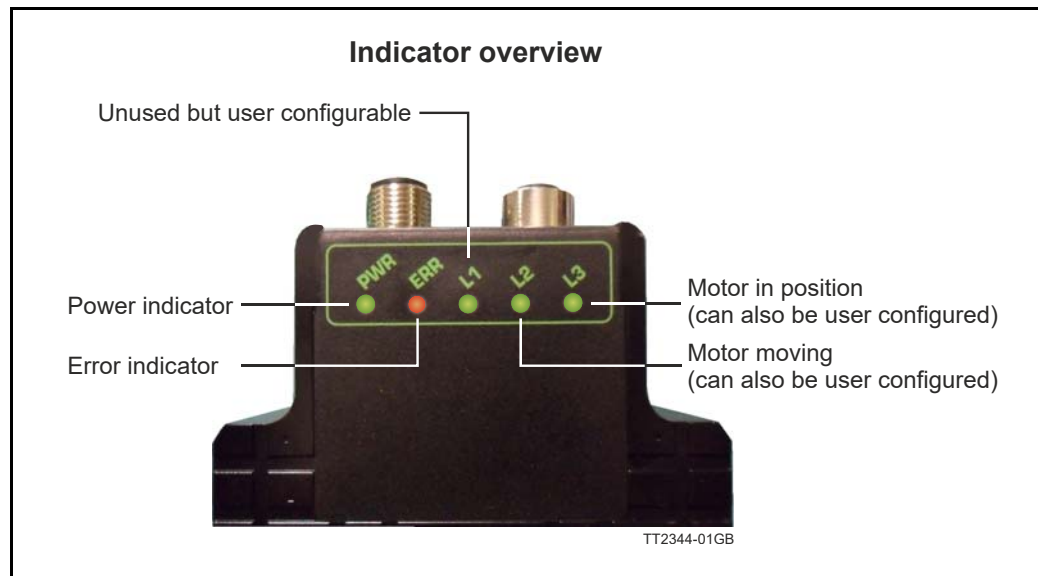


## 2.9 LED indicators basic motor

### 2.9.1 LED's - description for products without Ethernet or CANopen.

This description covers all MIS motors with basic configuration without any Ethernet or CANopen such as MISxxxxxx**Q5**xxxx, MISxxxxxx**Q9**xxxx, MISxxxxxx**FB**xxxx, or MISxxxxxx**EW**xxxx.

The LED's are used for indicating states and faults.



#### LED indicator descriptions (default setup)

LED Text	Colour	Constant off	Constant on	Blinking
L1	Green	Default	Only when user configured	Only when user configured
L2	Green	Motor not moving	Motor moving	-
L3	Green	Motor not in position	Motor in position	-
ERR	Red	No error	-	Error
PWR	Green	Power is not applied	Power is applied to both motor and module. Only <b>MIS17x</b> and <b>MIS23x</b> : The LED will lit red constantly if the supply is too low.	-

L1, Red, shows by default the Status bit I5: Closed loop lead/lag detected. It tells if the motor is overloaded, which can be caused of too low Running current or too heavy load.

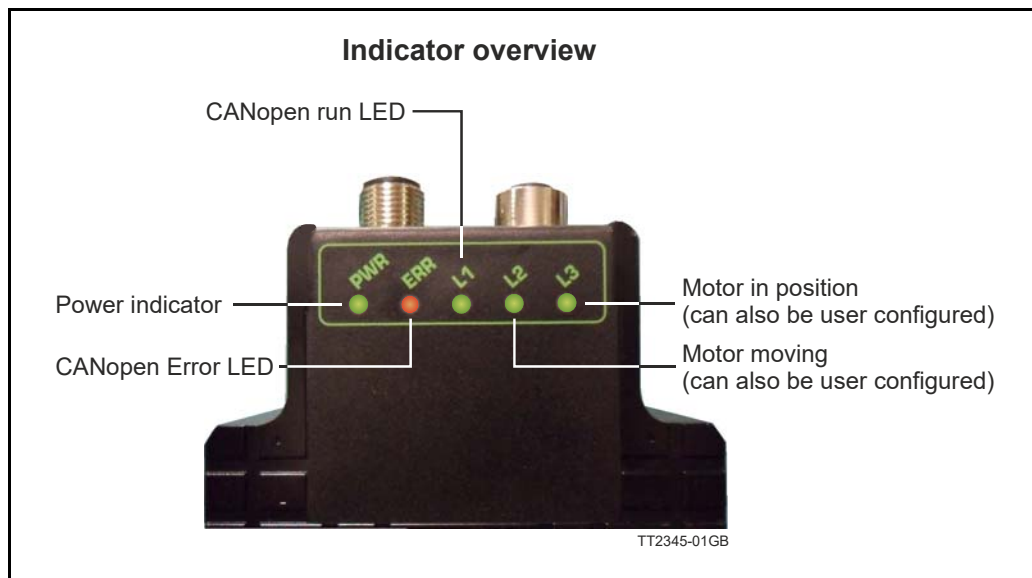
L1 to L3 can be configured to show the status of a almost any single bit from a user defined register. Please see [FlexLEDSetup I](#), page 237 for the details.

## 2.10 LED indicators using CANopen

### 2.10.1 Indicator LED's - description for products with CANopen.

This description covers all MISxxx products with build in CANopen option such as MIS34xxxxP6xxxx.

The LED's are used for indicating states and faults.



LED indicator descriptions (default setup)

LED Text	Function	Colour	Constant off	Constant on	Blinking
L1	CANopen run LED	Green	Please see below and optionally the DS303-3 standard		
L2	Motor moving	Green	Motor not moving	Motor moving	-
L3	Motor in position	Green	Motor not in position	Motor in position	-
ERR	CANopen error LED	Red	Please see below and Please see the DS303-3 standard		
PWR	Power	Green	Power is not applied.	Power is applied to both motor and module. Only <b>MIS17x</b> and <b>MIS23x</b> : The LED will lit red constant-ly if the supply is too low.	-

L2 to L3 can be configured to show the status of a almost any single bit from a user defined register. Please see [FlexLEDSetup I](#), page 237 for the details.

## 2.11 LED indicators using Ethernet

---

### 2.11.1 Indicator LED's - description for products with Ethernet.

The MIS motors offers optional 6 different Ethernet protocols.  
These are:

- **EtherCAT**
- **ModbusTCP**
- **Profinet**
- **Powerlink**
- **EthernetIP**
- **SercosIII**

This manual do only cover description of how to connect.

Concerning LED indicators, software and protocol setup and usage please consult a separate manual that can be found at [www.jvl.dk](http://www.jvl.dk) using this link: [www.jvl.dk](http://www.jvl.dk)

The LED descriptions are in the chapters "Commissioning" for each protocol.





### 3.1.1 Introduction

This section provides information for use of MIS stepper motors in functional safety applications. The section includes information about functional safety standards, the STO function, the related installation, commissioning and service and maintenance for STO.

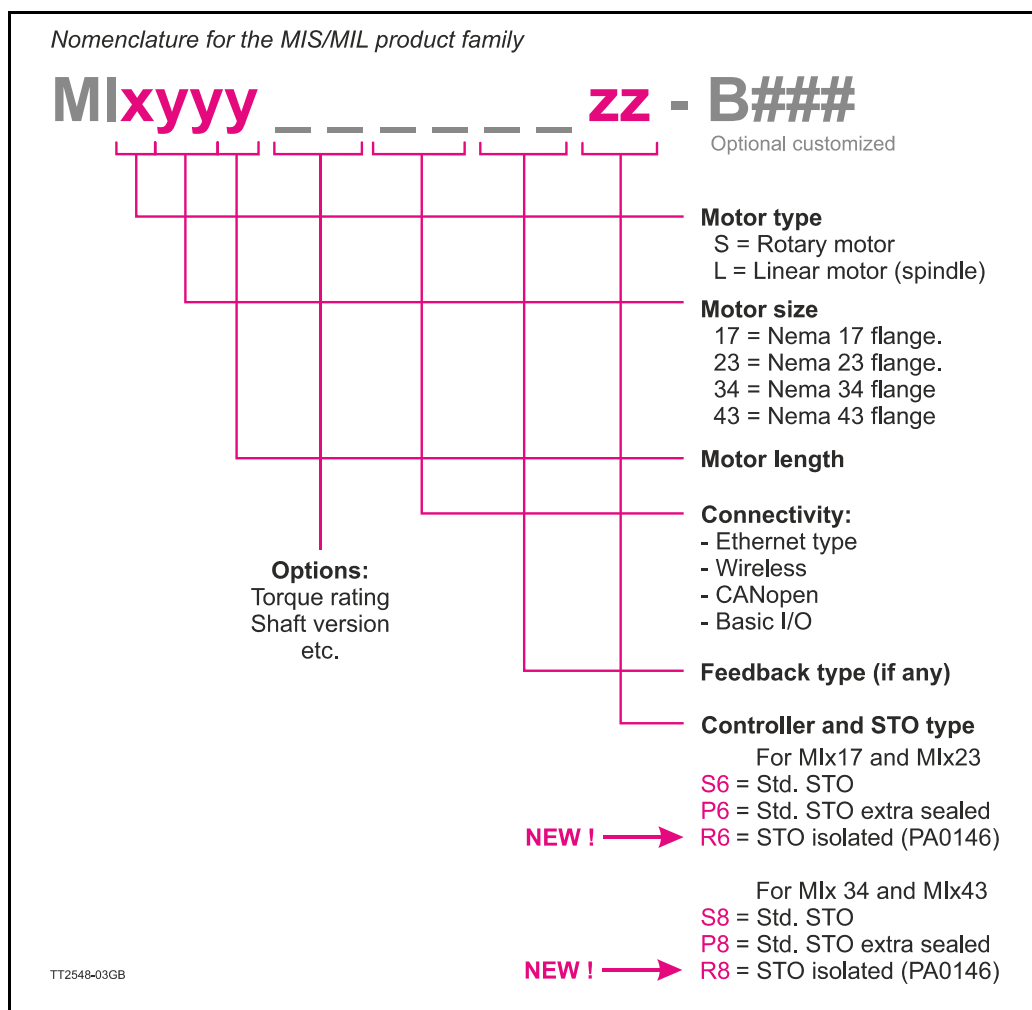
### 3.1.2 Functional overview

#### Introduction

As an option the MIS motor can be delivered with the so-called STO function which to a certain extent can simplify the safety system in the actual application. If the STO option is present in the motor an extra 4-pin M8 connector is available with 2 inputs. These 2 inputs need to be supplied in order to let the motor be energized and thereby operate. If one or both of the inputs are not supplied, the motor will be de-energized and no movement will happen regardless which commands are sent to the motor. The STO input simply overrule any other attempts to let the motor run. The default settings require manual restart of the motor when the STO signals are re-applied.

#### Products covered and identification

The whole range of MIS and MIL stepper motors offers the STO option. To identify whether a motor has the STO option installed, please check the last 2 digits of the type number:



## 3.1

# STO - Safe Torque Off

Option

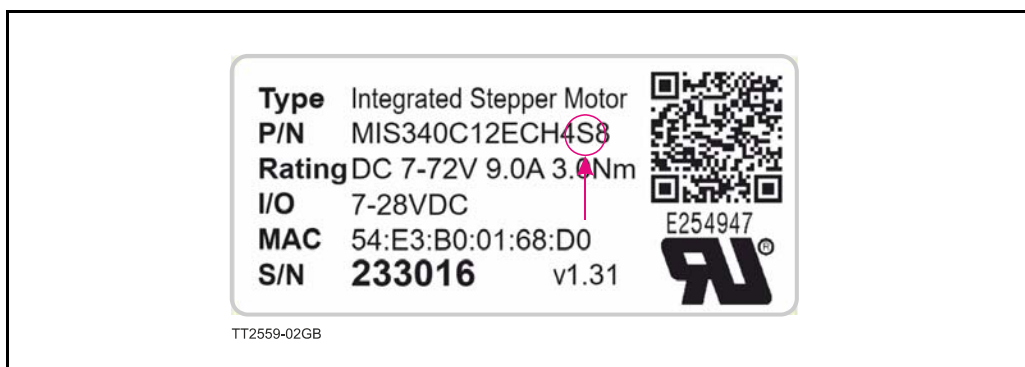
### Example:

Motor type: **MIS173S61EPH4R6-B033** offers following features and options:

- MIS:Rotating motor
- 173:Size Nema 17 with length = 3
- S61:Motor shaft type S61
- EP:ProfiNET communication
- H4:Feedback type H4
- R6:STO galvanically isolated with controller type SMC66.
- -B033:Custom option **Bxxx** means that some kind of customized change have been made. In this case it's another cover foil with alternative reference numbers for the connectors (acc. to ISO standard)

### Product label

Example of the physical label on the motor where the S8 is identified and marked with a red circle. See also the nomenclature for codes that defines other STO options.



### 3.1.3

### Abbreviations

Abbreviation	Reference	Description
Cat.	ISO13849-1	Category, level "1-4"
DC	-	Diagnostic coverage
FIT	-	Failure in time: 1E-9/hour
HFT	IEC61508	Hardware fault tolerance: HFT = n means that n+1 faults cause a loss of the safety function.
MTTFd	ISO13849-1	Mean time to failure – dangerous. Unit: Years
PFH	IEC61508	Probability of dangerous failures per hour. Consider this value if the safety device is operated in high demand or continuous mode of operation, where the frequency of demands for operation made on a safety-related system is greater than once per year.
PFD	IEC61508	Average probability of failure on demand, value used for low demand operation.
PL	ISO13849-1	Performance level, used to specify the ability of safety-related parts of control systems to perform a safety function under foreseeable conditions. Levels a–e.
SIL	IEC61508 IEC62061	Safety integrity level
STO	IEC61800-5-2	Safe Torque Off
SRECS	IEC62061	Safety related electrical control system
SRP/CS	ISO13849-1	Safety related parts of control systems
PDS/SR	IEC61800-5-2	Power drive system (safety related)

### 3.2.1 Qualified personnel

Only persons with proven skills are allowed to assemble, install, program, commission, maintain, and decommission the products.

Persons with proven skills

- Are qualified electrical engineers, or persons who have received training from qualified electrical engineers and are suitably experienced to operate devices, systems, plant, and machinery in accordance with the general standards and guidelines for safety technology.
- Are familiar with the basic regulations concerning health and safety/accident prevention.
- Have read and understood the safety guidelines given in this section and also the instructions given in the general operating instructions of the stepper motor.
- Have a good knowledge of the generic and specialist standards applicable to the specific application.

Users of power drive systems (safety related) (PDS(SR)) are responsible for:

- Hazard and risk analysis of the application.
- Identifying safety functions required and allocating SIL or PLr to each of the functions.
- Other subsystems and the validity of signals and commands from them.
- Designing appropriate safety-related control systems (hardware, software, parameterization, and so on).

Protective measures

- Only qualified and skilled personnel is allowed to install and commission safety engineering systems.
- Check the ingress protection (IP) level of the motor before installation in wet, humid environments - [Ordering information, page 421](#).
- Ensure short-circuit protection of the STO cable between the motor and the external safety device according to ISO 13849-2 table D.4.
- When external forces influence the motor axis (for example suspended loads), extra measures (for example a safety holding brake) are required to eliminate hazards.

### 3.2.2 Safety precautions



#### CAUTION

After installation of STO, perform a commissioning test as specified in chapter [STO commissioning test, page 75](#). A passed commissioning test is mandatory after first installation and after each change to the safety installation. In case that the test of STO fails: **DO NOT** use the motor, return it to JVL A/S for service.

The motor control electronics including the safety related parts must only be serviced and repaired by JVL A/S. Access to the inner parts of the motor will violate the guarantee.

**WARNING****RISK OF DEATH AND SERIOUS INJURY**

If external forces act on the motor, for example in case of vertical axis (suspended loads), and an unwanted movement, for example caused by gravity, could cause a hazard, the motor must be equipped with extra measures for fall protection. For example, install extra mechanical brakes.

**WARNING**

STO is suitable for performing mechanical work on the system or affected area of a machine only. It does not provide electrical safety.

**NOTICE**

Perform a risk assessment for each stop function to determine the selection of a stop category in accordance with EN 60204-1:

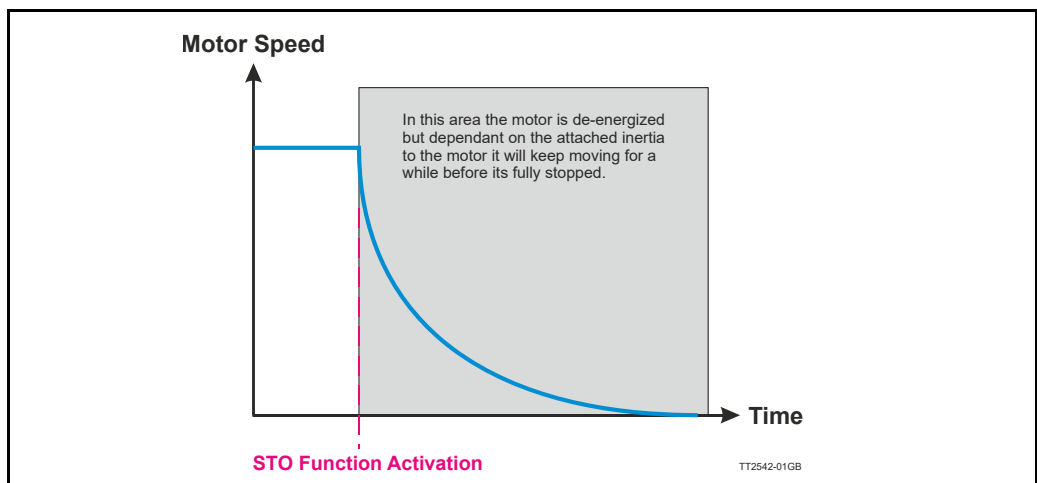
- Stop Category 0 is achieved with immediate removal of power to the actuator, resulting in an uncontrolled coast to stop.  
The STO according to EN 61800-5-2 accomplishes a Stop Category 0 stop.

This is the only category available for the MIS motors alone. Other measures must be made to achieve stop category 1 or 2:

- Stop Category 1 is achieved with power available to the machine actuators to achieve the stop. Power is removed from the actuators when the stop is achieved according to EN 61800-5-2 Safe Stop 1 (SS1).
- Stop Category 2 is a controlled stop with power available to the machine actuators. A holding position under power follows the stop.

**NOTICE**

When designing the machine application, timing and distance must be considered for a coast to stop (Stop Category 0 or STO). For more information regarding stop categories, refer to EN 60204-1.



**3.2.3 Important general information.**

If the motor is used outside the environmental limits given in this instruction the motor can not be expected to perform a safe stop when there is a demand on the STO-safety function.

**3.2.4 Using electromagnetic brake.**

The MIS motors can be supplied with an optional integrated brake or alternatively an external brake can be connected to one of the 8 user outputs (IO1 - IO8).

Please notice that the brake feature is not covered by the general TÜV safety approval since it is under software control.

Do therefore not use the motor with a brake in applications where the safety and injury of persons is depending of the function of the brake.

### 3.3.1 Safety instructions



#### CAUTION - ELECTRICAL HAZARD

The operator or electrical installer is responsible for proper grounding and compliance with all applicable national and local safety regulations. Please see [Power Supply Grounding and Earthing, page 18](#).

### 3.3.2 STO installation

Installation must be in accordance with the following steps and must be carried out by competent personnel, see [Qualified personnel, page 64](#). The Safe Torque Off option is intended to be part of the safety related control system. Before installation, perform a risk assessment that compares the Safe Torque Off option specifications and all foreseeable operational and environmental characteristics of the control system. A safety analysis of the machine section controlled by the drive is required to determine how often to test the safety function for proper operation during the life of the machine.

### 3.3.3 Enabling the STO function

Remember that the STO function is an option.  
The MIS motors must be ordered with the right type number to contain the STO option.  
Note that there are 2 sub-options for the STO.

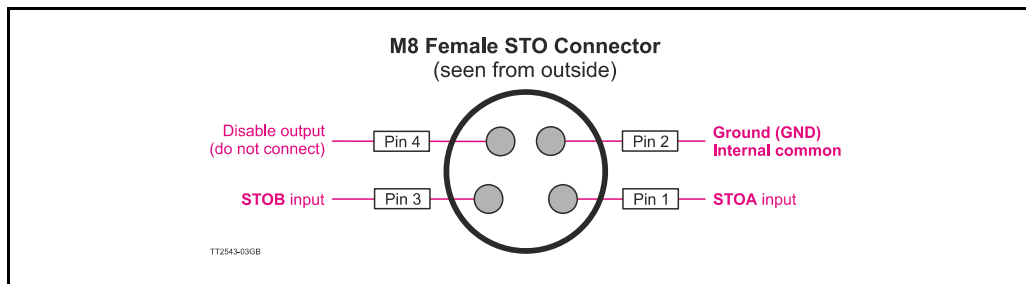
- “S” option: Non isolated STO interface. 2 inputs that must be connected to 2 source outputs from the safety controller.
- “R” option: Galvanic isolation through the PA0146 cable. 2 inputs where one input must be connected to a source output (24V nom.) and one input must be connected to a sink output (0V nom.).

See also the nomenclature overview in the section [Introduction, page 62](#)

Common for both STO options is that the STO option is enabled when the NON-STO plug is removed from the motor. Please see the section [Disabling the STO function, page 76](#) to learn more about the NON-STO plug.

### 3.3.4 Cables and connectors for the STO "S" option

The STO connector at the motor contains the two enable inputs STOA and STOB. Both inputs must be applied nominal +24VDC in order to energize the motor and make any motor movement possible. STO-signals must originate from a safety isolating transformer in accordance with IEC 61558-1 and IEC 61558-2-6 or from a switch mode power supply with equivalent property. If only one of the inputs is not applied +24VDC the internal STO circuit will remove the energy from the motor. The illustration below shows the pinout of the connector. A 100 mA fuse must be inserted in each channel.

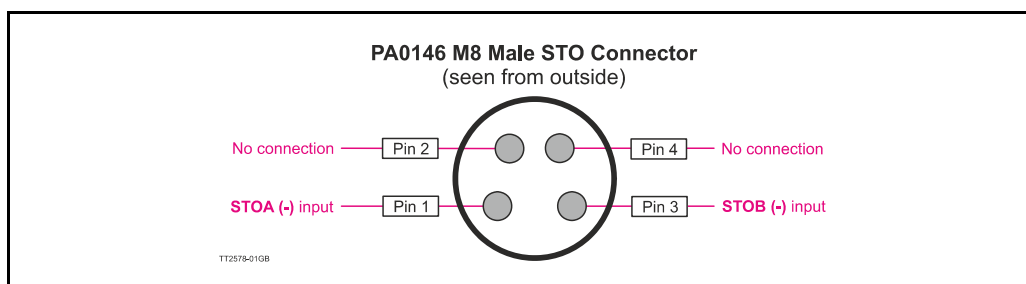


Use only the standard cable for the STO "S" function:  
JVL stock no. **W11010-M08M4V05P** (5m length) other lengths and versions also exist.

### 3.3.5 Cables and connectors for the STO "R" option

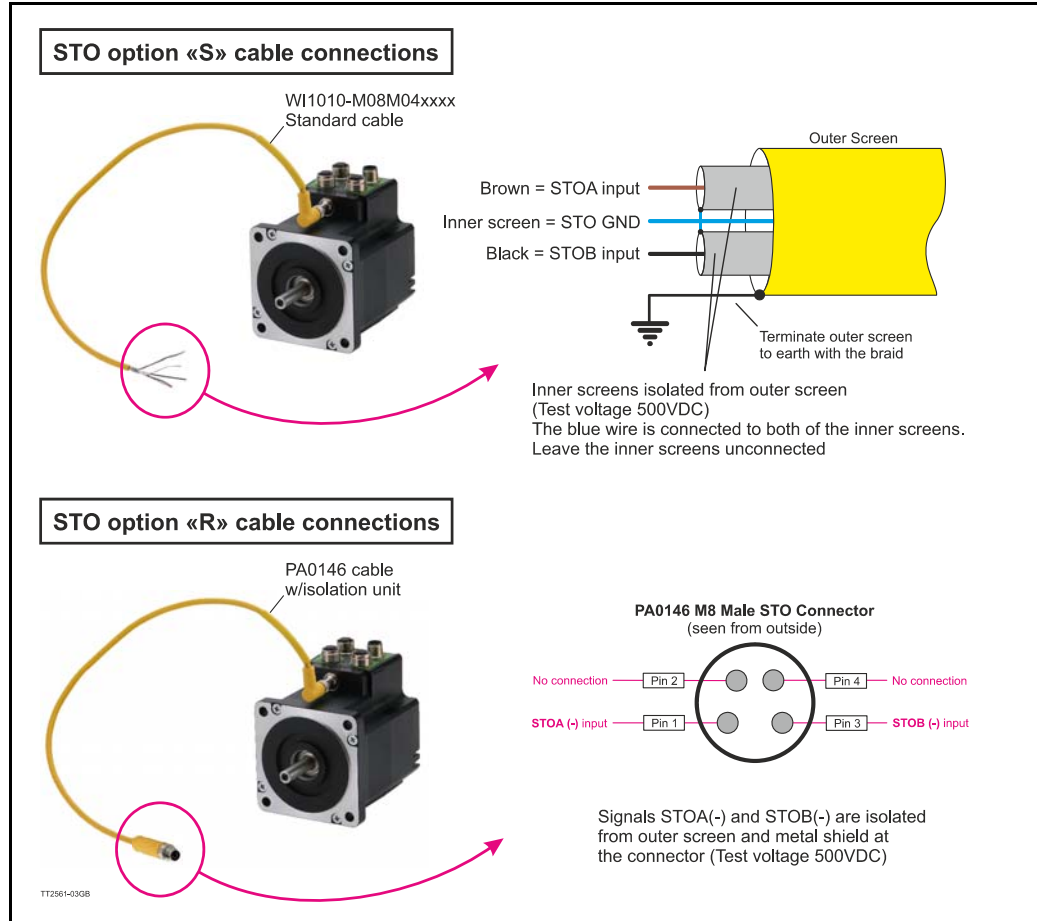
If the motor is having the "R" option the cable PA0146 is delivered with the motor. This cable offers galvanically isolation of the 2 STO inputs and have another input format. One input must be connected to a source output (24V nom.) and one input must be connected to a sink output (0V nom) in order to energize the motor and make any motor movement possible.

The STO-signals must originate from a safety isolating transformer in accordance with IEC 61558-1 and IEC 61558-2-6 or from a switch mode power supply with equivalent property. If only one of the inputs is not applied +24VDC the internal STO circuit will remove the energy from the motor. The illustration below shows the pin out of the connector. A 100 mA fuse must be inserted in each channel.



Use only the standard cable for the STO "R" function: JVL stock no. **PA0146**.

The standard STO cable is yellow to signal “safety”, and by using this official cable, the disable output (pin 4) is not connected inside. If using another standard cable, pin 4 might be connected and therefore 24 V is inside the cable and could be a risk in safety applications. Also risk of short connection of the 24V exist when using a standard cable.



**Please notice:** When connecting the M8 STO connector it must be tightened with 0.4 Nm to prevent ingress of fluids and/or particles in the connector which can cause malfunction.

### 3.3.6

#### Functional Proof tests

The functional safety standards require that functional proof tests be performed on the equipment used in the system. Proof tests are performed at user-defined intervals and are dependent upon PFD and PFH values.

#### IMPORTANT

The specific application determines the time frame for the proof test interval. Verify operation of safety function after drive installation, modification, or maintenance. Refer to [STO commissioning test](#), page 75 for more information about how to test the safety function.

### 3.4.1 Safety instructions

**CAUTION - ELECTRICAL HAZARD**

The operator or electrical installer is responsible for proper grounding and compliance with all applicable national and local safety regulations.

Please see [Power Supply Grounding and Earthing, page 18](#).

### 3.4.2 Activation of STO

The STO is activated by removing either signal STO\_A, STO\_B or both. In all 3 cases the STO will be active and the motor will not be energized.

Motors with the STO option "R" (PA0146 cable) the removal of the input signals STOA(+) or STOB(-) or both, will break the input loop and activate STO.

### 3.4.3 Parameter settings for STO (auto./manual restart behavior)

**AUTOMATIC RESTART**

Automatic restart behaviour is only allowed in 1 of the 2 situations:

- The unintended restart prevention is implemented by other parts of the STO installation.
- A presence in the dangerous zone can be physically excluded when STO is not activated. In particular, paragraph 6.3.3.2.5 of ISO 12100: 2010 must be observed.

**Compliance with UL requirements (UL61800)**

Only the MIS/MIL34x series are compliant with UL requirements.

M1x17x and M1x23x are UL pending.

If the motor is used in an application which require compliance with UL requirements the following precautions must be taken.

**Conditions of Acceptability:**

For use only in complete equipment where the acceptability of the combination is determined by UL LLC:

1. Supply voltage (P+) must maximum be set to 60VDC.
2. These devices are to be supplied from an isolating type DC power supply. The devices are evaluated for use in Overvoltage Category II only.
3. These devices are provided with terminals suitable for factory wiring only.
4. These devices are to be protected by supplementary fuse rated 10A and 60Vdc minimum.
5. These devices are not evaluated for functional safety.

## 3.4

# Commissioning

Option

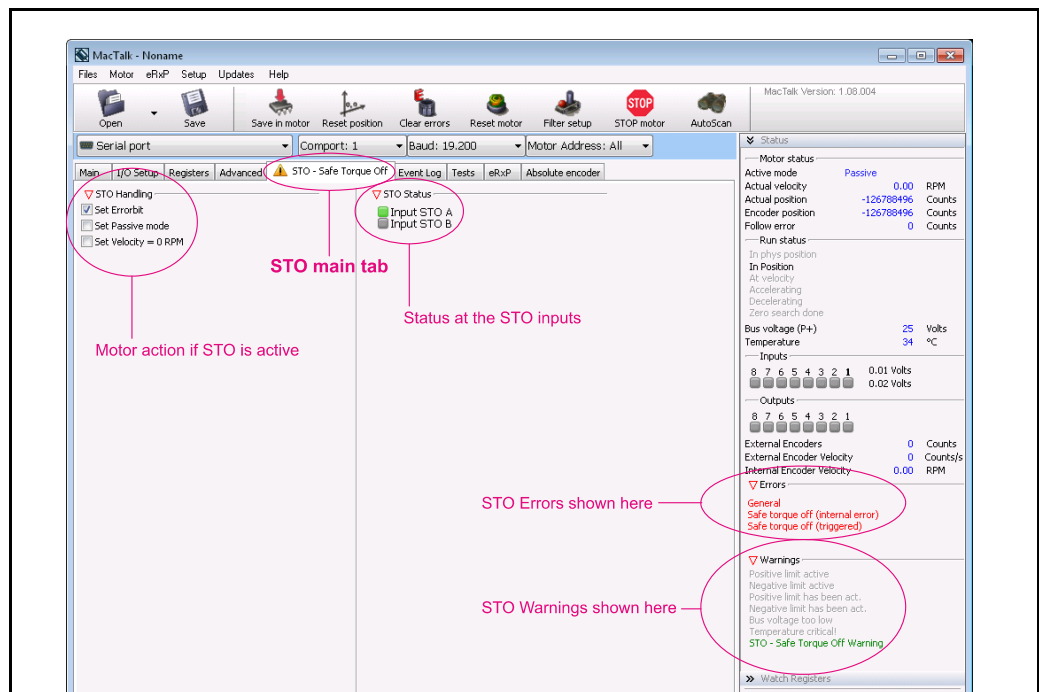
### 3.4.4 Operating the STO feature using MacTalk

If the motor has the Safe Torque Off option installed, The “STO – Safe Torque Off” tab will be shown in MacTalk.

The actual status of both channels can be monitored on this tab. When green, the channel is ON. Both channels must be ON at the same time in order to enable the motor.

When one or both of the channels are OFF, the Safe Torque Off is active and the motor is disabled. The motor will behave according to the settings made:

Below the STO tab in MacTalk.



### 3.4.5 Motor action if STO is active

Below a description of the 3 possible actions that can be done if the STO is activated. They refer to the screen dump above.

#### Set Errorbit

Default = OFF

The motor will set both the General (bit 0) and the STO (bit 29) error bit. To get the motor back in normal operation the STO channels must be ON, the error must be cleared and an active operation mode must be selected.

#### Set Passive mode

Default = ON

The motor will go to passive mode. To get the motor back in normal operation the STO channels must be ON and an active operation mode must be selected.

#### Set Velocity = 0 RPM

Default = OFF

The motor stays in the actual mode and the requested velocity will be set to 0 RPM. After the STO channels are ON, a velocity value (>0 RPM) must be written into the velocity register to get the motor moving again.

### 3.4.6 STO Error messages

2 error messages exist:

- **Safe torque off (triggered)**

When the Safe Torque Off is or has been active and the **Set Errorbit** setting, the STO Error bit is set, and in MacTalk this can be monitored on the right status panel as **“Safe torque off (triggered)”**. The error will only be cleared by sending the “Clear errors” command.



- **Safe Torque off (internal error)**

Will only be set if the STO self-diagnostic circuit has detected an internal error. In this case, the motor must be returned to the manufacturer (JVL) for repair. This error can not be cleared.

### 3.4.7 STO Warning messages

Warnings

When the Safe Torque Off is active, the STO Warning bit is set, and in MacTalk this can be monitored on the right status panel as “STO- Safe Torque Off Warning”.

**3.4.8 Safe Torque Off without using MacTalk**

If MacTalk is not used for monitoring the status of the STO function it must be done as follows:

The motor contains a number of registers which can be accessed from various protocols depending on which options the motor has.

Protocols available are for example Ethernet (EthernetIP, PROFINET etc.) and CAN-open, Modbus or the MacTalk protocol.

The status of the STO function is also available using above protocols.

Each field in MacTalk described earlier in this chapter can be monitored/read from a register in the motor.

The registers that are relevant for the STO function are:

**R25 STATUS\_BITS**

Status bits related to the STO function.

Note that other bits for many other functions in the motor are also represented in this register.

- Bit 23: Input status for the STOA input.  
Reading "0" means that the input is inactive (no voltage applied).
- Bit 24: Input status for the STOB input.  
Reading "0" means that the input is inactive (no voltage applied).

See also: [Status bits](#), page 205

**R35 ERROR\_BITS**

Error bits related to the STO function.

Note that other bits for many other functions in the motor are also represented in this register.

- Bit 27: STO\_ALARM.  
Will be set if there is an internal hardware defect in the STO circuitry.

**Important**

The STO\_ALARM will only be set if the STO self-diagnostic circuit has detected an internal error. In this case, the motor must be returned to the manufacturer (JVL) for repair.

- Bit 29: STO - This bit will be set if one of the 2 STO inputs are off (not applied an voltage). Also STO is set if the STO\_ACTION\_ERROR bit is set.

See also: [Err\\_Bits](#), page 208

**R36** WARNING\_BITS

Warning bits related to the STO function.  
Note that other bits for many other functions in the motor are also represented in this register.

- Bit 9: STO Warning  
The STO warning bit is always set if one of the STO channels are off (no voltage applied to the STOA and/or STOB input)

See also: [Warn\\_Bits](#), page 209

**R124** SETUP\_BITS

Setup bits related to the STO function.  
Note that other bits for many other functions in the motor are also represented in this register.  
If one or more of the following bits are set, the motor will behave as described when the STO function is activated:

- Bit 29: STO\_ACTION\_ERROR  
The motor will set both the General (bit 0) and the STO (bit 29) error bit.
- Bit 30: STO\_ACTION\_PASSIVE  
The motor will go to passive mode.
- Bit 31: STO\_ACTION\_V\_ZERO  
The motor stays in the actual mode and the requested velocity will be set to 0 RPM.

See also: [Setup\\_Bits](#), page 220

**3.4.9 STO commissioning test**

After installation and before first operation, perform a commissioning test of the installation, using STO. Perform the test again after each modification of the installation or application involving the STO.

**NOTICE**

A successful commissioning test of the STO function is required after the initial installation, and after each subsequent change to the installation.

To perform a commissioning test:

- Follow the instructions in chapter [Restart prevention for STO application, page 75](#) for applications without automatic restart after a safe stop, or
- Follow the instructions in chapter [Automatic restart of STO application, page 76](#) for applications with automatic restart after a safe stop.

**3.4.10 Restart prevention for STO application**

Application where SETUPBITS are set to default values STO\_ACTION\_ERROR=1, STO\_ACTION\_PASSIVE=0, STO\_ACTION\_V\_ZERO=0:

1. Set the motor in an active mode where it runs at well-defined velocity.
2. Remove STO channel A.
3. Check that:
  - 3a The motor coasts until stop.
  - 3b If any monitoring is connected, then check that ERRORBIT 0 and 29 are set. If MacTalk is connected, it shows "Safe Torque Off – triggered".
4. Reapply 24 V DC to STO channel A.
5. Verify that the motor remains stopped.
6. Clear the ERRORBITS either with MacTalk ("Clear errors" button) or by sending command 97 to register 24.
7. Set the motor in the active mode.
8. Check that the motor runs at the defined velocity.

Repeat the procedure with STO channel B.

The commissioning test is successfully completed when all the given steps are passed.

**3.4.11 Automatic restart of STO application**

Application where SETUPBITS are set to STO\_ACTION\_ERROR=0, STO\_ACTION\_PASSIVE=0, STO\_ACTION\_V\_ZERO=0:

1. Set the motor in an active mode where it runs at well-defined velocity.
2. Remove STO channel A.
3. Check that the motor coasts until stop.
4. Reapply 24 V DC to STO channel A.
5. Check that the motor runs at the defined velocity.

Repeat the procedure with STO channel B.

The commissioning test is successfully completed when all the given steps are passed.

**NOTICE**

See the warning on the restart behavior in chapter [Parameter settings for STO \(auto./manual restart behavior\)](#), page 70

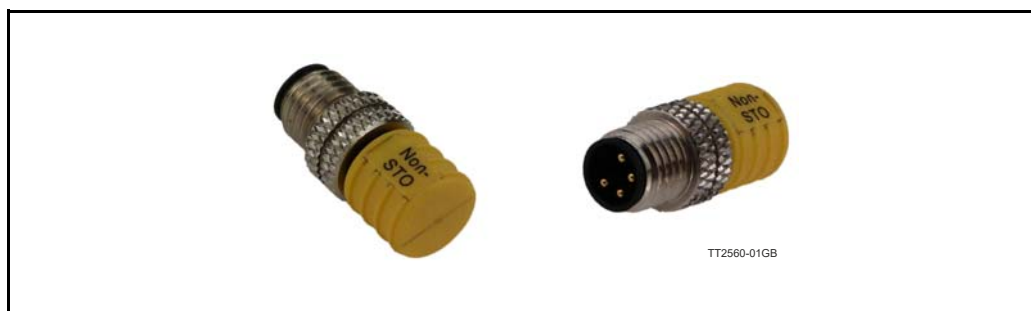
**3.4.12 Disabling the STO function**

If the STO function is not needed the plug JVL type **WI1010-M08M4SSTO** must be inserted in the STO connector at the motor.



The reason for this external plug to disable the STO function is to obtain a high safety level and make sure that no misunderstandings will occur concerning whether the STO function is active or not.

Pictures of the 4 pin M8 plug



**3.4.13 Service and maintenance**

For PL e or SIL3, it is a must, for lower PL or SIL it is recommended to conduct a functional test every 12 months to detect any failure or malfunction of the STO functionality. To conduct the functional test, perform the following steps (or a similar method suitable for the application):

1. When using the STO “S” option: Remove the 24 VDC voltage supply from the STO A and B inputs.  
When using the STO “R” option: Remove the input supply loop (24 VDC and 0V) from STOA(+) and STOB(-).
2. Check if the monitoring gives an error message.
3. Verify that the motor is coasting and comes to a complete stop.
4. When using the STO “S” option: Verify that the motor cannot be started by re-applying the 24 V DC to the STO A and B inputs.  
When using the STO “R” option: Verify that the motor cannot be started by re-applying 24 VDC and 0 VDC to the STOA(+) and STOB(-) inputs.
5. Verify that the motor is not started automatically until errors are cleared and the motor is set in an active mode.

**3.4.14 Capabilities**

Safe Torque Off performs a safety function such that when one of the STO inputs is not active, (i.e. open-circuit or set at nominally 0V), the motor will not develop torque. The STO function is implemented purely in simple solid state hardware for which substantial failure data exists, allowing meaningful quantitative FMEA to be carried out. The function does not use software or complex hardware.

When using the STO “S” option (STO cable **WI1010-M08M4xxx**) the probability of failure of the safety function due to a hardware fault has been estimated by JVL A/S as: 1,38e-10 per hour (IEC61508/IEC62061/IEC61800-5-2) or 4,29e-8 per hour (ISO 13849, Table K.1, MTTFd = 100 years)

When using the STO “R” option (**PA0146** cable) with galvanic isolation the results are: 1,01e-9 per hour (IEC61508/IEC62061/IEC61800-5-2) for the complete system, meaning PA0146 cable + Motor or 4,29e-8 per hour (ISO 13849, Table K.1, MTTFd = 100 years)

Above values are assessed by the independent notified body TÜV NORD. The input is compatible with self-testing digital outputs of controllers such as PLCs, where the test pulse is a maximum of 1 ms. This means that the drive is not disabled by logic-low input pulses with a maximum of 1 ms duration. This also applies when using the PA0146 cable.

**3.4.15 Limitations**

- STO uses solid-state techniques, it does not provide physical separation of electrical connections and is not intended to provide electrical isolation. For galvanic isolation of the STO signals use the STO “R” option - see also *Safety instructions*, page 70.
- STO does not provide braking, it disables the motor so no motor electrical braking is available. If motor braking is a requirement then an external arrangement must be made to stop the motor conventionally and then to safely remove the STO inputs to activate STO. Braking by the drive is not a high-integrity function, if braking is a safety requirement then an independent fail-safe brake must be provided.

## 3.4

## Commissioning

Option

### 3.4.16 STO Technical Data

NOTICE : The STO signal must be SELV or PELV supplied.

Following information is valid for motors with the STO “S” option.

European directives	Machinery Directive (2006/42/EC)	EN ISO 13849-1/2	
		EN IEC 62061	
		EN IEC 61800-5-2	
	EMC Directive (2014/30/EU)	EN 61000-6-3	
		EN 61800-3	
	Low Voltage Directive (2014/35/EU)	EN 61800-5-1	
Safety standards	Safety of Machinery	EN ISO 13849-1/2, IEC 62061, IEC 60204-1	
	Functional Safety	IEC 61508-1/2, IEC 61800-5-2	
Safety function		IEC 61800-5-2	IEC 60204-1
		Safe Torque Off (STO)	Stop Category 0
Safety performance	<b>ISO 13849-1</b>		
	Category	Cat 3	
	Diagnostic Coverage	DCavg: 98,5 % (Medium)	
	Mean Time to Dangerous Failure (per channel)	MTTFd: 100 years (High)	
	PFHd	4,29e-8/h	
	Performance Level	PLe	
	<b>IEC 61508 / IEC 62061 / IEC61800-5-2</b>		
	Safety Integrity Level	SIL 3, SIL CL3	
	Systematic capability (systematic integrity level)	SIL 3	
	Probability of Dangerous Failure per Hour	PFHd: 1,38e-10/h (High Demand Mode)	
	Diagnostic Coverage	DC: 96 %	
	Safe Failure Fraction	SFF: 96 %	
	Common Cause Failure Factor	CCF 5 %	
	Hardware Fault Tolerance	HFT: 1 (1oo2)	
	Proof Test Interval T1	20 Years	
	Mission time TM	20 Years	
Reaction time	Input to output response time	Maximum 8 ms.	
	Response time (internal fault)	Maximum 200 ms.	

## 3.4

## Commissioning

Option

Following information is valid for motors with the **STO "R" option** (isolated)

European directives	Machinery Directive (2006/42/EC)	EN ISO 13849-1/2	
		EN IEC 62061	
		EN IEC 61800-5-2	
	EMC Directive (2014/30/EU)		
		EN 61000-6-3	
		EN 61800-3	
Low Voltage Directive (2014/35/EU)			
	EN 61800-5-1		
Safety standards	Safety of Machinery	EN ISO 13849-1/2, IEC 62061, IEC 60204-1	
	Functional Safety	IEC 61508-1/2, IEC 61800-5-2	
Safety function		IEC 61800-5-2	IEC 60204-1
		Safe Torque Off (STO)	Stop Category 0
Safety performance	ISO 13849-1		
	Category	Cat 3	
	Diagnostic Coverage	DCavg: 98,6 % (Medium)	
	Mean Time to Dangerous Failure (per channel)	MTTFd: 100 years (High)	
	PFHd	4,29e-8/h	
	Performance Level	PLe	
	IEC 61508 / IEC 62061 / IEC61800-5-2		
	Safety Integrity Level	SIL 3, SIL CL3	
	Systematic capability (systematic integrity level)	SIL 3	
	Probability of Dangerous Failure per Hour	PFHd: 1,01e-9/h (High Demand Mode)	
	Diagnostic Coverage	DC: 95 %	
	Safe Failure Fraction	SFF: 98.9 %	
	Common Cause Failure Factor	CCF 5 %	
	Hardware Fault Tolerance	HFT: 1 (1oo2)	
	Proof Test Interval T1	20 Years	
	Mission time TM	20 Years	
Reaction time	Input to output response time	Maximum 8 ms.	
	Response time (internal fault)	Maximum 200 ms.	

The following temperature profile is used in safety value calculations:

- 1000 on/off cycles per year with  $T = 85\text{ °C}$
- $85\text{ °C}$  board temperature at 100 % of time.
- The STO is a type A safety component as defined in IEC 61508-2.
- Relevant failure modes:
  1. The STO trips spuriously (safe failure)
  2. The STO does not activate when requested

The analysis is based on an assumption that one failure occurs at one time. No accumulated failures have been analyzed. The conclusion gives that no single dangerous failures can remain undetected, and therefore the motor will always go to safe state in case of one dangerous failure at a time.

- Reaction times are described in section STO function activation and indication response times in the table above.

## 3.5 Specifications and Certifications

Option

### 3.5.1 General specifications

Attribute	Value
Standards	EN 60204-1, IEC 61508, EN 61800-3, EN 61800-5-1, EN 61800-5-2, EN 62061, EN ISO 13849-1
Safety category	Cat. 3 and PL e per EN ISO 13849-1; SIL CL3 per IEC 61508 and IEC 62061
Power supply (user I/O)	24V DC $\pm 10\%$ , 0.8...1.1 x rated voltage PELV or SELV
STO A	24 V DC, max. 10 mA
STO B	24 V DC, max. 10 mA
STO input ON voltage, min	5 V
STO input OFF voltage, max	2 V
STO input <b>absolute max</b> rating	60 V
Conductor type	Multi-conductor shielded cable
Conductor size	24 AWG
Strip length	10 mm

### 3.5.2 Environmental specifications


The installation must comply with all environmental, pollution degree, and drive enclosure rating specifications required for the operating environment.

Category	Specification
Ambient temperature	0...40 °C
Storage temperature	-40...70 °C
Relative humidity	5..93 % non-condensing
Shock	15 g, 30ms. (EN/IEC 60068-2-27)
Vibration	5 - 25 Hz: $\pm 1.6$ mm, 25 - 500Hz: 4 g, 1.0 oct./min. (EN/IEC 60068-2-6)
Surrounding environment	The ingress protection (IPxx) depends on the specific motor type number. Please see <a href="#">Ordering information, page 421</a>
EMC	Emission
	Noise immunity
	IEC61800-3 (Category 2)
	IEC61000-6-2

#### ATTENTION:

Failure to maintain the specified ambient temperature can result in a failure of the safety function.

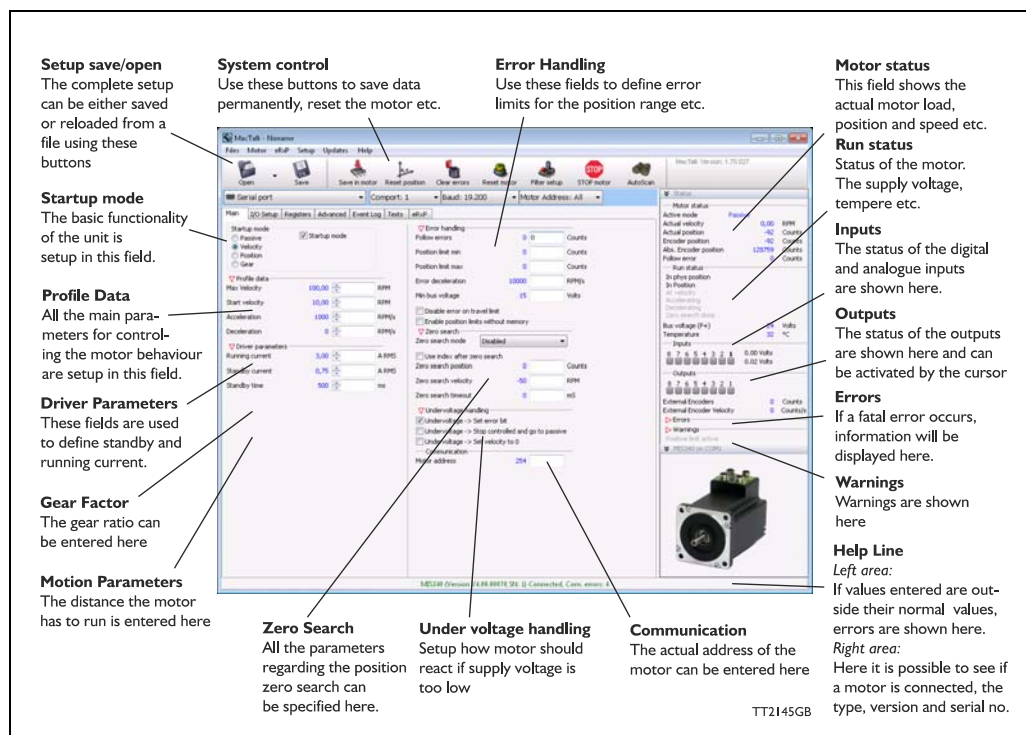
### 3.5.3 Certifications

Certification	Value
TÜV 	Certified by TÜV NORD for Functional Safety:  up to SIL CL3, according to IEC 61800-5-2, IEC 61508, and EN 62061;  up to Performance Level PLe and Category 3, according to EN ISO 13849-1;  when used as described in this User Manual.





## 4.1 Using the MacTalk software



### 4.1.1 MacTalk introduction

The MacTalk software is the main interface for setting up the MIS motor for a specific application.

The program offers the following features:

- Selection of operating mode of the MIS motor.
- Changing main parameters such as speed, motor current, zero search type, etc.
- Monitoring in real time the actual motor parameters, such as supply voltage, input status, etc.
- Changing protection limits such as position limits.
- Saving all current parameters to PC.
- Restoring all parameters from PC.
- Saving all parameters permanently in the motor.
- Updating the motor firmware or MacTalk software from the internet or a file.

The main window of the program changes according to the selected mode, thus only showing the relevant parameters for operation in the selected mode.

The following pages describe the actual window for each mode and how the parameters affect MIS motor operation.

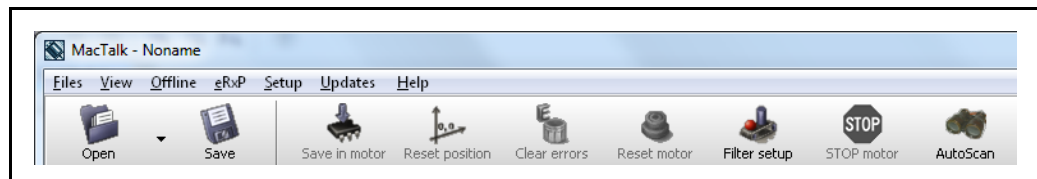
MacTalk is normally connected through the RS485 interface but may also work on Ethernet (if the Ethernet option is present in the actual motor).

## 4.1 Using the MacTalk software

---

### 4.1.2 Toolbar description

The toolbar at the top of MacTalk contains the most commonly used features.



#### **Open**

Opens a setup file from PC and downloads the setup to the motor. If no motor is connected, the setup is shown in MacTalk and can be edited and saved to the PC.

#### **Save**

Saves the actual setup from the motor to a file. If no motor is connected, the actual offline settings (including module setups and program) are saved.

#### **Save in motor**

The complete actual setup in the basic motor will be saved permanently in the flash memory. If the motor is powered down or reset, the saved setup will be used.

#### **Reset position**

Resets the position counter to 0. The content of the position counter can be monitored in the right side of the main screen as "Actual position".

#### **Clear errors**

Clears all the errors (if any). Please note that if an error is still present, the motor will remain in the actual error state.

#### **Reset motor**

Reset the motor. Same as performing a power off / on operation.

#### **Filter Setup**

For specifying the filter setup of the analogue inputs.

#### **STOP motor**

Stops the motor immediately using a controlled deceleration ramp and puts the motor into passive mode. If a program is present this is stopped as well.

This button shall be considered a functional stop button and is available using the keyboard shortcut F8.

Pressing the "Stop" button will immediately stop the motor by changing the currently running mode to "passive" using a fast controlled deceleration curve.

Using a quickstep motor or a module that enables the user to execute RxP programs this execution is also halted to prevent the motor from starting up if a startup-mode is setup from a program.

**Warning! Do not consider this button as an appropriate Emergency stop. Always fit an Emergency stop circuitry to your motor setup.**

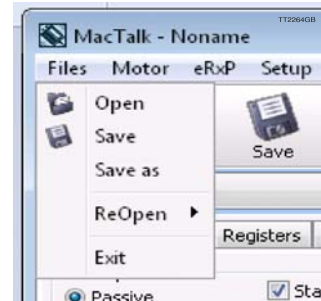
#### **Auto Scan**

If the actual COM port is not known or the motor is setup with an address different from default the Auto Scan feature can help finding the motor(s). Auto Scan is also available when connecting with one of the Ethernet protocols (optional)

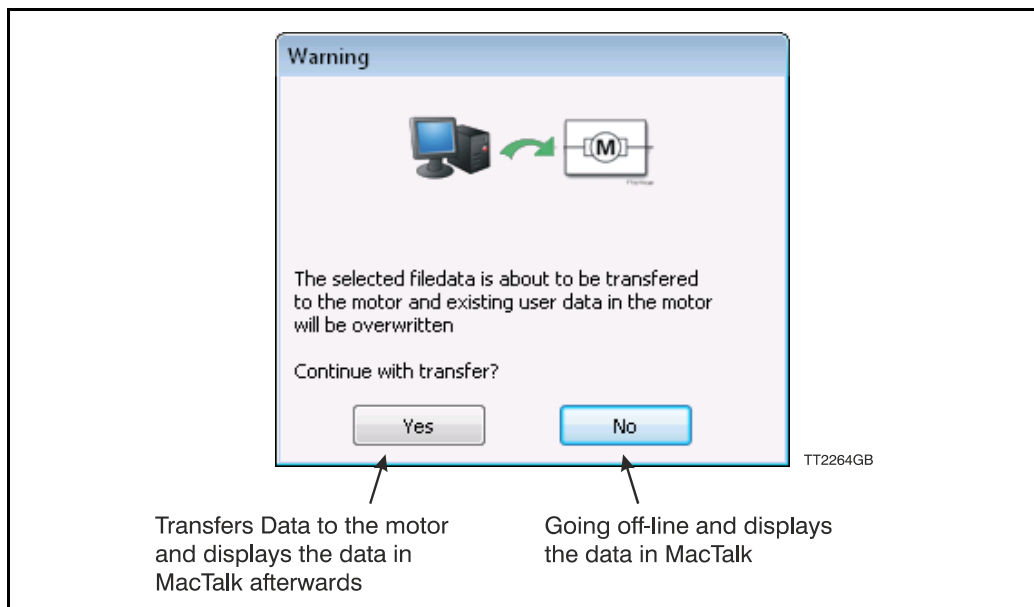
## 4.1 Using the MacTalk software

### 4.1.3 Saving or opening a setup file to/from PC

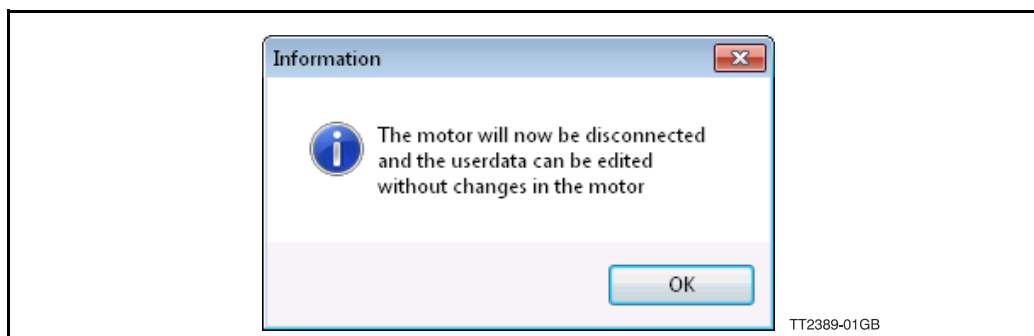
The complete motor setup can be saved to PC or opened from PC and transferred to the motor. Saving and opening a file over a network is also possible. The setup files use the extension MAC. By default, the setup files are saved in the same directory in which MacTalk itself is also installed. Other directories can be selected.



In case a motor is present and a PC file is opened the user is prompted for keeping the connection or going offline and displaying the file-content. The following message box appears.



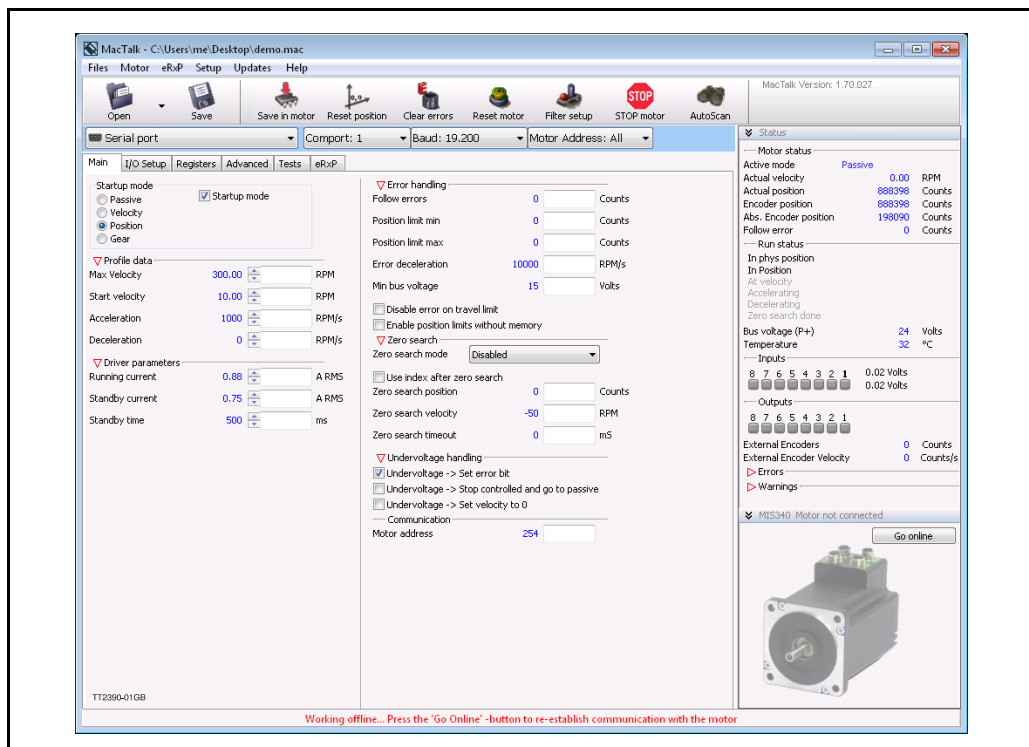
If the user decides to go offline the following text box is presented.



Pressing "OK" disconnects the motor from the PC-application and all data can be edited without any interruption in the motor.

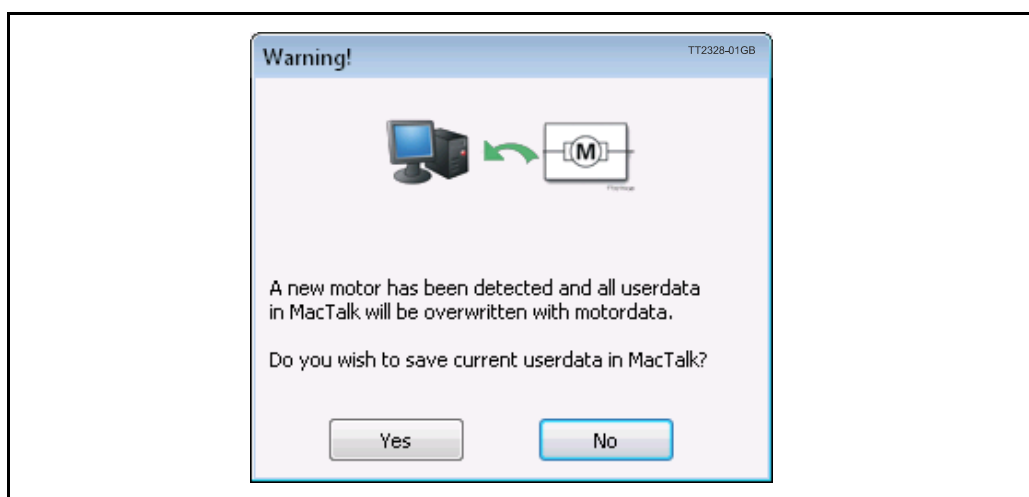
## 4.1 Using the MacTalk software

The following MacTalk view is presented.



As seen in the bottom info line, the motor is disconnected and the file data is currently present in MacTalk. To re-establish communication with the motor, simply press the "Go Online" -button and if any data has been changed a warning box appears enabling the user to save current data before re-establishing communication with the motor as this will overwrite existing data in MacTalk.

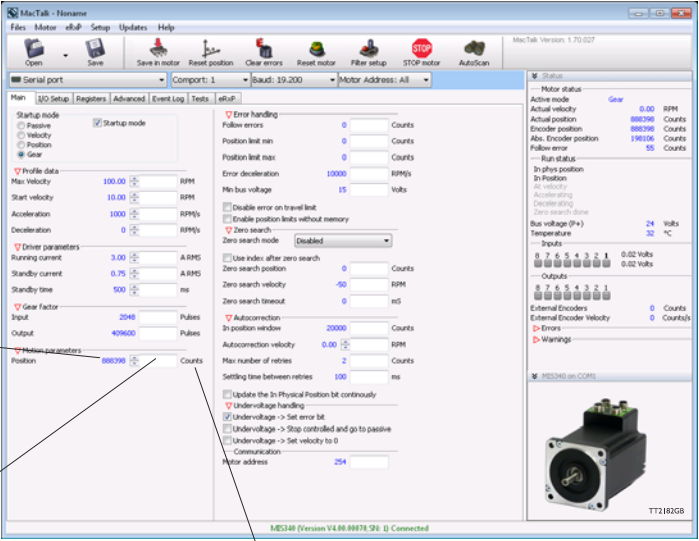
If data is changed in MacTalk the user is warned that current data in MacTalk may be overwritten and needs to be saved. The following warning box is presented.



Choosing "No" will immediately upload all motor data, pressing "yes" will save all data in the open file.

# 4.1 Using the MacTalk software

## 4.1.4 Main Screen

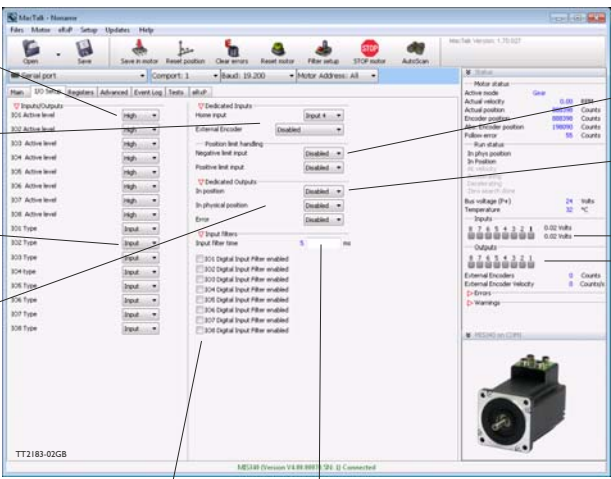


a) This field shows the register values in the controller

b) Here it is possible to key in new values. After pressing enter the value will be transferred to the motor and thereafter be read again from the controller and be shown at point a. Because of digitalizing of the keyed in value, the returned value in a) can be different from the value in b).

c) By pressing the unit field it is possible to change between internal unit in the controller and the unit shown to the user.  
E.g. If user unit for current is ARMS and the internal unit is 5.87mA (300ARMS correspond to 511 units.) Not all registers have different internal and user unit. Speed for example is always specified in RPM.

## 4.1.5 I/O Setup tab



**Active level for the I/Os**  
can be chosen to high or low on inputs

**Dedicated Inputs**  
Selection for Inputs HM, NKL and PL  
An external encoder can also be selected here and defined as either quadrature or pulse/direction type.  
Selection if it shall be Inputs or Outputs

**Dedicated Outputs**  
Selection for outputs "In position", "In Physical Position", "Error" output. It can also be selected if the pulses generated shall be used internally, externally or both and which output should be used for pulse and direction signals

**Input filters**  
Here the filter for the digital inputs can be selected.

**Filter time constant**  
The same value is used for all inputs if filter inputs are enabled.

Selection of Inputs for HM, NL and PL

Selection of output for In-Position and Errors

Status of the inputs \*)

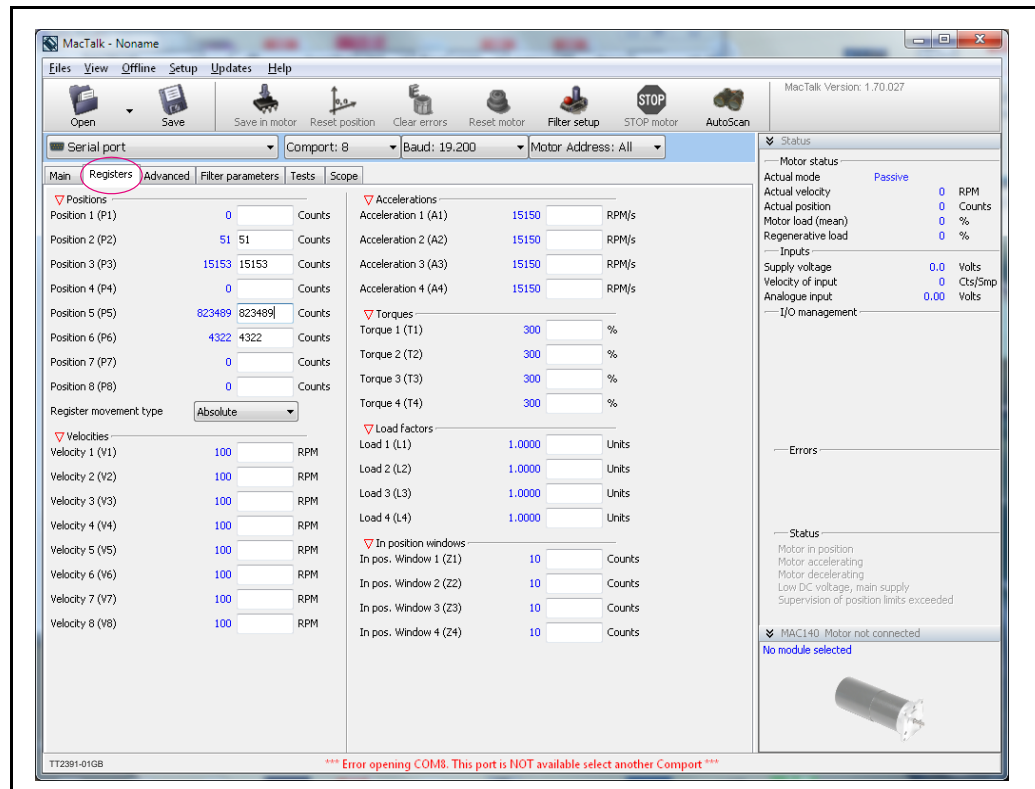
Status of the outputs \*\*)

\*) The analogue value of certain inputs can be read. Click at the input lamp and the analogue value will be shown. The upper value is the actual value and the lower value the filtered value.

\*\*) If the IO's are setup as output(s) they can be activated with a click at the output "LED" in the status window. At the same time the digital and analogue input are activated since they share the same I/O terminal. This can be very useful in test situations when the motor is not installed in the final application with all I/O signals connected.

## 4.1 Using the MacTalk software

### 4.1.6 Register Screen



These registers can be used with FastMac commands. For example, the motor can run to position P2 using velocity V2, acceleration/deceleration A2, running current T2, using only a one byte command.

These values are not updated automatically. To update, place the cursor at the specific register value to the left of the box for new values, and click. Otherwise they only update at motor reset or power up.

When relative moves are made using the MOVE command it uses the Position I (PI) register.

## 4.1 Using the MacTalk software

### 4.1.7 Advanced Screen

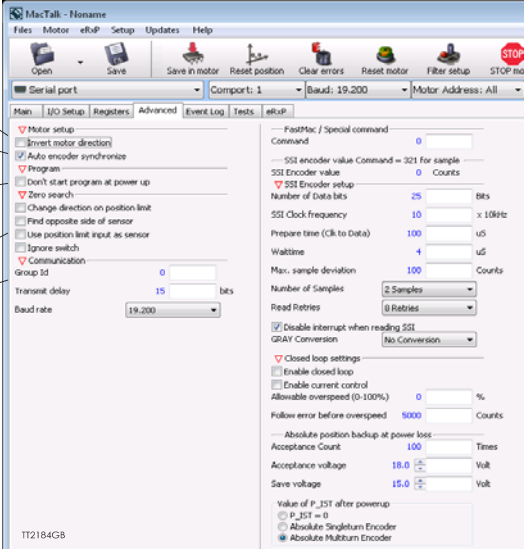
If it is desired to run the motor in the opposite direction it can be done by marking "Invert motor direction"

When this field is marked the motor runs to the AP (Actual position) from the encoder position when the motor goes from passive to position mode

Remove the mark in this field and the motor will start the program at start-up

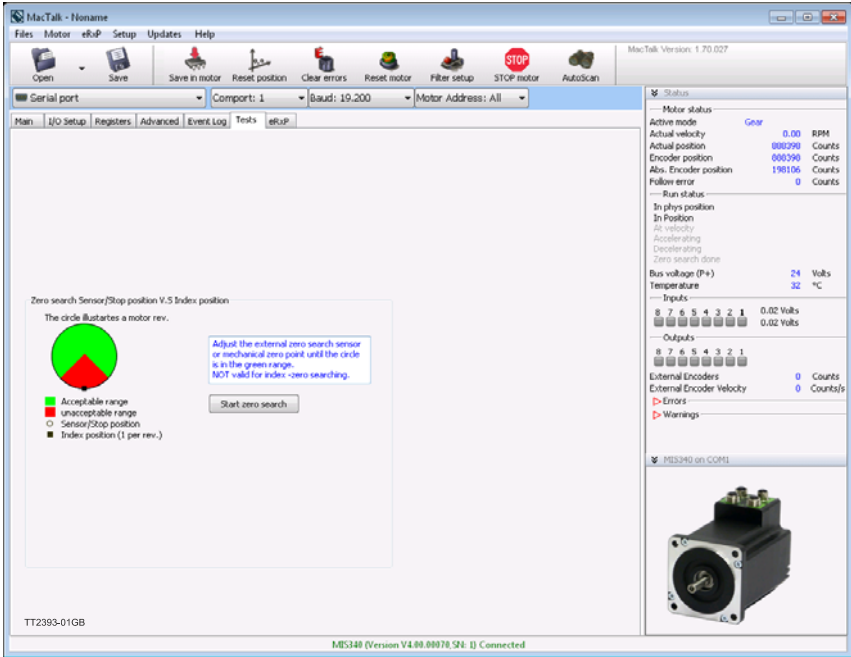
Zero search options can be selected here

It is possible to have a certain number of motors doing the same by giving them the same group id.



The screenshot shows the MacTalk software interface with the 'Advanced' tab selected. The 'Motor setup' section includes checkboxes for 'Invert motor direction', 'Auto encoder synchronize', and 'Don't start program at power up'. The 'Zero search' section has checkboxes for 'Change direction on position limit', 'Find opposite side of sensor', 'Use position limit input as sensor', and 'Ignore switch'. The 'Communication' section shows 'Group id' set to 0 and 'Baud rate' set to 19,200. The right panel shows various motor parameters like 'SS1 encoder value', 'SS1 Encoder setup', 'Number of Data bits', 'SS1 Clock frequency', 'Prepare time', 'Waittime', 'Max. sample deviation', 'Number of Samples', 'Read Retries', 'Disable interrupt when reading SS1', 'GRAY Conversion', 'Closed loop settings', 'Enable closed loop', 'Enable current control', 'Allowable overspeed', 'Follow error before overspeed', 'Absolute position backup at power loss', 'Acceptance Count', 'Acceptance voltage', 'Save voltage', and 'Value of P\_35T after power up'.

### 4.1.8 Test Screen



The screenshot shows the MacTalk software interface with the 'Test' tab selected. The 'Zero search Sensor/Stop position V.S Index position' section displays a circular diagram with a green area for 'Acceptable range' and a red area for 'unacceptable range'. A text box instructs the user to 'Adjust the external zero search sensor or mechanical zero point until the circle is in the green range. NOT valid for index-zero searching.' and includes a 'Start zero search' button. The right panel shows 'Motor status' including 'Active mode', 'Actual velocity', 'Actual position', 'Encoder position', 'Abs. Encoder position', 'Follow error', 'Run status', 'In phys position', 'In Position', 'At velocity', 'Accelerating', 'Decelerating', 'Zero search done', 'Bus voltage (V+)', 'Temperature', 'Inputs', 'Outputs', 'External Encoders', 'External Encoder Velocity', 'Errors', and 'Warnings'. A motor image is shown at the bottom right.

This screen is used for adjusting the Zero search sensor to the correct position when using the index pulse of an encoder. The index pulse should be in the green area. If not, the sensor has to be adjusted.

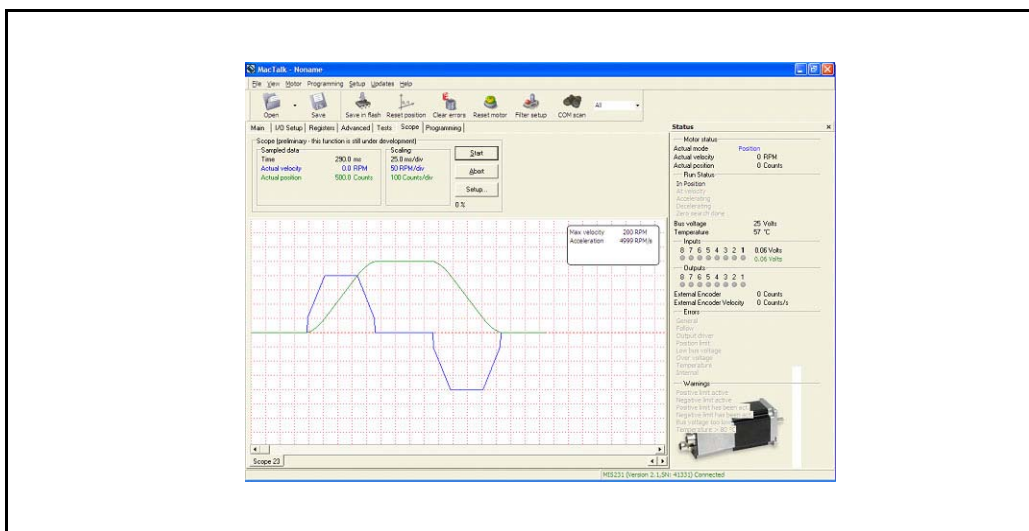
## 4.1 Using the MacTalk software

### 4.1.9 Scope Function

This function is not available yet!

The Scope function is an excellent and necessary function for testing a new application or finding errors in an existing system.

The Setup has to be selected to set up the Scope function correctly before use. Most registers in the MIS motors can be selected for viewing, different trigger functions can be selected, saving and loading scope pictures is possible, etc.



The screenshot shows the 'Sample setup' dialog box. It has a 'Trig type' section with radio buttons for 'Single move', 'Forward and backward move', 'Manual', and 'Register condition'. The 'Manual' option is selected. The 'Register condition' section has a dropdown menu set to '12 - Actual velocity', a comparison operator '>', a value of '10', and a unit of 'RPM'. There is also an 'Edge triggered' checkbox. The 'Time' section has a 'Samples' field set to '512' and a 'Total sample time' of '512 ms'. The 'Trig point setup' section has a 'Time before' field set to '0 ms' and a 'Time after' field set to '512 ms'. The 'Channels' section has four channels: Channel 1 is '12 - Actual velocity', Channel 2 is '10 - Actual position', Channel 3 is 'None', and Channel 4 is 'None'. Each channel has a 'Color...' button. An 'Ok' button is at the bottom right.

## 4.2 How to update MacTalk

MacTalk can be updated directly from the internet at any time. It is recommended always to use the latest version of MacTalk since it support the latest features and bugs may have been found and corrected. Below is shown how to make an update of MacTalk.

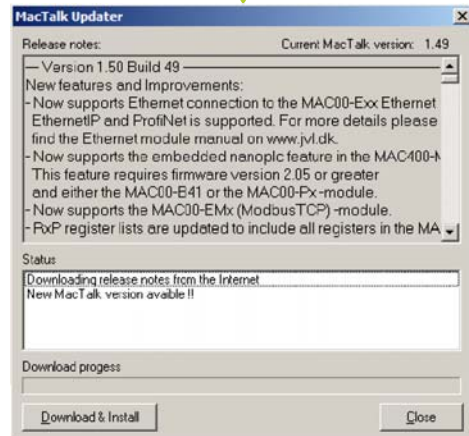
### Step 1

Choose the *Update MacTalk* in the *Updates* menu.



### Step 2

MacTalk will now check if newer version exist on the JVL server. If a newer version exist it will automatically be downloaded and the release notes can be seen in the window.



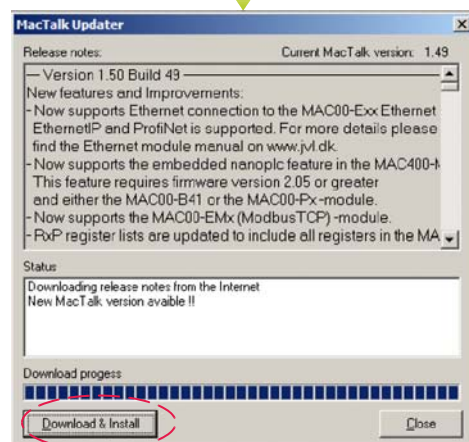
### Step 3

Press the *Download & Install* button to download the selected latest MacTalk.

The progress counter will now rise from 0 to 100%.

The new version is now located in the same directory as the MacTalk which was installed in the first place.

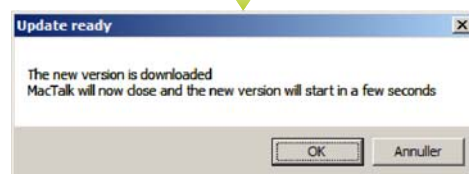
The old version is automatically deleted.



### Step 4

When the download process is finished, the status shows "*Update ready*".

Press "OK" in order to start the new version of MacTalk.



### Step 5

After MacTalk have restarted the version number of the new MacTalk can be observed in the top of the screen.

The complete update is finished !.



TT2342-02GB

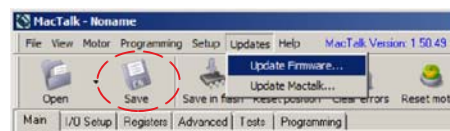
## 4.3 How to update the motor firmware

The firmware in the motor can be updated directly from the internet at any time by using MacTalk.

It is recommended always to use the latest version of the firmware available for the actual MIS motor used since it will contain the latest features and bugs may have been found and corrected. Below is shown how to make an update of the firmware. Notice that the screen dumps below is based on the update of a MIS34x but could be any other size of MIS motor.

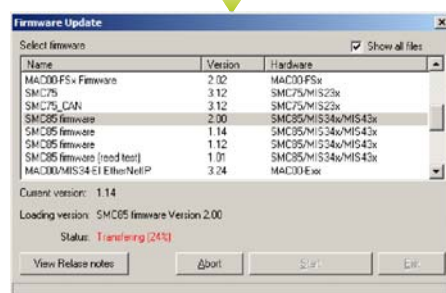
### Step 1

The firmware update will erase the existing user setup of the motor. Use the Save button to save the existing setup before updating the motor. Then choose the *Update Firmware* in the *Updates* menu.



### Step 2

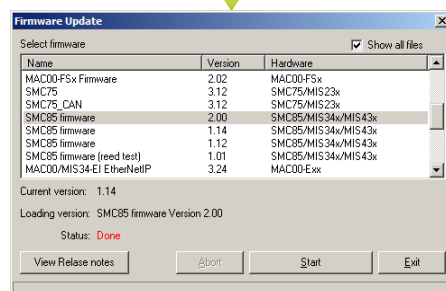
The first list shown is only the newest firmwares related to the actual motor connected. It may also contains encoder and/or Ethernet firmware if these features are present. To see all files also older versions enable the checkbox "Show all files".



Select the desired firmware, «SMCxxx firmware». Press *Start* to download the selected firmware. The progress counter will now rise from 0 to 100%.

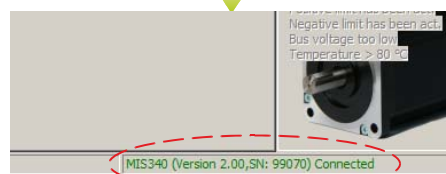
### Step 3

When the download process is finished, the status shows "Done".



### Step 4

The on-line information shown in the lower right corner of the MacTalk main window will now show the complete type of firmware and version.



The firmware update is now fully completed. Please remember that the settings of the motor is set back to default. But can be reinstalled by opening the user setup file made initially in this update sequence.

TT2343-01GB

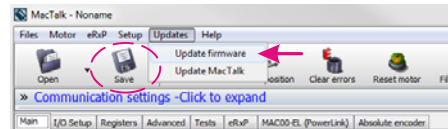
Hint!: Some older products may not start after pushing the "start" button showed above. If this is the case simply switch off power wait 5 seconds and reapply power. The update should now start.

## 4.4 How to update the encoder FW Only MISxxx---H3/H4

If the motor has the H3 or H4 (absolute multiturn encoder feature) then the firmware can be updated directly from the internet very easy at any time by using MacTalk. It is recommended always to use the latest version of the firmware available for the actual MIS motor used since it will contain the latest features and bugs may have been found and corrected. Below is shown how to make an update of the encoder firmware.

### Step 1

The encoder firmware update will erase the existing user setup of the motor. Use the **Save** button to save the existing setup before updating the motor. Then choose the **Update Firmware** in the **Updates** menu.

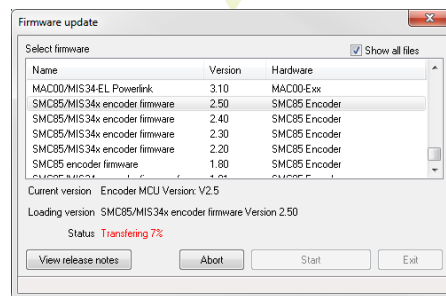


### Step 2

The first list shown is only the newest encoder firmwares related to the actual motor connected. It may also contains main and/or Ethernet firmware if these features are present.

To see all files also older versions enable the checkbox **"Show all files"**.

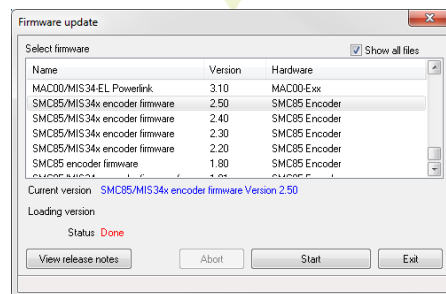
Select the desired firmware, **«SMC... encoder firmware»**. Press **Start** to download the selected firmware. The progress status counter will now rise from 0 to 100%.



### Step 3

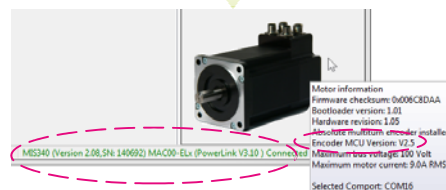
When the download process is finished, the status shows **"Done"**.

If the download process for some reason did NOT start/succed start from step 1 in this guide but switch off power until the **«Start»** button have been activated and then switch on power.



### Step 4

The on-line information shown in the lower right corner of the MacTalk main window will now show the complete type of main firmware and optionally the ethernet firmware version (if ethernet is present) including version. The encoder firmware version is shown by placing the mouse cursor at top of the motor for a short while.



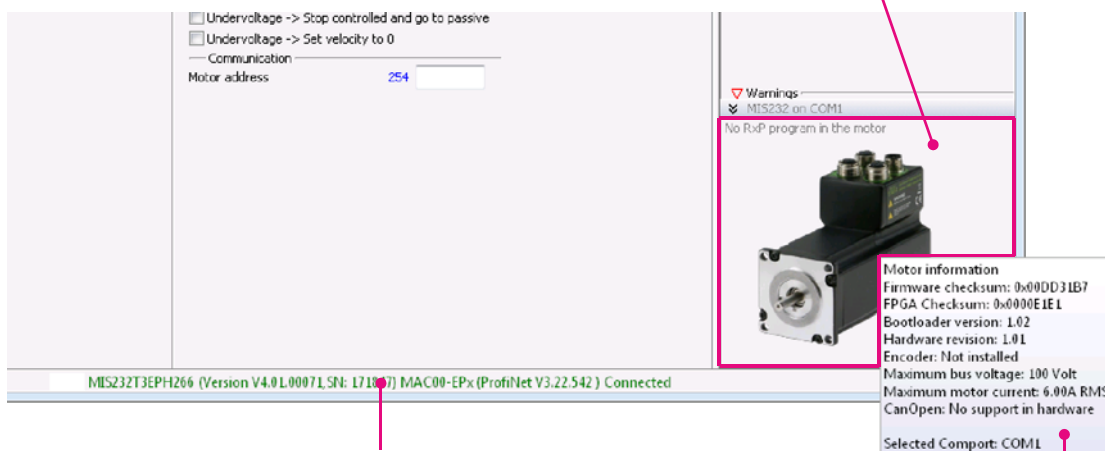
The firmware update is now fully completed. Please remember that the settings of the motor is set back to default. But can be reinstalled by opening the user setup file made initially in this update sequence.

TT2349-01GB

**Hint!:** Some older products may not start after pushing the "start" button showed above. If this is the case simply switch off power wait 5 seconds and re-apply power. The update should now start.

## 4.5 How to get SW/HW motor info

Place the mouse cursor in this field and the motor information box will show up



Info about which motor type and options that is connected with MacTalk - Also the serial number is monitored here

Info about SW and HW setups in the motor.

Description of contents in info box	Description	
Firmware checksum	Unique identifier for the installed firmware part 1. The firmware can be updated with MacTalk.	
FPGA checksum	Unique identifier for the installed firmware part 2. The firmware can be updated with MacTalk.	
Bootloader version	The program that handles firmware updates. The bootloader itself is installed during manufacturing.	
Hardware version Note: Internal controller SMC66 is used in MIS17x and MIS23x. Internal controller SMC85 is used in MIS34x and MIS43x.	The version of the controller/driver. All hardware versions supports the documented features in this manual except SMC85 (used in MIS34x and MIS43x) where that have been updated between version 1.05 and 1.07 (version 1.06 do not exist)	
	SMC85:	
	1.05	1.07
	H2 encoder is 10 bit and does not support closed loop.	H2 encoder is 10/12 bit and supports closed loop.
		H4 encoder option available.
	Event log and actual position are saved in flash at shutdown. If CVI drops too rapid, the informations are not saved.	Event log and actual position is saved every second and can be monitored in MacTalk during motor operation.
	Limited memory for future improvements.	Increased memory for future improvements.
	Only external brake is available.	Both external and integrated brake can be mounted.
	SMC66 From version 1.01 has the same features as SMC85 1.07.	

## 4.5 How to get SW/HW motor info

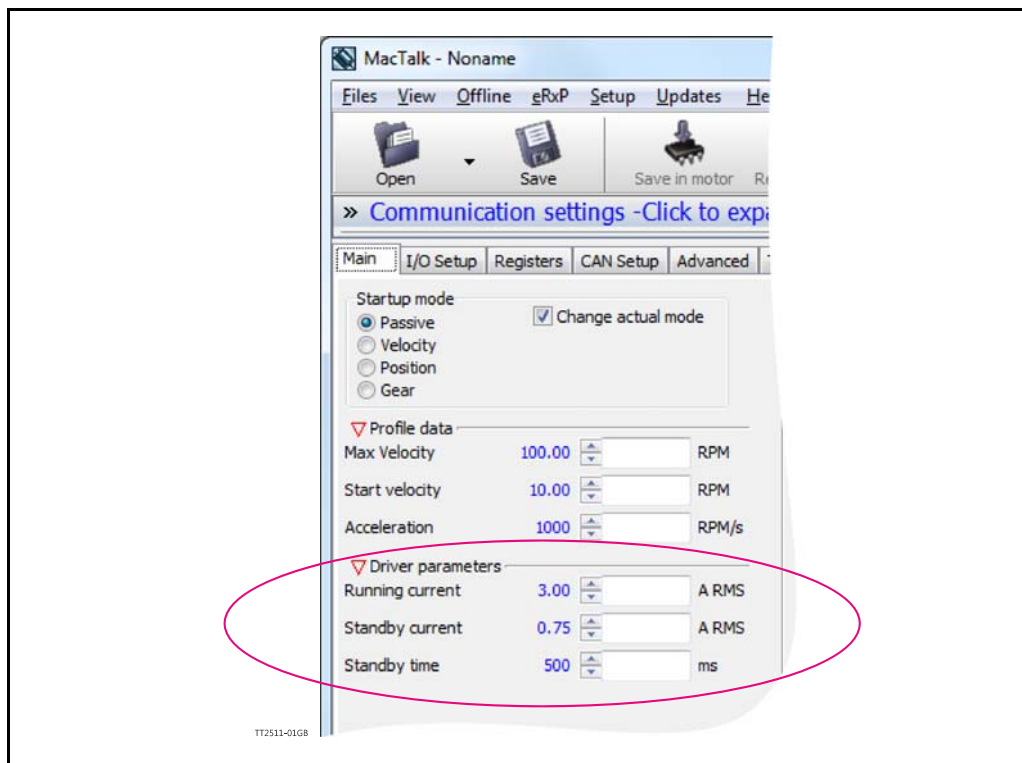
---

<b>Absolute multi turn encoder installed</b>	The absolute multiturn encoder option H3/H4 is installed.
<b>Encoder: Not installed</b>	An H2 encoder is not installed. Therefore the motor can only operate in open loop and will not be able to correct the position in case the motor is stalling.
<b>Encoder MCU Version</b>	The absolute multi turn encoder firmware version, can be updated with MacTalk.
<b>Maximum bus voltage</b>	The limit where the “Over voltage” error bit will be set.
<b>Maximum motor current</b>	The maximum motor Running and Standby current is defined by this value.
<b>CANopen</b>	An option which can be installed during the manufacturing.
<b>Selected Comport</b>	The actual serial com port the motor is connected to.



## 5.1 Setting up the motor current

The current supplied to each of the step motor's phases can be adjusted for standby and running currents by changing the values of standby and running currents under "Driver parameters" in the MacTalk program. The register is also accessible in general through the serial interface.



The electronics automatically switches between the two currents by detecting the presence of step-pulses. If a rising edge is detected at the step-clock, the "Move current" is selected. If no rising edge is detected during the period specified by "Standby time" at the step-clock input, the current is automatically switched back to "Standby current".

Values for the two currents are typically adjusted so that the Operating Current is significantly higher than the Standby Current, since the motor must be supplied with more power to drive its load during acceleration and constant operation than when it is stationary.

### 5.1.1 Standby current considerations

The main reason for having the Standby current setting is to optimise the heat produced by the motor.

Having a suitable standby current will make sure that the motor do no loose position but also make sure that the motor is not producing more heat than necessary.

A typical setting of the Standby Current typically is 30-40% of the Running current.

Normally the motor do not need to produce any significant torque during standby and therefore it makes sense to lower the standby current.

## 5.1 Setting up the motor current

### 5.1.2 MIS17x Current and torque relation

	MIS171	MIS173	MIS176				Unit
Standby Current	0 - 4000	0 - 4000	0 - 4000				mA RMS
Running Current	0 - 4000	0 - 4000	0 - 4000				mA RMS
Holding Torque	0 - 0.36 [0 - 51]	0 - 0.56 [0 - 79.3]	0 - 0.8 [0 - 113.3]				Nm [Oz-In]

### 5.1.3 MIS23x Current and torque relation

	Standard torque versions			High torque versions			Unit
	MIS231Q or MIS231S	MIS232Q or MIS232S	MIS234Q or MIS234S	MIS231T or MIS231R	MIS232T or MIS232R	MIS234T or MIS234R	
Standby Current	0 - 6000	0 - 6000	0 - 6000	0 - 6000	0 - 6000	n/a	mA RMS
Running Current	0 - 6000	0 - 6000	0 - 6000	0 - 6000	0 - 6000	n/a	mA RMS
Holding Torque	0 - 1.2 [0 - 170]	0-1.9 [0 - 269]	0 - 3.0 [0 - 425]	0 - 1.2 [0 - 170]	0 - 2.5 [0 - 354]	n/a	Nm [Oz-In]

### 5.1.4 MIS34x Current and torque relation

	MIS340	MIS341	MIS342	MIS343			Unit
Standby Current	0 - 6000	0 - 6000	0 - 6000	0 - 6000			mA RMS
Holding Torque	0 - 2.0 [0 - 283]	0 - 4.1 [0 - 575]	0 - 6.0 [0 - 849]	0 - 8.0 [0 - 1133]			Nm [Oz-In]
Running Current	0 - 9000	0 - 9000	0 - 9000	0 - 9000			mA RMS
Holding Torque	0-3.0 [425]	0-6.1 [863]	0-9.0 [1274]	0-12.0 [1699]			Nm [Oz-In]

### 5.1.5 MIS43x Current and torque relation

	MIS430	MIS431	MIS432				Unit
Standby Current	0 - 6000	0 - 6000	0 - 6000				mA RMS
Holding Torque	0 - 6.7 [0 - 944]	0 - 14.0 [0 - 1983]	0 - 16.7 [0 - 2360]				Nm [Oz-In]
Running Current	0 - 9000	0 - 9000	0 - 9000				mA RMS
Low speed torque 1 RPM	0-10.0 [0 - 1416]	0-21.0 [0 - 2974]	0-25.0 [0 - 3540]				Nm [Oz-In]

### 5.1.6 Current and torque relation

If a MIS232 motor is used and the current is set to 6000 mA, the motor will be able to deliver a torque of 1.9 Nm at low speed. If the current is set to 3000 mA, the motor will be able to deliver 0.85Nm.

In other words the torque produced is proportional to the current setup at low speeds. At increased speed the relation stops to be proportional since the motor current which is setup can not be achieved fully due to the motors electrical and mechanical timeconstants.

See also [Run\\_Current](#), page 198 for information about Running Current and [Standby\\_Current](#), page 199 for information about Standby Current.

### 5.2.1 Position "Auto correction"

This feature is only active when the motor is in Position Mode. Also the function is only active if the motor is equipped with one of the encoder options -H2/H3 or H4. The function can be called a semi-closed loop feature since it do only correct the motor position after a move have been done or tried done and not dynamically during a move like closed loop will do. See also [Introduction to closed loop operation.](#), page 104. The auto correction feature can be useful to assure that the motor reach its target position. The feature will take effect if for example the movement was physically blocked, the torque of the motor was insufficient, or a bad value for start velocity or acceleration were used. It might also be used to handle occasional mechanical oscillations.



**The Auto correction feature is not recommended to use if the motor also has enabled the closed loop regulation.**

### 5.2.2 "Auto correction" - basic function

The auto-correction feature is only used when the motor has stalled and not reached its final target position within the given position window.

Each time the motor has done a movement the "Actual position" counter and the "Encoder position" counter is compared.

If the difference without sign is within the value specified in the "In position window" as shown below no further action is taken.

If the difference is larger than the value in the "In position window" the motor will try to correct the position by doing a new motor movement. The "Max number of retries" is the number of times the motor will try to correct the position, and the "Settling time between retries" is the time the motor will wait between each retry.

The AutoCorrection system will first wait (unconditionally) for a certain time (settling time) to allow the initial movement to settle mechanically before testing for the target position. It will then attempt a normal movement, using the same values for velocities and acceleration as the movement that failed. It will continue until the encoder position is within the target window, or the selected number of retries has expired.

Note that AutoCorrection will only start after the value of the Position (P\_SOLL) register is changed. In other words, changing P\_SOLL (not just writing the same value again) will reload the maximum number of retries and set the Auto Correction Active status bit. The Auto Correction Active status bit will remain set until either the position is within the target window or the max number of retries has been exhausted.

Also note that if the motor is used to control other motors by sending out the pulse and direction signals on digital outputs, any extra movements caused by AutoCorrection will send out additional steps to the other motors.



**The Auto correction feature is not recommended to use if the motor also has enabled the closed loop regulation.**

## 5.2

## Auto Correction

Only MISxxx---H2/H3/H4---

### 5.2.3 Setup with MacTalk

Following parameters are available in MacTalk.

When selected the in position flag will realtime indicate if the motor is within the position window compared to a perfect move.

An autocorrection after end movement is performed if the difference between the actual position and the encoder position is more than the value specified in this field.

Velocity used during a retry

The max number of autocorrection retries per movement is specified here.

The settling (delay) time between the retries when doing autocorrection.

TT2338-01GB

#### In position window

Defines the window wherein the motor must be before the In Physical Position flag is set.

#### Autocorrection velocity

Defines the velocity used if a correction is done. Can be useful when Ethernet or other protocol is used since the main velocity register can be overwritten with a velocity value which is not optimal for auto correction.

#### Max number of retries

Defines how many auto correction retries that are allowed to be done

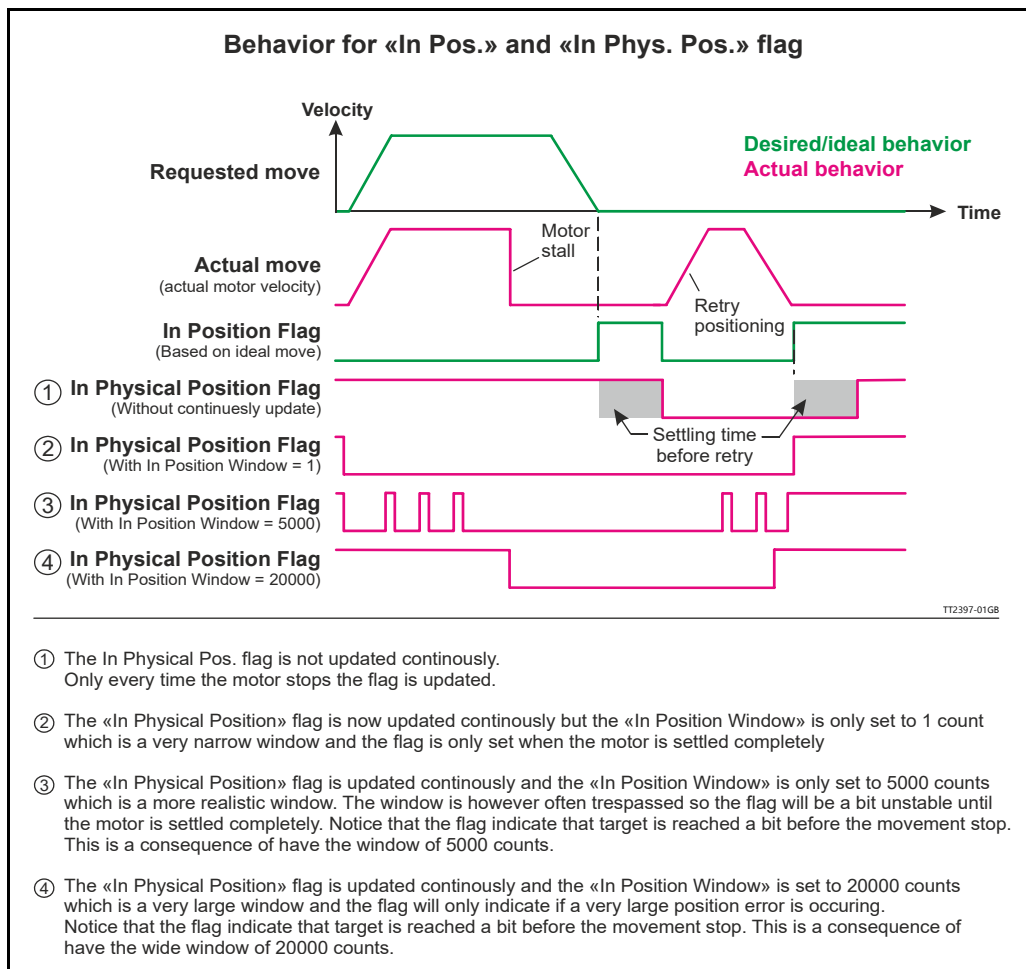
#### Settling time between retries

Defines defines the time between the auto correction retries

#### Update the In Physical Position bit continuously

Defines if the In Physical Position bit is updated continuously or only after the motor has stopped (default)

The relation between settings and behaviour of the In Physical Position flag can be seen below.



#### 5.2.4 Setup without MacTalk

If MacTalk is not used for setting up parameters/registers related to the auto correction feature it must be done as follows.

The motor contains a number of registers which can be accessed from various protocols depending at which options the motor has.

Protocols available are for example Ethernet (EthernetIP, Profinet etc.) and CANopen, Modbus or the MacTalk protocol.

Each field in MacTalk described earlier in this chapter is accessing a register in the motor. The registers that are relevant for auto correction operation are:

##### R33 - IN\_POSITION\_WINDOW

##### MacTalk name: "In Position Window".

Selects how close the internal encoder position must be to the target Position (P\_SOLL) to set the InPhysical-Position status bit and prevent further AutoCorrection.

See also: [IN\\_POSITION\\_WINDOW](#), page 207

**R236 - V\_SOLL\_AUTO****MacTalk name: “Auto correction velocity”.**

The auto correction is done per default with the velocity specified in the general velocity register. If an alternative velocity is intended the V\_SOLL\_AUTO register can be used. If V\_SOLL\_AUTO != 0 it will be used instead of the general velocity.

See also: [V\\_SOLL\\_AUTO](#), page 238

**R34 - IN\_POSITION\_COUNT****MacTalk name: “Max. number of retries”.**

Specifies the maximum number of auto correction retries before no further attempts are done. A value of 0 (zero) effectively disables AutoCorrection.

See also: [IN\\_POSITION\\_COUNT](#), page 207

**R110 - SETTLING\_TIME****MacTalk name: “Settling time between retries”.**

When the internal encoder option is installed and register 34, InPositionCount, is non-zero so AutoCorrection is enabled, the value in this register defines how many milliseconds to wait after each movement attempt before testing whether the encoder position is within the target window as defined in register 33. This waiting time is often necessary to allow mechanical oscillations to die out.

See also: [Settling Time](#), page 217

**R124 - SETUP\_BITS****MacTalk name: “Update the In Physical Position bit continuously”.**

Bit no. 6 defines if the In Physical Position bit should be updated continuously or not.

Default: Bit 6 = 0 = only update after motor stops.

See also: [Setup\\_Bits](#), page 220

**R25 - STATUSBITS****MacTalk name: (Run status area)**

This register contains 2 bits that are relevant for the auto correction feature.

**Bit 1: AutoCorrection Active**

If set an auto correction cycle is in progress because target position was not met.

**Bit 2: In Physical Position**

If set the motor position is physically within the In\_Physical\_Position\_Window

See also: [Status bits](#), page 205

### 5.3.1 Introduction to closed loop operation.

When running a stepper motor under normal operating conditions the load torque will cause a small displacement of the rotor from its nominal position, but normally such displacements do not lead to a loss of synchronization.

However, synchronization will be lost if the load exceeds the available motor torque.

This can cause the motor to stall with a position loss which must be corrected afterwards – for example with the *Auto correction* function (see also [Auto Correction](#), page 100) or by monitoring the position externally with an encoder.

The MISxxH2xx (only available from serial numbers 173000) and MISxxH4xx have the ability to run in **closed loop** and therefore always tracks the rotor displacement in real time. The control algorithm aligns the commutation angle and motor current when needed. This avoids that the motor is stalling and runs the motor at a lower current when possible with the advantage that the overall system efficiency is much better.

#### The closed loop offers 2 ways of operation

1. Closed loop with constant current. The current is maximum all the time regardless which load is applied to the motor.
2. Closed loop with dynamic current control. The current is adjusted real time to match the actual load. The advantage is that the motor runs more efficient (less heat) and the audible noise when running is much less.

**Note:** When "closed loop" is enabled, then the InPosition flag changes its behaviour so that it will act as the "InTargetPosition" bit, which means that it will tell if the encoder position and P\_SOLL are within the "in position window".

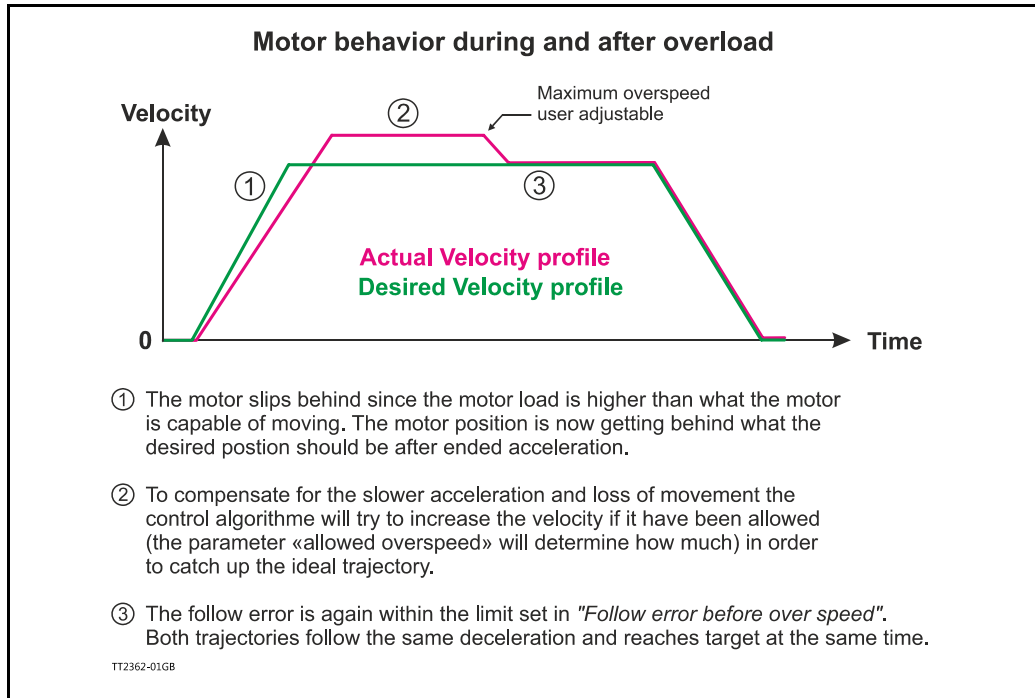
## 5.3 Closed loop operation

Only MISxxx---H2/H4---

### 5.3.2 Examples of motor behaviour in closed loop

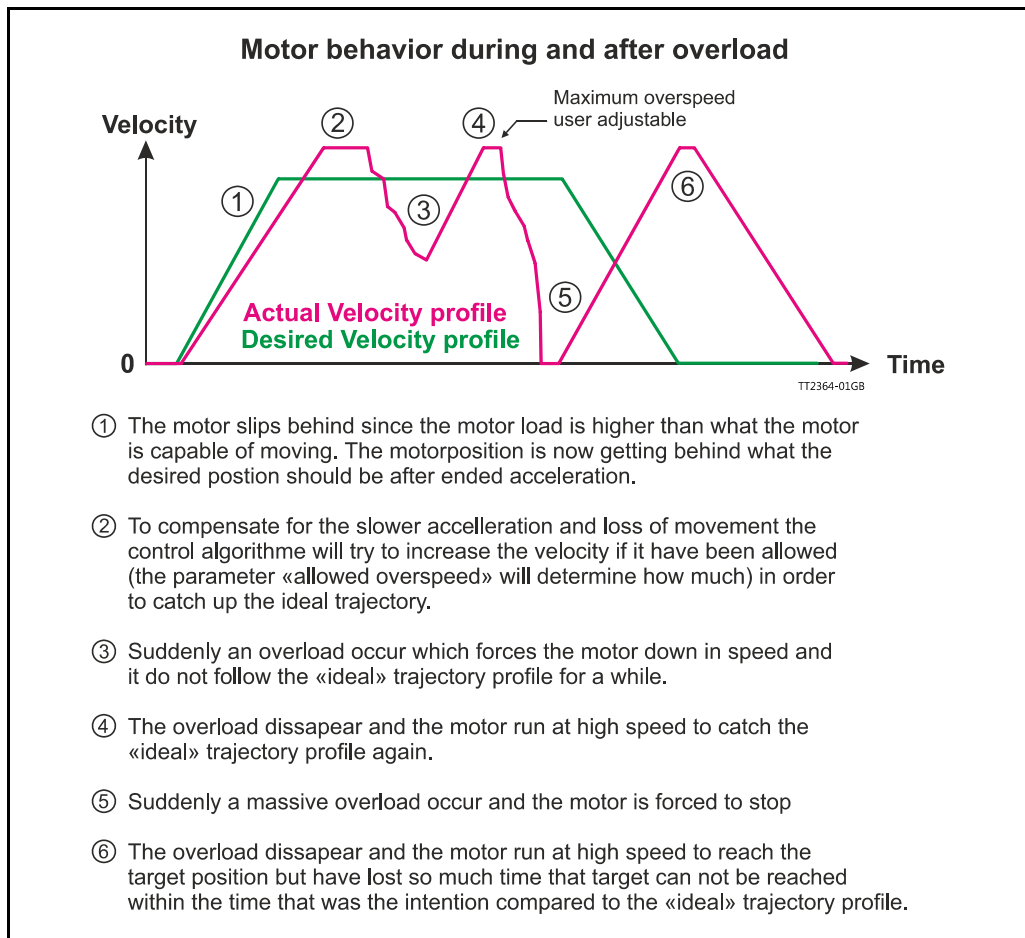
#### Example 1:

A too high acceleration has been set. The motor cannot accelerate the load fast enough and therefore a follow error will be incremented until the motor is able to maintain the right speed.



**Example 2:**

The motor is not able to follow the ideal acceleration and therefore increases the velocity to catch up the follow error. During the movement overloads also occur forcing the motor to go down in speed because lack of torque to overcome the load.

**5.3.3 Current control (optional)**

In a classic stepper motor system (not closed loop) the motor current is typically set to maximum to make sure that the motor do not stall and loose track of its position. The disadvantage is that the motor becomes hot and energy is wasted.

When running in closed loop, loss of position (motor stall) is not a concern and therefore the control algorithms current control will (if enabled) adjust the current to a level where the motor is able to follow the requested velocity and not loose the position.

The actual running current ("Actual torque" in MacTalk) is a percentage of the user defined "Running current". The motor can never run with a higher RMS current than the one specified in the "Running current" register, and the algorithm decreases the actual running current according to the follow error.

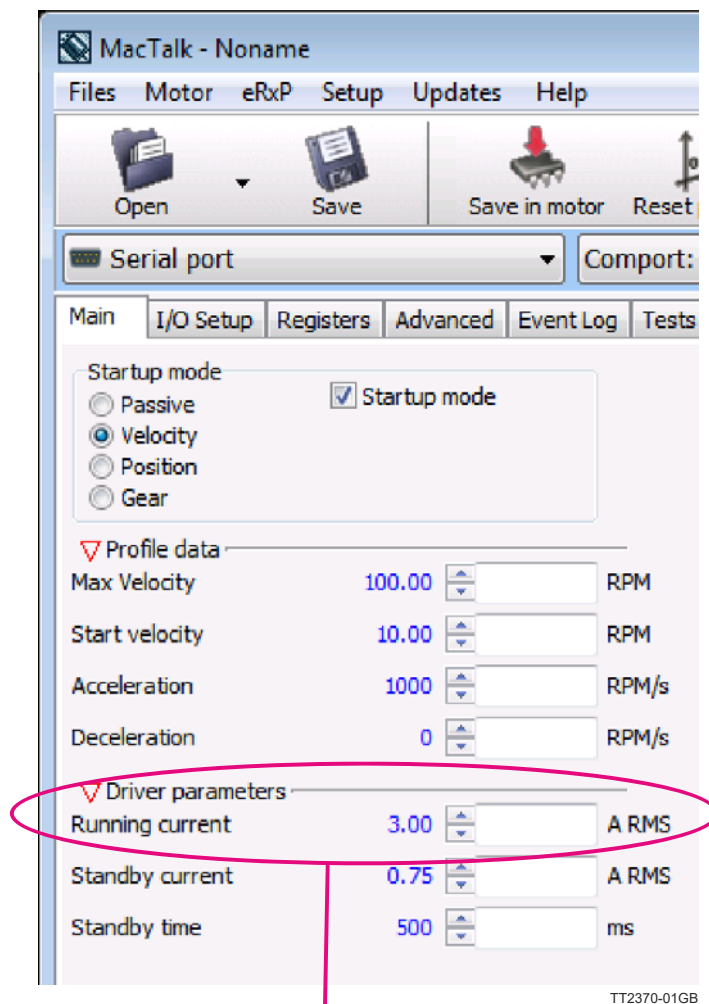
## 5.3

## Closed loop operation

Only MISxxx---H2/H4---

The allowed maximum current that can be used when the motor is running is setup in the “Running current” register which is found at the main tab in the left side. Notice that when using the closed loop current control feature only the “Running current setting is used”.

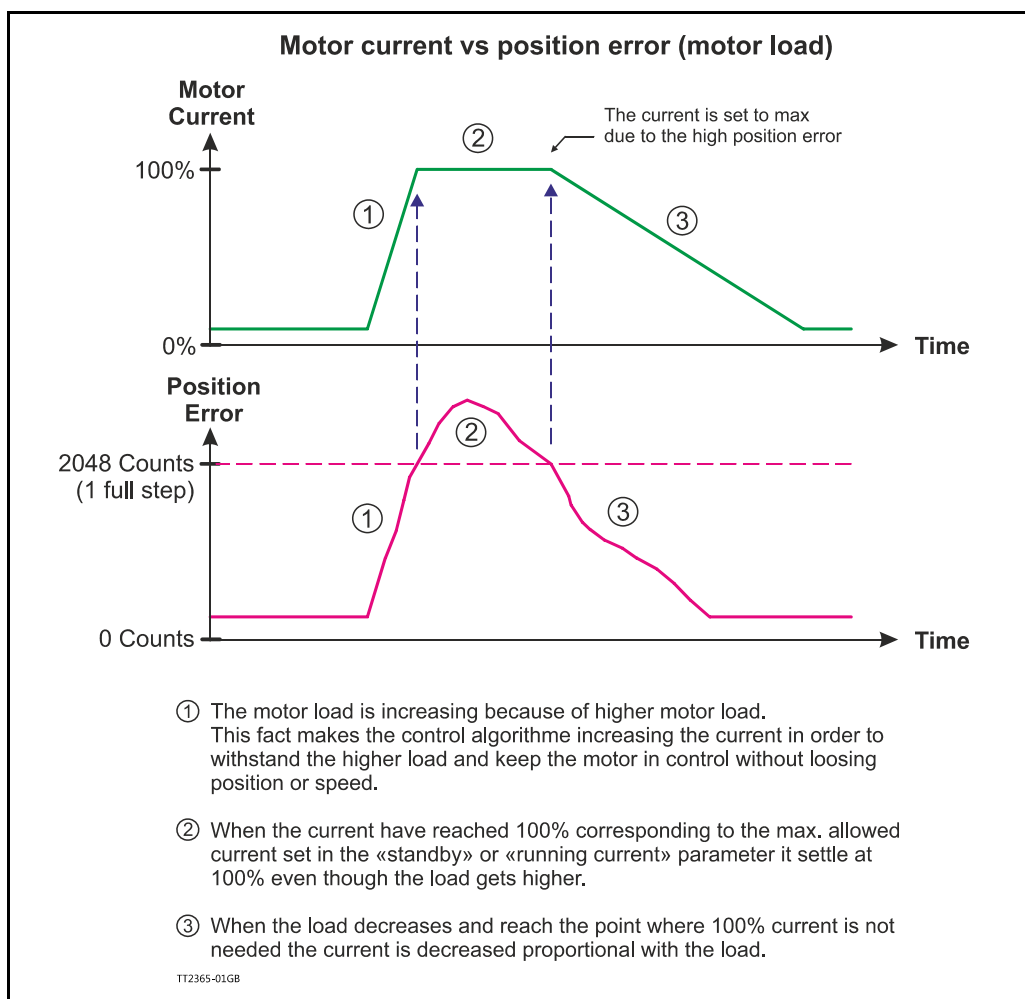
The “Standby current” and “Standby time” setting is **not** used at any time.



Running current can be set here  
Standby current and Standby time is not used  
when current control and closed loop is enabled.

**Example:**

The example shows a situation where the motor is running at a steady velocity and the follow error is stable. Suddenly the motor is temporarily overloaded, motor current is increased. When the current has reached 100% it settles since the electronics can not produce more than 100% and also have to respect the user current setting. The follow error still increases because the load on the shaft exceeds the available torque. In the last part of the cycle the load is lowered and when the position error becomes below 2048 counts (1 full step) the current is also proportionally decreased by the control algorithm and it settles at the same level as before the increased load/position error. As illustrated on the figure, the slope of the current increase and decrease are asymmetrical. This is to stabilize the current control.



## 5.3

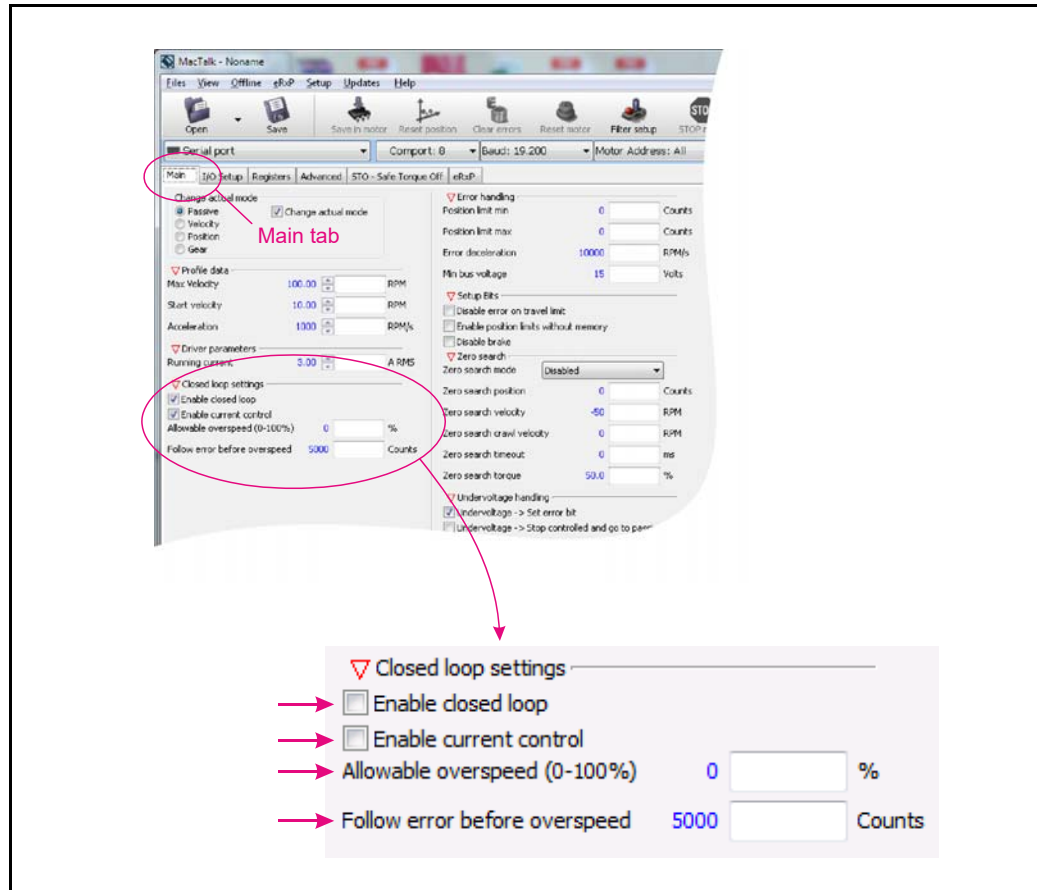
# Closed loop operation

Only MISxxx---H2/H4---

### 5.3.4 Setup with MacTalk

From firmware version 4.00 the closed loop control is available, if either the H2 or H4 encoder option is installed. Only motors with serial number 173000 or higher support the closed loop feature.

The closed loop function is by default enabled, also the current control is by default enabled.



Parameter explanations:

#### Enable closed loop

Check this field to activate the overall closed loop feature. Can be activated/deactivated on the fly and have immediate effect. Default: inactive.

#### Enable current control

Check this field to add current control. The current control will increase/decrease the current proportional to what is needed for driving the motor load. Can be activated/deactivated on the fly and have immediate effect. Default: inactive.

#### Allowable over speed (0-100%) and Follow error before overspeed

Allows the motor to run with a higher velocity if the encoder position deviate from the theoretical position by more than "Follow error before overspeed" counts.

The default for "Allowable overspeed..." is 0% which means that the maximum speed will never exceed the "Max velocity" setting.

The default for "Follow error before overspeed" is 5000 counts.

## 5.3

## Closed loop operation

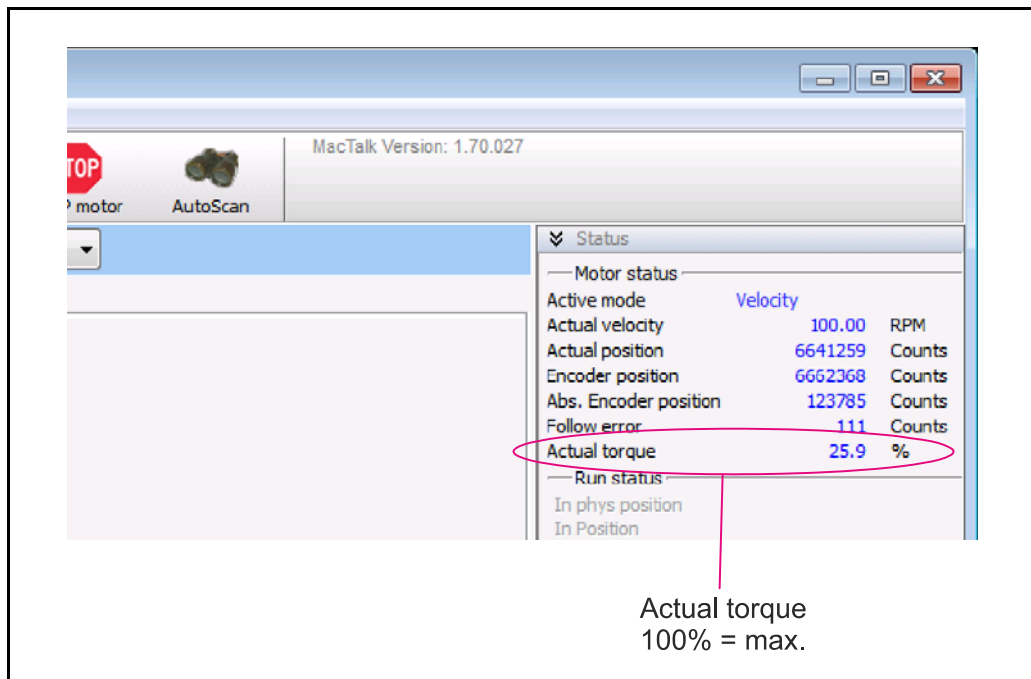
Only MISxxx---H2/H4---

### 5.3.5 Monitoring the actual motor torque

When using the closed loop feature and having the current control activated at the same time - the actual torque can be monitored.

The actual torque read-out is based on the actual motor current as a percentage of the “Running current” setting.

In MacTalk it is labelled “Actual torque”. Its monitored among the other motor status values in the right side of the main screen.



The “Actual torque” is defined from how many percentage of the “Running current” that is used.

**5.3.6 Setup without MacTalk**

If MacTalk is not used for setting up parameters/registers related to the closed loop feature it must be done as follows.

The motor contains a number of registers which can be accessed from various protocols depending at which options the motor has.

Protocols available are for example Ethernet (EthernetIP, PROFINET etc.) and CAN-open, Modbus or the MacTalk protocol.

Each field in MacTalk described earlier in this chapter is accessing a register in the motor. The registers that are relevant for closed loop operation are:

<b>R25</b>	<b>STATUS_BITS</b>	Overall status related to the closed loop feature. - Bit 15: Closed loop lead/lag - Bit 16: Closed loop activated - Bit 17: Internal encoder calibrated - Bit 20: Internal encoder ok See also: <a href="#">Status bits</a> , page 205
<b>R35</b>	<b>ERR_BITS</b>	Error status related to the closed loop feature. Following bits are available in the setup bit register. - Bit 12: Closed loop error. - Bit 14: Abs. single turn encoder (H2/H4) error. See also: <a href="#">Err_Bits</a> , page 208
<b>R124</b>	<b>SETUP_BITS</b>	This register is used to activate or deactivate the closed loop and current control feature. Following bits are available in the setup bit register. - Bit 24: Enable closed loop - Bit 25: Enable closed loop current control See also: <a href="#">Setup_Bits</a> , page 220
<b>R217</b>	<b>ACTUAL_TORQUE</b>	The actual motor current in closed loop with active current control. 2047 = 100% current. See also: <a href="#">ACTUAL_TORQUE</a> , page 236
<b>R245</b>	<b>CL_CATCH_UP</b>	In case the motor lags behind in closed loop mode this register determine which over speed that is allowed to catch up to the right position. It also determines how much the motor can slip behind before the overspeed is used to catch up. - Bit 0-7: Allowable overspeed in percent (0-100) - Bit 8-31: Follow error limit before overspeed is used (encoder counts). See also: <a href="#">CL_CATCH_UP</a> , page 240

**Other registers that may be relevant for the closed loop operation:**

<b>R5</b>	<b>V_SOLL</b>	The maximum (nominal) velocity allowed. See also: <a href="#">V_SOLL</a> , page 198
<b>R6</b>	<b>A_SOLL</b>	The acceleration/deceleration ramp to use. See also: <a href="#">A_SOLL</a> , page 198
<b>R7</b>	<b>RUN_CURRENT</b>	The maximum motor current is setup in this register. See also: <a href="#">Run_Current</a> , page 198

## 5.3.7 Special settings



**Please note:** The following parameters are optional and it is not recommended to modify them since they are optimized from factory.

The following 2 registers are relevant for the closed loop operation

**Motor commutation and encoder setup:**

1. KPHASE - register that offsets the magnetic field as function of the velocity  
See also: [Internal\\_Encoder\\_Setup](#), page 232
2. Internal\_Encoder\_Setup - register that define the encoder resolution and other encoder related parameters. See also: [Internal\\_Encoder\\_Setup](#), page 232

**Current control advanced registers:**

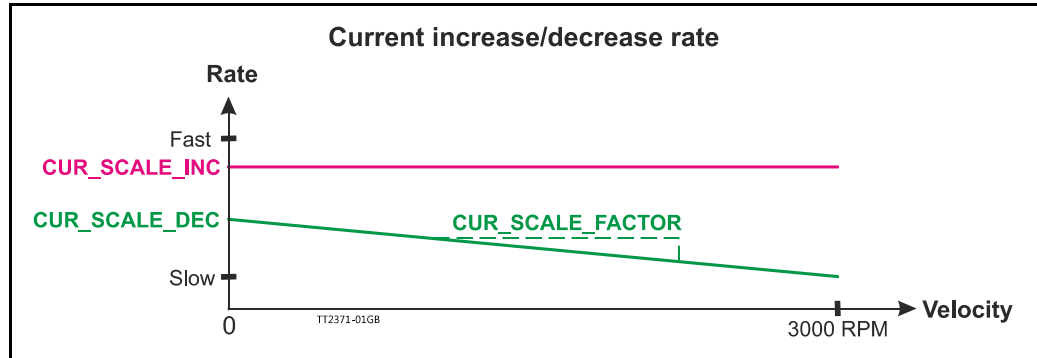
None of the below shown registers are accessible from MacTalk but only from other protocols such as Ethernet, Modbus, CANopen etc.

The registers shown below are all related to optimizing the current control performance.

<b>R212</b>	<b>CUR_SCALE_MAX</b>	The maximum allowable current. 2047 = 100% (of Running current). See also: <a href="#">CUR_SCALE_MAX</a> , page 234 !! - its recommended not to use this parameter. Better change the "Running current"
<b>R213</b>	<b>CUR_SCALE_MIN</b>	The minimum allowable current. 1 = 0,05% (of Running current). See also: <a href="#">CUR_SCALE_MIN</a> , page 234
<b>R215</b>	<b>CUR_SCALE_FACTOR</b>	The slope of the velocity dependent current decrement rate. See also: <a href="#">CUR_SCALE_FACTOR</a> , page 235
<b>R218</b>	<b>CUR_SCALE_INC</b>	Current increment rate (independent of velocity). See also: <a href="#">CUR_SCALE_INC</a> , page 236
<b>R219</b>	<b>CUR_SCALE_DEC</b>	Current decrement rate (at 0 RPM) See also: <a href="#">CUR_SCALE_DEC</a> , page 236

*Continued next page.*

The relationship between the current control rate and velocity is illustrated on the figure. The decrement rate is inverse proportional to the velocity because the motor is much more sensitive to variations in the running current at high speeds. It gives a more stable motor behaviour if the current is decreased slower at high speeds. The increase rate must always be fast enough to detect the worst case where the motor is blocked at 3000 RPM.



The start value and the slope can be modified with the registers `CUR_SCALE_INC`, `CUR_SCALE_DEC` and `CUR_SCALE_FACTOR`. The default values are already optimised to give the best overall performance on all velocities, some other values could be better for at specific velocity.

How to calculate how the registers affect the timing at a specific velocity (RPM):

$$DEC\_CNT(RPM) = \left( \frac{RPM \times 100}{4096} \times CUR\_SCALE\_FACTOR \right) + CUR\_SCALE\_DEC$$

$$T_{DEC} = \frac{1}{36MHz} \times DEC\_CNT$$

The increment timing is independent of velocity:

$$T_{INC} = \frac{1}{36MHz} \times CUR\_SCALE\_INC$$

Timing for the default factory settings:

$$DEC\_CNT(0) = \left( \frac{0 \text{ RPM} \times 100}{4096} \times 500 \right) + 4000 = 4000$$

$$T_{DEC\_0} = \frac{1}{36MHz} \times 4000 = 111\mu s$$

$$DEC\_CNT(3000) = \left( \frac{3000 \text{ RPM} \times 100}{4096} \times 500 \right) + 4000 = 40621$$

$$T_{DEC\_3000} = \frac{1}{36MHz} \times 40621 = 1,13ms$$

$$T_{INC} = \frac{1}{36MHz} \times 2000 = 55,56\mu s$$

The current is regulated in 2048 steps from 0-100 %.

This means that it will take  $T_{INC} \times 2048 = 113 \text{ ms}$  for the current to increase from 0 to 100 % - at all velocities.

It will take  $T_{DEC\_0} \times 2048 = 227 \text{ ms}$  for the current to decrease from 100 % to 0 % at 0 RPM and  $T_{DEC\_3000} \times 2048 = 2,3 \text{ s}$  at 3000 RPM.

## 5.4

## Absolute position back-up

Only MISxxx---H2---

### 5.4.1 Position back-up

This feature is specifically ment to be used when the motor is equipped with a H2 encoder and is ment to be a “low cost” solution compared to having the H3 or H4 encoder which is a full featured absolute multiturn encoder that takes care of keeping the motor position intact regardless if the motor is powered or not.

### 5.4.2 Functional description

The position backup feature makes it possible to save the last position before power was removed (the power supply at the CVI terminal). If the motor has not been moved more that half a revolution in either direction during power down.

An hardware improvement have been made on newer motors having a serial number (label at motor will tell) >SN I 73000. Before and after serial number I 73000 the MIS motor therefor have a different functionality. Motors with a serial number >SN I 73000 automatically saves the position every 1 second.

### 5.4.3 Setup position backup using MacTalk

— Absolute position backup at power loss —

Acceptance Count: 100 Times — Acceptance count Register 140

Acceptance voltage: 18.0 Volt — Acceptance voltage Register 139

Save voltage: 15.0 Volt — Save threshold voltage reg. 141

Value of P\_IST after powerup

☐ P\_IST = 0

☐ Absolute Singleturn Encoder

☒ Absolute Multiturn Encoder

— SetupRegister Register 124

Only valid if H2 encoder is present

Valid for H2, H3 or H4 encoder option. If no encoder option is present the position data stored by the backup system will be used.

TT2395-02GB MIS340 (Version V4.00.00070, SN: 1) Connected

The available options when using the absolute position backup feature is available in MacTalk as shown above.

The parameters available and their function are as follows:

Acceptance Count

Acceptance Voltage

Save Voltage

Value of P\_IST after power up

P\_IST=0

Absolute Single turn Encoder

Absolute Multiturn Encoder

## 5.4

## Absolute position back-up

Only MISxxx---H2---

The table below describes how various combinations of setup and encoder options will affect the value of the actual position register/counter (P\_IST) after the motor have powered power up.

Setup in MacTalk	Encoder type			
	None	-H2	-H3 *	-H4
"P_IST = 0"	0	0	0	0
"Absolute Singleturn Encoder"	0	0 - 409599 Only single rev. position restored	Invalid not available	0 - 409599 Only single rev. position restored
"Absolute Multiturn Encoder"	Last stored value	Last stored value corrected for the actual +/- 1/2 rev. position	Full abs. mult. turn position	Last stored value corrected for the actual +/- 1/2 rev. position

\* = Activating the absolute position back-up function makes no sense when the -H3 encoder option is installed since it will fully take care of tracking the position including multiturn in case power is not applied to the motor.

**5.4.4 Setup position backup NOT using MacTalk**

An hardware improvement have been made on newer motors having a serial number (label at motor will tell) > SN I 73000. Before and after serial number I 73000 the MIS motor therefor have a different functionality.

**Only MIS34x and MIS43x: Serial numbers < I 73000 or HW<VI.6:**

See also [How to get SW/HW motor info](#), page 95 to obtain this information.

The absolute position backup system is activated when a voltage goes under a level, defined by SAVE\_VOLTAGE (register I 41).

Then all absolute multiturn information is saved to the flash memory at once.

When power is applied again all data are recalled from the flash memory and the motor can start operating based on the actual position that it had before power down.

It is required that the supply control voltage drops relatively slowly to allow time to save the values to flash memory. This can be secured by adding, if necessary, a large capacitor on the CVI supply voltage and powering on/off the external power supply on the AC side. Beside the position information also the Event log information is saved. This is very helpful for later troubleshooting.

**Only MIS34x and MIS43x: Serial numbers < I 73000 or HW>VI.5:**

See also [How to get SW/HW motor info](#), page 95 to obtain this information.

The absolute position backup system is active all the time and is saving the absolute multiturn position data once every second. This feature is based on a relatively new memory technology called FRAM (Ferroelectric RAM) and no battery backup is involved.

If suddenly the supply power is removed and the motor stay within half a revolution during power down the position is valid. If the motor was running during power down the position data is probably invalid.

When power is applied again all data are recalled from the memory and the motor can start operating based on the actual position that it had before power down.

Beside the position information also Event log information is saved. This is very helpful for later troubleshooting.

**5.4.5 Registers involved****Register I 41,**

Save Threshold Voltage, selects the voltage threshold, that will trigger the flash backup save operation (and stop all other motor operation).

When register I 42 has the value I 2, the scaling/unit of register I 41 is the same as register 97, Bus Voltage (4095 = I I I.4V).

The register I 42 has the values of I-8 or 8I-88, the scaling/unit of register I 41 is the same as registers 8I-96 (4095 = 5.0V)

**Register I 39,**

Acceptance Voltage, selects the voltage threshold that defines when the power supply is ready to use for erasing flash memory after power up. The scaling/unit is the same as register I 41.

**Register 140,**

Acceptance Count, selects the number of times the Acceptance Voltage must be measured after power up before the flash erase operation is started.  
The count is in units of 1 ms.

**Register 124,**

SetupBits, selects to use Flash-based Absolute Multi turn Encoder functionality when bit 11 is set.

All data storage done by the absolute position backup function can also be monitored in the event log - see also : [Reading the Event log](#), page 174

## 5.5

# Multifunction I/O setup

The MIS motor contain a dual RS485 port also called the Multifunction port, since it can be setup for many purposes such as data I/O, pulses out from the internal encoder, and pulses in from an external encoder. By RS485 means that its a bi-directional balanced interface port which can also be configured as RS422 input or output used to for example to transmit the internal encoder pulses to a connected PLC or similar.

The setup is not yet fully integrated in MacTalk but the following guide shows how to setup the most common combinations.



Notice that the Multifunction port is 5V levels and it is destructive if operated with voltages outside the voltage range 0 to 5V.

**Following setups are described in this chapter:**

- 1 Internal encoder (H2 or H4 option only) is send out at multifunction channel A and B. Link to description: [Internal encoder \(H2/H4\) signals](#) , page 119
- 2 Quadrature signals from the internal pulse generator is sent to the multifunction channels. Link to description: [Quadrature signals from the internal pulse generator](#), page 119
- 3 Same as #2 but in pulse/direction format instead of quadrature. Link to description: [Pulse/direction signals from the internal pulse generator](#), page 120

### 5.5.1

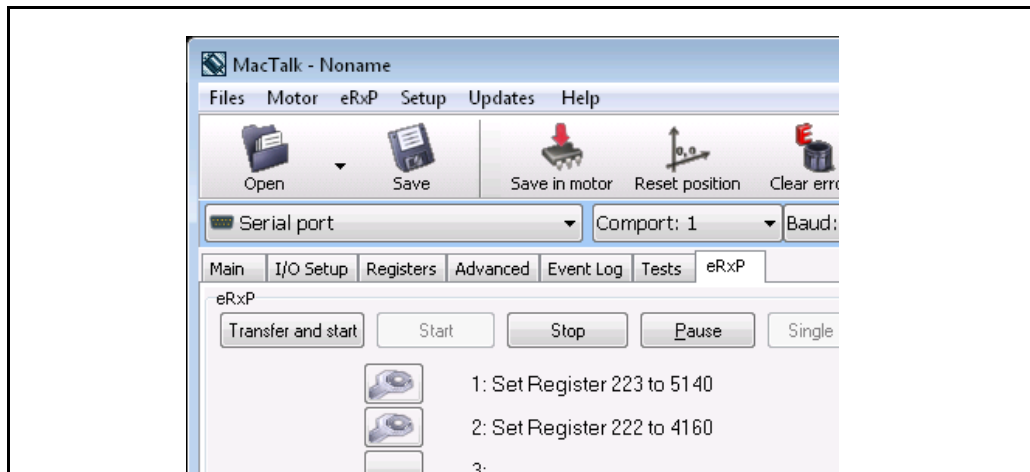
#### Configuration of the Multifunction port in general

The RS485 interface (dual) can be setup to output internal encoder signals, pulse/direction or quadrature signals from the internal pulse generator etc.

The registers mentioned below can be written by any supported protocol/interface, but are not visible as fields in MacTalk. By using the RxP program, the 2 registers can be written with 2 lines in the beginning of the program:

**R223 XFIELD\_DATA**  
**R222 XFIELD\_ADDR**

Data for the internal switch board/cross field setup.  
Address for the internal switch board/cross field setup.



Following text describe specifically which values that must be written into register 222 and 223 to obtain various outputs from the motor.

## 5.5 Multifunction I/O setup

### 5.5.2 Internal encoder (H2/H4) signals

The internal encoder quadrature signal generates 1024 pulses (4096 counts) per revolution. To output the signals A and B on the RS422/485 A and B port, write to the following registers via a little program in the RxP program:

Set register 223 to 5140

Set register 222 to 4160

The Multifunction interface is thereby set up as a permanent RS422 output interface.

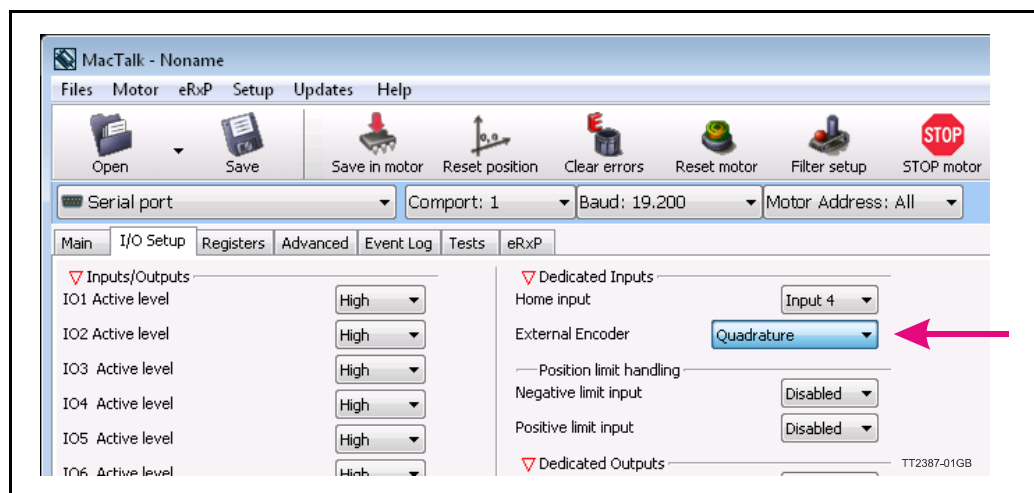
### 5.5.3 Quadrature signals from the internal pulse generator

The internal pulse generator has a resolution of 409600 cpr (20 MHz at 3000 RPM), but for the output channel it is scaled by 16 in order to limit the max output frequency to 1.28 MHz at 3000 RPM.

Set register 223 to 4626

Set register 222 to 4160

Then set the internal pulse generator to quadrature format:



Which can also be done without MacTalk:

#### RI24 SETUP\_BITS

A general setup register for many settings in the motor. Only bit 2 and 3 are used for this setting, all other bits must not be changed. This table shows how the combo box in MacTalk is setting the bits:

Bit 2	Bit 3	Setting
0	0	None
0	1	Quadrature
1	0	Pulse/direction

## 5.5 Multifunction I/O setup

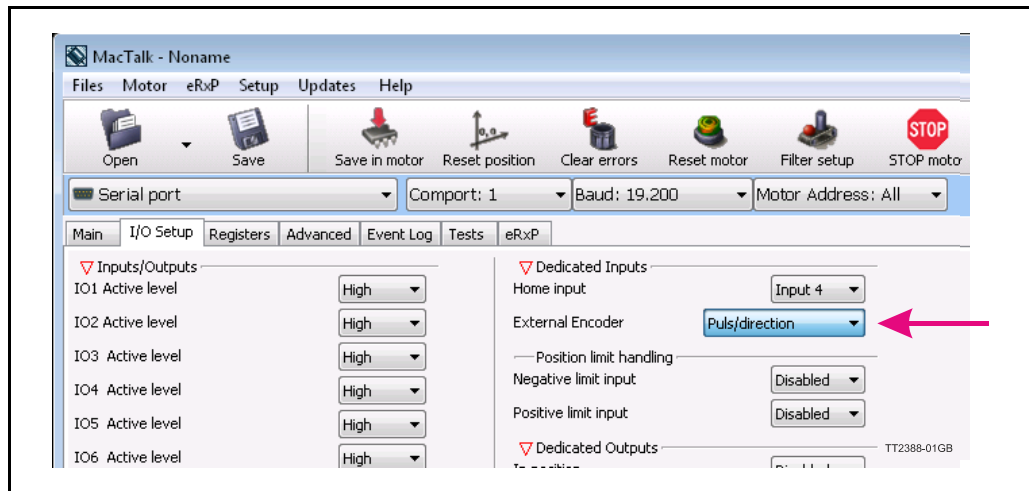
### 5.5.4 Pulse/direction signals from the internal pulse generator

The internal pulse generator has a resolution of 409600 cpr (20 MHz at 3000 RPM), but for the output channel it is scaled by 16 in order to limit the max output frequency to 1.28 MHz at 3000 RPM.

Set register 223 to 4626

Set register 222 to 4160

Then set the internal pulse generator to pulse/direction:



Which can also be done without MacTalk:

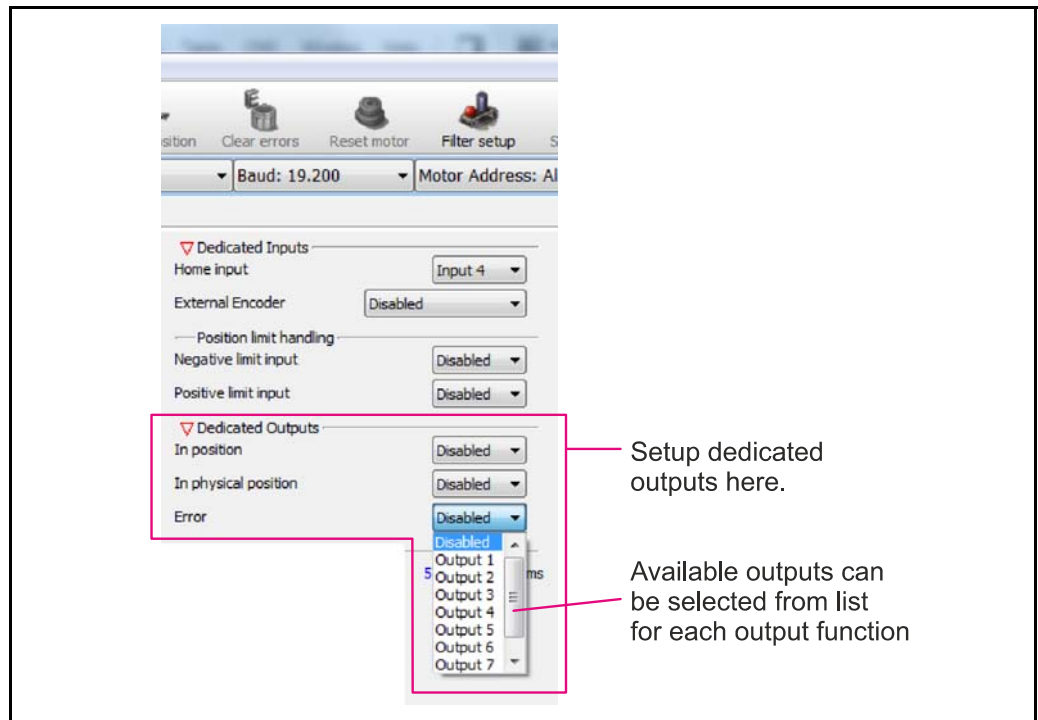
#### R124 SETUP\_BITS

A general setup register for many settings in the motor. Only bit 2 and 3 are used for this setting, all other bits must not be changed. This table shows how the combo box in MacTalk is setting the bits:

Bit 2	Bit 3	Setting
0	0	None
0	1	Quadrature
1	0	Pulse/direction

## 5.6

## Dedicated outputs



### 5.6.1

#### Error Output

The internal flag that indicate when a fatal error have occurred can be copied to a physical output. This setup is done in MacTalk - please see illustration above or by setting a bit in register `Error_Mask`. See also : [Error\\_Mask](#), page 223.

This feature can be used for signalling to a PLC or other equipment in a motion control system that an error have occurred. Under normal operation the Error Output is active. If an error occur such as temperature too high the output is set to passive.

### 5.6.2

#### In Position Output

The internal flag that indicate when the motor has reached its target position can be copied to a physical output. This selection is done in MacTalk or by setting a bit in register 137 (bit 0-7) `InPos_Mask`, See also [Inpos\\_Mask](#), page 223.

Function at output: When the motor is running, the output will be inactive. When the motor has reached target position and is at stand-still, the output will be activated.

### 5.6.3

#### In Physical Position Output” (Only valid for H2, H3, H4 options)

This signal can be used only if the MIS motor is equipped with an internal encoder (H2, H3 or H4) or an external encoder for measuring the actual position of the motor.

This signal can be selected to be continuously updated and will then indicate if the motor is inside the “In Position Window” all the time.

If continuous update of the “In Physical Position” is not selected and the autocorrection is used, this signal is changed after a move and when a check has been made of the position after the “settling time between retries” if the motor is inside the “In Position Window”.

The signal can be copied to a physical output

This selection is done in MacTalk or by setting a bit in register 137 (bit 8). See also [Inpos\\_Mask](#), page 223.

## 5.6

## Dedicated outputs

---

### 5.6.4 Brake output.

An external electro mechanical brake can be controlled by one of the user outputs. The output goes high when the brake is released, and it goes low when the brake is engaged. This is controlled automatically at the switch between an active mode (Position, Velocity, Gear, Zero search) and Passive mode (motor not powered).

This selection is done in MacTalk or by setting a bit in register 179.

See also the chapter: *Electro Mechanical brake*, page 142

## 5.7 SSI encoder/sensor interface

### 5.7.1 General information on how to connect the SSI device.

The SSI interface is based on 2 differential lines. Both lines are available in the M12 connectors and are named AI +, AI - and BI +, BI - (4 wires) -

In order to see the exact physical location of the signals please consult the pages:

- [Connector overview for the MIS motors](#), page 41

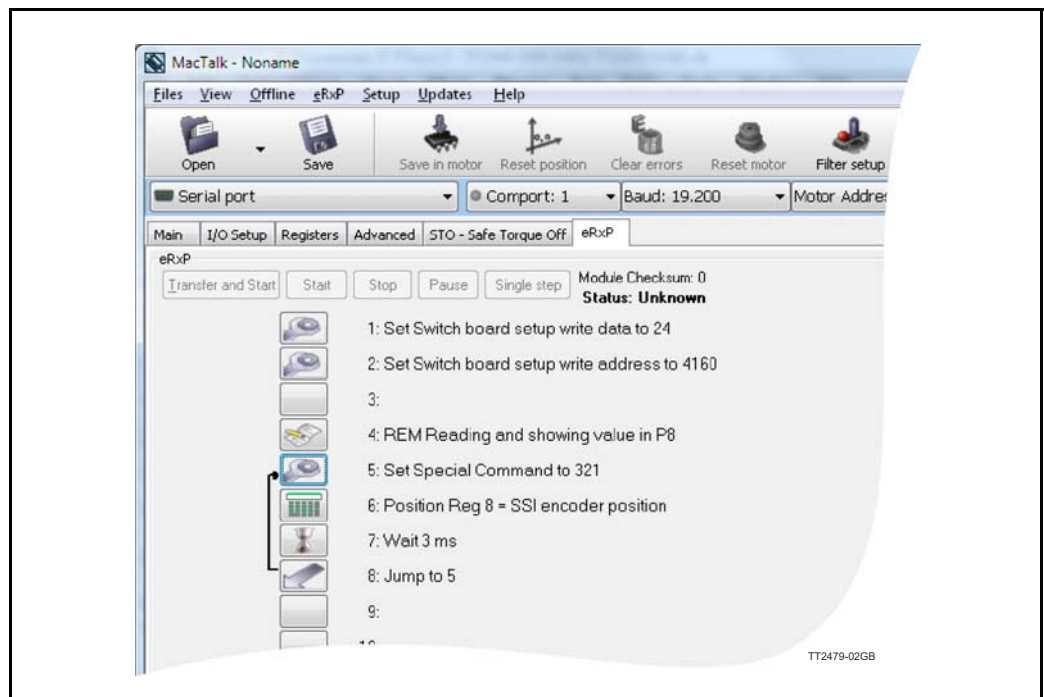
The function of the signals is as follows:

- Line **AI +** and **AI -** transmit a clock signal to the SSI device.
- Line **BI +** and **BI -** receives the data stream from the SSI device.

### 5.7.2 Setup and operation of the SSI function when using MacTalk.

When using the MacTalk Windows program supplied by JVL the following descriptions must be used.

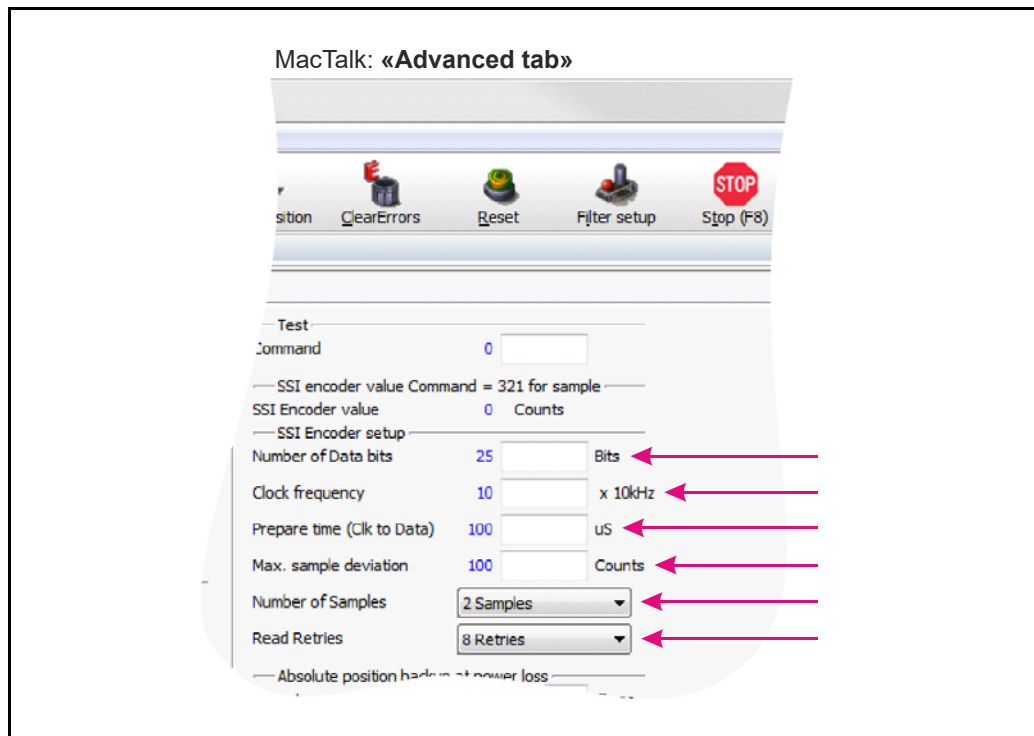
In order to setup for SSI support 2 codes need to be executed in a RxP program. These 2 commands setup the RS485 multifunction interface covering the 4 lines to the SSI device for transmitting a clock and receiving data from the SSI device.



## 5.7 SSI encoder/sensor interface

From MacTalk all configurations and settings are accessible.

Choosing the “**Advanced**”-tab gives access to the “SSI encoder value” and the “SSI encoder setup”.



### Field: “Number of Data bits”

Selects the number of data bits in each SSI transfer. The valid range is 1 to 31.

Set this value according to the actual SSI device connected. Please consult the data sheet for the actual SSI device concerning which value to be used.

### Field: “Clock frequency”

Selects the maximum clock speed in units of 10 kHz. The valid range is 1 to 255, corresponding to 10 kHz to 2.55 MHz.

Set this value according to the actual SSI device connected. Please consult the data sheet for the actual SSI device concerning which value to be used.

### Field: “Prepare time” (Clk to Data)

A typical SSI device needs a so called prepare time to sample the position data before transfer. This field is dedicated to type in the prepare time in micro seconds at the start of an SSI transfer. The valid range is 1 to 255, corresponding to 1 to 255 micro seconds. Set this value according to the actual SSI device connected. Please consult the data sheet for the actual SSI device concerning which value to be used.

### Field “Max. sample deviation”

Selects the maximum allowed deviation between two samples. The valid range is 1 to 8191. This function is ment to be an extra safety to avoid invalid reading of position data caused by noise influencing the signal. Please bear in mind that if the external SSI device is tracking the position of something that moves the value and thereby the deviation from one sample to the next can be significant.

## 5.7

## SSI encoder/sensor interface

### Field: “Number of Samples”

Selects the number of samples in each SSI measurement. If all samples stay below the “Max. sample deviation” value (described earlier in this text), no retry is required. If one pair of samples fails the whole measurement fails and a retry is attempted if allowed according to the “Read Retries” parameter.

### Field: “Read Retries”

Selects the number of retries before time out and reporting an error.

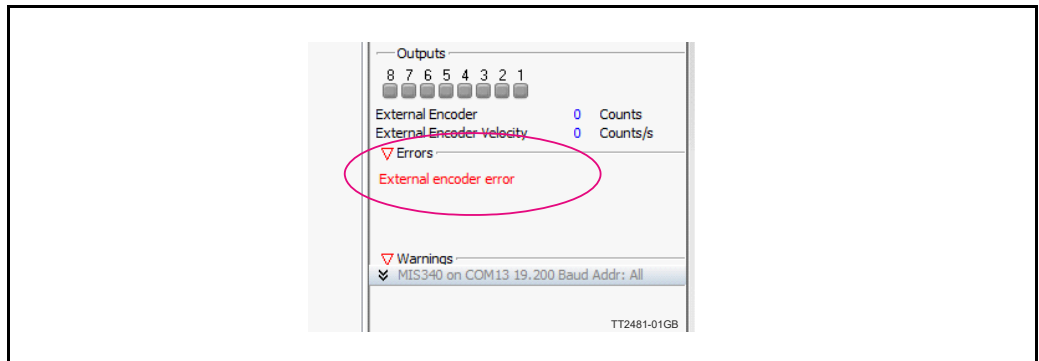
The MISxxx / SMC66 and SMC85 has a build in data validation function which automatically compares the current sample with the previous and makes sure that the deviation is within the “Max. sample deviation” limit. “Number of samples” determines how many samples one measurement contains. If the measurement fails, a retry is attempted if number of retries has not exceeded the “Read Retries” value.

$$\begin{aligned} |\text{Sample}(x) - \text{Sample}(x+1)| < (\text{Max\_sample\_deviation}) &= \text{OK} \\ |\text{Sample}(x) - \text{Sample}(x+1)| > (\text{Max\_sample\_deviation}) &= \text{ERROR} \end{aligned}$$

If the number of retries has exceeded the “Read Retries” value and the measurement still reports an error, the reading will be discarded and “SSI Encoder value” still shows the last correct measurement.

An error bit will be set in the ERR\_BITS register (34) on position 11.

MacTalk will report this error :



### Additional hardware settings:

Some LIKA SSI encoders offer 2 additional hardware settings, for instance “Zero Setting” and “Invert Counting Direction”. These settings can be controlled by having user I/O 5 and 6 set as output. Consult the data sheet for the specific encoder to read more about the behaviour of these settings and to make sure they are available.

If counting direction has been changed - the zero set (clearing the position counter) must be done to make sure that position data is valid.

We suggest performing the zero set when the encoder is stationary (no rotation).

## 5.7 SSI encoder/sensor interface

### 5.7.3 Setup and operation of the SSI function when NOT using MacTalk.

Following lines describe how to access relevant registers when having a SSI device connected.

When reading the data from the SSI device the data will be places as a signed 32 bit integer in register 47 shortly after the read command have been executed.

The time before data is present can be calculated after following formula:

Total time from the read command 32I is executed until valid data is present is equal:

$$\text{Time} = \text{Prepare time (timer)} + (1 / \text{Clock frequency} * (\text{Number of Data bits} + 1))$$

#### Example:

Encoder used have following setup:

Prepare time = 100uS (0.0001 sec.)

Clock frequency = 10kHz (10000Hz)

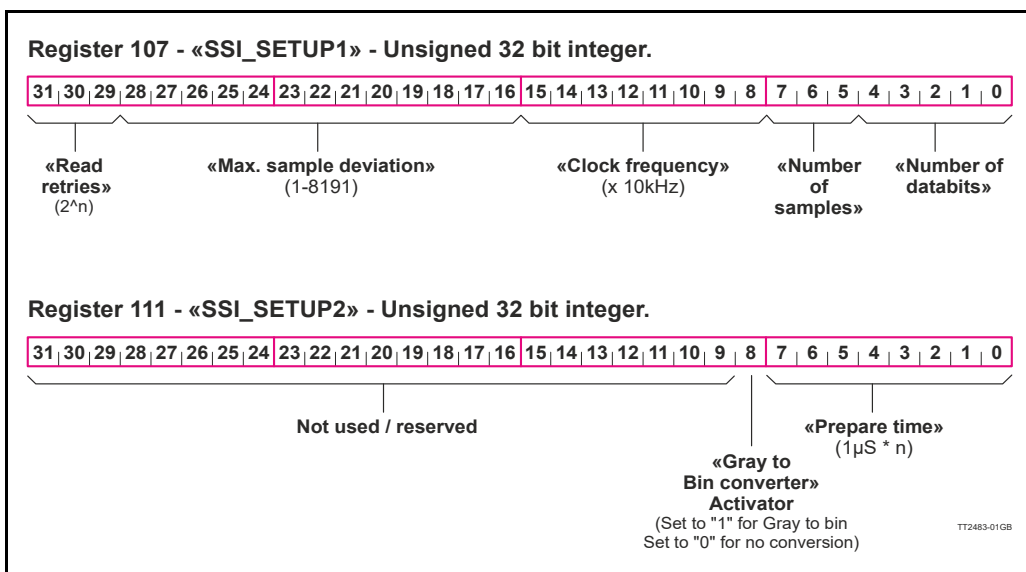
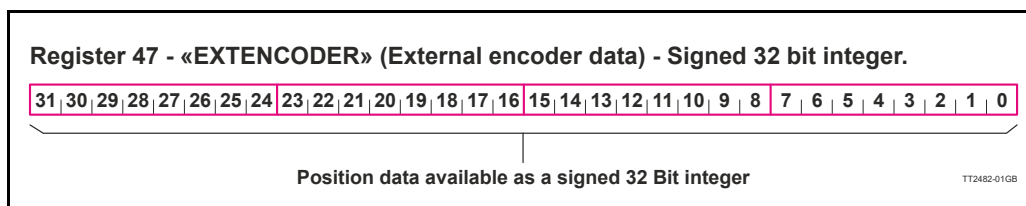
Number of data bits = 25

$$0.0001 + (1 / 10000 * (25 + 1)) = 0.0027 \text{ sec.} = 2.7\text{ms}$$

#### Note:

If noise have affected the signal the time before valid data is presented will be longer. Depending on the specific setting for the data validation function.

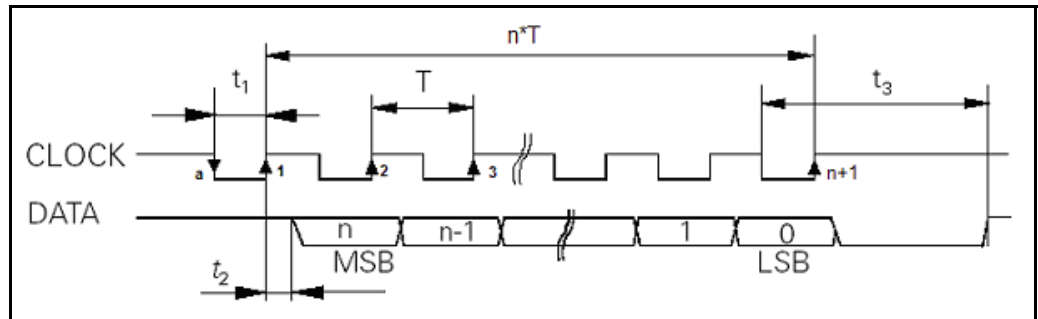
The position data from the encoder is presented in register 47 as shown below.



## 5.7 SSI encoder/sensor interface

### 5.7.4 The SSI interface principle of operation.

When the differential lines are used for SSI, the AI +/- lines work as a clock signal to the encoder, while the BI +/- signals work as a data signal from the encoder to the controller (MISxxx/SMCxx).



The figure above shows the SSI protocol principle.

The clock line is normally high. When a reading is requested, the clock goes low for  $t_1$  micro seconds (see illustration above) to allow the encoder to sample and prepare a value.

On the first rising edge of the clock (1), no sampling is done but on the second rising edge of the clock (2) the first data bit is read from the data line.

Shortly after reading the bit value, the MISxxx/SMCxx will set the clock high and execute another cycle, where the data bit is sampled just before each rising clock.

After the last data bit has been sampled, the clock stays high.

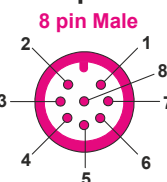
See also *Setup and operation of the SSI function when using MacTalk.*, page 123 where a sample program is shown.

### 5.7.5 Connecting an external SSI encoder to SMC66 or SMC85

Below is shown how an external LIKA encoder type SMA5 is connected to a SMC66 or SMC85 stand-alone controller equipped with the optional encoder connector.

No.	Color	LIKA	SMC	SMC 8-Pin Male
1	White	Zero Setting	IO5	1
2	Grey	Complementary	IO6	2
3	Yellow	Clock+	A1+	3
4	Black	GND	GND	4
5	Orange	Data Out -	B1-	5
6	Green	Data Out +	B1+	6
7	Blue	Clock-	A1-	7
8	Red	10...30 VDC supply	CVO	8

Encoder connector  
at the SMC66 or 85  
(Front view)



TT2621-01GB

## 5.8 Absolute Multi-turn Encoder

Only MISxxx---H3/H4--

### 5.8.1 Introduction

Optional the MIS motors can be equipped with the absolute multi-turn encoder (MISxxx-zzyyH3/H4nn option) which offers the possibility to keep track of the position regardless if power is connected or not.

When this option is present a mechanical zero search is only necessary one time after installation and the system will keep track of the actual motor position at any time afterwards.

The encoder is based on a magnetic principle which do not offer the same resolution and precision as the more expensive optical based solutions. It do however offer the advantage to keep track of the position without power applied. The magnet principle is also much more tolerant to mechanical stress such as shock and extreme temperatures.

The absolute multi-turn option offers the following main features.

#### Encoder:

Resolution	409.600 counts per revolution displayed (internal 1024 cpr)
Accuracy	+/- 0,1% of full scale
Repeatability	+/- 0,1% of full scale
Position range	$\pm 5242$ motor revolutions ( $\pm 2^{31}$ motor counts)

#### Motor:

Resolution	409.600 counts per revolution (standard)
------------	--

As seen above the motor resolution is much higher than the encoder resolution. The encoder option is however mostly used for stall detection at the motor (the motor loose its position) and for this situation the encoder will be adequate since the motor can only stall in multiples of a 1/50 shaft revolution corresponding to 7.2 mechanical degrees.

## 5.8 Absolute Multi-turn Encoder

Only MISxxx---H3/H4--

### 5.8.2 How to Preset the encoder position.

After installing the motor it is normally desired to make an alignment of the encoder position which represent the "Actual position" compared to the actual physical position of the motor and its load.

This operation is recommended to do in the following manner.

1. Set the motor in a known position
2. Insert the corresponding position value as shown below.
3. Press the "Set position" button to preset the new position value.
4. The encoder position and all other relevant position registers are now preset with the new value. No further action is needed. The motor will remember this change also during power off.

Select the «Absolute encoder» tab

MacTalk - Noname

Files Motor eRxP Setup Updates Help

Open Save Save in motor Reset position Clear errors Reset motor Filter setup STO

Serial port Comport: 1 Baud: 19,200 Motor Address: All

Main I/O Setup Registers Advanced Event Log Tests eRxP Absolute encoder

Absolute encoder position

-2147483648 0 2147483647

Encoder position: 5694000

Change encoder position

New encoder position:

0 Set position

Information

When the position reaches the limitation in the position register, the bar turns red to indicate that the position will wrap within a short distance. Special care on the position handling needs to be taken when wrapping has occurred.

The position registers are 32bit signed and the motor resolution is 409600 counts/rev. This limits the travel distance to appx. 5243 motor revolutions in each direction.

First the new value that the encoder must be preset to must be typed here

Secondly press «Set position» in order to preset the encoder counter with the new value.

TT2339-01GB

## 5.8 Absolute Multi-turn Encoder

Only MISxxx---H3/H4--

### 5.8.3 Position "Auto-correction"

This feature is only active when the motor is in position mode.

The auto-correction feature is only used when the motor has stalled and not reached its final target position within the given position window.

Each time the motor has done a movement the "Actual position" counter and the "Encoder position" counter is compared.

If the difference without sign is within the value specified in the "In position window" as shown below no further action is taken.

If the difference is larger than the value in the "In position window" the motor will try to correct the position by doing a new motor movement. The "Max number of retries" is the number of times the motor will try to correct the position, and the "Settling time between retries" is the time the motor will wait between each retry.

The screenshot shows the 'Absolute encoder' configuration window. The 'Autocorrection' section includes the following settings:

- In position window:** 20000 Counts. Callout: "An autocorrection after end movement is performed if the difference between the actual position and the encoder position is more than the value specified in this field."
- Autocorrection velocity:** 0.00 RPM. Callout: "Velocity used during a retry"
- Max number of retries:** 2 Counts. Callout: "The max number of autocorrection retries per movement is specified here. If 0 is inserted the auto correction is disabled."
- Settling time between retries:** 100 ms. Callout: "The settling (delay) time between the retries when doing autocorrection."

Other settings visible include:

- Error handling:** Follow errors (0 Counts), Position limit min (0 Counts), Position limit max (0 Counts), Error deceleration (10000 RPM/s), Min bus voltage (15 Volts).
- Zero search:** Zero search mode (Disabled), Use index after zero search (unchecked), Zero search position (0 Counts), Zero search velocity (-50 RPM), Zero search timeout (0 ms).
- Update the In Physical Position bit continuously:** (unchecked)
- Undervoltage handling:** (unchecked)

When selected the in position flag will realtime indicate if the motor is within the position window compared to a perfect move.

TT2338-02GB

See also [Position "Auto correction"](#), page 100

## 5.8 Absolute Multi-turn Encoder

Only MISxxx---H3/H4--

### 5.8.4 Operation of the encoder when NOT using MacTalk.

The following description is useful for reading the encoder or presetting a new encoder value if MacTalk is not used for the communication.

This could for example be when using one of the Ethernet options.

#### Reading the encoder position.

The encoder position is read from the register 46 (AbsEncPos). The value is scaled to match the motor resolution which means 409600 counts per motor revolution.

#### Presetting a new encoder position.

Write the new encoder value to register 144 (P\_NEW) and afterward write 316 into the register 24 (Command reg). Notice that this value must match the motor resolution which means 409600 counts per motor revolution.

Optionally the desired position value (P\_SOLL and P\_IST) can be set to the same position by writing the value 119 to the register 24 (Command reg).

### 5.8.5 Updating the Encoder firmware

The the H3/H4 encoder contain a firmware which is programmed from factory.

This firmware normally follow the encoder through the products lifetime.

But if it have been lost for some reason or need an update because a newer firmware have been released having additional features it described in details how to do in [How to update the encoder FW](#), page 94.

### 5.8.6 Position retention time

A special high temperature "Poly-carbon mono fluoride" lithium cell is used as backup for the absolute multi-turn encoder. Compared to standard cells this one has extremely good performance at high temperatures up to 125 degrees Celsius.

Normal lithium cells self-discharge very rapidly just being exposed to a temperature of 60 degree celsius.

The retention time of the position when the motor is not supplied is better than **10 years**. This depends of how long time the motor is in use and the temperature.

The self-discharge at 85 degrees celsius (185 degrees Fahrenheit) is 6% per year, and 2% per year at 40 degrees celsius (104 degrees Fahrenheit).

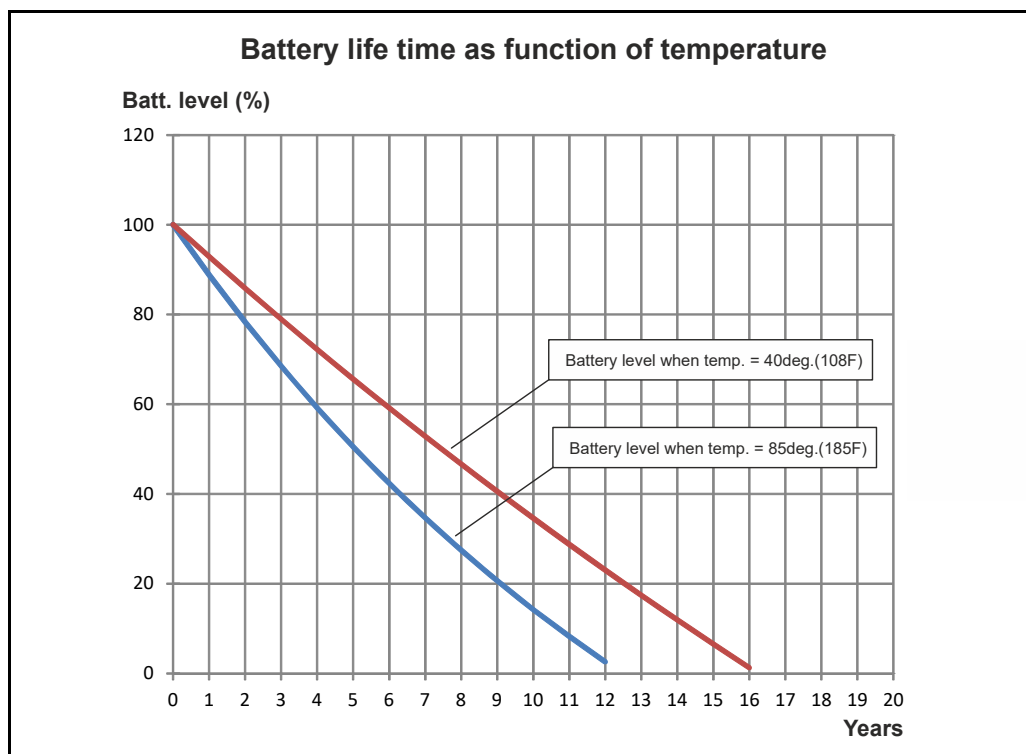
Concerning the retention time, there is no big difference if the motor is in use at a high temperature or it is powered down (low temperature).

The absolute multi-turn encoder's current consumption from the battery when the motor is not externally powered, is max. 1.5µA.

The curves below shows what is considered as a worst case scenario for the retention time. The curves are based on 40 and 85 degree celsius (108/185F) **without any external power applied to the motor** meaning that the internal battery have to supply current to the encoder circuitry all the time. The current consumption of the encoder circuitry is not dominant compared to the internal leakage current in the battery.

#### Hints to optimise the battery lifetime:

1. Avoid to place the motor in an environment with high temperatures.
2. Set the running and especially the standby motor current as low as possible in order not to heat up the motor unnecessarily.
3. Keep the external power applied as much as possible.
4. Connect an external battery to the terminal "EXTBACKUP" at CN4 (17 pin connector pin 16) - see also: *Connecting an external power source for the H3/H4 encoder, page 133*



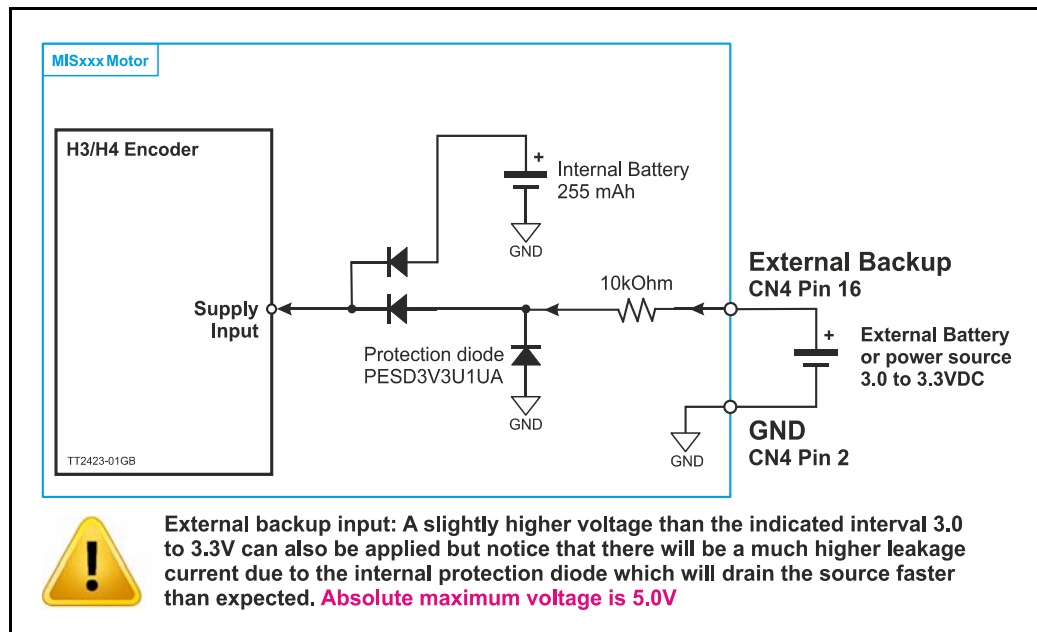
## 5.8 Absolute Multi-turn Encoder

Only MISxxx---H3/H4--

The battery level can not be measured in the present version of the MIS motors. If the internal battery is empty no pre-warnings will be given but the error “lost encoder position” will be reported in MacTalk and the error register.

### 5.8.7 Connecting an external power source for the H3/H4 encoder

An external power source such as a battery can also be connected. The illustration below shows how. See also *How to connect a MIS motor, page 41*.



The lifetime of the internal battery will not get significant longer when connecting an external battery since a significant part of the lifetime is based on the internal leakage current in the battery.

### Frequently asked questions

Q1 How can i see the status level of the internal battery ?

A1 Sorry but not possible in the present hardware version

Q2 What happens if the internal battery is empty ?

A2 The absolute position will not be retained but the motor can still function. The error “Encoder lost position” will however be reported. See also [Error message 'Encoder lost position', page 178](#) for details how to reset the error. If the internal battery is empty an external can be connected and the motor will operate normally again.

Q3 If i connect an external battery - which size must it be ?

A3 The internal battery is 3.0V/255mAh. The external battery is recommended to be minimum the same size. Do only use batteries with a nominal voltage of 3.0 to 3.3V.

Q4 Can i replace the internal battery ?

A4 No this is not possible. The motor need to be serviced by JVL.

Q5 If the internal battery is empty how long time is the position then kept ?

A5 The position data is not kept if the internal battery is empty.

## 5.9

## Position Limits

### 5.9.1 Position limitation features.

The MIS motor family offers 2 different methods of limiting the movement of the motor. In some applications it may be fatal or critical if the position of the motor passes a certain mechanical position range.

The 2 methods are as follows:

- **End of Travel Limit Inputs.**  
Limit switch inputs for detecting the physical position of the mechanics.
- **Software Position Limits.**  
Limits in software to prevent the motor to move outside a certain position range.

The next pages describe the function and how to use these 2 methods.

### 5.9.2 End-of Travel Limit Inputs

Any of the 8 general inputs (IO1 to IO8) can be used as limit inputs. The input can be set from MacTalk or via register [NL\\_Mask](#), page 221 or [PL\\_Mask](#), page 222.

#### Positive limit (PL)

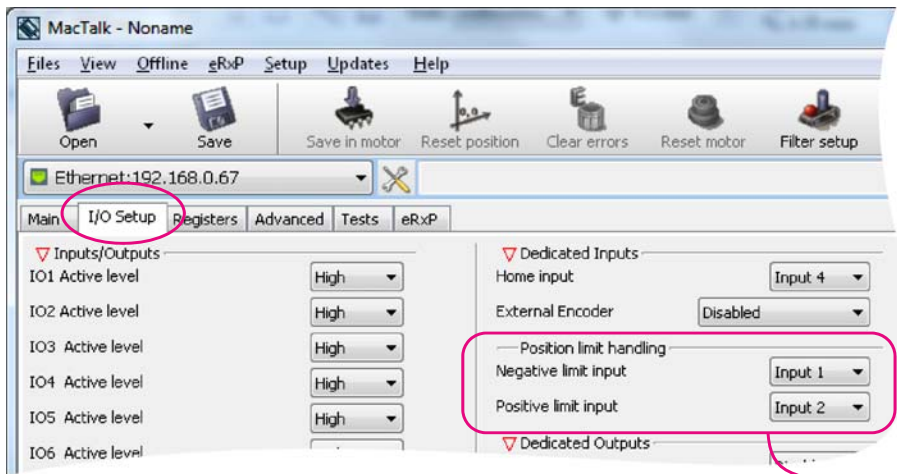
Activation of the Positive limit (PL) Input will halt motor operation if the motor is moving in a positive direction. The motor can however operate in a negative direction even if the PL Input is activated.

#### Negative limit (NL)

Activation of the Negative limit (NL) Input will halt motor operation if the motor is moving in a negative direction. The motor can however operate in a positive direction even if the NL Input is activated.

Below is shown how to select the desired input(s) to be used for the limit switch(es). Notice that the inputs default are set to disabled. Its also possible to select only one input for one of the directions and keeping the opposite input disabled. Please use the general chapter [I/O Setup tab](#), page 88 for setting up the active level, optional input filter etc.

**How to select the input(s) for the End of Travel Limit**



MacTalk - Noname

Files View Offline eRxP Setup Updates Help

Open Save Save in motor Reset position Clear errors Reset motor Filter setup

Ethernet:192.168.0.67

Main I/O Setup Registers Advanced Tests eRxP

Inputs/Outputs

IO1 Active level High

IO2 Active level High

IO3 Active level High

IO4 Active level High

IO5 Active level High

IO6 Active level High

Dedicated Inputs

Home input Input 4

External Encoder Disabled

Position limit handling

Negative limit input Input 1

Positive limit input Input 2

Dedicated Outputs

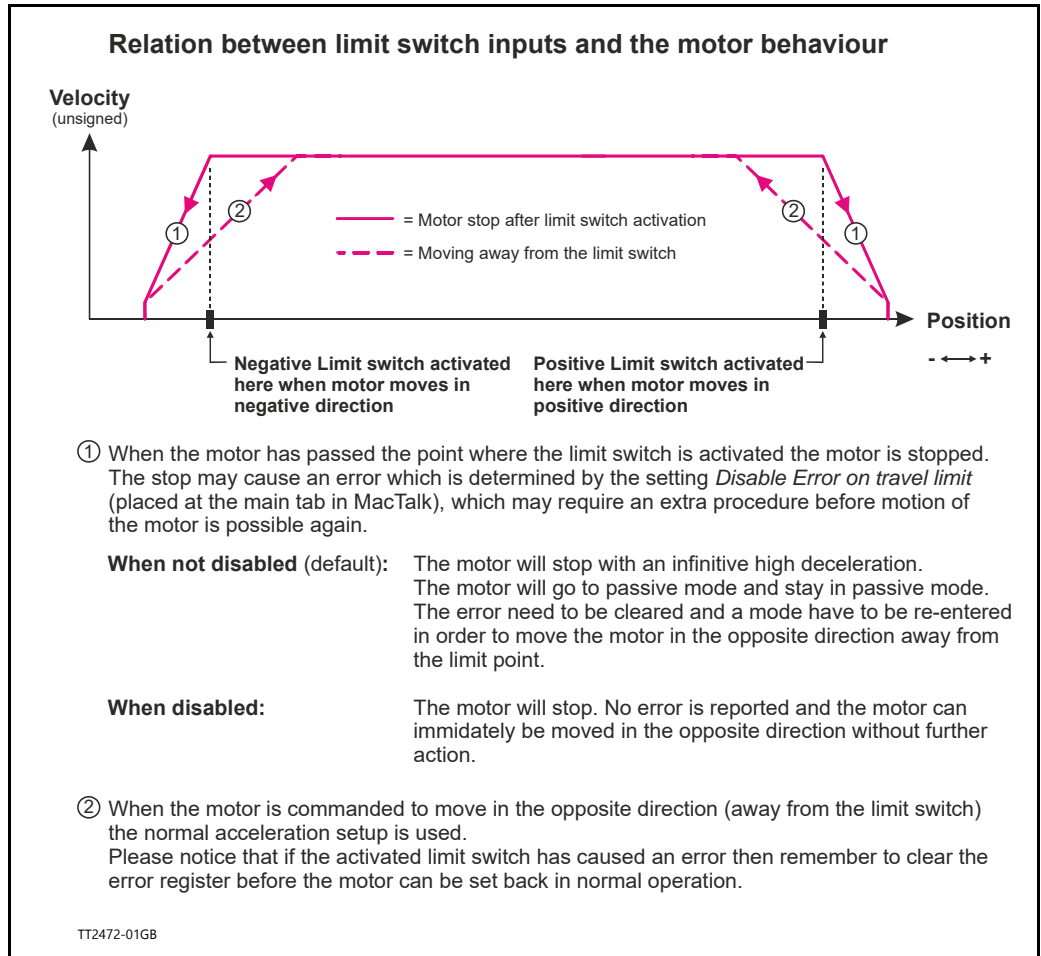
TT2471-01GB

Select the negative and/or positive limit input(s) in these fields.

## 5.9

## Position Limits

The following illustration shows the timing and motor behaviour when the limit switch inputs are activated.



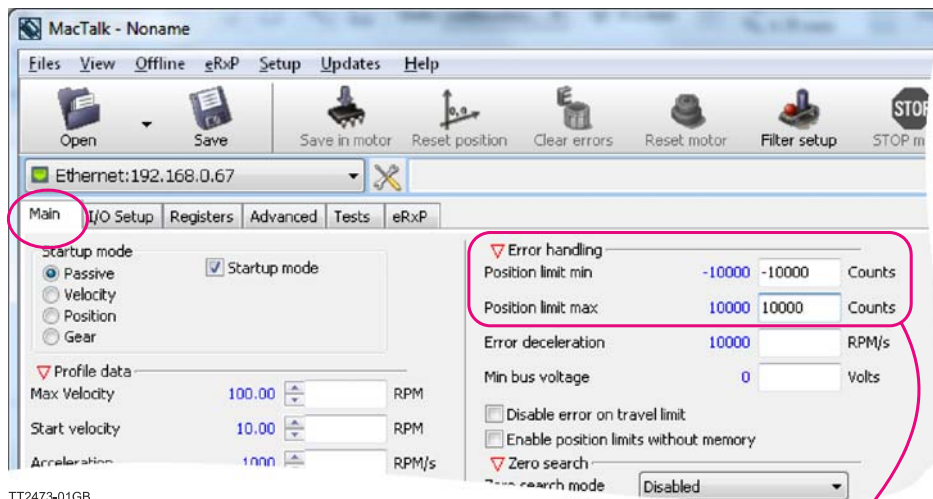
## 5.9

## Position Limits

### 5.9.3 Software Position Limits.

These limits are software limits and are can optionally be used to prevent the motor to move outside a certain position range in cases where for example faulty position commands are send to the motor or similar cases.

#### How to setup the Position limits



Select the negative and/or positive limit(s) in these fields.



**CAUTION** - Please notice that the motor may start to move if the position limits are changed after a situation where they have been passed and have forced the motor to stop. Its recommended to set the motor in passive mode before changing the limits in order to avoid personal injuries.

The limits has the same function as the physical End of travel inputs but is a pure software limitation. Default for both parameters is 0 meaning that the feature is disabled. Notice that if one of the parameters are different from 0, both values are activated.

#### Positive Position Limit (PLS)

When the motor is moving in a positive direction in position mode or gear mode, the motor will stop at Position Limit Max. In velocity mode the speed will internally be set to 0 when passing Position Limit Max, causing the motor to decelerate and stop.

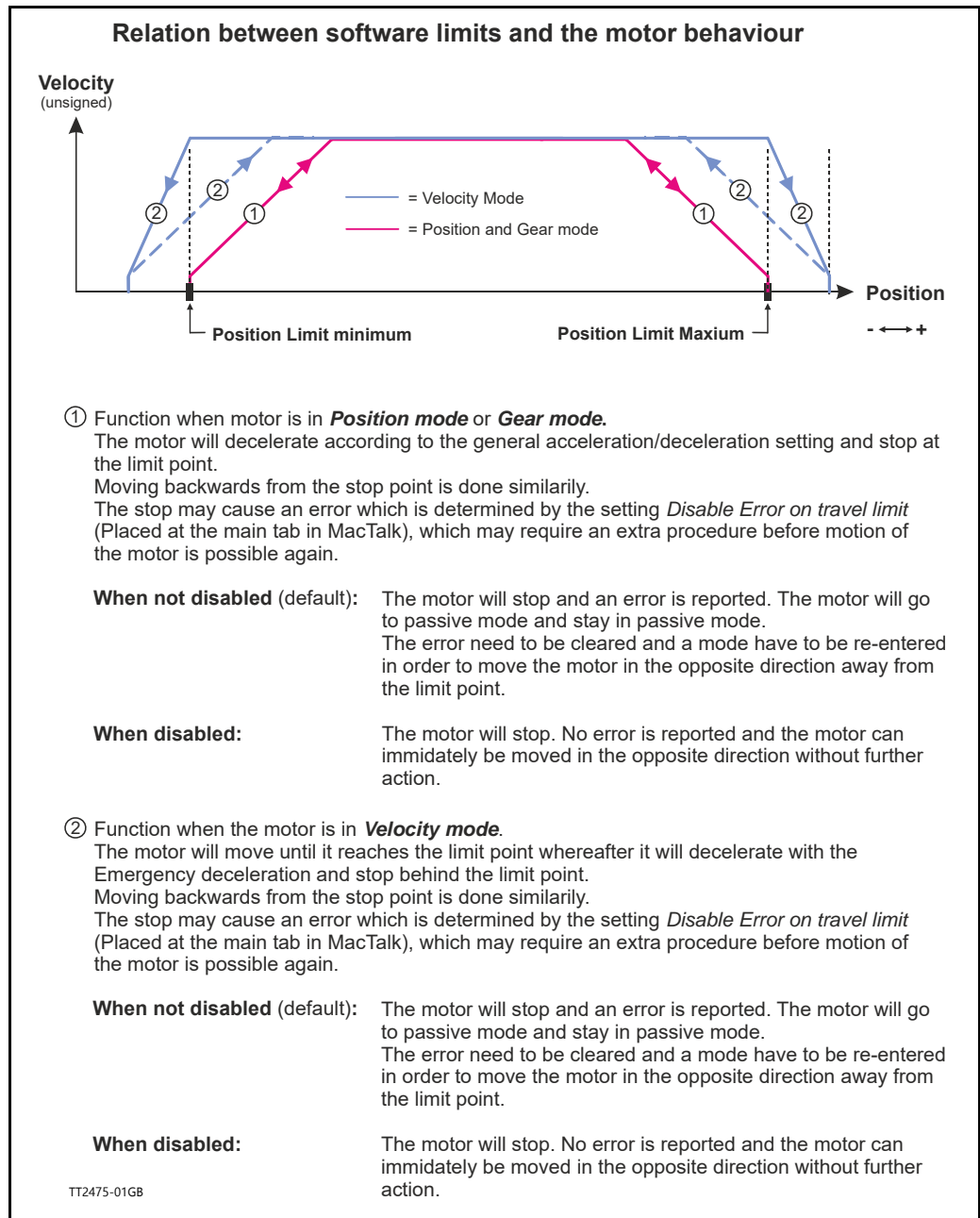
#### Negative Position Limit (NLS)

When the motor is moving in a negative direction in position mode or gear mode, the motor will stop at Position Limit Min. In velocity mode the speed will internally be set to 0 when passing Position Limit Min, causing the motor to decelerate and stop.

## 5.9

## Position Limits

The illustration below shows how the software limits take effect at the motor movement.



For further information about the internal registers that are behind the fields in MacTalk see also: [MIN\\_P\\_IST](#), page 206 and [MAX\\_P\\_IST](#), page 207.

## 5.9

## Position Limits

---

### 5.9.4 Limit Error handling

A bit will be set in the Controller's warning register if either the NL, PL, NLS or PLS has been activated or are active. See [Warn\\_Bits](#), page 209.

Bits 0 and 2 are common for PL and PLS. Bits 1 and 3 are common for NL and NLS.

The motor will stop and activate an error, when reaching a limit. When a limit error is active the motor is forced into passive mode, and further movements are impossible. To reset the error press the "Clear Errors" button in MacTalk. Clearing errors automatically also clears warnings too.

### 5.9.5 Limit handling - optional

The MIS motors can be configured to stop and stay in the current mode when reaching a limit. This can be done by setting the *DisableErrorOnTravelLimit-bit* in the *SETUP\_BITS* register. See [Setup\\_Bits](#), page 220.

When the *NoErrorOnPositionLimit-bit* is set the motor decelerates with the "Error deceleration" on travel limits in all modes and on position limits in velocity mode. The normal acceleration is used on position limits in position and gear mode.

### 5.9.6 Simple mode: Position limits without memory

When using position limits without memory, the sensor has-been warning is automatically cleared when the direction of the motor is reversed and the limit switch is no longer active.

"Position limits without memory" is enabled in the [Setup\\_Bits](#), page 220, **bit 28**.

The MIS motors offer the possibility to define the behaviour during and after the P+ bus voltage (main supply) disappear. This situation could for example be during an emergency stop, which causes the P+ supply to be cut while the control voltage (CVI) is still applied to the motor.

3 options available:

**Under voltage -> Set error bit**

Default = ON.

If this option is selected an under voltage will be handled like an error situation and the corresponding error bit will be set. The motor is stopped using the "Error deceleration" before the motor is switched to Passive mode, like any other error situation. When P+ is re-applied to the motor it will stay in Passive mode and report a "Low bus voltage" error. To get the motor back in normal operation the error must be cleared and an operation mode must be selected. If this function is activated it will have first priority and the "Error deceleration" will be used compared to the 2 other options "Under voltage -> Stop controlled and go to passive" and "Under voltage -> Set velocity to 0" which both use normal deceleration.

**Under voltage -> Stop controlled and go to passive**

Default = Off.

This option makes the motor decelerating according to the normal acceleration parameter and go to Passive mode when P+ is removed. When P+ is re-applied the motor stays in Passive mode. To get the motor back in normal operation an operation mode must be selected.

**Under voltage -> Set velocity to 0**

Default = Off.

This option simply just sets the velocity to 0 causing the motor to decelerate and stay stationary when P+ is removed. The velocity setting will stay at 0 also after P+ is re-applied. A velocity value (>0 RPM) must be written into the velocity register to get the motor moving again.

*Continued next page*

## 5.10 Under voltage Handling

### 5.10.1 Setup with MacTalk

The 3 options that define the behaviour of the motor when P+ bus voltage is lower than what is set in the “Min bus voltage” field can all be accessed from MacTalk as shown below.

The screenshot shows the MacTalk software interface with the following settings:

- Startup mode:** ☒ Passive, ☒ Startup mode, ☐ Velocity, ☐ Position, ☐ Gear
- Profile data:**
  - Max Velocity: 100.00 RPM
  - Start velocity: 10.00 RPM
  - Acceleration: 1000 RPM/s
  - Deceleration: 0 RPM/s
- Driver parameters:**
  - Running current: 3.00 A RMS
  - Standby current: 0.75 A RMS
  - Standby time: 500 ms
- Error handling:**
  - Follow errors: 0 Counts
  - Position limit min: 0 Counts
  - Position limit max: 0 Counts
  - Error deceleration: 10000 RPM/s
  - Min bus voltage: 15 Volts
  - ☐ Disable error on travel limit
  - ☐ Enable position limits without memory
  - ☒ Zero search
  - Zero search mode: Disabled
  - ☐ Use index after zero search
  - Zero search position: 0 Counts
  - Zero search velocity: -50 RPM
  - Zero search timeout: 0 mS
- Undervoltage handling:**
  - ☒ Undervoltage -> Set error bit
  - ☐ Undervoltage -> Stop controlled and go to passive
  - ☐ Undervoltage -> Set velocity to 0
- Communication:**
  - Motor address: 254

The undervoltage level is defined in this field. Default = 15 Volt.

The 3 fields that define which action that should be taken in case of undervoltage

## 5.10 Under voltage Handling

---

### 5.10.2 Setup without MacTalk

If MacTalk is not used for setting up parameters and registers related to the under voltage feature it must be done as follows.

The motor contains a number of registers which can be accessed from various protocols depending at which options the motor has.

Protocols available are for example Ethernet (EthernetIP, PROFINET etc.) and CAN-open, Modbus or the MacTalk protocol.

Each field in MacTalk described earlier in this chapter is accessing a register in the motor.

The registers that are relevant for the under voltage feature are:

<b>R98</b>	MIN_BUSVOL	The voltage level that defines when an under voltage situation is present at the P+ (main power). See also: <a href="#">Min_Busvol</a> , page 213
------------	------------	--

<b>R124</b>	SetupBits	SetupBitsThe 3 bits that define what action should be taken in case of an under voltage situation. <b>Bit 21:</b> Under voltage -> Set error bit <b>Bit 22:</b> Under voltage -> Stop controlled and go to passive <b>Bit 23:</b> Under voltage -> Set velocity to 0 See also: <a href="#">Setup_Bits</a> , page 220
-------------	-----------	--

If none of the above bits are set the motor will continue to run when the main power (P+) is back.

## 5.11 Electro Mechanical brake

### 5.11.1 Brake Introduction

The motor can be equipped with a electro mechanical brake to hold the position in Passive mode and power off situations where the motor has no torque. This is often desired to keep mechanics in position for example if it's a vertical movement.

The brake control always takes care that the brake is activated (hold the motor) if a situation occurs where the motor is not powered and therefore can slip away from the desired position.

This will typically be in situations like when the motor is in Passive mode or an error has occurred which will cause the motor to be power less and not able to keep its position in a controlled manner.

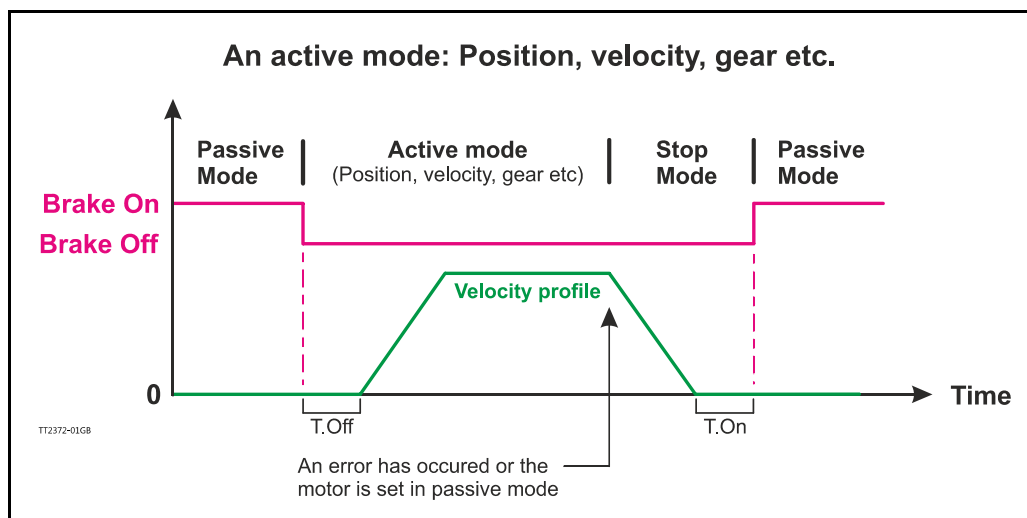
No involvement from users side is needed to activate and de-activate the brake.

It is also possible to connect an external brake via one of the user outputs I/O1 to 8.

### 5.11.2 Brake timing

The brake is always active in Passive mode because the motor has zero torque. When changing to an active mode, the standby current is applied and the brake is disabled. The brake needs some milliseconds to release and therefore there will be a short delay (typically ~40 ms) before the motor can move. The brake is always off in active modes.

When changing to Passive mode the motor goes into Stop mode to decelerate according to the "Deceleration"-ramp. When "Actual velocity" is 0, the brake is activated and also here a short delay makes sure that the brake is active before the motor goes passive.



### 5.11.3 Connecting an external brake

An external brake can also be connected to one of the eight user outputs I/O1 to 8.

The external brake will be operated in parallel with the internal brake without any time delay.

MacTalk do not support setup of the external brake but an internal register is available for setting up the brake. Please refer to [Access without MacTalk](#), page 144

## 5.11 Electro Mechanical brake

### 5.11.4 Brake registers

2 registers in the motor are relevant for the operation of the brake.

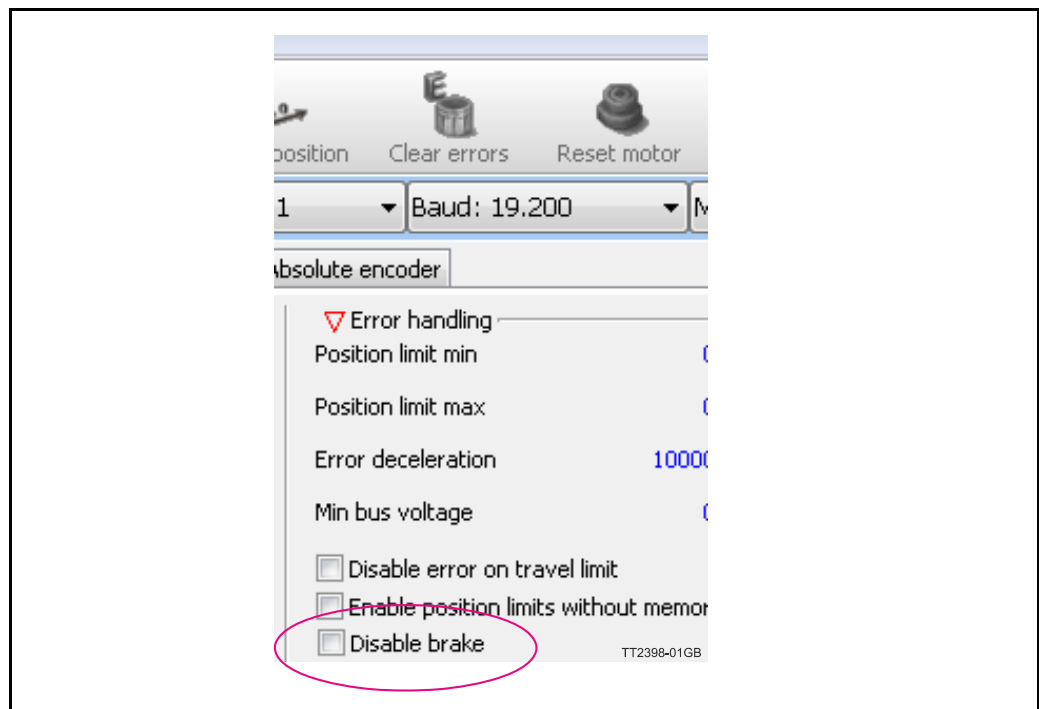
One register makes it possible to disable the brake so that the motor can run freely regardless which mode or condition the motor is in.

The other makes it possible to verify if the brake is active or passive.

### 5.11.5 Access from MacTalk

A field named "Disable brake" at the main tab in MacTalk makes it possible to disable the brake.

In the status bar at the right side its also possible to see whether the brake is active or not.



## 5.11 Electro Mechanical brake

---

### 5.11.6 Access without MacTalk

If MacTalk is not used for setting up parameters/registers related to the brake feature it must be done as follows.

The motor contains a number of registers which can be accessed from various protocols depending which options the motor has.

Protocols available are for example Ethernet (EthernetIP, PROFINET etc.) and CAN-open, Modbus or the MacTalk protocol.

Each field in MacTalk described earlier in this chapter is accessing a register in the motor.

The registers that are relevant for operating the brake are:

#### **R25 STATUS\_BITS** **MacTalk name:** N/A

Bit 14 in this register monitors the actual brake status. If Bit 14 is 0 the brake is not active (motor can run freely) and if bit 14 is 1 the brake is active and keeps the motor in position.

Notice that other bits in this register are used for other purposes. See also: [Status bits](#), page 205

#### **R124 SETUP\_BITS** **MacTalk name:** Disable brake

Bit 19 in this register set to 1 will disable the brake which will allow the machine operator to move the shaft.

Notice that other bits in this register are used for other purposes. See also: [Setup\\_Bits](#), page 220

Connect an external brake

#### **R179 BRAKE** **MacTalk name:** N/A

The user outputs I/O1-8 can be used to control an external brake. The BRAKE register sets this up:

##### Bit 0-7: **Brake output mask**

Defines which of the eight outputs that controls the brake.

##### Bit 8-15: **Brake\_T\_ON** (ms)

Brake turn on time

##### Bit 16-23: **Brake\_T\_OFF** (ms)

Brake turn off time

See also: [Status bits](#), page 205

## 5.12

## Turn Table Mode

---

### 5.12.1 General description



**Please note:** The Turn Table Mode is only available under following conditions:

1. MIS34x/43x: Hardware need to be version 1.7 or newer. MIS17x/23x: all hardware versions support turn table mode.
2. Firmware need to be version 4.06 or newer.
3. Serial number need to be 173.000 or newer.

All 3 conditions have to be met. See also [How to get SW/HW motor info, page 95](#)

The Turn Table Mode, also sometimes called Dividing head, Indexing table or Rotary Table, makes the motor work in three different ways that limits the actual position and target position to a smaller position range than normal. This is useful for supporting some mechanical systems.

One of the strong features of this is the motor's ability to calculate the shortest movement to a new target position automatically.

The option can also be used to have the motor always move in the same direction for any target position value even if the new target position value is below the old position in this case the motor will turn in the configured CW or CCW direction until it returns to the lower target position.

The actual position will not overflow even when always running in the same position for any length of time. Notice that when using the -H3 or -H4 encoder option the total position range must be within  $(-2^{31}) - (2^{31}) - 1$  counts from the initial zero position. When using the -H2 encoder option there is no restrictions.

The general idea is to define a range of positions by a Minimum and a Maximum value that can be freely selected within the normal position range of the motor ( $\pm 2^{31}$  counts). Whenever the actual position exceeds the maximum it will wrap around and continue from the minimum position while still running in the same direction.

Also when moving below the minimum position it will wrap around and continue to count down from the maximum position.

During operation the motors main operation mode is usually set to Position mode (MODE\_REG = 2), and the motor is commanded to a new target position by writing a new value to the Target Position, register 3, P\_SOLL. See also - [Modes other than Position mode](#), page 150.

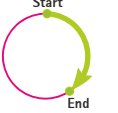
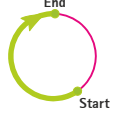
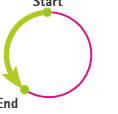
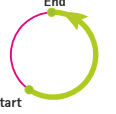
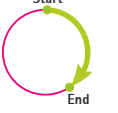
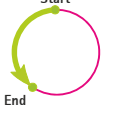

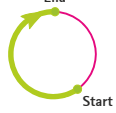
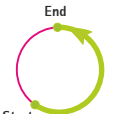

The value written to P\_SOLL should be within the Turn Table position range. If the value is outside the working range specified by "Turn table pos. min." or "Turn table pos. max" it will be clipped to be within the values specified by "Turn table pos. min." or "Turn table pos. max". To make a full turn in CW or CCW mode, a multiturn operation is supported see following illustration. The registers normally used for software position limits are used to define the Turn Table working range in Turn Table Mode operation.

(Continued next page)

## 5.12

## Turn Table Mode

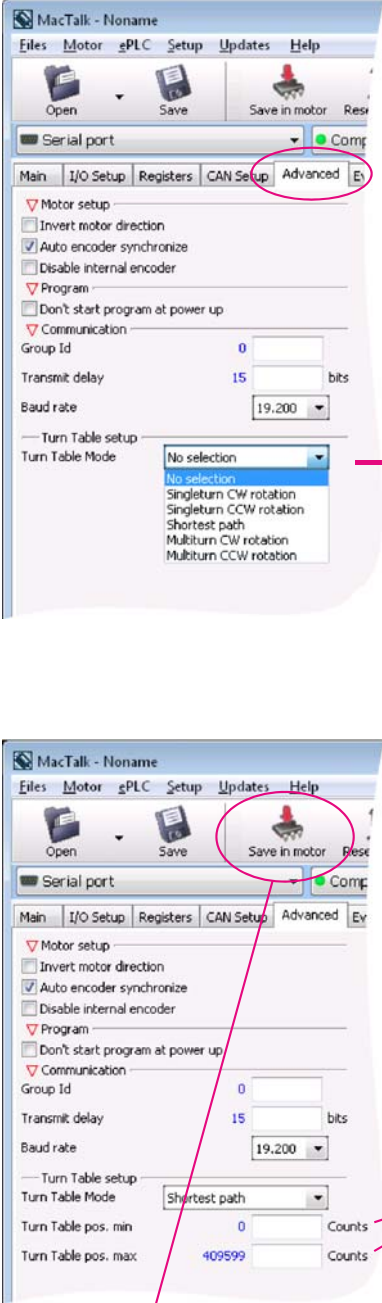
The 5 operation modes for the Turn Table are shown in the following table:

<b>Basic setup:</b> Working range min./max= 0 to 409599 counts equal to one revolution.	<b>Situation 1</b>	<b>Situation 2</b>
<b>1</b> Singleturn CW rotation	 <p>Setup : Start position 0 Target position = 150000</p>	 <p>Setup : Start position 150000 Target position = 0</p>
<b>2</b> Singleturn CCW rotation	 <p>Setup : Start position 0 Target position = 250000</p>	 <p>Setup : Start position 250000 Target position = 0</p>
<b>3</b> Shortest path	 <p>Setup : Start position 0 Target position = 150000</p>	 <p>Setup : Start position 0 Target position = 250000</p>
<b>4</b> Multiturn CW rotation	 <p>Setup: Start position 0 Target position = 970000 Result: P_SOLL = 150800 since the remainder after 97000/409600 is 150800.</p>	 <p>Setup : Start position 150000 Target position = -1000 Result: Actual position (P_SOLL) = 0 since the value is limited to the working range minimum</p>
<b>5</b> Multiturn CCW rotation	 <p>Setup : Start position 250000 Target position = 12345678 Result: Actual position (P_SOLL) = 409599 since the value is limited against the working range maximum</p>	 <p>Setup: Start position 0 Target position = -970000 Result: P_SOLL = 150800 since the remainder after 970000/409600 is 150800.</p>
<p>Note 1: In general, Actual position (P_SOLL) will be modified to be within the valid rotary table position working range if a value outside this range is written to the Actual position register (P_SOLL).</p> <p>Note 2: When using the multiturn options (4 and 5) the position is limited as follows:</p> <ul style="list-style-type: none"> <li>- Multiturn CW rotation (4). Values below working range minimum are limited to the minimum.</li> <li>- Multiturn CCW rotation (5). Values above working range maximum are limited to the maximum.</li> </ul> <p style="text-align: right;">TT2555-01GB</p>		

## 5.12 Turn Table Mode

### 5.12.2 Configuration via MacTalk.

When using MacTalk for configuration the following parameters are used:



The desired *Turn Table Mode* is selected here at the *Advanced* tab

The working range can be defined here after selecting the type of turn table mode. Minimum is the lowest position possible to reach and maximum is the highest position that can be reached. Both in encoder counts. Both values are referring to the zero position found during the optional zero search or alternatively the zero point defined if using an absolute multiturn encoder.

After the setup is done please remember to save it in the permanent memory by pressing the *Save in motor* button.

TT2563-01GB

## 5.12

## Turn Table Mode

---

### 5.12.3 Configuration via motor registers.

When the configuration is done by channels other than MacTalk such as Ethernet, Profibus, CANopen etc. or just a simple serial interface connection to the basic motor the following description must be followed.

The main Turn Table Mode operation is selected by Register 126.

See also [Turntable\\_Mode](#), page 221.

#### Register 28, MIN\_P\_IST

Sets the minimum position for the Turn Table working range.

#### Register 30, MAX\_P\_IST

Sets the maximum position for the Turn Table working range.

Note that the firmware will swap the min. and max. values if the minimum is larger than the maximum at startup.

#### Register 126, TURNTABLE\_MODE

The value of this register defines the Turn Table Mode:

Mode	Operation selected
0	No Turn Table Mode operation (normal motor operation with full position range)
1	Always CW Turn Table Mode operation.
2	Always CCW Turn Table Mode operation.
3	Shortest path Turn Table Mode operation.
4	Always CW Turn Table Mode operation with multiturn operation.
5	Always CCW Turn Table Mode operation with multiturn operation.
>5	Illegal setting.

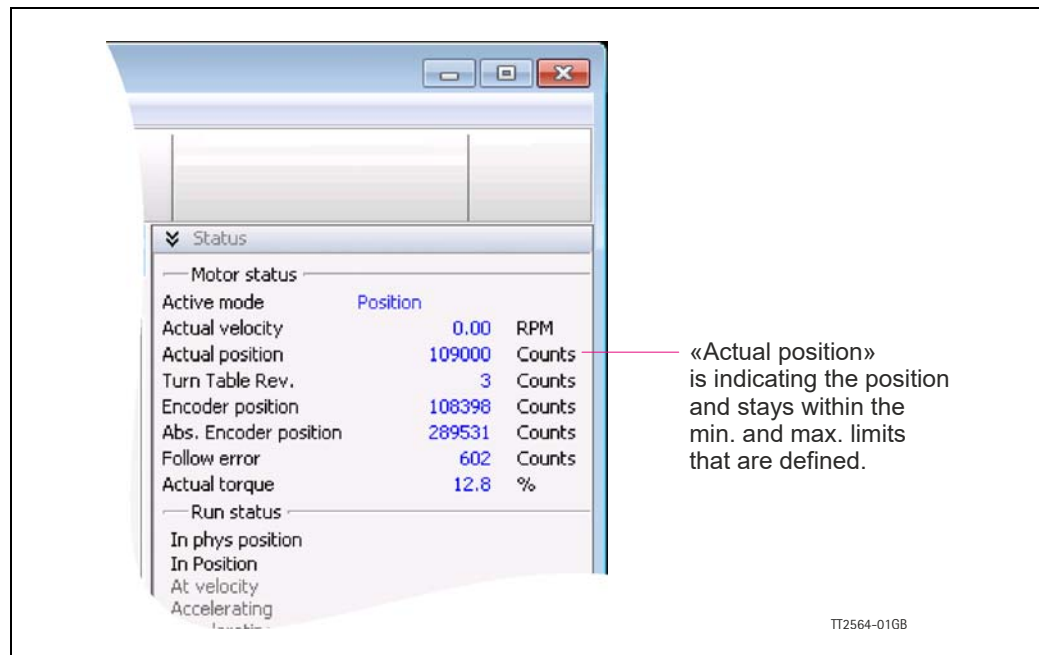
**Important:** Please remember to save the setup in permanent memory (save in flash) before the setup is applied and used by the motor.

## 5.12

## Turn Table Mode

### 5.12.4 Status read out via MacTalk.

When status read out is done via the MacTalk program, data is presented as follows:



### 5.12.5 Status read out via motor registers.

When status read out is done by channels other than MacTalk such as Ethernet, Profibus, CANopen etc. or just a simple serial interface connection to the basic motor the following description must be followed.

#### Register 3, P\_SOLL

P\_SOLL will be modified to be within the valid Turn Table position working range if a value outside this range is written to P\_SOLL.

#### Register 10, P\_IST

"Actual position" is the actual motor position and stays within the min. and max. limits defined.

#### Register 27, TURNTAB\_COUNT

Is a counter that keeps track of wrap-arounds since the last reset of the motor. This register can be written to another value by the user at any time.

#### Register 46, ABSENCPOS position

For H2 encoder: Gives the absolute single turn position, 0-409500. Not corrected for the turntable.

For H3 and H4 encoder: Gives the absolute multi turn position,  $(-2^{31}) - (2^{31}) - 1$ . Not corrected for the turn table.

## 5.12

## Turn Table Mode

---

### 5.12.6 Turn Table notes

#### - Start up position.

The motor will start up with the position according to the settings described in section [Absolute position back-up, page 114](#) but will be adjusted to be within the turn table range.

#### - Modes other than Position mode

Software position limits are disabled when a turntable mode is selected.

Register 25, P\_IST\_TURNTAB is still updated to show the actual position relative to the Turn Table position range - even while P\_IST gets outside MIN\_P\_IST or MAX\_P\_IST.

One common way to do manual adjustment of the motor is to make a JOG function by using Velocity mode. After a JOG function the P\_IST may be left outside the Turn Table position range, so switching back to Position mode may cause the motor to move 'unexpectedly'. It is generally recommended to set the maximum velocity V\_SOLL, to zero when exiting a JOG operation and update P\_IST and P\_SOLL to desired values before setting V\_SOLL back to a non-zero value.

The QuickStep motor offers the following modes of operation:

**Passive mode:**

The motor will be in a completely passive state but communication is active and internal registers can be set up.

**Velocity mode:**

The motor velocity can be controlled using MacTalk software or by setting register 5 (V\_SOLL) using serial or program commands.

**Position mode:**

The motor position can be controlled using MacTalk software or by setting register 3 (P\_SOLL) using serial or program commands.

**Gear mode:**

The motor position and velocity can be controlled by pulse and direction or encoder signals at IN1 and IN2.

The gear ratio can be set to a large ratio using register 14 (GEAR1) and register 15 (GEAR2).

**Zero search mode type 1, type 2 and torque:**

Searches for sensor to define a zero position (Reference point).

**Cyclic Synchronous Position mode (CSP):**

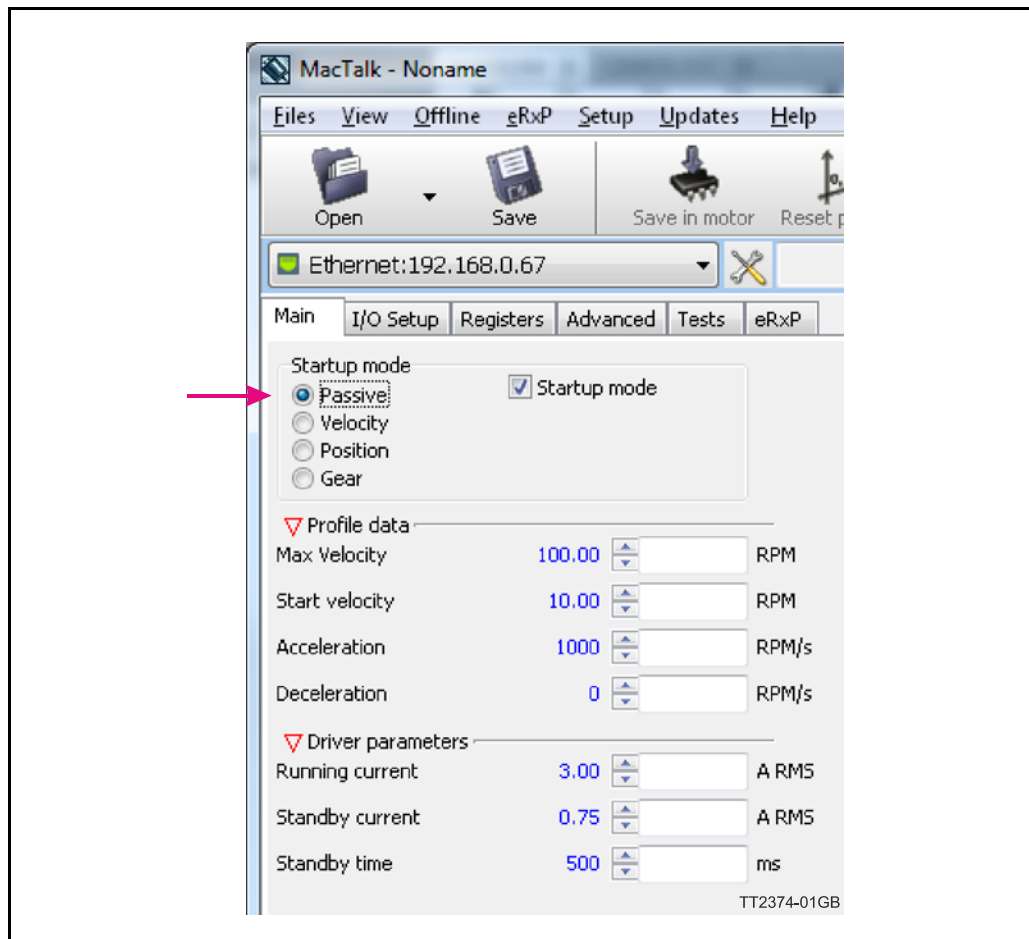
This mode is entered by the Ethernet module and enables very accurate positioning from the PLC. Cycle times down to 1 ms are supported and means that P\_SOLL is updated once every ms. Please consult the Ethernet manual for details.

## 6.1

## Passive Mode

### 6.1.1 Passive Mode

After power up, the controller will start up in passive mode. This means that it is possible to communicate and read/write to/from registers, but no current is supplied to the motor. It should thus be possible to turn the motor shaft as no voltage is connected to the motor. If there is encoder feed-back, the encoder counter will always register the correct position.

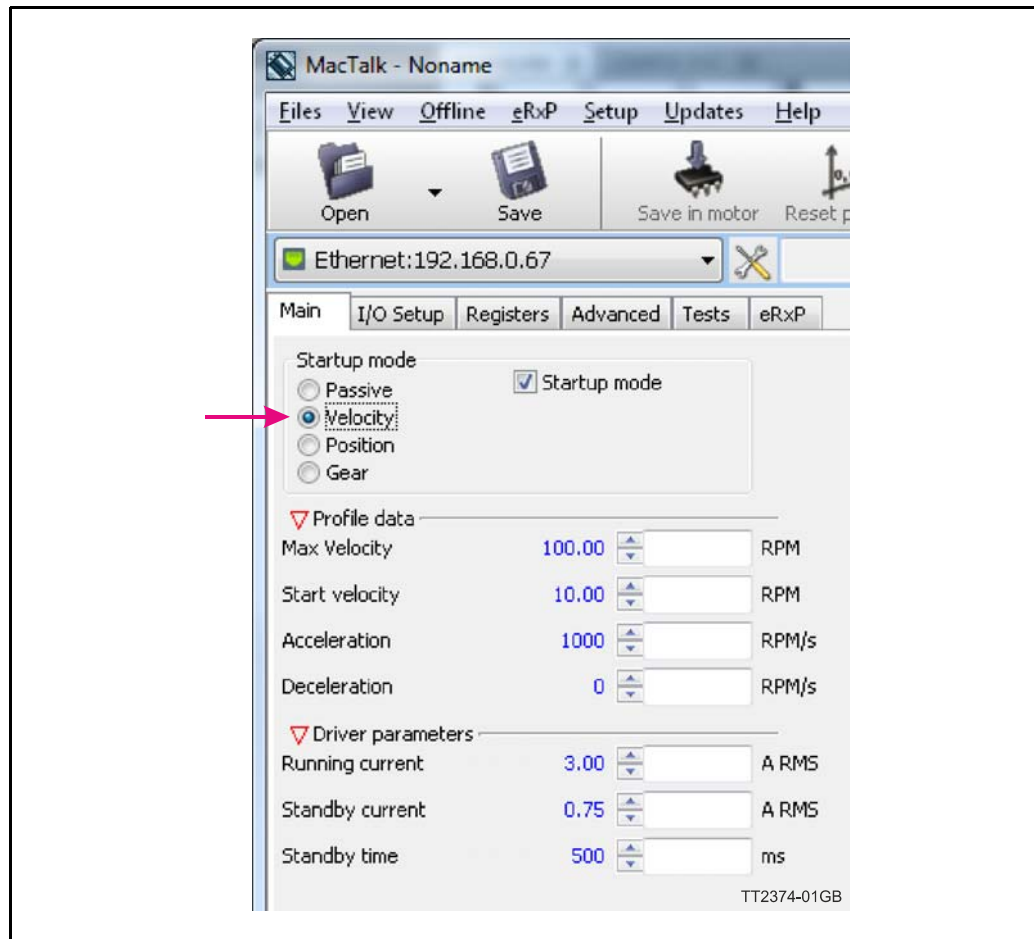


## 6.2

## Velocity Mode

### 6.2.1 Velocity Mode

In this mode, the QuickStep motor controls the motor velocity via the Max Velocity setting. This mode is typically used for simple tasks or for applications in which an overall unit, such as a PC-board or PLC, controls velocity and positioning.

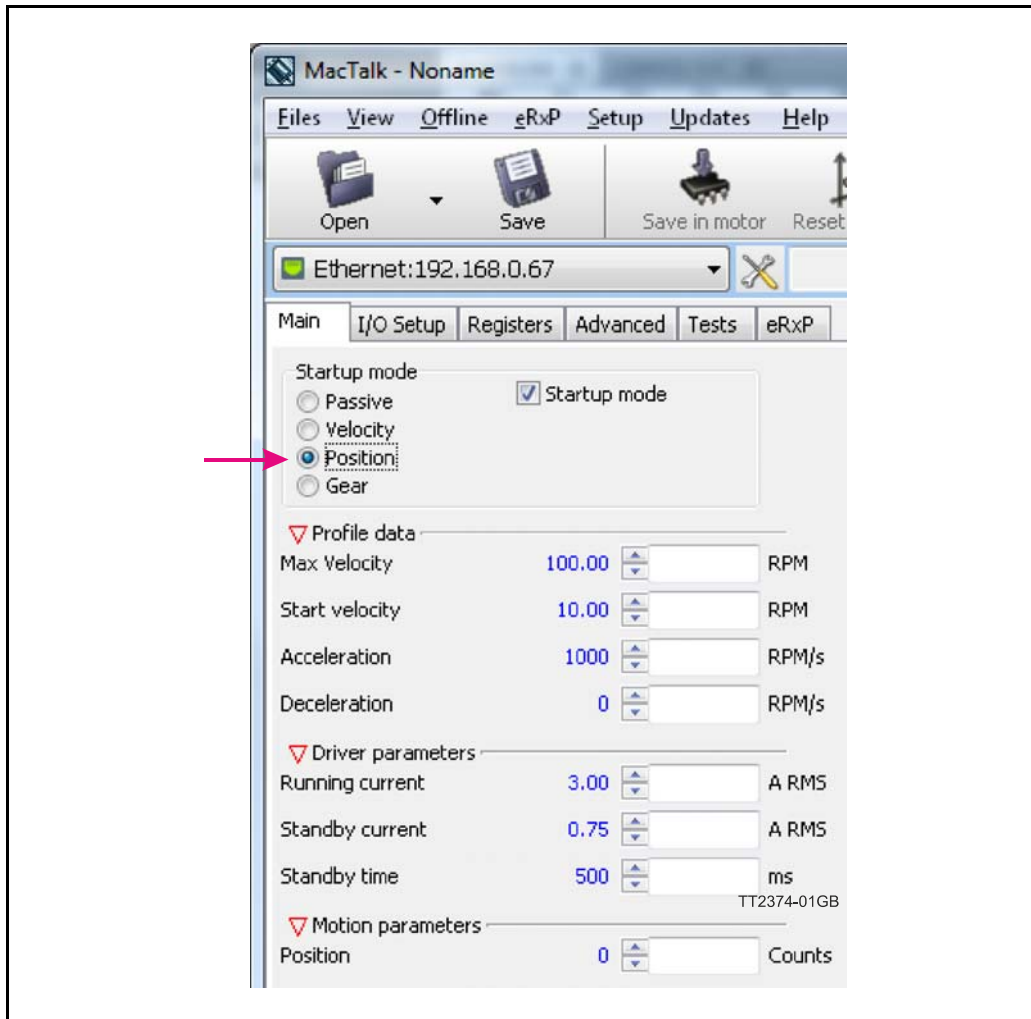


## 6.3

# Positioning Mode

### 6.3.1 Positioning Mode

In this mode, the QuickStep motor positions the motor via commands sent over the serial interface. Various operating parameters can be changed continuously while the motor is running. This mode of operation is used primarily in systems where the Controller is permanently connected to a PC/PLC via the interface. This mode is also well suited for setting up and testing systems. The mode is also used when programming is done.



## 6.4

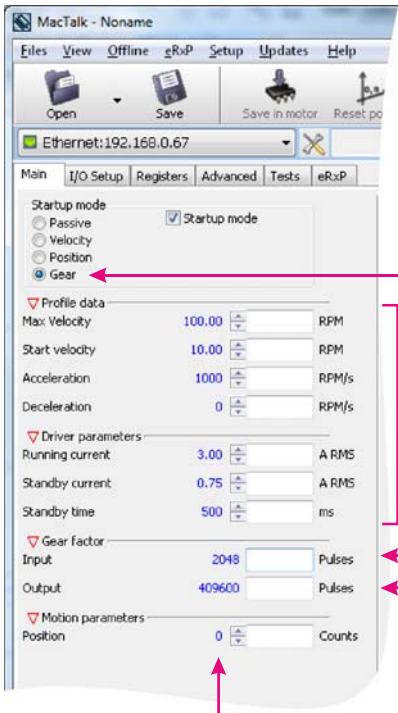
## Gear Mode

### 6.4.1 Gear Mode.

In this mode, the QuickStep motor functions as in a step motor driver. The motor moves one step each time a pulse is applied to an input.

Velocity, acceleration and deceleration are determined by the external frequency but can be limited and controlled by the QuickStep motor. In addition, the QuickStep motor also provides a facility for electronic gearing at a keyed-in ratio in the following intervals:

- **MIS17x, 23x, 34x, and 43x:** 1/2147483647 to 2147483647/1.



**Main parameters used in Gear Mode**

Select gear mode here.

Make sure that all these parameters are set to proper values in order not to cause any limitations in the motors ability to move.

Insert the resolution for the pulse source in this field.

Insert the resolution for the motor in this field.

Notice that if other ratios than 1:1 between pulse source and motor is desired either the input and/or output value must be scaled to match the desired ratio.

This value reflect the target position (P\_SOLL) which is controlled directly from the pulse source. This value should normally match the actual position unless the motor has stalled or some of the motion parameters have been set to a limiting value.

TT2462-01GB

#### Example:

A MIS231 motor has a resolution of 409600 steps/rev. and an encoder is connected which has a resolution of 2048 counts/rev.

If one revolution of the encoder should result in one motor revolution the Input must be set to 2048 and the Output to 409600.

If the motor must run 5 revolutions because there is a gear with a reduction of 5:1 the output must be set to  $5 \times 409600 = 2048000$  instead.

Notice that a very high ratio may cause unstable behaviour in the motor speed.

## 6.4

## Gear Mode

### 6.4.2 Signal formats supported.

If gear mode is selected an external pulse source can control the position of the motor. Following 2 formats are supported in all the MIS motors:

#### 1. Pulse and direction control

One input is applied with a pulse signal. Each rising edge at this input will cause the motor to move with a certain ratio (length) according to the gear registers “input” and “output” described at the previous page (see [Gear Mode.](#), page 155).

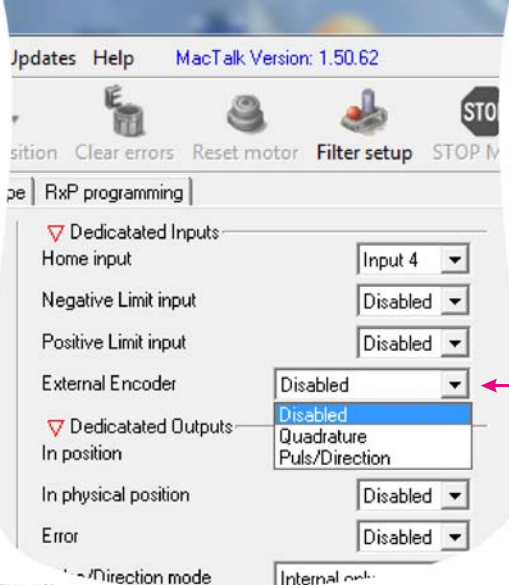
A secondary input controls in which direction the motor moves.

A falling edge at the pulse input do not have any effect.

#### 2. Quadrature control

When selecting this format 2 square wave signals (channel A and B) 90 degree phase shifted is applied to 2 inputs. Each transition (count) at the A or B channel will cause the motor to move with a certain ratio (length) according to the gear registers “input” and “output” described at the previous page (see [Gear Mode.](#), page 155).

The formats can be selected in MacTalk at the “I/O setup” tab.



MacTalk Version: 1.50.62

Position Clear errors Reset motor Filter setup STOP M

pe RXP programming

▼ Dedicated Inputs

Home input Input 4

Negative Limit input Disabled

Positive Limit input Disabled

External Encoder Disabled

▼ Dedicated Outputs

In position Disabled

In physical position Disabled

Error Disabled

TT2463-01GB

Direction mode Internal only

### How to setup the input format.

**Select input format here.**

- Quadrature is typically used for incremental encoders and supports a 2 channel 90 degree phase shifted signal. The direction is defined by the polarity on the phase shift.
- Pulse/direction is typically used as format in stepper motor systems. A pulse signal is applied to one input and the direction to another input.

## 6.4

## Gear Mode

### 6.4.3 How to connect the pulse source.

The present firmware require that the external pulse source is applied to specific inputs. These inputs differ from motor family to motor family.

Other input options are under development.

The external source must be connected to the following inputs:

Quadrature format	I/O terminal	Pulse/direction format	I/O terminal
Channel A	<b>A1+</b> and <b>A1-</b>	Pulse	<b>A1+</b> and <b>A1-</b>
Channel B	<b>B1+</b> and <b>B1-</b>	Direction	<b>B1+</b> and <b>B1-</b>

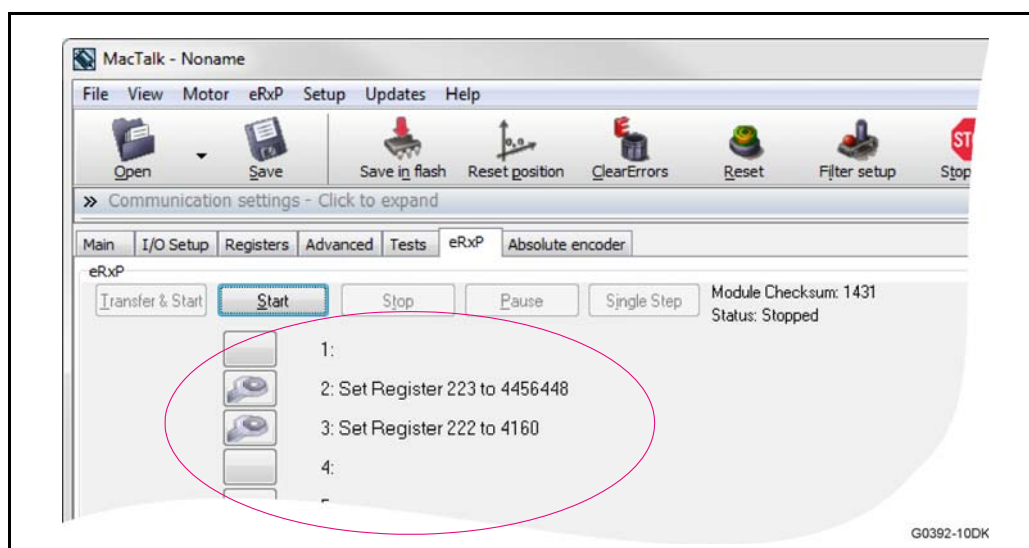


Notice that the Multifunction port pins mentioned above only operate at 5V levels and it is destructive if voltages outside the voltage range 0 to 5V is applied.

### 6.4.4 Single ended signals

If the external encoder signals are only available as single ended and with voltage levels up to CVO (typical 24V) it is possible to use IO2+3 as inputs.

To make this change, a small RxP program is required in addition to the settings in [Signal formats supported.](#), page 156 Signal formats supported:



Continued next page

## 6.4

## Gear Mode

---

After this program has been executed, the external source must be connected to the following inputs:

Quadrature format	I/O terminal	Pulse/direction format	I/O terminal
Channel A	<b><i>IO2</i></b>	Pulse	<b><i>IO2</i></b>
Channel B	<b><i>IO3</i></b>	Direction	<b><i>IO3</i></b>

Please note that the MIS motor family offers different connector configurations. For specific details about at which connectors the I/O terminals are available please consult: - [Connector overview for the MIS motors](#), page 41.

### 6.4.5 General considerations concerning cabling.

It is strongly recommended that shielded cable is always used when connecting the external pulse source to the pulse input to ensure that no noise from the surroundings affect the quality of the signal and worst case cause the motor movement to get affected. Also a good, solid ground wire between the motor and the source is recommended since any differences in the potential between the pulse source and the motor can affect the motor movement.

General guidelines concerning the I/O's are given in the following chapters:

- [User Inputs](#), page 25 or [General](#), page 36.

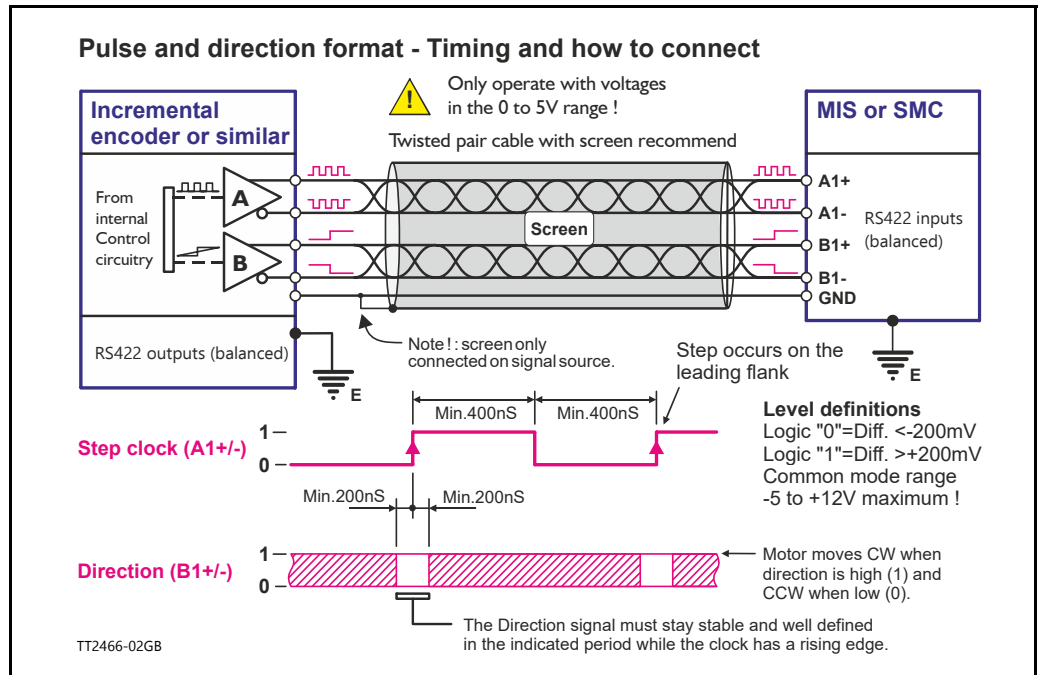
## 6.4

## Gear Mode

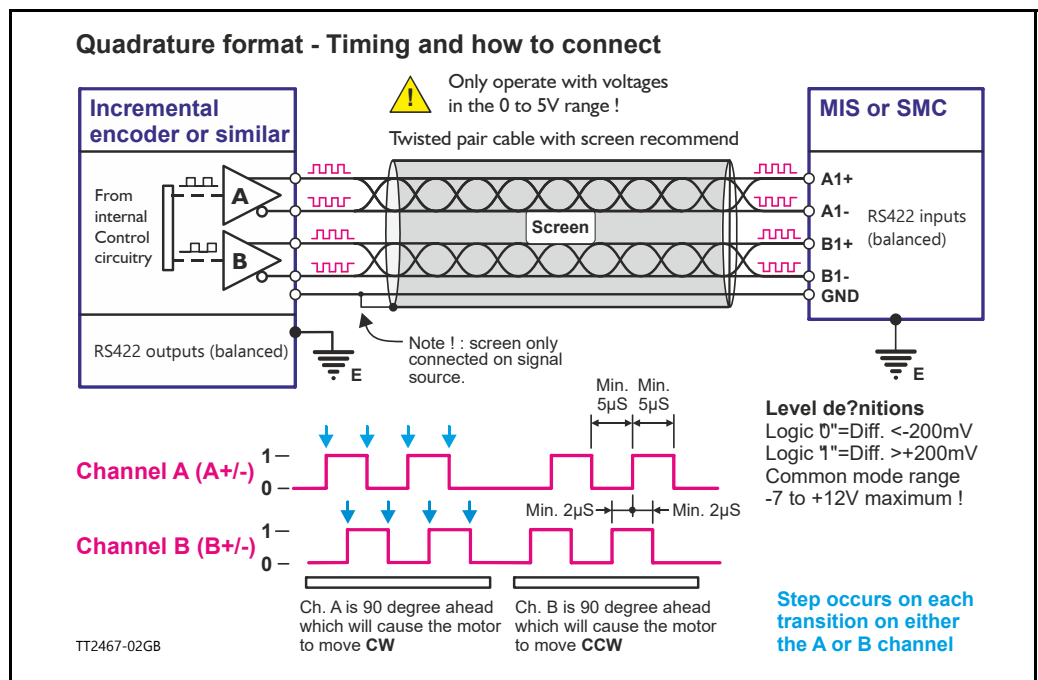
### 6.4.6

#### Signal function and timing.

The description below shows how to connect the pulse source when using the pulse and direction format. Also the timing is shown. Please be aware that if the indicated minimum timing is not respected the motor may lose some of the step clocks and the position of the motor will end up being out of synchronism with the pulse generator.



The description below shows how to connect the pulse source when using the quadrature format. Also the timing is shown. Please be aware that if the indicated minimum timing is not respected the motor may lose some of the step clocks and the position of the motor will end up being out of synchronism with the pulse generator.



## 6.4

## Gear Mode

### 6.4.7 Principle of gear mode.

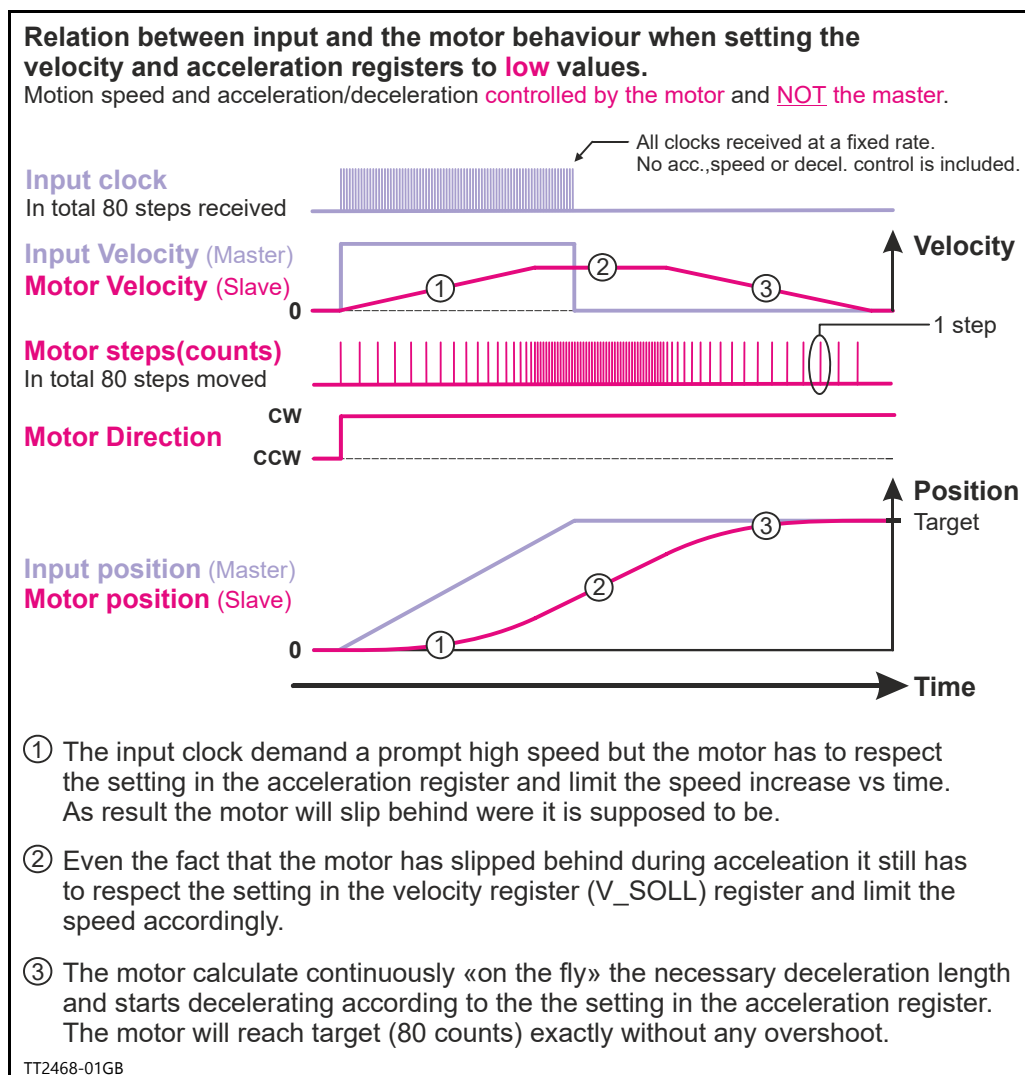
As mentioned on the previous pages the motor will follow the input signal synchronously to a certain extend according to the gear ratio setup. There are, however, a few other parameters which are vital for success in the actual application.

These parameters are:

**Velocity** The value of the velocity register will be respected at any time when the motor is in gear mode. Therefore, care must be taken if this is set lower than the equivalent speed of the external source producing clocks to the gear input since it will be a limiting factor.

**Acceleration** Similar to the velocity register the acceleration setting will be respected at any time and if set lower (slower speed rise time) than the external source producing the clocks it will start to be a limiting factor.

Below can be seen an example of the relation between applied clocks and the actual motor movement when having velocity and acceleration set to low values.

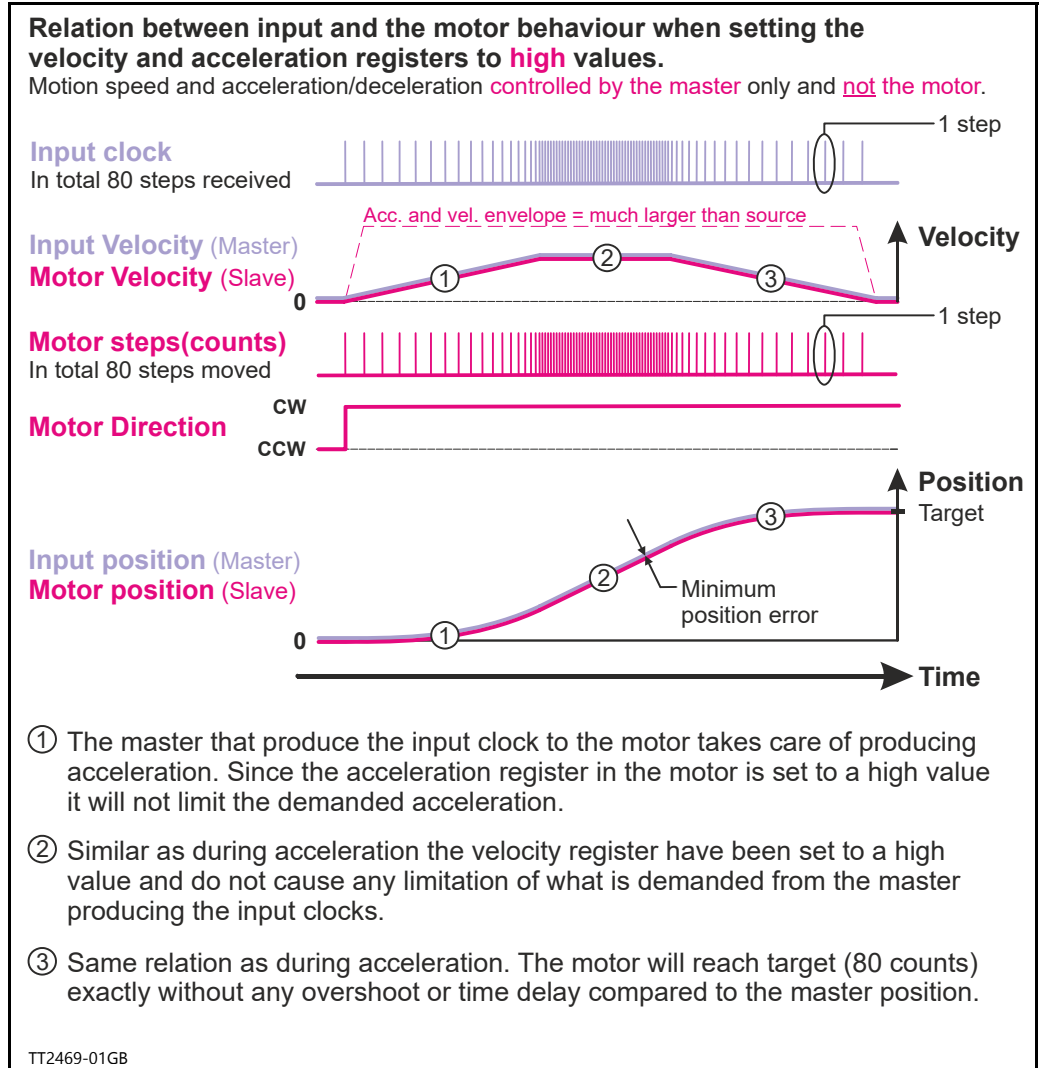


## 6.4

## Gear Mode

As an alternative to the previous illustration the full control can be done by the master by setting the velocity and acceleration registers to a significant higher value than the equivalent value of the clock source which will assure that the motor follows each clock with a very narrow timing and no delays.

This solution must be used if the master that produces the clocks do generate the full motion profile with acceleration to a desired top speed and make sure to decelerate and hit target.



## 6.4

## Gear Mode

### Example 1:

#### Encoder (quadrature) input.

An external encoder feeds the MIS. The I/O type is set to "Pulse input" and "Input type" is set to "Quadrature" in order to decode the encoder signal. The encoder is connected to the 2 pulse input terminals.

See also [How to connect the pulse source.](#), page 157.

The resolution of the external encoder is 2048 cpr.

The MIS motor itself has 409600 cpr. If this application requires that the MIS motor rotates 1 rev. each time the external encoder has rotated 1 rev., the *Input* parameter is set to 2048

(external encoder) and the *Output* parameter is set to 409600.

Now the ratio between the external encoder and the MIS motor will be 1:1. Ensure the "Profile data" is set to proper values in order not to limit motor operation unintentionally.

### Example 2:

#### Pulse and direction input.

A traditional step motor system with separate driver and motor is replaced by the integrated MIS motor, meaning that the MIS motor receives a pulse and direction signal which is a very common signal format in step motor applications.

The I/O type is set to "Pulse input" and "Input type" is set to "Pulse-direction" in order to decode the input signal. The pulse signal is connected to the 2 pulse input terminals. See also [How to connect the pulse source.](#), page 157.

The MAC motor is replacing a step motor system with 400 steps per revolution, which means that when the pulse source produce 400 pulses, it expects the MIS motor to rotate one revolution.

The MIS motor itself has 409600 cpr. If this application requires that the MIS motor rotates 1 revolution each time 400 pulses are received, the *Input* parameter is set to 400 since the MIS motor interpret every rising edge at the applied pulse signal as one count (step). The *Output* parameter is set to 409600 since this is the number of counts (steps) on one revolution.

Now the MIS motor will move 1 revolution for every 400 pulses that are applied to the pulse input. Ensure the "Profile data" is set to proper values in order not to limit motor operation unintentionally.

The following table can be used as guide for setting up typical gear ratios:

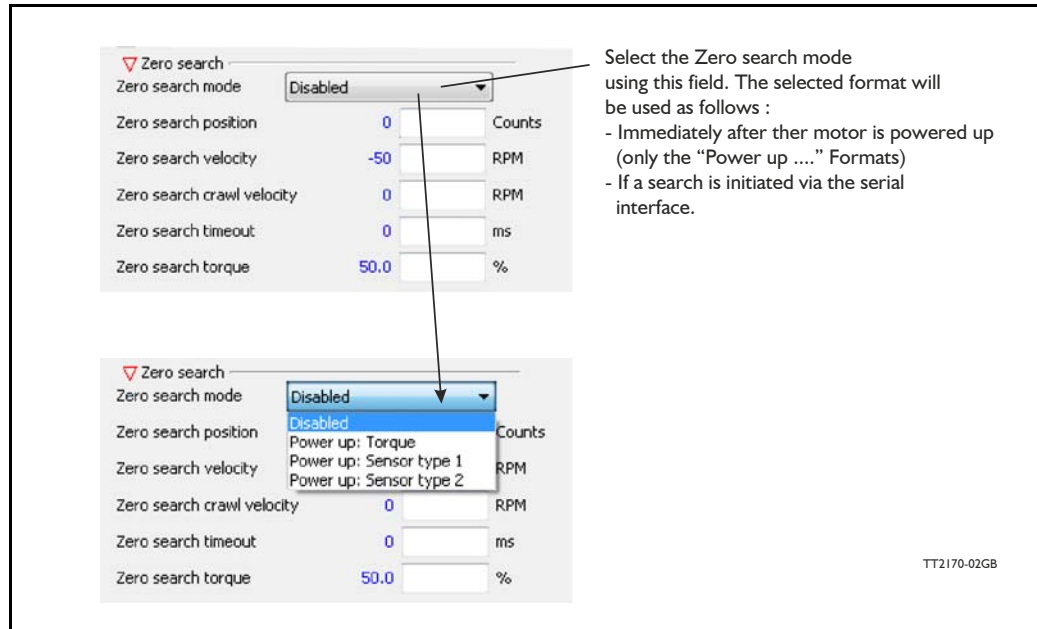
Pulse and direction gear ratio setup - "Commonly used ratios"		
Applied number of pulses(clocks) per desired MIS motor revolution.	"Input" register	"Output" register
200	200	409600
400	400	409600
500	500	409600
800	800	409600
1000	1000	409600
1600	1600	409600
2000	2000	409600
409600 (Equal to the MISxxx resolution)	409600	409600

## 6.5

## Zero search modes

### 6.5.1 Mechanical zero search modes

In all positioning systems there is a requirement to be able to find a mechanical zero position after the system is powered up or at specific times during operation. For this purpose the MIS motor offers 3 different Zero search modes which can be selected from the MacTalk main window or by sending a command via one of the serial interfaces.



The menu offers 3 options:

**Disabled** (default)

The Zero search is disabled.

**Power up: Sensor type 1**

The *Zero search* function will start seeking for Zero until an external sensor is activated. The point at which the sensor is activated is defined as the zero.

**Power up: Sensor type 2**

Like above (Sensor type 1) but after the sensor is activated the direction of movement is reversed and the point at which the sensor is disabled is defined as zero.

**Power up: Torque**

Only available with H2/H4 encoder installed and with Closed loop and Current control enabled. Please see "5.3 Closed loop". The Zero search will start searching for Zero until a mechanical "collision" occurs. The point at which the motor torque is equal to the specified value of the Zero search torque is defined as the zero position.

The following sections explain in detail the functionality of the 3 fundamental Zero search modes.

## 6.5 Zero search modes

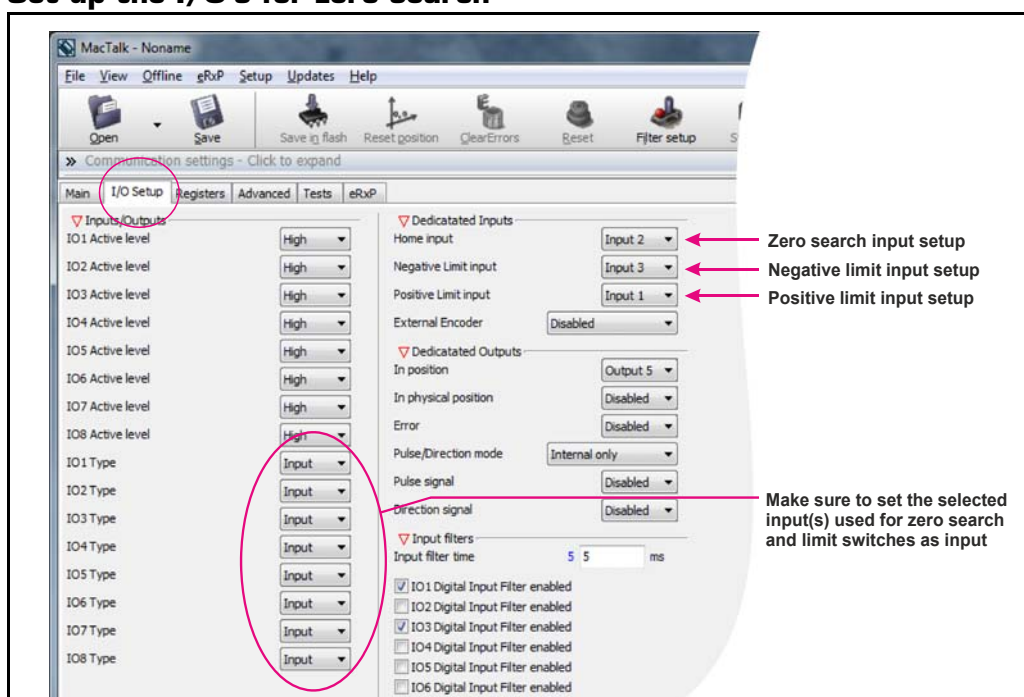
### 6.5.2 Starting a Zero search

If the Zero search mode is set to *Disabled*, no Zero search is done at any time unless written in a program or ordered from an external interface such as RS485, CANopen or ethernet.

If one of the 3 modes *Power up: Sensor type 1*, *Sensor type 2* or *Torque* is selected, the respective Zero search mode will be executed every time the MIS motor is powered up unless a eRXP program exist. If a program has been made and is running, the Zero search command must be executed within the program to execute a Zero search.

The MIS motor's zero search facility is very flexible. The inputs for reference and limit switches must be set up correctly before use.  
The active levels must also be set up correctly.

### 6.5.3 Set up the I/O's for zero search



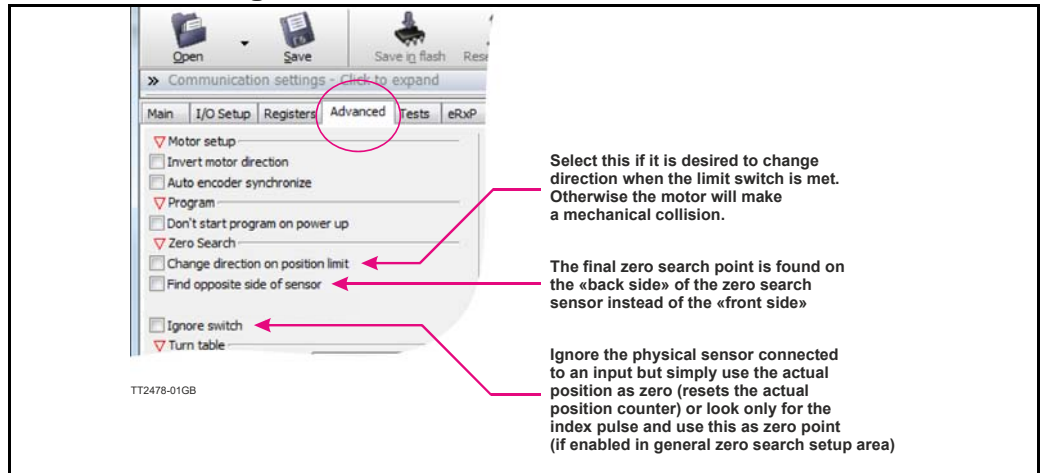
**Important information:** Each of the 8 pins can be defined as inputs or outputs. The active digital input level for each input is also defined in the above screen. Furthermore, it is possible to set up a filter for each input to avoid noise interfering with the program. The inputs for Home, Negative Limit and Positive Limit are selected here.

## 6.5

## Zero search modes

### 6.5.4

#### Advanced settings



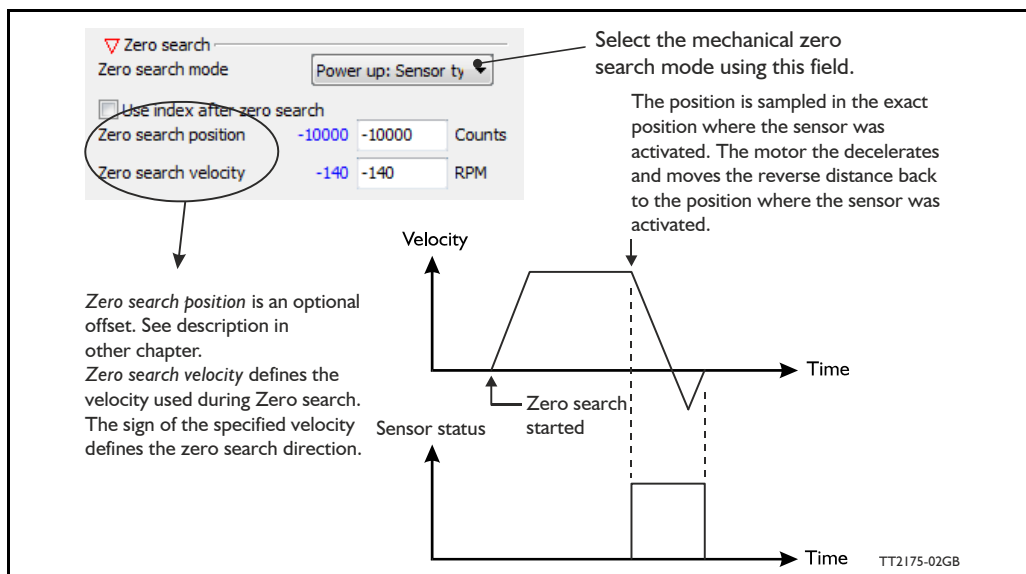
There are several ways to perform a Zero search:

- Start from both sides of the reference sensor in a system with limit switches without having position limit problems.
- to go to the opposite side of the sensor and use this position as zero position.
- use a position limit as reference position. In this case the zero search position must be different from 0 or the motor enters passive mode.
- ignore the reference switch input and use the actual position or index pulse as zero position before using the zero search position.

## 6.5 Zero search modes

### 6.5.5 “Sensor type 1” Zero search

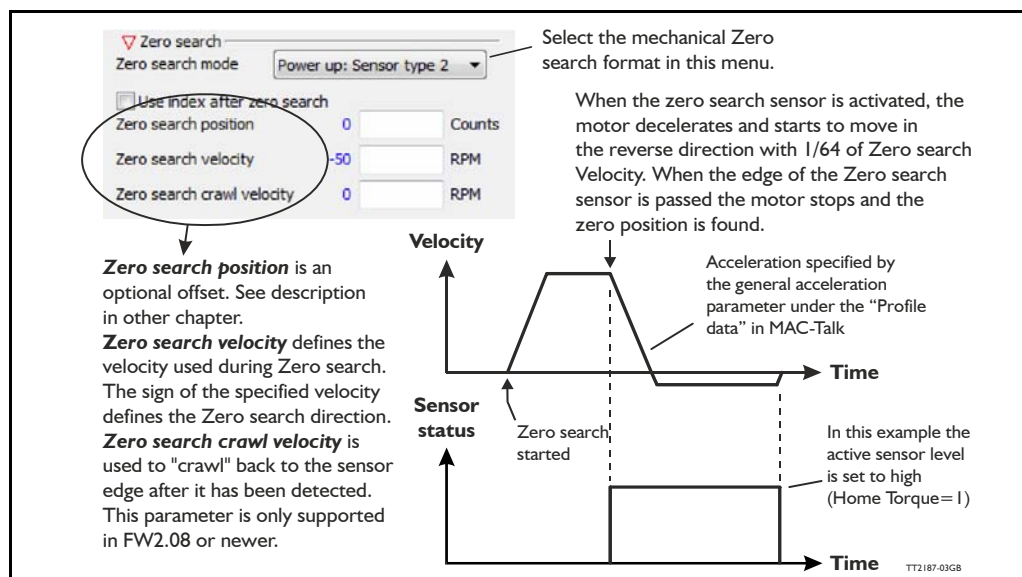
Sensor type 1 zero search is carried out according to the following illustration:



The Zero sensor must be connected to a user input  
For connection information, see [User Inputs](#), page 25

### 6.5.6 “Sensor type 2” Zero search

Sensor type 2 zero search is carried out according to the following illustration.  
This type of zero search is more precise than the other types of zero search.



The Zero sensor must be connected to a user input. For connection information, see [User Inputs](#), page 25.

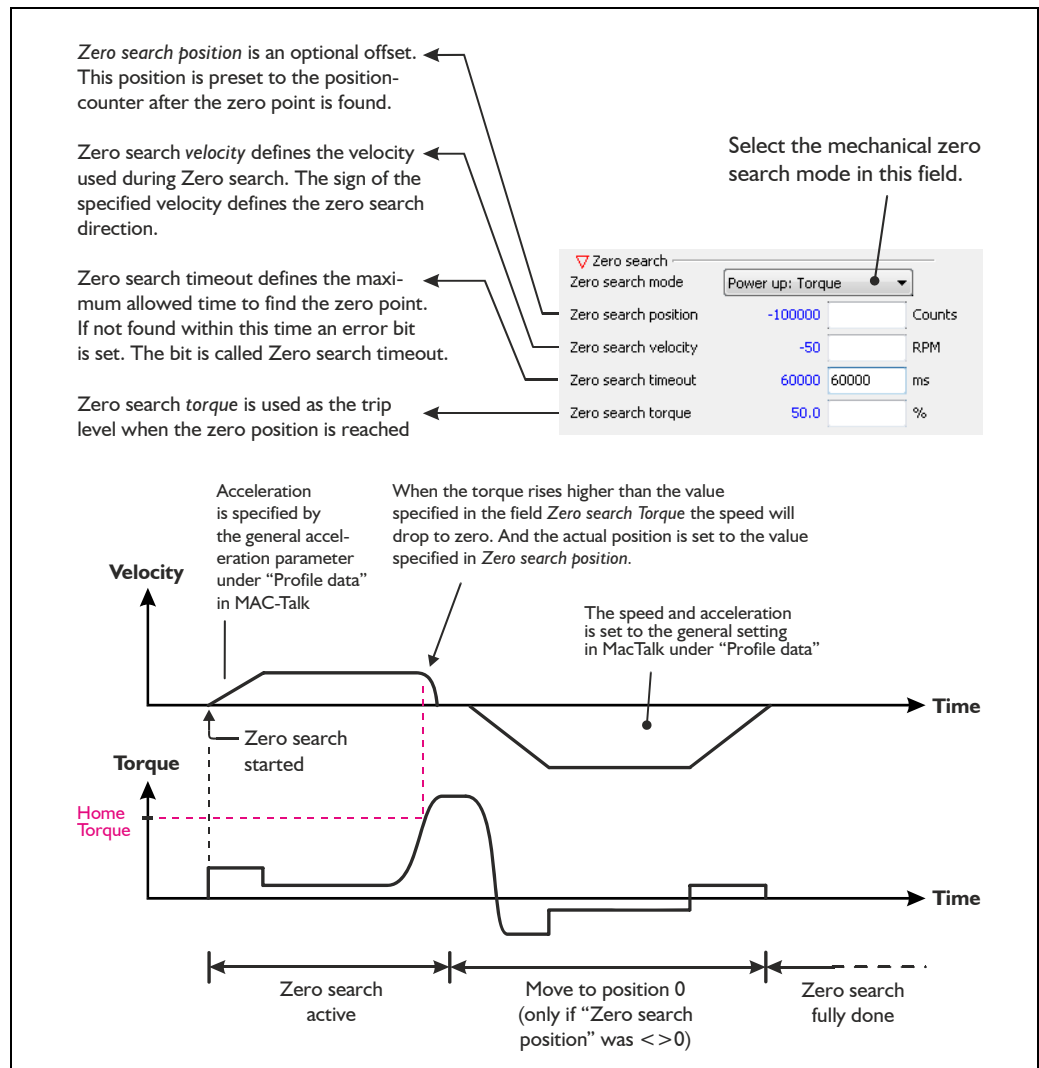
**Hint:** Make sure the acceleration/deceleration is set to an appropriate value which stops the motor when the Zero search switch is detected but before mechanical collision.

## 6.5

## Zero search modes

### 6.5.7 “Torque” Zero search

Torque Zero search is carried out according to the following illustration.



The Zero search method using a torque as a reference is a cheap, simple way to find the mechanical zero position, but please be aware of following critical points.

- Make sure that the Zero search torque is set to a proper value higher than the mechanical friction in the system in order to avoid a faulty zero point being found. It is a good idea to let the motor run in velocity mode with the same velocity and observe what the actual motor torque is. This value can be observed in the status area in the right side of the main window. Set the Zero search torque to a value 10-20% higher than the actual torque observed during this procedure.
- To improve the repeatability precision of the zero point make sure that the mechanical “collision” point is as stiff and well-defined as possible.

## 6.5

## Zero search modes

### 6.5.8 Making a Zero point offset

Common for all the zero search modes, it is possible to optionally define the zero point as a value other than zero (position 0).

When is it useful to use the zero point offset?

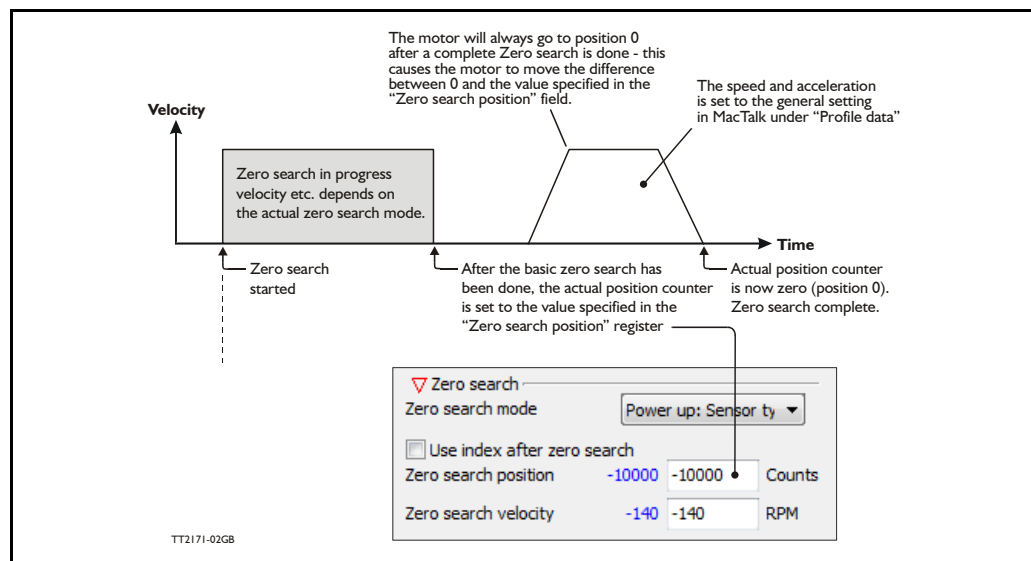
- If it is required that the position interval under normal operation is always convenient values from 0 to x instead of a mixture of negative and positive values. This can happen if the zero point sensor is placed a long distance away from the normal positioning interval or inside the normal positioning interval.
- If an automatic move to an initial position is desired after a power up zero search.

The offset value must be specified in the “Zero search position” field.

The complete zero search will be performed in the following order:

1. The zero search is started either automatically (power up) or initiated by a command from the serial interface.
2. The basic Zero search is completed and the position counter is set to the value specified in the “Zero search position” field.
3. If the zero search position value is different from position, the motor will now move to position 0.
4. The zero search is now complete and the motor will switch to normal operation, i.e. the mode selected in the “Start up mode” field in the main window.

The illustration below shows the complete zero search cycle.



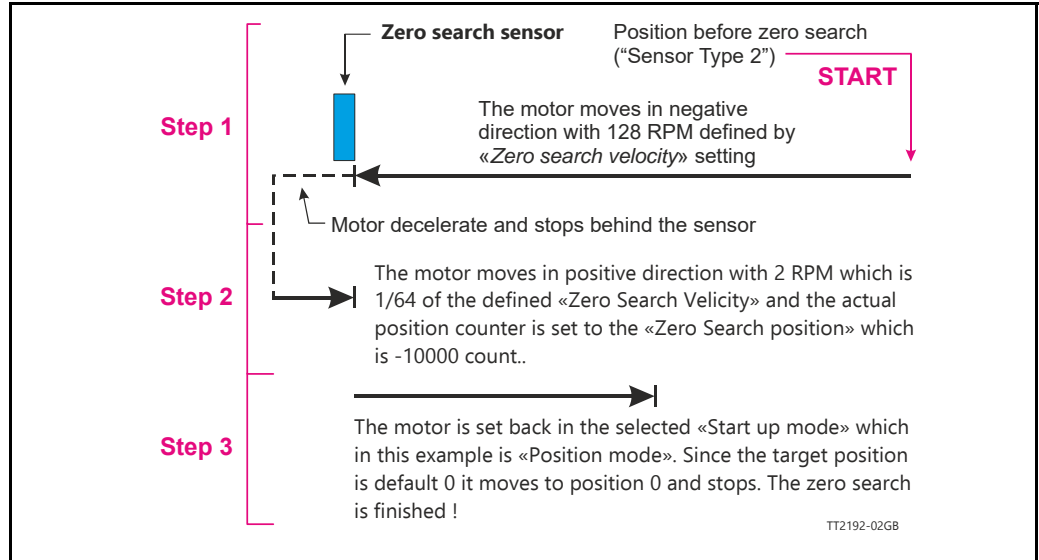
## 6.5

## Zero search modes

### Zero point offset Example.

Setup done before start:

- Zero search velocity = -128 rpm
- Zero search position = -10000 counts



### 6.5.9 Setting up zero search without MacTalk

If MacTalk is not used for setting up parameters/registers related to the zero search feature it must be done as follows.

The motor contains a number of registers which can be accessed from various protocols depending at which options the motor has.

Protocols available are for example Ethernet (EthernetIP, PROFINET etc.) and CAN-open, Modbus or the MacTalk protocol.

Each field in MacTalk described earlier in this chapter is accessing a register in the motor. The registers that are relevant for zero search operation are:

#### Zero search basic settings:

- |                     |  |
|---------------------|--|
| <b>R38 P_HOME</b>   | <b>MacTalk name: “Zero search position”</b><br>The found zero point is offset with this value.<br>See also <a href="#">P_Home</a> , page 210   |
| <b>R40 V_HOME</b>   | <b>MacTalk name: “Zero search velocity”</b><br>The velocity to use during zero search.<br>Set a negative velocity to search in the negative direction.<br>See also <a href="#">V_Home</a> , page 210 |
| <b>R41 T_HOME</b>   | MacTalk name: "Zero search torque (0-100%)"<br>The measured torque where the zero search must end.<br>See also <a href="#">T_Home</a> , page 210   |
| <b>R42 HOMEMODE</b> | <b>MacTalk name: “Zero search mode”</b><br>Selects the zero search type that should start on power up.<br>See also <a href="#">Home mode</a> , page 210  |

## 6.5

## Zero search modes

---

- R120 INDEX\_OFFSET** **MacTalk name: N/A**  
The position of the zero sensor relative to the encoder index. This is set after a zero search where the index is used. See also [Index\\_Offset](#), page 218
- R242 V\_HOME\_CRAWL** **MacTalk name: “Zero search crawl velocity”**  
In Zero Search type 2, the “crawl” velocity is V\_HOME/64 by default.  
If register 242:V\_HOME\_CRAWL is <>0, a user defined velocity is used – independent of V\_HOME. Please note that overshoot can occur if this velocity is set too high.  
See also [V\\_HOME\\_CRAWL](#), page 239
- R243 V\_HOME\_TIMEOUT** **MacTalk name: “Zero search timeout”**  
The default zero search time out is 60 s. This register sets another time out in milliseconds. If the time out is passed the motor will return to startup mode.  
If the timeout has passed, the motor will set an error.  
Please see register 35 - *Err\_Bits*, page 208  
See also [V\\_HOME\\_TIMEOUT](#), page 239

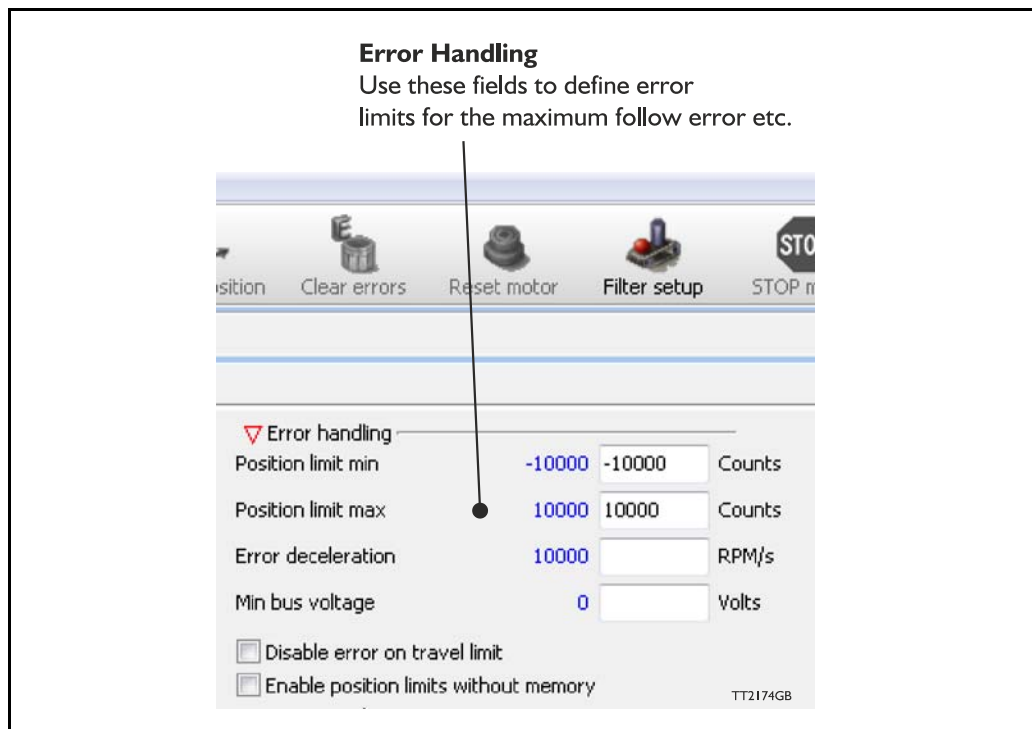
### Zero search advanced settings:

- R122 Zero\_Search\_BITS** **MacTalk names:** (multiple - see below)  
“Use index after zero search”, bit 0  
“Change direction on position limit”, bit 1  
“Find opposite side of sensor”, bit 2  
“Ignore switch”, bit 4  
“Disable zero search timeout”, bit 5  
Explanation of the individual bits see [Advanced settings](#), page 165. See also [Zero\\_Search\\_Bits](#), page 219

### Zero search I/O setup:

- R125 IOSETUP** **MacTalk names: “I/O Setup” tab**  
Bit 0-7 Sets the I/O active level.  
Bit 8-15 Enables the I/O as an output.  
See also [IOsetup](#), page 220
- R132 HOME\_MASK** **MacTalk name: “Home input”**  
Input mask for home sensor input(s), each bit select which of the I/O 1-8 to use as input for the home sensor signal. See also [Home\\_Mask](#), page 222
- R135 INPUT\_FILTER\_MASK** **MacTalk names: “Input filters”**  
Input mask for the digital inputs with input filter.  
Bits set use the input filter time in register 136, bits cleared use a fixed update time of 100 us.  
See also [Input\\_Filter\\_Mask](#), page 223
- R136 INPUT\_FILTER\_CNT** **MacTalk name: “Input filter time”**  
The number of milliseconds the filtered digital inputs must be stable before accepting a change.  
See also [Input\\_Filter\\_Cnt](#), page 223





The MIS motor contains 5 fundamental parameters which are used for protection related purposes. They all have effect regardless of which mode of operation the motor is set to use.

#### Position limit min. and max.

Same as physical limit switches but implemented in software. Default is 0 meaning that the feature is disabled. If one parameter is different from 0, both values are activated. See also [End-of Travel Limit Inputs](#), page 134

#### Error acceleration

If a fatal error occurs, it can be convenient to use a controlled deceleration instead of a sudden stop. If the inertia in the system is high and the mechanical parts are weak, a sudden stop can cause damage and unintended behaviour. Use this parameter to define the deceleration used during a fatal error. Default is 0, meaning that the feature is disabled.

#### Min. bus voltage

This is the level of P+ at which the motor goes into error state “low bus voltage”.

#### Disable error on travel limit

If one of the position limits (external sensors) are activated no error is generated. This will avoid that the motor enter passive mode and make the motor power less. See also [End-of Travel Limit Inputs](#), page 134

#### Enable position limits without memory

See [Simple mode: Position limits without memory](#), page 138

## 7.2 Error messages

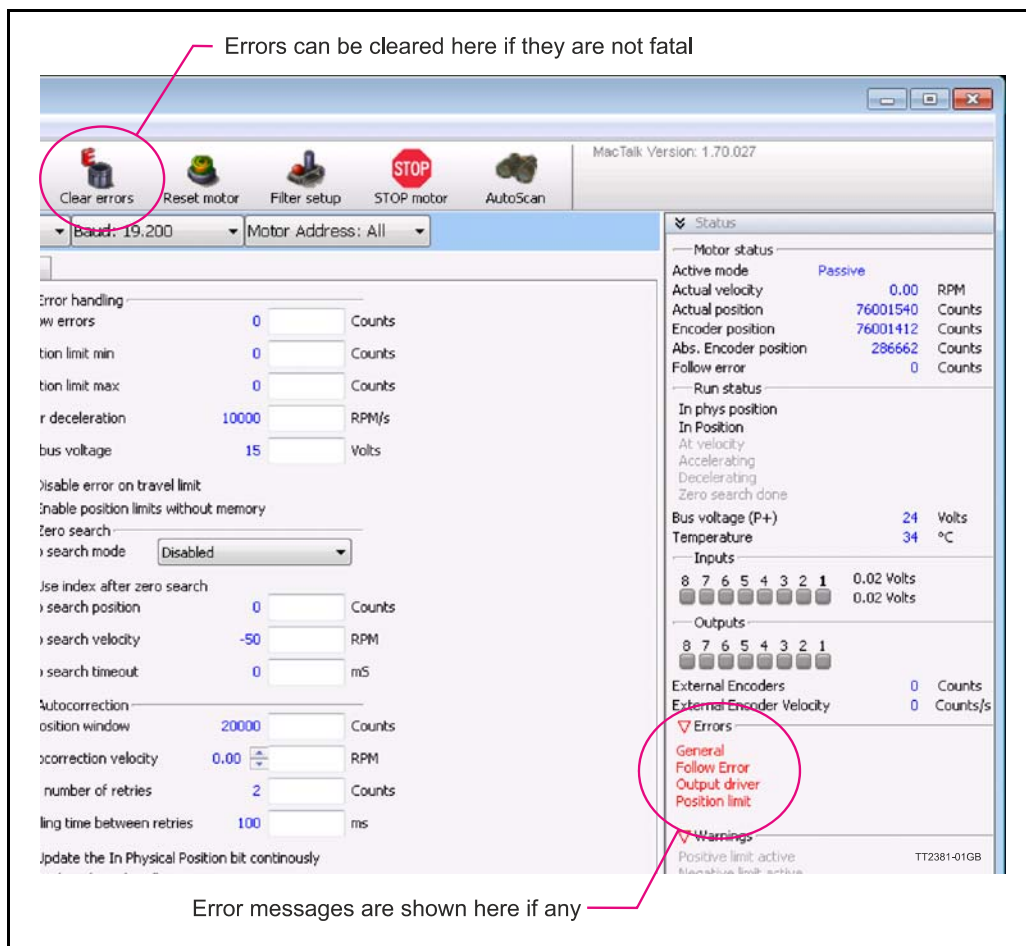
### 7.2.1 How to monitor Errors

Any error that occur will show up in the right side of the MacTalk screen.

Some errors can be cleared by pressing the “Clear Error” button.

Other errors are fatal and need other actions.

For example will a temperature error not be possible to clear before the temperature is lower than what is accepted.



The next pages describe every error including cause for the error and how to solve the error situation.

## 7.2

## Error messages

### 7.2.2 Reading the Event log

In MacTalk it is possible to monitor all the saved data. This tab shows the total amount each error has occurred, the last 20 errors with time stamps, total amount of revolutions the motor has run during the entire lifetime etc.

The motor do not have a real time clock so all time stamps are based on the active time where the motor has been powered also showed as “Up time”.

MacTalk - Noname

Files Motor eRXP Setup Updates Help

Open Save Save in motor Reset position Clear errors Reset motor Filter setup STOP motor AutoScan

MacTalk Version: 1.70.027

Serial port Comport: 1 Baud: 19,200 Motor Address: All

Main I/O Setup Registers Advanced **Event Log** Tests eRXP

Error(s) and events logged in the motor

Error type	Number of errors	Last error time
Follow error	9	27h:49m:11s
Output driver	2	23h:37m:21s
Position limits	2	23h:37m:21s
Low bus voltage	4	23h:37m:21s
Overvoltage	2	23h:37m:21s
Temperature	2	23h:37m:21s
Internal error	4	23h:37m:21s
Encoder lost position	2	23h:37m:21s
Encoder reed sensor	2	23h:37m:21s
Encoder no communication	2	23h:37m:21s
External encoder	2	23h:37m:21s
Closed loop lost sync	2	23h:37m:21s
Saved encoder position	0	0h:0m:0s
Saved P_IJT	0	0h:0m:0s

Up time [Hours:Minutes:Seconds]  
**28h:47m:1s**

Saved events

Saved event	Saved value
Saved encoder position	193990
Saved P_IJT	000390
Powerdown count	52
Total runtime [hours:minutes:seconds]	27h:59m:27s
PLC Flash savings	2
Userdata flash savings	2
Saved SSI data	0
Saved Ext. Encoder data	0
Total amount of rotations [1000: rev]	36101

Error type Last error time

Follow error	27h:49m:11s
Follow error	27h:11m:37s
Closed loop lost sync	23h:37m:21s
External encoder	23h:37m:21s
Encoder no communication	23h:37m:21s
Encoder reed sensor	23h:37m:21s
Encoder lost position	23h:37m:21s
Internal error	23h:37m:21s
Temperature	23h:37m:21s
Output driver	23h:37m:21s
Internal error	23h:37m:21s
Follow error	23h:37m:21s

Motor status

Active mode Gear 0.00 RPM

Actual velocity 888398 Counts

Actual position 888398 Counts

Encoder position 198106 Counts

Follow error 55 Counts

Run status

In phys position

In Position

At velocity

Accelerating

Decelerating

Zero search done

Bus voltage (P+) 24 Volts

Temperature 32 °C

Inputs

8 7 6 5 4 3 2 1 0.02 Volts

Outputs

8 7 6 5 4 3 2 1 0.02 Volts

External Encoders 0 Counts

External Encoder Velocity 0 Counts/s

Errors

Warning MIS340 on COM1

MIS340 (Version V4.00.00070.SN: 1) Connected

TT2394-01GB

## 7.2

## Error messages

### 7.2.3 Error messages and error handling

The following list show the possible error messages, the cause of the error and possible actions to prevent the error from happening. Each error can also be monitored by reading the Error register (register 35) by using software packages like the OCX driver or MacRegIO.

### 7.2.4 Error message 'Follow error'

Message no. / Message	1 / 'Follow error'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The Follow error (register 20) has exceeded the value specified in "Follow error max" (register 22).
Possible cause of this error	"Follow error max" is set to a too low value and therefore the Follow error exceeds this value at normal operation. The motor has stalled because of too much load or a too low "Running current" (register 7). Faulty encoder
Solutions to avoid error	Set "Follow error max" to a "much" greater value than the average "Follow error" while running at the desired velocity (V_SOLL, register 5). Please allow some overhead in order to avoid Errors because of small spikes in the "Follow error". Please notice that 1 revolution is 409600 counts.  Make sure that the "Running current" (register 7) is set high enough to avoid step loss or stalling.  In Passive mode, the shaft can be moved freely to check that the encoder is counting properly – 1 revolution should make 409600 counts. Looking at the front of the motor, the positive counting direction is clockwise.
How to return to normal operation	Clear the error bits in register 35. Return to the desired active mode. or Reset the motor / Cycle the power
Error bit / Firmware name	Bit 1

### 7.2.5 Error message 'I/O Output driver'

Message no. / Message	2 / 'I/O Output driver'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	1 or more of the 8 IO's has a hardware fault.
Possible cause of this error	An IO has been setup as an output, but 24 V is applied to that specific output.  Output pins are overloaded.
Solutions to avoid error	Make sure that the setup is correct. The easiest way to do this is by connecting to MacTalk and go to the I/O setup tab. Here the actual setting for each pin is shown. The status on the Inputs and Outputs can be monitored in the right "Status" panel.  Always take care not to load the outputs by more than 300 mA per channel. Please consult the <a href="#">User Outputs</a> , page 35 section.  Check that none of the wires are short-circuited.
How to return to normal operation	Reset the motor / Cycle the power
Error bit / Firmware name	Bit 2

## 7.2 Error messages

### 7.2.6 Error message 'Position limit'

Message no. / Message	3 / 'Position limit'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	<p>This error relates to both the Hardware and software position limits. It is set when one of the 4 conditions is true:</p> <ul style="list-style-type: none"><li>• The positive sensor has been active</li><li>• The negative sensor has been active</li><li>• Actual Position is greater or equal to Max position (register 30)</li><li>• Actual Position is less or equal to Min position (register 28)</li></ul>
Possible cause of this error	<ul style="list-style-type: none"><li>• One of the position limits is reached.</li><li>• Noise on the hardware limit input.</li><li>• Faulty limit sensor.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• When position limits are activated, make sure that the motor does not reach the limits. An internal absolute multi turn encoder can be a good help to avoid reaching the limits.</li><li>• If the motor must reach the limits, but also stay in the active mode, it is possible to disable the 'Position limit' error message by setting Setupbit 17 in register 124. Please see <a href="#">Position Limits</a>, page 134</li><li>• If noise has triggered the hardware position limits, it is recommended to add some digital filtering on the inputs. The easiest way to do this is by connecting the motor to MacTalk and setup the filter for the specific input on the I/O setup tab. See also <a href="#">Digital input filter setup with MacTalk</a>, page 27</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Clear the error bits in register 35.</li><li>• Return to the desired active mode. It is now only possible to run the motor in the opposite direction of the limit.</li><li>• Reset the motor / Cycle the power</li></ul>
Error bit / Firmware name	Bit 3

### 7.2.7 Error message 'Low bus voltage'

Message no. / Message	4 / 'Low bus voltage'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	<ul style="list-style-type: none"><li>• The measured bus voltage is lower than the level Min_Busvol (register 98).</li><li>• The voltage of the P+ bus voltage has been measured to be lower than the limit selected in the register 'Min_Busvol' (register 98). This has resulted in an error as configured in the setup of 'Undervoltage handling'. See also <a href="#">Under voltage Handling</a>, page 139.</li></ul>
Possible cause of this error	<ul style="list-style-type: none"><li>• The current rating of the external power supply is too small.</li><li>• The power supply is not able to deliver the required peak currents that the motor need. This is a typical problem when using switch mode power supply.</li><li>• The power cable is under dimensioned.</li><li>• The under voltage min. setting must be decreased.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Use a power supply with a higher current rating.</li><li>• Use a power cable with at least 0,75mm<sup>2</sup> wires (up to cable lengths of 10m. If the power cable is longer, use 1,5mm<sup>2</sup> or use multiple wires in parallel.</li><li>• Connect a capacitor across the supply line close to the motor. Especially if using a switch mode power supply this will help.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Reset the motor, clear the error bit(s) in register 35 or cycle the power.</li></ul>
Error bit / Firmware name	Bit 4

## 7.2

## Error messages

### 7.2.8 Error message 'Over voltage'

Message no. / Message	5 / 'Over voltage'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	A too high P+ voltage (> 100 V) has been measured.
Possible cause of this error	<ul style="list-style-type: none"><li>• A too high P+ voltage has been connected.</li><li>• The returned amount of energy from the motor has been too high. This can typically happen if:</li><li>• The motor decelerate a large load inertia too fast. - The motor is turned by an external force.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Decrease the load inertia.</li><li>• Decrease the top speed and/or the acceleration value.</li><li>• Make sure that the supply voltage is within the limits.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Reset the motor, clear the error bit(s) in register 35 or cycle the power.</li></ul>
Error bit / Firmware name	Bit 5

### 7.2.9 Error message 'Temperature'

Message no. / Message	6 / 'Temperature'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The temperature has been higher than 90°C (194F) which is not allowed.
Possible cause of this error	<ul style="list-style-type: none"><li>• The ambient temperature is higher than allowed - max is +40°C/ 104°F.</li><li>• The motor is build into an environment where it can not dissipate enough heat.</li><li>• The motor is not mounted on a proper mechanical structure where heat can be dissipated.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Make precautions to decrease the surrounding ambient temperature.</li><li>• Lower the speed and or load on the motor.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Reset the motor, clear the error bit(s) in register 35 or cycle the power.</li></ul>
Error bit / Firmware name	Bit 6

### 7.2.10 Error message 'Internal error'

Message no. / Message	7 / 'Internal error'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The firmware consists of 2 parts, and only one part is working.
Possible cause of this error	<ul style="list-style-type: none"><li>• Firmware update process has been interrupted.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Use the recommended USB-RS485 converter with part number RS485-USB-ATC-820. See <a href="#">MISxxxxxxQ5xxxx connector description</a>, page 42</li><li>• Let MacTalk finish the firmware update.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Try firmware updating again and follow the recommendations above.</li></ul>
Error bit / Firmware name	Bit 7

## 7.2 Error messages

### 7.2.11 Error message 'Encoder lost position'

Message no. / Message	8 / 'Encoder lost position'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The absolute multi turn encoder (H3/H4) has lost the position.
Possible cause of this error	<ul style="list-style-type: none"><li>• The reset of the encoder can be caused by a firmware update.</li><li>• The battery level is low and the encoder cannot remember the position.</li></ul>
Solutions to avoid error	Hints to optimise the battery lifetime: <ol style="list-style-type: none"><li>1. Avoid to place the motor in an environment with high temperatures.</li><li>2. Set the running and especially the standby motor current as low as possible in order not to heat up the motor unnecessarily.</li><li>3. Keep the external power applied as much as possible.</li></ol> See also <a href="#">Position retention time</a> , page 132 for further info
How to return to normal operation	<ul style="list-style-type: none"><li>• Reset the position (special command 354 in register 24), clear the error bit(s) in register 35 or cycle the power.</li><li>• If the battery level is low, this error will re-appear every time after power has been off for a while. In that case, the motor must be returned for service.</li><li>• A the cause have been a encoder firmware update from an older version an encoder calibration may probably rescue the motor from this error state. See <a href="#">Procedure to calibrate encoder</a>, page 422 for further info</li></ul>
Error bit / Firmware name	Bit 8

### 7.2.12 Error message 'Encoder Reed error'

Message no. / Message	9 / 'Encoder Reed error'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	<ul style="list-style-type: none"><li>• The absolute multi turn encoder (H3/H4) has detected a wrong sequence in the positioning algorithm.</li><li>• This error also occurs after firmware update.</li></ul>
Possible cause of this error	<ul style="list-style-type: none"><li>• This can be caused by a mechanical shock on the shaft or an external magnetic field.</li><li>• Because the encoder has been reset during a firmware update.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Do not place the motor inside a strong magnetic field.</li><li>• Do not expose the shaft or the motor for mechanical shocks.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Reset the position (special command 354 in register 24), clear the error bit(s) in register 35 or cycle the power.</li></ul>
Error bit / Firmware name	Bit 9

## 7.2

## Error messages

### 7.2.13 Error message 'Encoder COM error'

Message no. / Message	10 / 'Encoder COM error'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The internal communication to the Absolute multi turn encoder (H3/H4) does not work.
Possible cause of this error	<ul style="list-style-type: none"><li>• A firmware update of either the SMC66/85 or the Absolute multi turn encoder (H3/H4) has gone wrong.</li><li>• Hardware error.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• This error should not occur during normal operation, but can happen if something went wrong in a firmware update process.</li><li>• If a new firmware update does not clear the error: Return the motor for service.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• This error will only be set during startup and can be cleared afterwards. Then the motor operation will be the same, but the multi turn encoder will not work.</li><li>• Clear the error bit(s) in register 35.</li></ul>
Error bit / Firmware name	Bit 10

### 7.2.14 Error message 'External encoder'

Message no. / Message	11 / 'External encoder'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	An external SSI encoder has been enabled but communication with the encoder has failed.
Possible cause of this error	<ul style="list-style-type: none"><li>• The encoder is not connected correctly.</li><li>• The format chosen is not compatible with the actual encoder.</li><li>• Improper cabling have been used.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Use proper cabling between the motor and the external SSI encoder. A screened cable with twisted pair wires is recommended.</li><li>• Make sure that the right SSI format is selected.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Reset the motor, clear the error bit(s) in register 35 or cycle the power.</li></ul>
Error bit / Firmware name	Bit 11

## 7.2

## Error messages

### 7.2.15 Error message 'Closed loop'.

Message no. / Message	12 / 'Closed loop'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The encoder and motor is misaligned.
Possible cause of this error	The motor has been updated to a newer firmware where a new calibration is required for closed loop operation.
Solutions to avoid error	The error only occurs when activating the closed loop. If closed loop is not needed, then just deactivate it and save the setting by using "Save in Flash".
How to return to normal operation	<p>The motor must be calibrated. This is done by following procedure:</p> <ol style="list-style-type: none"><li>1. Disconnect any load from the motor shaft.</li><li>2. Write 383 to register 24.</li><li>3. Wait ~60 seconds for the calibration to finish.</li><li>4. When the motor stands still and register 51 (P2) reads 7, then the motor can be reset by writing 267 to register 24 or simply by cycling the power (CVI).</li><li>5. Now the motor should run in closed loop.</li></ol> <p>See also: Calibrating an the H2(H4) single turn encoder option <a href="https://www.jvl.dk/files/pdf-1/instructions/mis_h2_fw_update_guide-2.pdf">https://www.jvl.dk/files/pdf-1/instructions/mis_h2_fw_update_guide-2.pdf</a></p>
Error bit / Firmware name	Bit 12

### 7.2.16 Error message 'External memory'

Message no. / Message	13 / 'External memory'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The controller has build in memory for the "Event log".
Possible cause of this error	The memory self test has failed because of faulty hardware inside the motor.
Solutions to avoid error	Return the motor for service.
How to return to normal operation	<ul style="list-style-type: none"><li>• This error will only be set during startup and can be cleared afterwards. Then the motor operation will be the same, but the "Event log" will not work.</li><li>• Clear the error bit(s) in register 35.</li></ul>
Error bit / Firmware name	Bit 13

## 7.2 Error messages

### 7.2.17 Error message 'Single turn encoder error'

Message no. / Message	14 / 'Singleturn encoder error'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The absolute single turn encoder (H2/H4) has failed.
Possible cause of this error	<ul style="list-style-type: none"><li>• Wrong setup.</li><li>• The distance between the internal magnet and the internal encoder sensor is outside the limits caused by too high force at the motor shaft in forward or backward direction.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Take care when changing settings in Internal_Encoder_Setup (register 175).</li><li>• Do not expose the shaft or the motor for mechanical shocks.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• If the error appeared after changing some settings, please try to "Load factory defaults" in MacTalk.</li><li>• Cycle the power to the motor.</li></ul>
Error bit / Firmware name	Bit 14

### 7.2.18 Error message 'Safe Torque Off'

Message no. / Message	15 / 'Safe Torque Off'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	"Safe Torque Off" has been triggered.
Possible cause of this error	<ul style="list-style-type: none"><li>• One of the two inputs STO channel A or STO channel B has been measured less than 18 V.</li><li>• A hardware fault in the STO circuitry.</li></ul>
Solutions to avoid error	Make sure that both STO channels are connected to 24 V with a stable power supply.
How to return to normal operation	<ul style="list-style-type: none"><li>• Apply a stable 24 V to both STO channels and clear the error bit(s) in register 35.</li><li>• If the Safe Torque Off error is still present, the STO circuit has a hardware fault and the motor must be sent to JVL for inspection.</li></ul>
Error bit / Firmware name	Bit 27

### 7.2.19 Error message 'Zero search timeout'

Message no. / Message	16 / 'Zero search timeout'
Type / Motor action	Error / Motor is set back in previous mode.
Error condition	Zero search have taken longer than allowed.
Possible cause of this error	<ul style="list-style-type: none"><li>• The collision point has not been met within the time specified in the zero search time out register. Default of 60 sec. has been used unless it have been changed by user.</li></ul>
Solutions to avoid error	After this error occur the motor is set back in the mode it was before a zero search was started. In order to do a successfull zero search make sure that the time out interval is set to a proper value or increase the zero search speed.
How to return to normal operation	The motor is already in "normal operation" but the zero search have failed.
Error bit / Firmware name	Bit 16

See also *Zero search modes*, page 163 for further details.

## 7.2 Error messages

### 7.2.20 Error message 'CVI unstable'

Message no. / Message	17 / 'CVI unstable'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	The supply voltage at the CVI terminal have been below what is specified as minimum voltage.
Possible cause of this error	<ul style="list-style-type: none"><li>• Unstable power supply</li><li>• Supply cable have been infected by surges/noise from other cables nearby</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Make sure that the power supply is dimensioned safely.</li><li>• Make sure to use screened cable to avoid influence by noise from external sources or neighbour cables with a significant high level of noise such as supply cables to large motors or ultra-sound welders.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Cycle power</li><li>• Make a software reset</li></ul>
Error bit / Firmware name	Bit 17

### 7.2.21 Error message 'Motor driver overload'

Message no. / Message	18 / 'Motor driver overload'
Type / Motor action	Unrecoverable error / Motor is set in Passive mode.
Error condition	A too high current in the motor windings have been detected.
Possible cause of this error	<ul style="list-style-type: none"><li>• The motor have been forced up to an extreme speed by an external force making it impossible to control the current.</li><li>• Unstable power supply.</li><li>• Internal error.</li></ul>
Solutions to avoid error	<ul style="list-style-type: none"><li>• Make sure to avoid that external forces to exceed 3000 RPM since the MIS motor is defenceless in this situation and can cause permanent damages.</li><li>• Make sure the power supply is properly dimensioned.</li><li>• If none of above - consult your JVL representative.</li></ul>
How to return to normal operation	<ul style="list-style-type: none"><li>• Cycle power</li><li>• Make a software reset</li></ul>
Error bit / Firmware name	Bit 18



## 8.1 Introduction to registers

---

All of the motor registers can be accessed either through the RS485 interface which is the standard interface in the MIS motors.

Optionally the registers can also be accessed through the optional CANopen, or Ethernet interface. A separate manual LB0056 exist for the industrial ethernet protocols.

The Ethernet manual can be found at [www.jvl.dk](http://www.jvl.dk) using this link : [www.jvl.dk](http://www.jvl.dk)

When accessing registers over CANopen, they are mapped to object indexes 2012 and 2014 (hex) with the sub-index equal to the register number 1...255. Use index 2012 for the 32-bit registers and index 2014 for the 16-bit registers.

For example to access all 32 bits of P\_SOLL, use index 2012, subindex 3. To access 16 bits of V\_SOLL use index 2014, subindex 5. This is described in more detail in [CANopen \(optional\)](#), page 277.

All of the registers can be accessed over CANopen with the same Read/Write access restrictions as when using the RS485 interface.

Some registers are tagged as R for Read-only. There are different reasons for this, such as protecting the serial number from being changed or indicating that the value in registers, such as analogue Inputs, will never be read by the motor but always overwritten using the latest sampled values.

## 8.2

## Internal registers

### 8.2.1 Register Overview

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
1	PROG_VERSION	32bit	R	-	-	Major*16 + Minor + 16384 + $17 \cdot 2^{14}$	The firmware version. The Bit 14 is set to indicate that the type is a stepper motor controller, while bits [19:14] are set to the specific motor type, where 17 means SMC85xx.	"Status bar"
2	MODE_REG	32bit	R/W	0, 1, 2, 13, 14	0	-	Controls the operating mode of the motor. 0 : Passive 1 : Velocity mode 2 : Position mode 13 : Zero search type 1 14 : Zero search type 2 32: Cyclic Synchronous Position mode (Ethernet only)	Current Mode
3	P_SOLL	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	The desired position. When in position mode, the motor will move to this position. This value can be changed at any time.	Position
4	Reserved						(intended for 64-bit P_SOLL hi-word)	
5	V_SOLL	32bit	R/W	-300,000-300,000	10000	0.01 RPM	The maximum allowed velocity. When in velocity mode the motor will run constantly at this velocity. Specify a negative velocity to invert the direction. This value can be changed at any time. Example: The value 25000 selects 250 RPM	Max velocity
6	A_SOLL	32bit	R/W	1-500,000	1000	RPM/s	The acceleration/deceleration ramp to use. If this value is changed during movement it will first be active when the motor stops or changes direction.	Acceleration
7	RUN_CURRENT	32bit	R/W	0-1533	511	C: 5.87 mA B: 3.91 mA A: 1.96 mA	Current to use when the motor is running. The unit depends on the driver: C = 9 A, B = 6 A, A = 3 A.	Running Current
8	STANDBY_TIME	32bit	R/W	1-65535	500	ms	Number of milliseconds before changing to standby current.	Standby Time
9	STANDBY_CURRENT	32bit	R/W	0-1533	128	C: 5.87 mA B: 3.91 mA A: 1.96 mA	The standby current. The unit depends on the driver: C = 9 A, B = 6 A, A = 3 A.	Standby Current
10	P_IST	32bit	R/W	$(-2^{31})-(2^{31}-1)$	-	Steps	The actual position. This value can be changed at any time.	Actual position
11	Reserved						(intended for 64-bit P_IST hi-word)	
12	V_IST	32bit	R	-300,000 - 300,000	-	0.01 RPM	The current velocity.	Actual velocity
13	V_START	32bit	R/W	1-300,000	1000	0.01 RPM	The start velocity. The motor will start the acceleration at this velocity.	Start velocity
14	GEAR1	32bit	R/W	$(-2^{31})-(2^{31}-1)$	409600	Counts	The multiplier of the gear factor	Output
15	GEAR2	32bit	R/W	$(-2^{31})-(2^{31}-1)$	2048	Counts	The divider of the gear factor	Input
16	ENCODER_POS	32bit	R/W	$(-2^{31})-(2^{31}-1)$	-	Steps	If the encoder option is installed, this shows the position feedback from the encoder.	Encoder position
17	Reserved						(intended for 64-bit ENCODER_POS hi-word)	
18	INPUTS	32bit	R	-	-	Special	The current status of the digital inputs.	"Status bar"
19	OUTPUTS	32bit	R/W	-	0	Special	The current status of the digital outputs, can be written to change the outputs.	"Status bar"
20	FLWERR	32bit	R	$(-2^{31})-(2^{31}-1)$	-	Steps	When the encoder option is installed this shows encoder deviation from the calculated position (P_IST).	Follow error
21	Reserved						(intended for 64-bit FLWERR hi-word)	

## 8.2

## Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
22	FLWERRMAX	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	The maximum allowed value in FLWERR before an error is triggered. If FLWERRMAX = 0, the error is disabled.	Error handling -> Follow error
23	Reserved						(intended for 64-bit FLWERRMAX hi-word)	
24	COMMAND	32bit	R/W	FastMac commands: 0-127  Other: 256-	0	-	Used to issue commands to the motor. 0-127 is the normal FastMac commands.	Special command
25	STATUSBITS	32bit	R	-	-	Special	Status bits: Bit 0: Reserved Bit 1: AutoCorrection active Bit 2: In Physical Position Bit 3: At velocity Bit 4: In position Bit 5: Accelerating Bit 6: Decelerating Bit 7: Zero search done Bit 8: PassWord lock Bit 9: Magnetic encoder error Bits 10-13: Reserved Bit 14: Electromech. brake active (Int./Ext.) Bit 15: Closed loop lead/lag detected Bit 16: Closed loop activated Bit 17: Internal encoder calibrated (ready for closed loop) Bit 18: Standby current is used Bit 19: STO enabled Bit 20: Internal encoder ok Bit 21: Ethernet sync activated Bit 22: In target position Bit 23: STO channel A ok Bit 24: STO channel B ok Bit 25-31: Reserved	Run Status
26	TEMP	32bit	R		-	-2.27 – uses offset	Temperature measured inside the motor. See the detailed description for information on the value scaling.	Temperature
28	MIN_P_IST	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	Negative software position limit	Position limit min
29	Reserved						(intended for 64-bit MIN_P_IST hi-word)	
30	MAX_P_IST	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	Positive software position limit	Position limit max
31	Reserved						(intended for 64-bit MAX_P_IST hi-word)	
32	ACC_EMERG	32bit	R/W	1-500,000	10,000	RPM/s	Acceleration to use when performing an emergency stop when an error has occurred.	Error acceleration
33	IN_POSITION_WINDOW	32bit	R/W	$0-(2^{32}-1)$	20000	Steps	Selects how close the internal encoder position must be to P_SOLL to set the InPhysical-Position status bit and prevent further AutoCorrection.	In position window
34	IN_POSITION_COUNT	32bit	R/W	0-100	2	Counts	The number of times to attempt AutoCorrection. A value of zero disables AutoCorrection.	Max. number of retries

## 8.2

## Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
35	ERR_BITS	32bit	R/W		0	Special	Error bits: Bit 0: General error (always set together with another error bit) Bit 1: Follow error Bit 2: Output driver Bit 3: Position Limit Bit 4: Low bus voltage Bit 5: Over voltage Bit 6: Temperature >90 °C Bit 7: Internal (Self diagnostics failed) Bit 8: Absolute multiturn encoder lost position Bit 9: Absolute multiturn encoder sensor counting Bit 10: No comm. to absolute multiturn encoder Bit 11: SSI encoder counting Bit 12: Closed loop Bit 13: External memory Bit 14: Absolute single turn encoder Bit 15: H4 Internal Encoder error Bit 16: Zero search timeout Bit 17: CVI unstable Bit 18: Motor driver overload Bit 27: STO_ALARM Bit 29: STO	Errors
36	WARN_BITS	32bit	R/W		0	Special	Warning bits: Bit 0: Positive limit active Bit 1: Negative limit active Bit 2: Positive limit has been active Bit 3: Negative limit has been active Bit 4: Low bus voltage Bit 5: Reserved Bit 6: Temperature >80 °C Bit 7: SSI encoder Bit 8: Driver overload Bit 9: Safe torque off active	Warnings
37	STARTMODE	32bit	R/W	0, 1, 2, 3	0	-	The motor will change to this mode after power up. This is also the mode that is used after a zero search is completed. See MODE_REG for a list of possible modes.	Startup mode
38	P_HOME	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	The found zero point is offset with this value.	Zero search position
39	Reserved						(intended for 64-bit P_HOME hi-word)	
40	V_HOME	32bit	R/W	-300,000-300,000	-5000	0.01 RPM	The velocity to use during zero search. Set a negative velocity to search in the negative direction.	Zero search velocity
41	T_HOME	32bit	R/W	0-2047 (0-100 %)	1024 (50 %)	-	Only used during Torque homing. Defines the torque trigger level where the zero point is set.	Zero search torque
42	HOMEMODE	32bit	R/W	0,13,14	0	-	Select the zero search that should start on power up.	Zero search mode
43-45	Reserved	32bit	R/W	1-8	0		Planned - Not supported yet!	
46	AbsEncPos	32bit	R	0-409,500	0	Steps	The position last read from the internal magnetic encoder. This is the absolute single-turn position.	Abs. encoder position
47	EXTENCODER	32bit	R	$(-2^{31})-(2^{31}-1)$	0	Counts	The value from an external encoder, eg. SSI.	SSI Encoder value
48	FlexReg	32bit	R	-	0	-	A mix of 16 bits from different registers. The user can set this up.	
49-64	Pn	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	8 position registers (odd numbered registers)	Position n (Pn)

## 8.2

## Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
65-72	Vn	32bit	R/W	0-300,000	10000	0.01 RPM	8 Velocity registers	Velocity n (Vn)
73-76	An	32bit	R/W	1-500,000	1000	RPM/s	4 Acceleration registers	Acceleration n (An)
77-80	Tn	32bit	R/W	0-1533	511	5.87 mA	4 Run current registers	Current n (Tn)
81-88	Analog Filtered	32bit	R	0-4095	0	1.221 mV	The voltage on inputs 1 to 8 after being filtered in firmware. See the AFZUP_xxx registers for filter parameters. 5V is equal to a value of 4095.	N/A
89-96	AnalogInput	32bit	R	0-4095	-	1.221 mV	The unfiltered voltage on inputs 1 to 8. 5V is equal to a value of 4095.	N/A
97	BUSVOL	32bit	R	0-4095	-	26.525 mV	Bus voltage	Bus voltage
98	MIN_BUSVOL	32bit	R/W	0-4095	565	26.525 mV	Trigger point for under voltage	Min bus voltage
99	ENCODER_TYPE	32bit	R	0-10	-	-	Internal encoder type 0: No encoder 1: H2 (Single turn encoder 10 bit) 2: H3 (Absolute multi turn encoder 10 bit) 3: H2 (Single turn encoder 12 bit) 4: H4 (Singleturn encoder 12 bit + absolute multi turn encoder.	"Tooltip on motor"
100	AFZUP_WriteBits	32bit	R/W	-	0	Special	Bits 0-7: Bit mask for which of the analog inputs that will use the current value of the ConfMin/Max, MaxSlope and Filter registers. Bit 15: Set when values have been copied and used.	N/A – handled on the Filter Setup screen.
101	AFZUP_ReadIndex	32bit	R/W	0, 1-8, 32768-32775	0	Special	Bits 0-7: Index (1-8) of the analog input whose ConfMin/Max, MaxSlope and filter values to load into the corresponding AFZUO_xxx registers (for read-back). Bit 15 gets set after the registers have been updated.	N/A – handled on the Filter Setup screen.
102	AFZUP_ConfMin	32bit	R/W	0-4094	0	1.221 mV	Minimum confidence limit for analog inputs.	Confidence Min
103	AFZUP_ConfMax	32bit	R/W	1-4095	4095	1.221 mV	Maximum confidence limit for analog inputs.	Confidence Max
104	AFZUP_MaxSlope	32bit	R/W	2-4095	4095	1.221 mV	Maximum slope limit for analog inputs.	Max Slope
105	AFZUP_Filter	32bit	R/W	1-64	64	64 <sup>th</sup> of new sample	Filter value for analog inputs.	Filter (on the Filter Setup screen)
106	FilterStatus	32bit	R	0-65535	0		Individual status bits for 50% of samples outside confidence limits (high 8 bits) and 50% of samples violated the slope limit. (low 8 bits)	N/A (shown graphically)
107	SSI_Setup1	32bit	R/W	-	-	Special	SSI setup bits: Bit 0-4: No. of data bits Bit 5-7: No. of samples Bit 8-15: SSI clk. frequency Bit 16-28: Max. sample deviation Bit 29-31: Read retries	SSI Encoder setup
110	SettlingTime	32bit	R/W	0-32676	0	ms	Number of milliseconds to wait after an AutoCorrection attempt before testing for the position being within the target window.	Settling time between retries

## 8.2

## Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
111	SSI_Setup2	32bit	R/W	-	-	Special	SSI setup bits: Bit 0-7: Prepare time Bit 8: Gray to bin conversion Bit 9: Reserved Bit 10: Disable interrupts Bit 11-18: Wait time	SSI Encoder setup
112-115	SAMPLE1-4	32bit	R/W	-	0	-	Select what register(s) to sample – part of the sample/scope function.	N/A
116	REC_CNT	32bit	R/W	-	0	-	Number of samples to make – part of the scope/sample function.	N/A
117	S_TIME	32bit	R/W	-	1	ms	Sampletime – part of the scope/sample function.	N/A
118	S_CONTROL	32bit	R/W	-	0	-	Controls the scope/sample system.	N/A
120	INDEX_OFFSET	32bit	R	0-409600	-	Steps	The position of the zero sensor relative to the encoder index. This is set after a zero search where the index is used.	Tests tab
121	Modbus_Setup	32bit	R/W	-	0	Special	Modbus setup bits: Bit 0: Enabled Bit 1: Type Bit 2-3: Parity Bit 4: Data bits Bit 5: Stop bits	N/A
122	Zero_Search_BITS	32bit	R/W	-	0	Special	Bits to control Zero Search: Bit 0: Search for index. Bit 1: Change direction on limit. Bit 2: Search for opposite side of sensor. Bit 3: Reserved Bit 4: Ignore switch (Used for searching only for index). Bit 5: Disable the 60 s Zero Search time out.	Advanced → Zero search
124	SETUP_BITS	32bit	R/W	-	0	Special	Bit 0: Invert motor direction. Bit 1: Don't start program after power up. Bit 2-3: External encoder input type Bit 5: Synchronize to encoder after passive Bit 6: In phys. Position update continuously Bit 10: Startup: Transfer single turn position to P_IST Bit 11: Startup: Transfer multi turn position to P_IST Bit 12: Startup: Keep External Encoder Bit 13: Startup: Keep SSI Value Bit 14: CANopen: Beckhoff mode Bit 15: Disable internal encoder Bit 16: External Encoder counting direction Bit 17: Disable position limit error Bit 19: Disable brake (int./ext.) temporarily Bit 20: Disable SSI encoder error Bit 21: Low bus voltage → Error Bit 22: Low bus voltage → Passive Bit 23: Low bus voltage → 0 RPM Bit 24: Enable closed loop Bit 25: Enable closed loop current control Bit 28: Position limits without memory	0: Invert motor direction  1: Don't start program after power up  2-3: 0 = Disabled, 1 = Quadrature, 2 = Puls/direction  17: No error if position limit is detected

## 8.2

## Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
125	IOSETUP	32bit	R/W	-	0	Special	Bit 0-7 sets the I/O active level. Bit 8-15 enables the I/O as an output.	Inputs/Outputs
126	TURNTABLE_MODE	32bit	R/W	0 - 6	0	Mode	Turn table mode	Turn table mode
127	TURNTABLE_SIZE	32bit	R/W	0 - (2 <sup>32</sup> -1)	0	Counts	Turn table size	N/A
129	NL_MASK	32bit	R/W	-	0	IO Mask	Input mask for Negative limit input.	Dedicated inputs - Negative limit input
130	PL_MASK	32bit	R/W	-	0	IO Mask	Input mask for Positive limit input.	Dedicated inputs - Positive limit input
132	HOME_MASK	32bit	R/W	-	0	IO Mask	Input mask for home sensor input(s), each bit set select which I/O 1-8 to use.	Dedicated inputs - Home input
135	INPUT_FILTER_MASK	32bit	R/W	-	0	IO Mask	Input mask for the digital inputs with input filter. Bits set use the input filter time in register 136, bits clear use a fixed update time of 100 us.	IOx digital input filter enabled
136	INPUT_FILTER_CNT	32bit	R/W	-	5	ms	The number of milliseconds the filtered digital inputs must be stable before accepting a change.	Input filter time
137	INPOS_MASK	32bit	R/W	-	0	IO Mask	Output mask for In position output	Dedicated outputs - In position
138	ERROR_MASK	32bit	R/W	-	0	IO Mask	Output mask for error output.	Dedicated outputs - Error
139	ACCEPT_VOLTAGE	32-bit	R/W		2052	8.764 mV	The voltage that must be measured before the current status log is erased.	Acceptance voltage
140	ACCEPT_COUNT	32-bit	R/W		100	Counts	The number of times the ACCEPT_VOLTAGE must be measured before starting the processor	Acceptance count
141	SAVE_VOLTAGE	32-bit	R/W		1710	8.764 mV	The voltage that determines how low the CVI can be before shut down.	Save voltage
143	CVI_VOLT	32-bit	R	-	-	8.764 mV	The measured control voltage	N/A
144	P_NEW	32bit	R/W	(-2 <sup>31</sup> )- (2 <sup>31</sup> -1)	0	Counts	Used with FastMac commands 23 and 24 for changing both the actual and requested position in one operation either absolute or relative.	N/A
145	Reserved						(intended for 64-bit P_NEW hi-word)	
146	BAUD_RATE	32bit	R/W	0-5	1	-	The baud rate on the serial port. 0 : 9600 baud 1 : 19200 baud (default) 2 : 38400 baud 3 : 57600 baud 4 : 115200 baud 5 : 230400 baud 6 : 460800 baud 7 : 921600 baud	Baud rate
147	TX_DELAY	32bit	R/W	1-255	15	Bits	The time to wait before the response is transmitted. The unit corresponds to the time of one bit at the current baud rate.	Transmit delay
148	GROUP_ID	32bit	R/W	0-255		-	The group id of the motor – used for the GroupWrite telegram on the MacTalk protocol.	Group Id
149	GROUP_SEQ	32bit	R	0-255	-	-	The last received group write sequence – part of the MacTalk serial protocol.	N/A
150	MY_ADDR	32bit	R/W	0-254	254	-	The motor address. Used on the MacTalk serial protocol.	Motor address

## 8.2

## Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
151	MOTORTYPE	32bit	R	80-254		-	The motor type. Examples: 80: SMC85, 81: MIS340, 82: MIS341, 83: MIS342 120: MIS17, 150: SMC66, 151: MIS230, 152: MIS231 250: MIL340	"Status bar"
152	SERIAL-NUMBER	32bit	R	-	-	-	The serial number of the motor.	"Status bar"
154	CHECKSUM_1	32bit	R	0-65535	-		Firmware checksum part 1	"Tooltip on motor"
155	CHECKSUM_2	32bit	R	0-65535	-		Firmware checksum part 2	"Tooltip on motor"
156	HARDWARE_REV	32bit	R	0-65535	-	Major*16 + Minor	The revision of the hardware	"Tooltip on motor"
157	MAX_VOLTAGE MAX_CURRENT	32bit	R	0-100 [VDC] 0-9000 [mARMS]	*	Volt	Bit 0-15: Max voltage on bus If the bus voltage exceeds this value, the motor will go in error. Bit 16-31: Full scale motor current in mARMS	"Tooltip on motor"
158	AVAILABLE_IO	32bit	R	-	-	IO Mask and max current from 1-1532.	Bit 0-15: Defines what IO that are available on the connector – programmed during manufacturing. Bit 16-31: The max current to the motor.	N/A
159	BOOTLOADER_VER	32bit	R	0-65535	-	Major*16 + Minor	The version of the boot loader	"Tooltip on motor"
160	NOTSAVED	32bit	R/W	0-65535	0	-	This register is not used internally, but will always be 0 after power-on. Please notice that MacTalk uses this register.	N/A
165	OPTIONS_BITS	32bit	R	0-65535	-	-	This register contains information about what options that are available. Bit 0-7 defines the options available in the hardware (or licensed). Bit 8-15 defines the options available in the firmware. Bit 0,8 : CANopen fieldbus	"Tooltip on motor"
166	FBUS_NODEID	32bit	R/W	1-127	5	Node id	The node id on the CANopen fieldbus interface.	CANopen -> Node id
167	FBUS_BAUD	32bit	R/W	0-8	2	-	The baudrate used on the CANopen fieldbus interface. 0 : 1000 kbit/s 2 : 500 kbit/s 3 : 250 kbit/s 4 : 125 kbit/s 5 : 100 kbit/s 6 : 50 kbit/s 7 : 20 kbit/s 8 : 10 kbit/s	CANopen -> Baud rate
168	ModuleType	32bit	R	0	0	-	Tells which type of module is connected to the internal 1Mbit/s Modbus channel. 0 = No module 0x34 = EthernetIP 0x35 = EtherCAT 0x36 = PowerLink 0x37 = Profinet 0x38 = Modbus/TCP	Dedicated tab
170	EXT_ENCODER	32bit	R/W	$(-2^{31})-(2^{31}-1)$	-	Counts	This register counts the external encoder.	External encoder
171	Reserved						(intended for 64-bit EXT_ENCODER hi-word)	

## 8.2 Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
172	EXT_ENCODER_VEL	32bit	R	$(-2^{31})-(2^{31}-1)$	-	Counts/16ms	This register is updated with the velocity of the external encoder input. The velocity is measured every 16ms.	External encoder Velocity
174	D_SOLL	32bit	R/W	1-500,000	1000	RPM/s	The deceleration ramp to use. If this value is changed during at movement it will first be active when the motor stops or changes direction. If 0, A_SOLL is used for deceleration.	Deceleration
175	Internal_Encoder_Setup	32bit	R/W	-	-	Special	Bit 0-1: Hysteresis (0, 0.17, 0.35, 0.70 deg) Bit 2-4: Resolution (16,15,14,13,12*,11,10*,9) Bit 5: Filter cutoff (16 kHz, 3 kHz) Bit 6: Filter time (0, 1.2 us)  *Closed loop compatible	N/A
176	FW_BUILD	32bit	R	$0-(2^{32}-1)$	-	Counts	Current firmware build number.	"Status bar"
177	InTargetPositionTime	32bit	R/W	$0-(2^{32}-1)$	10	ms	Time the motor must stand still before InTargetPosition flag is set.	N/A
179	BRAKE	32bit	R/W	$0-(2^{32}-1)$	-	Special	Selects which one of the eight I/O pins to use for the external brake.	N/A

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
The following parameters are only available when the CanOpen option is installed and only used for DSP-402								
<b>NOTE: DSP-402 is NOT supported yet!</b>								
180	ControlWord	32bit	R/W	0-65535	0	-	Object 6040 subindex 0	
181	StatusWord	32bit	R	0-65535	0	-	Object 6041 subindex 0	
182	ModeOf-Operation	32bit	R/W	0-255	0	-	Object 6060 subindex 0	
183	ModeOfOperationDisplay	32bit	R	0-255	0	-	Object 6061 subindex 0	
184	Target-Position	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	-	Object 607A subindex 0	
185	Reserved							
186	Actual-Position	32bit	R	$(-2^{31})-(2^{31}-1)$	0	-	Object 6064 subindex 0	
187	Reserved							
188	Target-Velocity	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	-	Object 60FF subindex 0	
189	Reserved							
190	ActualVelocity	32bit	R	$(-2^{31})-(2^{31}-1)$	0	-	Object 606C subindex 0	
191	Reserved							
192	Digital-Outputs	32bit	R/W	0-65535	0	-	Object 60FE subindex 1 (Low 16bit)	
193	Reserved							
194	DigitalInput	32bit	R	0-65535	0	-	Object 60FD subindex 1 (Low 16bit)	
195								

## 8.2

## Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
202	TICKS	32bit	R/W	0-(2 <sup>32</sup> -1)	0	ms	Timer. Increments at a fixed rate of one count per mS. Starts at zero after the motor has been reset	N/A
212	CUR_SCALE_MAX	32bit	R/W	0-2047	2047	Counts	Closed loop: Max current in closed loop with current control. 2047 = 100 % of RUN_CURRENT.	N/A
213	CUR_SCALE_MIN	32bit	R/W	0-2047	1	Counts	Closed loop: Min current in closed loop with current control. 2047 = 100 % of RUN_CURRENT.	N/A
215	CUR_SCALE_FACTOR	32bit	R/W	1-10,000	500	Counts	Closed loop: The slope of the velocity dependent current decrement rate.	N/A
216	KPHASE	32bit	R/W	0-200	-	Counts	Closed loop: A motor dependent factor which optimizes the commutation angle at high speeds.	N/A
217	ACTUAL_TORQUE	32bit	R	0-2047	-	Counts	Closed loop: The actual motor current in closed loop with active current control. 2047 = 100 % of RUN_CURRENT.	Actual torque
218	CUR_SCALE_INC	32bit	R/W	1-100,000	2000	Counts	Closed loop: Current increment rate in closed loop with current control. (1=fastest)	N/A
219	CUR_SCALE_DEC	32bit	R/W	1-100,000	4000	Counts	Closed loop: Current decrement rate in closed loop with current control. (1=fastest)	N/A
222	XFIELD_ADDR	32bit	R/W	-	0	Special	Address for the internal switch board/cross field setup.	N/A
223	XFIELD_DATA	32bit	R/W	-	0	Special	Data for the internal switch board/cross field setup.	N/A
224-231	FlexRegSetup	32bit	R/W		0	-	Each register in this range sets up 2 bits in the FlexRegister 48 = 16 bits in total.	N/A
232	FlexLEDSetup1	32bit	R/W		0	-	Sets up LED L3 and L2 on the motor.	N/A
233	FlexLEDSetup2	32bit	R/W		0	-	Sets up LED L1 GREEN and L1 RED on the motor.	N/A
236	V_SOLL_AUTO	32bit	R/W	-300,000-300,000	0	0.01 RPM	In position mode the auto correction is run with V_SOLL, but if V_SOLL_AUTO != 0 it will be used instead.	Auto correction velocity
237	V_IST_CALC	32bit	R	-300,000-300,000	0	0.01 RPM	The theoretical actual velocity.	Actual velocity
238	MOTOR_REV	32bit	R		0	Rev	Number of motor revolutions the motor has run since last power on.	Event log -> Motor rev
239	EX_CYCLIC_SETUP	32bit	R		0	Special	The actual cyclic setup from the Ethernet module. Bit 0-15: Cycle period (us) Bit 16-31: Sync0 offset in percent.	N/A
241	EX_CRC_ERR	32bit	R		0	Counts	CRC error counter of the internal communication between controller and Ethernet module.	N/A
242	V_HOME_CRAWL	32bit	R/W	0-300,000	0	0.01 RPM	In Zero Search type 2, the "crawl" velocity is V_HOME/64 by default. If register 242 is !=0, a user defined velocity is used.	Zero search crawl velocity
243	V_HOME_TIMEOUT	32bit	R/W		0	ms	If 0, the Zero Search time out is 60000 ms. Else the value in this register is used.	Zero search time out

## 8.2 Internal registers

Reg	Name	Size	Access	Range	Default	Unit	Description	MacTalk name
244	TEMP_LIMITS	32bit	R		0	Special	The actual temperature limits in the motor: Bit 0-15: Warning limit (unit: degC) Bit 16-31: Error limit (unit: °C)	N/A
245	CL_CATCH_UP	32bit	R/W	-	0	Special	Bit 0-7: Allowable overspeed in percent (0-100) Bit 8-31: Follow error limit before overspeed is used.	Allowable overspeed Follow error before overspeed
246	TEMP_HIGHRES	32bit	R		-	°C/ 1000	Like register 26 but the measured temperature is in full resolution and presented in degree celcius x 1000.	Temperature
252	LOWBUSCVI_CNT	32bit	R/W		10	Counts	Number of times in a row the voltage can be too low before error is set. Time between each measurement = 100 us.	N/A
253	V_ENCODER	32bit	R	-300,000- 300,000	-	0.01 RPM	The actual internal encoder velocity.	Internal encoder velocity

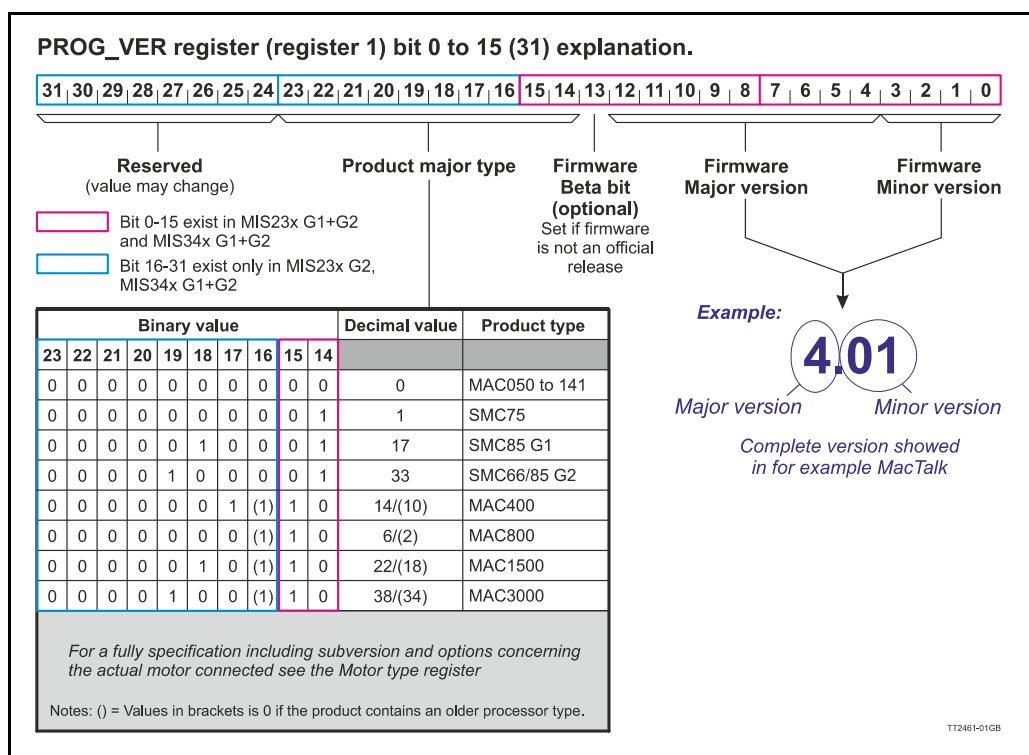
## 8.2 Internal registers

### 8.2.2 Prog\_Vers

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
1	PROG_VERSION	32bit	R	-	*	-	"Status bar"

**Description:** The firmware version. The Bit 14 is set to indicate that the type is SMC75 or SMC85. Bit 0-3 is the minor version and bit 4-12 is the major version. Bit 13 is set if the actual firmware is a beta version (not officially released). Bit 14 to 23 indicate the overall motor type. For specific motor type see also the register [Motor type](#), page 227

Detailed description of the individual bits:



## 8.2 Internal registers

### 8.2.3 Mode\_Reg

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
2	Mode_Reg	32bit	R/W	0,1,2,3,11, 13,14,15	0	-	Current Mode

Description: Controls the operating mode of the motor. The following modes can be selected:

- 0: Passive
- 1: Velocity mode
- 2: Position mode
- 3: Gear mode
- 11: Stop mode
- 13: Zero search type 1
- 14: Zero search type 2
- 15: Safe mode

#### Passive mode (0)

In this mode, the motor current is turned off and the motor will not react to any position/velocity commands.

#### Velocity mode (1)

When the motor is in velocity mode, the controller accelerates the motor to the velocity in V\_SOLL. V\_SOLL can be changed at any time and the move will decelerate/accelerate accordingly.

It is permissible to change A\_SOLL and V\_START during a movement, but the changes will first take effect after the motor has stopped. Please note that if the motor needs to change direction, it will decelerate and stop, and the new A\_SOLL and V\_START will be activated.

#### Position mode (2)

When the motor is in position mode, the controller will always try to move until P\_IST = P\_SOLL.

The movement will follow the profile specified by V\_SOLL, A\_SOLL and V\_START.

P\_SOLL can be changed at any time and the motor will move accordingly.

V\_SOLL can also be changed during a movement.

It is permissible to change A\_SOLL and V\_START during a movement, but the changes will first take effect after the motor has stopped. Please note that if the motor needs to change direction, it will decelerate and stop, and the new A\_SOLL and V\_START will be active.

## 8.2

## Internal registers

### Gear mode (3)

The GEAR mode works as position mode, but has an additional feature. The input on the external encoder is multiplied with GEAR1/GEAR2 and added to P\_SOLL. Any remainder of the result is saved and used next time the external encoder changes.

The result is that this mode can be used as an electronic gear.

When using gear mode, it is not recommended to set V\_START below 10 rpm. This can give problems at low speeds, because the motor will lag behind when doing the first step. It will then accelerate in order to catch up.

**NOTE:** Time from the first input pulse to the first step is typically 30-60µs if not on standby. 72-102µs if on standby.

### Stop mode (11)

When changing from an active mode (Velocity, Position, Gear) to passive mode the motor decelerates with A\_SOLL (or D\_SOLL if not = 0) before it goes passive.

### Zero search type 1 (13)

When the operation mode is set to 13, the controller will start the search for the zero point. See [“Sensor type 1” Zero search](#), page 166 for details.

### Zero search type 2 (14)

When the operation mode is set to 15, the controller will start the search for the zero point. See [“Sensor type 2” Zero search](#), page 166 for details.

### Safe mode (15)

This mode is similar to passive mode, but also allows the “save in flash” and “reset” commands. Safe mode cannot be entered/exited directly; this must be done using the serial commands ENTER/EXIT SAFEMODE.

#### Example:

Writing MODE\_REG=2 will set the motor in position mode. When P\_SOLL is changed, the motor will move to this position with the specified max velocity (V\_SOLL) and acceleration (A\_SOLL).

Writing MODE\_REG=13 will start a zero search for a sensor. When the search is completed, the MODE\_REG will automatically be changed to the mode specified in START\_MODE.

### 8.2.4 P\_SOLL

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
3	P_SOLL	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Counts	Position

**Description:** The desired position. When in position mode, the motor will move to this position. This value can be changed at any time. The maximum possible position difference is  $2^{31}-1$ . If relative movement is used, the P\_SOLL will just wrap at  $2^{31}-1$  and the motor will move correctly.

The MISxxx motor family all have 409600 counts per motor revolution.

**Example:** If P\_SOLL = 0 and then P\_SOLL is set to 409600, the motor moves one revolution forward.

## 8.2 Internal registers

### 8.2.5 V\_SOLL

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
5	V_SOLL	32bit	R/W	±1-300000 (0.01-3000RPM)	10000 (100 RPM)	RPM/100	Max velocity

Description: The maximum velocity allowed. When in velocity mode, the motor will run constantly at this velocity. Specify a negative velocity to invert the direction. This value can be changed at any time.

Example: V\_SOLL = 25000, will limit the velocity to 250 RPM.

### 8.2.6 A\_SOLL

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
6	A_SOLL	32bit	R/W	1-500000	1000	RPM/s	Acceleration

Description: The acceleration/deceleration ramp to use. If this value is changed during at movement, it will first be active when the motor stops or changes direction.

Example: A\_SOLL = 100, will set the acceleration to 100 RPM/s.

### 8.2.7 Run\_Current

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
7	RUN_CURRENT	32bit	R/W	0-1533	511	5.87mA	Running Current

Description: This register sets the running current for the motor. The software is made for controlling motors up to 9 ARMS per motor phase but the maximum allowed current setting is different from motor to motor size.

Motor type	Max. current	Max. Run_Current setting
MIS17x	4 ARMS	4 ARMS / 5.87 mA = <b>681</b>
MIS23x	6 ARMS	6 ARMS / 5.87 mA = <b>1022</b>
MIS34x	9 ARMS	9 ARMS / 5.87 mA = <b>1533</b>
MIS43x	9 ARMS	9 ARMS / 5.87 mA = <b>1533</b>
MIL34x	6 ARMS	6 ARMS / 5.87 mA = <b>1022</b>

The running current is active when the motor is running and after it stops until the specified standby time has elapsed. See [Standby Time](#), page 199.

When a new value is written to the RUN\_CURRENT register, the new motor current will be set instantly.

Example: RUN\_CURRENT = 100, will set the running current to 0.587 ARMS.

## 8.2 Internal registers

### 8.2.8 Standby\_Time

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
8	STANDBY_TIME	32bit	R/W	1-65535	500	ms	Standby Time

**Description:** This register sets the standby time. This time is the time from the last step has been performed until the current changes from running to standby. When a new request for a move is received the current changes from standby to running with no delay.

**Example:** STANDBY\_TIME = 200, will result in the controller switching to the standby current after 200ms.

### 8.2.9 Standby\_Current

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
9	STANDBY_CURRENT	32bit	R/W	0-1533	128	5.87 mA	Standby Current

**Description:** The current range is defined similar to the running current. Please see [Run\\_Current](#), page 198. The standby current is active when the motor has stopped and the specified Standby time has elapsed. See [Standby\\_Time](#), page 199. When the STANDBY\_CURRENT is changed, the new standby current be set instantly.

**Example:** STANDBY\_CURRENT = 50, will set the standby current to 0.285 ARMS.

### 8.2.10 P\_IST

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
10	P_IST	32bit	R/W	$(-2^{31})-(2^{31}-1)$	-	Counts	Actual Position

**Description:** This register shows the actual position of the motor. This is updated each time the motor makes a step. If P\_IST is changed when in position mode or gear mode, the motor will move until  $P\_IST = P\_SOLL$ . When P\_IST reaches  $2^{31}-1$ , it will wrap around to  $-2^{31}$ .

**Example:**  $P\_IST = 1000$ ,  $P\_SOLL = 1000$ . P\_IST is set to 500. The motor will move 500 steps forward and P\_IST will again be 1000.

### 8.2.11 V\_IST

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
12	V_IST	32bit	R	$\pm 1-300000$ (0.01-3000RPM)	-	RPM/100	Actual Velocity

**Description:** This register shows the actual velocity of the motor. The velocity is positive when running in a positive direction and negative when running in a negative direction.

**Example:** If  $V\_SOLL = 40000$  (400 RPM) and a movement of -10000 steps is done, V\_IST will be -40000 (400 RPM) during the move and when the move is complete V\_IST will be 0.

## 8.2 Internal registers

### 8.2.12 V\_START

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
13	V_START	32bit	R/W	$\pm 1-300000$ (0.01-3000RPM)	10000 (100 RPM)	RPM/100	Start Velocity

**Description:** The start velocity. The motor will start the acceleration at this velocity. It will also stop the deceleration at this velocity. If  $|V\_SOLL|$  is lower than  $V\_START$  the motor will not accelerate at all, but start to run at  $V\_SOLL$  instantly. The motor will actually start the movement with an internal  $V\_START = V\_SOLL$ .  
If  $V\_START$  is changed during a movement, it will first be active when the motor stops or changes direction. This also means that if  $V\_SOLL$  is changed to a value below  $V\_START$ , while the motor is in motion, the motor will decelerate to  $V\_START$  and run at that velocity.

**Example:**  $V\_START = 10000$  (100 RPM),  $V\_SOLL = 20000$  (200 RPM),  $MODE\_REG = 1$ . The motor will accelerate from 100 RPM to 200 RPM.  
 $V\_SOLL$  is now changed to 5000 (50RPM). The motor will decelerate to 100 RPM and continue at 100 RPM.  
 $V\_SOLL$  is now changed to -50 RPM. The motor will stop and start at -50 RPM.

### 8.2.13 GEAR1

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
14	GEAR1	32bit	R/W	$(-2^{31})-(2^{31}-1)$	409600	Counts	Output

**Description:** When the gear mode is active, the input from the external encoder is multiplied by GEAR1 and divided by GEAR2.

**Example:**

1.  $GEAR1 = 409600$ ,  $GEAR2 = 2048$ . If 2048 counts are applied to the input, the motor will turn 1 revolution.
2. If one step is applied, the motor will move 200 counts.

### 8.2.14 GEAR2

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
15	GEAR2	32bit	R/W	$(-2^{31})-(2^{31}-1)$	2048	Counts	Input

**Description:** The denominator of the gear factor. See GEAR1 for details.

### 8.2.15 Encoder\_Pos

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
16	ENCODER_POS	32bit	R/W	$(-2^{31})-(2^{31}-1)$	-	Steps	Encoder position

**Description:** If the internal encoder option is installed, this register shows the position feedback from the encoder. This value is initialized to zero at power-up and modified by the firmware when a zero search is performed. The value can be used internally by the AutoCorrection system to retry a movement in position and gear modes.

## 8.2 Internal registers

### 8.2.16 Inputs

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
18	INPUTS	32bit	R	-	-	Special	Inputs

Description: This register shows the status of the digital inputs. Bit 0-7 shows whether IO 1-8 is active or inactive. The active level can be set using IOSETUP. See [IOsetup](#), page 220. Bits 8-15 are not used and will always be 0. The inputs can be filtered or unfiltered. See [Input\\_Filter\\_Mask](#), page 223.

Note that all of the inputs have a digital state and an analogue value at the same time. This register shows their digital state only. Note that the digital inputs can be filtered by setting bits in register 135 ([Input\\_Filter\\_Mask](#), page 223).

Bit	7	6	5	4	3	2	1	0
Input	IO8	IO7	IO6	IO5	IO4	IO3	IO2	IO1

### 8.2.17 Outputs

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
19	OUTPUTS	32bit	R/W	-	0	Special	Outputs

Description: This register shows the status of the outputs. Bit 0-7 shows whether IO 1-8 is active or inactive. The active level can be set using IOSETUP. See [IOsetup](#), page 220. Please note that the output driver for each output also has to be enabled. This is also done using IOSETUP. The register can be changed in order to change the status of the outputs.

### 8.2.18 Flwerr

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
20	FLWERR	32bit	R	$(-2^{31})-(2^{31}-1)$	-	Steps	Follow Error

Description: When the encoder option is installed, this register shows the encoder deviation from the calculated position (P\_IST).

### 8.2.19 Flwerrmax

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
22	FLWERRMAX	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	Follow Error Max

Description: The maximum allowed value in FLWERR before an error is triggered. If FLWERRMAX = 0, the error is disabled. See register 35 ([Err\\_Bits](#), page 208) for a description of the error bit.

## 8.2 Internal registers

### 8.2.20 Command

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
24	COMMAND	32bit	R/W	0-127, 255-1000	0	-	Special command

Description: Used to issue commands to the motor.  
There are 2 kind of commands

- **FastMac commands:**  
A FastMac command typically executes multiple operations by only one command. These commands is therefore very useful if timing is critical. Often FastMac commands are used in a ePLC program.
- **General commands:**  
These commands are a kind of system commands which can be used to activate or read special functions/values inside the motor/controller.

All the FastMac and general commands is defined as a number which is simply written to the COMMAND register.

Number 0-128 are the FastMac commands.  
The values 128-255 are reserved.  
Commands number 256 and higher is the Normal commands.

### 8.2.21 Table of FastMac commands

Number	Description
97 (FastMac 1)	Clear errors and warnings
98 (FastMac 2)	Target position = 0 (P_SOLL = 0)
99 (FastMac 3)	Actual position = 0 (P_IST = 0)
101 (FastMac 5)	Target velocity = 0 (V_SOLL = 0)
103 (FastMac 7)	Clear flags: "In Position", "Acceleration" and "Deceleration"
108 (FastMac 12)	Set P_SOLL = Position register 1 Set V_SOLL = Velocity register 1 Set A_SOLL = Acceleration register 1 Set Running current = Current register 1
109 (FastMac 13)	Set P_SOLL = Position register 2 Set V_SOLL = Velocity register 2 Set A_SOLL = Acceleration register 2 Set Running current = Current register 2
110 (FastMac 14)	Set P_SOLL = Position register 3 Set V_SOLL = Velocity register 3 Set A_SOLL = Acceleration register 3 Set Running current = Current register 3
111 (FastMac 15)	Set P_SOLL = Position register 4 Set V_SOLL = Velocity register 4 Set A_SOLL = Acceleration register 4 Set Running current = Current register 4

## 8.2

## Internal registers

Number	Description
113 (FastMac 17)	Set P_SOLL = P_IST + Position register 7
114 (FastMac 18)	Set P_SOLL = P_IST + Position register 8
116 (FastMac 20)	Clear flag RelativeMove
117 (FastMac 21)	Set flag RelativeMove
118 (FastMac 22)	Set flag RelativeMove
119 (FastMac 23)	P_IST = P_NEW (register 144) P_SOLL = P_NEW (register 144) P_Encoder = P_NEW (register 144)
120 (FastMac 24)	P_IST += P_NEW (register 144) P_SOLL += P_NEW (register 144) P_Encoder += P_NEW (register 144)
121 (FastMac 25)	P_IST += P_NEW (register 144) P_Encoder += P_NEW (register 144)

### 8.2.22 Table of User commands

Number	Description
257	Re-sync P_IST and P_ENCODER position.
267	Reset the CPU.
268	Save to flash memory then reset the CPU.
269	Save to flash memory, then continue normal execution. NOTE: Some registers used only during startup! Take care not to reach an infinite loop if used in RxP, flash memory can be worn out very fast. Use 268 if possible.
316	Preset H3 encoder position (encoder opt. H3) with P_NEW.
320	Set up the RS422 to support SSI encoder.
321	Read SSI encoder.
322	Read SSI encoder and convert from Gray code to binary.
342	Clear all flash sectors in the RXP area.
350 *, **	Linearise/calibrate H3 encoder using Internal Reference, P_IST.
354	Preset encoder opt. H2, H3 and H4, P_IST and P_SOLL with P_NEW, Reset position, Follow error disabled temporarily to avoid errors.
383 *, **	H2 and H4 encoder calibration start. Includes gain, offset, and linearisation.
398	Emergency stop with deceleration.
399	Emergency stop without deceleration.
406	Activate closed loop configuration registers.

\* = See also following links which may be useful:

\*\* = Protected from accidental writes. Must be preceded by 999 command  
Calibrating the H2(H4) single turn encoder option.

[https://www.jvl.dk/files/pdf-1/instructions/mis\\_h2\\_fw\\_update\\_guide-2.pdf](https://www.jvl.dk/files/pdf-1/instructions/mis_h2_fw_update_guide-2.pdf)

Calibrating the H3(H4) absolute multiturn encoder option.

[Encoder calibration, page 422](#)



**Please note:** Several of the commands access the flash memory.

Please notice that the flash memory have restricted number of write cycles (100000 write cycles) and can be permanent damaged if this number is exceeded.

The primary commands that access (write) in the flash memory are:  
Command 268, 269 and 342.

## 8.2 Internal registers

### 8.2.23 Table of User commands (continued)

Number	Description
720 **	Saves K-Phase parameters from register 327-330 in flash
721 **	Load the default k-phases to register 327-330 from flash.
901 **	<p>SSI encoder linearisation</p> <p>Make sure SSI setting is set according to JVL SSI closed loop recommendations. Set the running current to the match the motors rated current. Set special command to 901. P2 will have value 10 during the Linearization part of the calibration, and 20 during the offset detection part of the calibration. When P2 becomes 100 the calibration is done successfully. Power cycle the motor. NOTE: Do not press the “save in motor” button before a power cycle is performed.</p>
902 **	<p>Motor K-Phase detection</p> <p>The KPHASE parameter is different for each motor type. This parameter is used to control the motor, and each motor has different electrical characteristics at different RPM. Set a speed in Max velocity Register 5 before starting a KPHASE detection. It can sometimes be an advance to choose a lower RPM than the maximum RPM the motor will run. Set the running current to the match the motors rated current. Set the speed and direction for the test in Max velocity register (register 5) Set special command to 902. P2 will have value 30 during the KPHASE detection. When P2 becomes 100 the KPHASE detection is done successfully, and the result is stored as the default value. Press save in motor to store the KPHASE value detected for the test. Power cycle the motor. If you are unhappy with the performance of the motor when using closed loop. Try to run the test with another max velocity.</p>

\*\* = Commands protected from accidental writes. Must be preceded by 999 command



**Please note:** Several of the commands access the flash memory. Please notice that the flash memory have restricted number of write cycles (100000 write cycles) and can be permanent damaged if this number is exceeded.  
The command that access (write) in the flash memory is Command 720.

## 8.2

## Internal registers

### 8.2.24 Status bits

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
25	STATUSBITS	32bit	R	-	-	Special	Run Status

Description: This register contains a number of status bits that indicate status of various functions in the motor. The status bit are:

- Bit 0: Reserved
- Bit 1: AutoCorrection Active
- Bit 2: In Physical Position
- Bit 3: At velocity
- Bit 4: In position
- Bit 5: Accelerating
- Bit 6: Decelerating
- Bit 7: Zero search done
- Bit 8: Reserved
- Bit 9: Internal encoder error
- Bit 10: H3 calibration data present
- Bit 11: H3 linearisation table recorded
- Bit 12: General Error (Same as register 35 bit 0).
- Bit 13: H3 calibration data locked
- Bit 14: Electromechanical brake active (Int./Ext.)
- Bit 15: Closed loop lead/lag detected. Bit also activates the LI LED if no Ethernet or CANopen option is present.
- Bit 16: Closed loop activated
- Bit 17: Internal encoder calibrated (ready for closed loop)
- Bit 18: Standby current is being used in stead of Running current
- Bit 19: Safe Torque Off is enabled in the motor.
- Bit 20: Internal encoder OK
- Bit 21: Ethernet Sync is activated. Motor will only change the velocity and position when sync pulse is received.
- Bit 22: In target position if encoder position and P\_SOLL are within the window.
- Bit 23: STO channel A status
- Bit 24: STO channel B status
- Bit 25-26: External memory size: 0 = 0 kbit, 1=4kbit, 2=64kbit

## 8.2 Internal registers

---

### 8.2.25 Temp

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
26	TEMP	32bit	R	0...127	-	-2.27 - uses offset	Temperature

Description: Temperature measured inside the motor electronics.  
The approximate temperature in degrees Celsius is calculated from the value in this register using the formula:  $T_c = 2.27 * \text{Value}$ .

### 8.2.26 MIN\_P\_IST

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
28	MIN_P_IST	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	Position Limit Min

Description: Position limit for movement in the negative direction. The motor can be configured to stop automatically when it reaches this position.  
The MIN\_P\_IST is also used when using the Turn Table Mode to define the lower position limit of the turn table.  
Please also see [Turn Table Mode](#), page 145 for detailed description.

## 8.2 Internal registers

### 8.2.27 MAX\_P\_IST

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
30	MAX_P_IST	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	Position Limit Max

Description: Position limit for movement in the positive direction. The motor can be configured to stop automatically when it reaches this position.  
The MAX\_P\_IST is also used when using the Turn Table Mode to define the upper position limit of the turn table. Please also see [Turn Table Mode](#), page 145 for detailed description.

### 8.2.28 Acc\_Emerg

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
32	ACC_EMERG	32bit	R/W	1-500000	10000	RPM/s	Error Acceleration

Description: The motor will use this acceleration during an emergency stop.

### 8.2.29 IN\_POSITION\_WINDOW

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
33	IN_POSITION_WINDOW	32bit	R/W	$0 - (2^{31}-1)$	20000	Counts	In position window

Description: Selects how close the internal encoder position must be to the target Position (P\_SOLL) to set the InPhysical-Position status bit and prevent further AutoCorrection.

### 8.2.30 IN\_POSITION\_COUNT

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
33	IN_POSITION_COUNT	32bit	R/W	0 - 100	2	Counts	Max. number of retries

Description: The number of times to attempt AutoCorrection. A value of zero disables AutoCorrection.

## 8.2

## Internal registers

### 8.2.31 Err\_Bits

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
35	ERR_BITS	32bit	R/W		0	Special	Errors

Description: This register contains all information about present errors - if any.

#### Error bits:

- Bit 0: General error. Will always be set together with one of the other bits.
- Bit 1: Follow error
- Bit 2: Output driver. Bit is set if one of the user outputs is short circuited.
- Bit 3: Position Limit
- Bit 4: Low bus voltage
- Bit 5: Over voltage
- Bit 6: Temperature too high ( $>90^{\circ}\text{C}$ )
- Bit 7: Internal error (Self diagnostics failed)
- Bit 8: Encoder Lost Position (Absolute Multi-turn Encoder option, H3).
- Bit 9: Encoder Reed Error (Absolute Multi-turn Encoder option, H3).
- Bit 10: Encoder Communication Error (Absolute Multi-turn Encoder option, H3).
- Bit 11: SSI encoder.
- Bit 12: Closed loop.
- Bit 13: External memory.
- Bit 14: Single turn encoder error (H2).
- Bit 15: H4 Internal Encoder error
- Bit 16: Zero search has timed out.
- Bit 17: Control voltage (CVI) has been too low.
- Bit 18: Motor driver overload (only available in SMC85/MIS34x/MIS43x).
- Bit 27: STO\_ALARM.
- Bit 29: STO.

#### Important Bit 27 - Functional safety related !

The STO\_ALARM will only be set if the STO self-diagnostic circuit has detected an internal error. In this case, the motor must be returned the manufacturer (JVL) for repair.

#### In general

If any of these bits are set, the motor is in a state of error, and will not move until all the errors have been cleared.

Some of the errors can be cleared by writing zero to this register.

Other errors will require hardware fixes or intervention, such as allowing the motor cool down or adjusting the power supply voltage.

## 8.2 Internal registers

---

### 8.2.32 Warn\_Bits

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
36	WARN_BITS	32bit	R/W		0	Special	Warnings

Description: Warning bits:

- Bit 0: Positive limit active. This bit will be set as long as the positive limit is active.
- Bit 1: Negative limit active. This bit will be set as long as the negative limit is active.
- Bit 2: Positive limit has been active.
- Bit 3: Negative limit has been active.
- Bit 4: Low bus voltage.
- Bit 5: Reserved.
- Bit 6: Temperature has been above 80°C.
- Bit 7: SSI encoder.
- Bit 8: Driver overload.
- Bit 9: Safe torque off is active.

These bits provide information on both the actual state and remembered state of the end position limits, the supply voltage and the temperature. These are used for diagnostic purposes as well as handling position limit stops, also after the motor may have left the end position mechanically.

## 8.2 Internal registers

### 8.2.33 Start mode

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
37	STARTMODE	32bit	R/W	-	0	-	Startup Mode

Description: The motor will switch to this mode after power up. This is also the mode that is used when a zero search has been completed. See [Mode\\_Reg](#), page 196 for a list of possible modes.

### 8.2.34 P\_Home

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
38	P_HOME	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	Zero Search Position

Description: The zero point found is offset with this value.

### 8.2.35 V\_Home

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
40	V_HOME	32bit	R/W	$\pm 1-300000$ (0.01-3000RPM)	5000 (50 RPM)	RPM/100	Zero Search Velocity

Description: The velocity used during zero search. Set a negative velocity to search in the negative direction.

### 8.2.36 T\_Home

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
41	T_HOME	32bit	R/W	0-2047 (0-100 %)	1024 (50 %)	RPM/100	Zero Search Torque

Description: The torque trigger point when doing a torque zero search.

### 8.2.37 Home mode

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
42	HOMEMODE	32bit	R/W	0,13,14	0	-	Zero Search Mode

Description: Selects the zero search that should start on power up.  
A value of 13 will use sensor type 1, while a value of 14 will use sensor type 2.  
Select 0 (default) if no automatic zero search must be done after power up.

## 8.2 Internal registers

### 8.2.38 Absolute encoder position

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
46	ABSENCODER	32bit	R	H2 (0-409500) H3 $((2^{31})-((2^{31})-1))$ H4 $((2^{31})-((2^{31})-1))$	0	-	Absolute Encoder Position

Description: If one of the encoder options are present in the motor this register monitors the position value. The value are shown in 2 different formats depending on which encoder option that is present.

**H2 encoder option:**

The register contains the absolute single turn position shown in the range 0-409500 counts.

**H3+H4 encoder option:**

The register contains the absolute multi turn position for the whole 32 bit signed range.

### 8.2.39 EXTENCODER2

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
47	EXTENCODER2	32bit	R	$(-2^{31})-(2^{31}-1)$	0	-	SSI Encoder Value

Description: This is the actual encoder position data received from the external SSI encoder. Some SSI encoders output Gray coded values. The firmware offers the possibility to do the Gray code to binary conversion before updating the EXTENCODER2 register with the actual position.

Example: An SSI encoder outputs the position in binary. We want to sample, and update the EXTENCODER2 register 47 without any conversion. This can be done by use command 321.

If the SSI encoder outputs the position in Gray code, the value can be converted to binary before updating the EXTENCODER2 register by using command 322 instead.

For further description of the external encoder interface using SSI format please consult [The SSI interface principle of operation.](#), page 127

### 8.2.40 FlexRegister

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
48	FlexRegister	32bit	R	$(-2^{15})-(2^{15}-1)$	0	-	

Description: A register that can be set up to contain different bits from several registers. 16 bits are available.

## 8.2 Internal registers

### 8.2.41 Pn

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
49,51, 53,55, 57,59, 61,63	Pn	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Steps	Position n (Pn)

Description: These eight general-purpose position registers are referred to as P1... P8 and can be used to make absolute or relative movements in several different ways, either from the user program or via the serial interfaces. See also the sections on FastMac commands, and the P\_NEW register description ([P\\_New](#), page 225).

### 8.2.42 Vn

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
65-72	Vn	32bit	R/W	$\pm 1-300000$ (0.01-3000RPM)	25000 (250 RPM)	RPM/100	Velocity n (Vn)

Description: These eight general-purpose Velocity registers are referred to as V1...V8 and can be used to change the velocity in several different ways, either from the user program or via the serial interfaces. See also the sections on FastMac commands.

### 8.2.43 An

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
73-76	An	32bit	R/W	1-500000	1000	RPM/s	Acceleration n (An)

Description: These four general-purpose Acceleration registers are referred to as A1... A4 and can be used to change the acceleration in several different ways, either from the user program or via the serial interfaces. See also the sections on FastMac commands.

### 8.2.44 Tn

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
77-80	Tn	32bit	R/W	0-511	511	5.87 mA	Current n (Tn)

Description: These four general-purpose Torque registers are referred to as T1...T4 and can be used to change the Running current in several different ways, either from the user program or via the serial interfaces.  
See also the sections on FastMac commands. They select the current in the motor windings used during movement.

## 8.2 Internal registers

### 8.2.45 Analogue Filtered

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>81-88</b>	Analogue Filtered	32bit	R	0-4095	0	1.221mV	N/A

Description: These eight registers hold the software-filtered analogue value of each of the eight I/O's: IO-1 to IO-8. Their values are updated every ten milliseconds. See the AFZUP\_xx registers 100-106 for the filter parameters. Important: Also read the section on analogue filters in this manual.

To use the unfiltered values of the inputs for faster updates, but with no noise immunity, use registers 89-96 instead ([Analogue In](#), page 213).

An input voltage of 5.00 Volts corresponds to a register value of 4095.

See also: [Analogue input filters](#), page 30

### 8.2.46 Analogue In

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>89-96</b>	Analogue Input	32bit	R	0-4095	-	1.221 mV	N/A

Description: These eight registers hold the unfiltered analogue value of each of the eight I/Os: IO-1 to IO-8. Their values are updated approximately every 1 ms.

To use the filtered values of the inputs for better noise immunity, use registers 81-88 instead ([Analogue Filtered](#), page 213).

An input voltage of 5.00 Volts corresponds to a register value of 4095.

See also: [Analogue input filters](#), page 30

### 8.2.47 Busvol

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>97</b>	BUSVOL	32bit	R	0-4095	-	26.67 mV	Bus Voltage

Description: The supply voltage inside the motor is continually measured and stored in this register. This value is the basis for the warnings and errors of Low Bus Voltage and Over Voltage.

### 8.2.48 Min\_Busvol

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>98</b>	MIN_BUSVOL	32bit	R/W	0-4095	15	26.67 mV	Min Bus Voltage

Description: Trigger point for under-voltage

## 8.2 Internal registers

### 8.2.49 Encoder\_Typ

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
99	ENCODER_TYPE	32bit	R	0-10	-	-	"Tooltip on motor"

Description: This register monitor which encoder option that is installed in the motor.

- 0 = No encoder
- 1 = Absolute single turn encoder 10 bit (H2)
- 2 = Absolute multi turn encoder (H3)
- 3 = Absolute single turn encoder 12 bit (H2)
- 4 = Absolute single turn + multi turn encoder (H4)

### 8.2.50 Afzup\_WriteBits

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
100	AFZUP_WriteBits	32bit	R/W	-	0	Special	N/A handled on the Filter Setup screen

Description: When changing values for the analogue input filter parameters, this register is used in combination with registers 102-106. First, all of the registers 102-106 must be loaded with the values to be used for one or more analogue input filters. Then the lower eight bits in this register are set to select which inputs the parameters in registers 102-106 should control.

The firmware will detect this and copy the parameter values from registers 102-106 to internal storage. Once this has been completed, the firmware sets bit 15 in this register to show that registers 102-106 are free to receive new values for programming the remaining inputs with other filter parameters. To use the same filtering for all analogue inputs, this register can be loaded with 255 (hex FF).

### 8.2.51 Afzup\_ReadIndex

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
101	AFZUP_Read Index	32bit	R/W	0, 1-8, 32768-32775	0	Special	N/A handled on the Filter Setup screen

Description: This register makes it possible to read back the analogue input filter parameters for one analogue input at a time. To select a new input, write a value of 1 to 8 to this register and wait for bit 15 to be set high. When bit 15 has been set by the firmware, the registers 102-106 have been loaded with the filter parameters currently used by that analogue input.

### 8.2.52 Afzup\_ConfMin

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
102	AFZUP Conf Min	32bit	R/W	0-4094	0	1.221 mV	Confidence Min

Description: The minimum confidence limits for analogue inputs are set and read back using this register in combination with the read and write 'command' registers 100 and 101.

If a new raw sample value is less than the value in this register, it is simply discarded and the filtered input value in registers 81-88 will not change. A value of zero in this register will effectively disable the minimum confidence check.

## 8.2

## Internal registers

### 8.2.53 Afzup\_ConfMax

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
103	AFZUP_Conf Max	32bit	R/W	1-4095	4095	1.221 mV	Confidence Max

Description: The maximum confidence limits for analogue inputs are set and read back using this register in combination with the read and write 'command' registers 100 and 101. If a new raw sample value is larger than the value in this register, it is simply discarded and the filtered input value in registers 81-88 will not change. A value of 4095 in this register will effectively disable the maximum confidence check.

### 8.2.54 Afzup\_MaxSlope

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
104	AFZUP_Max Slope	32bit	R/W	2-4095	4095	1.221 mV	Max Slope

Description: The maximum slopes per sample for analogue inputs are set and read back using this register in combination with the read and write 'command' registers 100 and 101. If a new raw sample value on an analogue input lies farther from the previous filtered value in registers 81-88, the new sample will be modified to lie at most MaxSlope units from the filtered value. This is used to suppress noise and limit acceleration. Note that the value is optionally filtered after being slope limited, in which case the effective slope limitation will be divided by the filter ratio. A value of 4095 will effectively disable slope limitation.

### 8.2.55 Afzup\_Filter

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
105	AFZUP_Filter	32bit	R/W	1-64	64	64 <sup>th</sup> of new sample	Filter (on the Filter setup screen)

Description: The final filtering of new samples on the analogue inputs can be selected using this register in combination with the read and write 'command' registers 100 and 101. The final filtered value results from taking Filter/64 of the new sample plus (64-Filter)/64 of the old value and storing the result in registers 81-88. A value of 64 effectively disables this filtering, so the new sample simply replaces the old value.

### 8.2.56 FilterStatus

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
106	FilterStatus	32bit	R	0-65535	0		N/A (shown graphically)

Description: This register contains status bits for the analogue input filters. The lowest eight bits hold confidence errors for each of the eight inputs, while the highest eight bits hold the status of their slope errors. The filter status is updated each second. The confidence error bit will be set if more than half of the samples within the last second fell outside either of the confidence limits. The slope errors will be set if more than half of the samples within the last second were slope limited.

## 8.2 Internal registers

### 8.2.57 SSI\_SETUP1

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
107	SSI_Setup1	32bit	R/W	32Bit	25bit, 100kHz frequency pre- pare time = 100µs	*	Number of data bits SSI Clock Frequency Wait time Max. sample deviation Number of samples Read retries

\* Number of data bits. Clock frequency, Disable interrupts when Reading SSI

Description: SSI encoder interface setup bits:

- Bit 0-4: Number of data bits in each SSI transfer
- Bit 5-7: Number of samples for each SSI position reading
- Bit 8-15: SSI clock frequency in units of 10 kHz
- Bit 16-28: Max. sample deviation between each sample
- Bit 29-31: Read retries

See also: [SSI encoder/sensor interface](#), page 123

## 8.2 Internal registers

### 8.2.58 Settling Time

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>110</b>	Settling Time	32bit	R/W	0-32676	0	ms	Settling time between retries

Description: When the internal encoder option is installed and register 34, InPositionCount, is non-zero so AutoCorrection is enabled, the value in this register defines how many milliseconds to wait after each movement attempt before testing whether the encoder position is within the target window as defined in register 33. This waiting time is often necessary to allow mechanical oscillations to die out.

### 8.2.59 SSI\_SETUP2

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>111</b>	SSI_Setup2	32bit	R/W	32 bit	25bit, 100kHz frequency prepare time=100µs	-	Prepare time GRAY conversion

Description: SSI encoder interface setup bits:  
Bit 0-7: Prepare time in milliseconds  
Bit 8: Gray to bin conversion (1=on, 0=off)

### 8.2.60 Sample 1-4

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>112-115</b>	SAMPLE1-4	32bit	R/W	-	0	-	N/A

Description: Up to four registers can be set up to be sampled into buffers for diagnostic purposes. These registers define which registers are sampled. All of the registers 1-255 can be sampled.  
A value of zero in any of these four registers will cause the corresponding sample buffer to contain zeroes.  
See registers 116-119 for more information on the sampling system.

### 8.2.61 Rec\_Cnt

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
<b>116</b>	REC_CNT	32bit	R/W	-	0	-	N/A

Description: This value specifies the number of samples to take for each of the sampled registers selected in registers 112-115. This value must never be set larger than the value in the read-only register 119. Sampling will stop automatically after the specified number of samples has been taken.

## 8.2 Internal registers

### 8.2.62 S\_Time

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
117	S_TIME	32bit	R/W	-	1	-	N/A

Description: This value selects the time in milliseconds between samples of the registers selected in registers 112-115.

### 8.2.63 S\_Control

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
118	S_CONTROL	32bit	R/W	-	0	-	NA

Description: This value controls the sample system. It can assume three different values:  
A value of zero is set by the firmware after all sampling has completed.  
A value of one will initialize the sample system.  
A value of two will start a new sample sequence and set this register to zero at completion.  
The sampled values are read back using the command hex 53 *SMC\_READSAMPLE*.

### 8.2.64 Buf\_Size

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
119	BUF_SIZE	32bit	R	-	-	-	N/A

Description: This read-only register contains the maximum length of the sample buffers used to sample the registers selected in registers 112-115.  
Register 116 should never be set to a value higher than the value in this register.

### 8.2.65 Index\_Offset

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
120	INDEX_OFFSET	32bit	R	0-1599	-	Steps	Tests-

Description: This register can be selected to receive the absolute value of the internal encoder where the Zero search/home position was found during Zero Search. This is selected by bit 0, Use Index, in register 122. It requires that the internal encoder option is installed.

## 8.2 Internal registers

### 8.2.66 Modbus\_setup

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
121	Modbus_setup	32bit	R/W	-	-	-	-

Description: The traditional MacTalk channel can be setup to run Modbus protocol according to these settings:

Bit description:

Bit 0: Enabled

Bit 1: Type (0 = RTU, 1 = ASCII)

Bit 2-3: Parity (0=None, 1=Odd, 2=Even)

Bit 4: Data bits (0=7 bits, 1=8 bits)

Bit 5: Stop bits (0=1 bit, 1=2 bit)

When enabled, the motor can still be connected with the MacTalk protocol the 1.st. second after power on. This way the Modbus settings can be disabled again if necessary.

### 8.2.67 Zero\_Search\_Bits

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
122	Zero_Search_Bits	32bit	R/W	-	0	Special	Advanced-Zero Search

Description: This register contains configuration bits, that define how Zero search should be carried out.

Bit 0: Search for index

Bit 1: Change direction on limit.

Bit 2: Search for opposite side of sensor

Bit 3: (reserved)

Bit 4: Ignore switch (Used for searching only for index)

Bit 5: Disable zero search time out

## 8.2

## Internal registers

### 8.2.68 Setup\_Bits

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
124	SETUP_BITS	32bit	R/W	-	0	Special	Don't start program after power up. Invert motor direction. External Encoder Support Auto encoder synchronize etc. etc. (se below)

Description:

- Bit 0: Invert motor direction
- Bit 1: Do not start RxP program after power up.
- Bit 2-3: Select encoder input type. 0 = Disabled, 1 = Quadrature, 2 = Pulse/direction
- Bit 4: *Reserved*
- Bit 5: Synchronize encoder position to P\_IST after change to active mode.  
Follow error = 0
- Bit 6: InPhysPosMode (If set, recalc InPhysPos continuously. If 0, only after stop)
- Bit 7-9: *Reserved*
- Bit 10: EncoderToP\_IST (Automatically transfer the absolute single turn encoder position to P\_IST at power up)
- Bit 11: Multiturn (Automatically transfer the multi turn encoder position to actual P\_IST at power up)
- Bit 12: KeepExtEncoder (Do not zero the external encoder count on startup)
- Bit 13: KeepSSIValue (Do not zero the SSI data register on startup)
- Bit 14: UseBeckhoff (use the Beckhoff variant of CAN - required by TwinCAT)
- Bit 15: *Reserved*
- Bit 16: External Encoder counting direction (1=inverse)
- Bit 17: Disable position limit error. Motor stays in active mode on position limit.
- Bit 19: Disable brake (int./ext.) temporarily in order to move the shaft in passive mode.
- Bit 20: Disable SSI encoder error. Motor can stay in active mode even if SSI position is wrong.
- Bit 21: "Low bus voltage" gives an Error
- Bit 22: "Low bus voltage" sets the motor in Passive mode.
- Bit 23: "Low bus voltage" sets V\_SOLL to 0 RPM.
- Bit 24: Enable closed loop
- Bit 25: Enable closed loop current control
- Bit 28: Position limits without memory.

These individual bits are used to control various functions in the firmware. Bits marked in grey are not fully available - consult JVL !.

### 8.2.69 IOsetup

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
125	IOSETUP	32bit	R/W	-	0	Special	Inputs/Outputs

Description: This register controls the eight IO's: IO-1 to IO-8. These pins can be used either in input mode as combined digital and analogue inputs or used in output mode as digital outputs. The lowest eight bits in this register can be used to individually invert the active level of the digital inputs. The highest eight bits are used to select the corresponding pin as an output.

## 8.2 Internal registers

### 8.2.70 Turntable\_Mode

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
126	TURNTABLE_MODE	32bit	R/W	0 - 6	0	Special	Turn Table Mode

**Description:** In turntable mode, the motor controls the revolution of a turntable that has the number of positions specified in register 127, Turntable\_Size. This means the same position will be reached after rotating this number of steps in either direction.

This register selects one of three modes that define how the motor should move to a new position when the P\_SOLL register is changed.

If the value of this register is zero, the motor will not operate in turntable mode.

Several modes of operation exist. Please also see [Turn Table Mode](#), page 145 for detailed description.

### 8.2.71 Turntable\_Size

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
127	TURNTABLE_SIZE	32bit	R/W	-	0	Steps	Turn Table - Size

**Description:** If turn table mode is selected in register 126, the number of steps needed for a full revolution of the turn table is set in this register. Note that the register P\_SOLL must always have a value between zero and the value in this register minus one. Negative values are not allowed for P\_SOLL or Turntable\_Size.

Please also see [Turn Table Mode](#), page 145 for detailed description.

### 8.2.72 NL\_Mask

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
129	NL_MASK	32bit	R/W	-	0	IO Mask	Dedicated Inputs Negative Limit Input

**Description:** Selects which one of the eight IO pins to use for the dedicated function of Negative Position Limit. Exactly one bit must be set, and the IO pin must be configured in register 125 as an input.

**Example:** If input 7 is to be used for the Negative Input Limit, write  $2^7 = 128$  to this register.

## 8.2 Internal registers

---

### 8.2.73 PL\_Mask

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
130	PL_MASK	32bit	R/W	-	0	IO Mask	Dedicated Inputs - Positive Limit Input

Description: Selects which one of the eight IO pins to use for the dedicated function of Positive Position Limit.  
Exactly one bit must be set, and the IO pin must be configured in register 125 as an input.

Example: If input 8 is to be used for the Positive Input Limit, write  $2^7 = 128$  to this register.

### 8.2.74 Home\_Mask

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
132	HOME_MASK	32bit	R/W	-	0	IO Mask	Dedicated inputs. Home Input

Description: Selects which one of the eight IO pins to use for the dedicated function of Home Input.  
Exactly one bit must be set, and the IO pin must be configured in register 125 as an input.

Example: If input 2 is to be used for the Home Input, write  $2^1 = 2$  to this register.

## 8.2 Internal registers

### 8.2.75 Input\_Filter\_Mask

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
135	INPUT_FILTER_MASK	32bit	R/W	-	0	IO Mask	IOx digital input filter enabled

**Description:** This register controls filtering of each of the eight IO pins that are used as digital inputs. If the bit corresponding to the input number is set in this register, the input value will be filtered to a new logical level is only accepted after that level has been measured on the hardware pin for the number of milliseconds specified in register 136. If the bit is not set, the input will be updated directly from the hardware value every 100 microseconds. Please read the section on Digital Input filters in this manual.

### 8.2.76 Input\_Filter\_Cnt

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
136	INPUT_FILTER_CNT	32bit	R/W	-	5	ms	Input filter time

**Description:** The filtering of all of the eight digital inputs is controlled by the value in this register together with register 135. The input must be sampled at the same value for the specified number of milliseconds in this register to be accepted as the new filtered value. See also the section on Digital Input Filters in this manual.

### 8.2.77 Inpos\_Mask

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
137	INPOS_MASK	32bit	R/W	-	0	IO MASK	Dedicated Outputs - In Position

**Description:** Selects which one of the eight IO pins to use for the dedicated function of In Position Output.

Exactly one bit must be set, and the IO pin must be configured in register 125 as an output.

The In Position output will then be set after a movement has completed.

**Example:** If output “n” is to be used for the In Position Output, write  $2^{(n-1)}$  to this register.

### 8.2.78 Error\_Mask

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
138	ERROR_MASK	32bit	R/W	-	0	IO Mask	Dedicated Outputs - Error

**Description:** Selects which one of the eight IO pins to use for the dedicated function of Error Output. Exactly one bit must be set, and the IO pin must be configured in register 125 as an output.

The Error Output will set be set when any error is set.

See register 35 ([Err\\_Bits](#), page 208) for more information on errors.

**Example:** If output “n” is to be used for the Error Output, write  $2^{(n-1)}$  to this register.

## 8.2 Internal registers

---

### 8.2.79 Acceptance voltage

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
139	Acceptance Voltage	32bit	R/W	32bit	2052 (18 Volt)	8.764 mV	Acceptance Voltage

Description: The acceptance Voltage, is the voltage required at the CVI supply terminal (PWR connector) for the program to start up.  
The typical and recommended nominal voltage is 24VDC but if a lower voltage is used in for example battery powered applications please make sure that the acceptance voltage also covers that the battery is much lower than if its fully charged.

### 8.2.80 Acceptance count

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
140	Acceptance Count	32bit	R/W	32bit	100	-	Acceptance Count

Description: Acceptance Count is the number of times a voltage above the acceptance voltage must have been measured before the program starts.  
The basic idea behind this register/function is to make sure that the start-up is completed and the supply voltage is stable.

## 8.2 Internal registers

### 8.2.81 Save voltage

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
141	Save voltage	32bit	R/W	32bit	1710 (15 Volt)	8.764 mV	Save Voltage

Description: This register sets the voltage level where the program shuts down and all motor activity stops.

### 8.2.82 CVI\_VOLT

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
143	CVI_VOLT	32bit	R			8.764 mV	N/A

Description: The measured control voltage.

### 8.2.83 P\_New

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
144	P_NEW	32bit	R/W	$(-2^{31})-(2^{31}-1)$	0	Counts	N/A

Description: This register can be used to change both of the registers P\_SOLL and P\_IST in one operation. This can be used to correct or offset the current position without performing a movement. The register value can be copied to P\_IST and P\_SOLL using FastMac command 23, or it can be added with sign to both of these registers using FastMac command 24.

### 8.2.84 Baud\_Rate

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
146	BAUD_RATE	32bit	R/W	0-7	1	-	Baud Rate

Description: The baud rate on the serial port.

- 0: 9600 baud
- 1: 19200 baud (default)
- 2: 38400 baud
- 3: 57600 baud
- 4: 115200 baud
- 5: 230400 baud
- 6: 460800 baud
- 7: 921600 baud

The firmware will automatically update the baud rate after this value is changed over the serial interface (RS485) once the motor has finished transmitting all data bytes that are queued.

## 8.2 Internal registers

### 8.2.85 Tx\_Delay

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
147	TX_DELAY	32bit	R/W	1-255	15	Bits	Transmit Delay

Description: The time to wait before the response is transmitted. The unit corresponds to the time of one bit at the current baud rate.

Many PLCs and communications processors require a minimum delay after they have sent a command to the motor before they are able to receive the response.

### 8.2.86 Group\_Id

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
148	GROUP_ID	32bit	R/W	0-255	-	-	Group Id

Description: The group ID of the motor. The motor will accept data from a group write command only if the group ID number in the command matches this number. The idea is that several motors can have the same group ID so they can be updated with new register values in parallel to save transmission time.

### 8.2.87 Group\_Seq

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
149	GROUP_SEQ	32bit	R	0-255	-	-	N/A

Description: The last received group write sequence.

### 8.2.88 My\_Addr

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
150	MY_ADDR	32bit	R/W	0-254	254	-	Motor Address

Description: The motor address. Data communicated over the serial interface will only be accepted if the address byte in the command is either equal to this value or has the value 255, which means broadcast to all motors.

## 8.2

## Internal registers

### 8.2.89 Motor type

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
151	MOTORTYPE	32bit	R	64-xx		-	"Status Bar"

Description: The actual motor type or controller type. Please notice that this register is in common for all motor families and sizes from JVL including the MAC servomotor range. The list show all available types within stepper motors including versions with build in brake. The motor type register will contain the following values dependent at the actual motor type.

Motor type	Technology	Decimal value	Hex value
SMC85	Stepper motor controller	80	
MIS340	Stepper motor rotary	81	
MIS341	Stepper motor rotary	82	
MIS342	Stepper motor rotary	83	
MIS343	Stepper motor rotary	84	
MIS344	Stepper motor rotary	85	
MIS345	Stepper motor rotary	86	
MIS430	Stepper motor rotary	90	
MIS431	Stepper motor rotary	91	
MIS432	Stepper motor rotary	92	
MIS433	Stepper motor rotary	93	
MIS434	Stepper motor rotary	94	
MIS435	Stepper motor rotary	95	
MIS511	Stepper motor rotary	100	
MIS512	Stepper motor rotary	101	
MIS513	Stepper motor rotary	102	
MIS514	Stepper motor rotary	103	
MIS515	Stepper motor rotary	104	
MIS170	Stepper motor rotary	120	
MIS171	Stepper motor rotary	121	
MIS172	Stepper motor rotary	122	
MIS173	Stepper motor rotary	123	
MIS174	Stepper motor rotary	124	
MIS175	Stepper motor rotary	125	
MIS176	Stepper motor rotary	126	
SMC66	Stepper motor controller	150	
MIS230x	Stepper motor rotary	151	
MIS231x	Stepper motor rotary	152	
MIS232x	Stepper motor rotary	153	
MIS233x	Stepper motor rotary	154	
MIS234x	Stepper motor rotary	155	
MIL230x	Stepper motor linear	200	
MIL231x	Stepper motor linear	201	
MIL232x	Stepper motor linear	202	
MIL233x	Stepper motor linear	203	
MIL234x	Stepper motor linear	204	
MIL340x	Stepper motor linear	250	
MIL341x	Stepper motor linear	251	
MIL342x	Stepper motor linear	252	
MIL343x	Stepper motor linear	253	
MIL344x	Stepper motor linear	254	

This value is read-only and is programmed into the motor during manufacturing.

## 8.2 Internal registers

### 8.2.90 Serial\_Number

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
152	SERIAL-NUMBER	32bit	R	-	-	-	"Status Bar"

Description: The serial number of the motor.  
This value is read-only and is programmed into the motor during manufacturing.

### 8.2.91 Checksum

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
154-155	CHECKSUM	32bit	R	0-65535	-		"Tooltip on motor"

Description: Firmware checksum.  
This value is read-only and is programmed into the motor during firmware update.

### 8.2.92 Hardware\_Rev

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
156	HARDWARE_REV	32bit	R	0-65535	-	Major*16+ Minor +16384	"Tooltip on Motor"

Description: The revision of the hardware. This value is read-only and is programmed into the motor during manufacturing.

### 8.2.93 Max\_Voltage

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
157	MAX_VOLTAGE	32bit	R	0-100	*	Volt	"Tooltip on Motor"

Description: The maximum allowed voltage on the bus. If the bus voltage exceeds this value, the motor will enter an error state.  
This value is read-only and is programmed into the motor during manufacturing. It reflects the rating of the hardware components. Supplying a higher voltage can damage the electronics components permanently. If in doubt, it is strongly recommended to first supply 24 Volts and connect the motor to MacTalk. In MacTalk this value can be read by holding the mouse cursor over the image of the motor in the lower right of the main window.

Bit 0-15: Max voltage on bus  
Bit 16-31: Full scale motor current in mARMS

## 8.2 Internal registers

### 8.2.94 Available\_IO

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
158	AVAILABLE_IO	32bit	R	-	-	IO MASK	N/A

Description: Defines what IO that are available on the connector.  
This value is read-only and is programmed into the motor during manufacturing. Service personnel may ask for this value to identify the type of connector board mounted on the motor. The values are not documented here.

Bit 0-15: Defines what IO that are available on the connector

Bit 16-31: The max current to the motor in the same units as Running current register 7

### 8.2.95 Bootloader\_Ver

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
159	BOOTLOADER_VER	32bit	R	0-65535	-	Major*16+ Minor +16384	"Tooltip on Motor"

Description: The version of the boot-loader.  
This value is read-only and is programmed into the motor during manufacturing

### 8.2.96 Not saved

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
160	NOTSAVED	32bit	R/W	0-65535	0	-	N/A

Description: This register is not used internally, but will always be 0 after power on. Please note that MacTalk uses this register

### 8.2.97 Option\_Bits

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
165	OPTION_BITS	32bit	R	0-65535	-	-	"Tooltip on motor"

Description: This register contains information about what options are available. Bit 0-7 defines the options available in the hardware (or licensed). Bit 8-15 defines the options available in the firmware.

Bit 0,8: CANopen fieldbus

Bit 1,9: DeviceNet fieldbus

## 8.2 Internal registers

### 8.2.98 Fbus\_Node\_Id

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
166	FBUS_NODE_ID	32bit	R/W	1-127	5	-	CANopen Node ID

Description: The node id on the CANopen interface.

### 8.2.99 Fbus\_Baud

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
167	FBUS_BAUD	32bit	R/W	0-8	2	-	Fieldbus - Baud Rate

Description: The baudrate used on the CANopen interface (optional).

0: 1000 kbit/s  
1: 800 kbit/s (unsupported)  
2: 500 kbit/s  
3: 250 kbit/s  
4: 125 kbit/s  
5: 100 kbit/s  
6: 50 kbit/s  
7: 20 kbit/s  
8: 10 kbit/s

### 8.2.100 Module Type

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
168	MODULE_TYPE	32bit	R				(dedicated tab in MacTalk when module is present)

Description: Tells which type of module is connected to the internal 1Mbit/s Modbus channel.

0= No module  
0x34 = EthernetIP  
0x35 = EtherCAT  
0x36 = PowerLink  
0x37 = Profinet  
0x38 = Modbus/TCP

### 8.2.101 Ext\_Encoder

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
170	EXT_ENCODER	32bit	R/W	$(-2^{31})-(2^{31}-1)$	-	Counts	External Encoder

Description: This register counts the external encoder input at the multifunction I/O.

## 8.2 Internal registers

---

### 8.2.102 Ext\_Encoder\_Vel

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
172	EXT_ENCODER_VEL	32bit	R	$(-2^{31})-(2^{31}-1)$	-	Counts 16ms	External Encoder Velocity

Description: This register is updated with the velocity of the external encoder input. The velocity is measured every 16ms.

## 8.2 Internal registers

### 8.2.103 Internal\_Encoder\_Setup

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
175	Internal_Encoder_Setup	32bit	R/W	-	-	Special	N/A

The internal encoder has different settings available:

#### Hysteresis

Is used to prevent flickering of the angular position LSBs. Bit 0-1 set the hysteresis.

#### Resolution

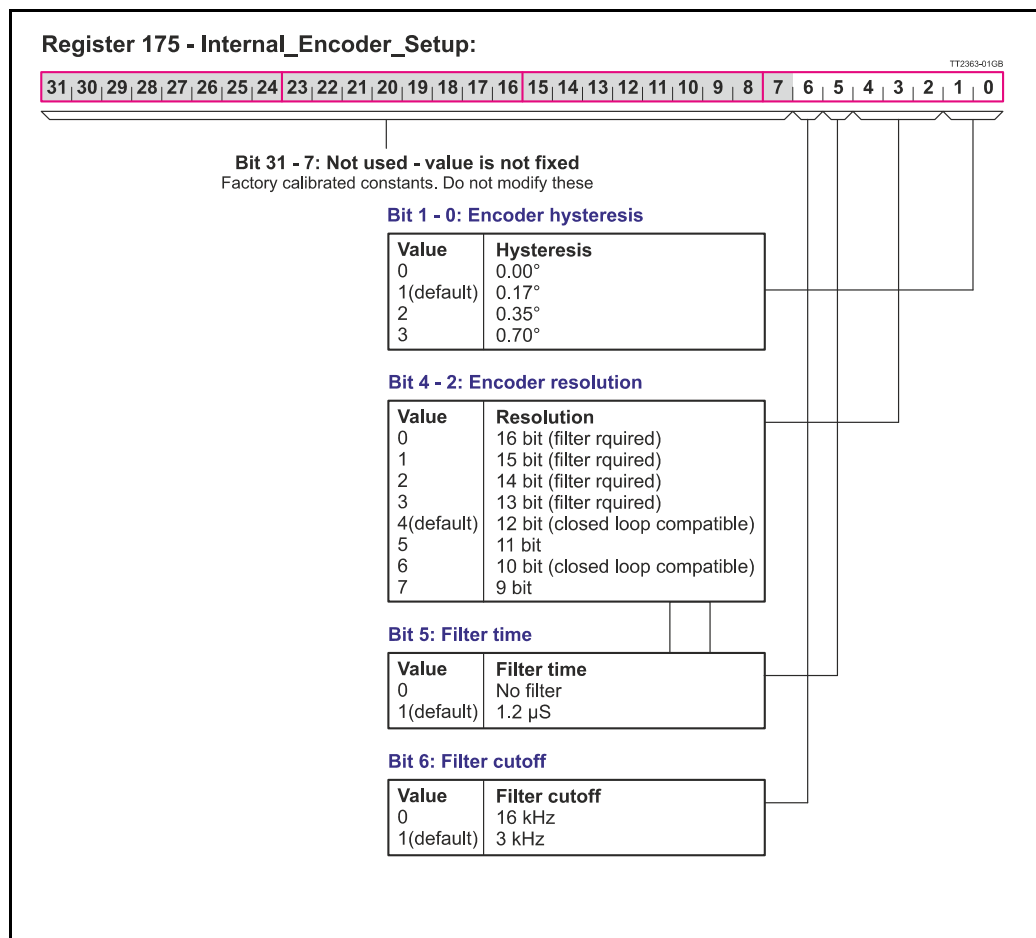
Bit 2-4 determines the resolution, i.e. number of counts in 1 revolution. If the motor is set up to output the encoder pulses, this will also be affected by changing the resolution.

#### Filter

The filter can be enabled to allow resolutions above 12 bits. Bit 5 set the filter.

#### Filter cutoff frequency.

A filter cutoff frequency on 3 kHz is recommended in the entire velocity range from 0 to 3000 RPM. The cutoff frequency is controlled by bit 6.



## 8.2 Internal registers

### 8.2.104 FW\_BUILD

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
176	FW_BUILD	32bit	R	0 - ( $2^{32}-1$ )	-	Counts	"Status bar"

Description: The actual firmware build number. This number is unique for each beta and released version.

### 8.2.105 InTargetPositionTime

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
177	InTargetPositionTime	32bit	R/W	0 - ( $2^{32}-1$ )	10	ms	N/A

Description: Time the motor must stand still before InTargetPosition (reg 25: Statusbits) flag is set.

### 8.2.106 BRAKE

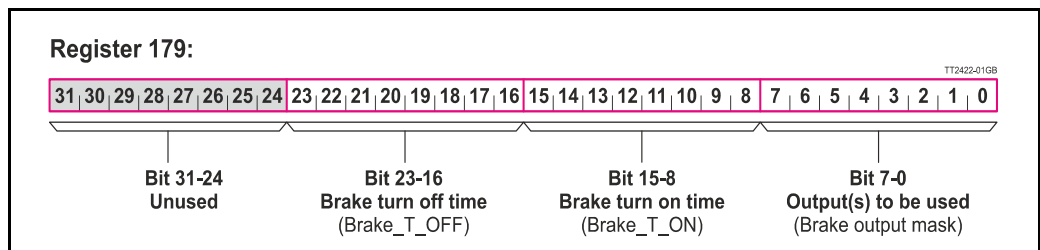
Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
179	BRAKE	32bit	R/W	0 - ( $2^{32}-1$ )	0	Special	N/A

Description: This register selects which one of the eight IO (IO1 to IO8) pins to use for the external brake.

Each of the first 8 bits in this register corresponds to 1 output pin. The selected IO pin must be configured in register 125 as an output.

See also [User outputs](#), page 35.

The bits have following function:



#### Example:

Output 4 is used for the Brake Output. T\_ON time is 40 ms and T\_OFF is 50 ms:

The following string will define this (shown in groups of 8 bits)

00000000 / 00110010 / 00101000 / 00001000 = in hex : 00 32 28 08

Bit 0-7: Brake output mask = 8, define that output 4 controls the brake.

Bit 8-15: Brake\_T\_ON - Time from motor is stopped until brake is activated = 40 ms

Bit 16-23: Brake\_T\_OFF - Time from the motor is activated until the brake is released = 50 ms.

## 8.2 Internal registers

---

### 8.2.107 TICKS

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
202	TICKS	32bit	R/W	0 - ( $2^{32}-1$ )	0	ms	N/A

Description: Increments at a fixed rate of 1 count per millisecond. Starts at zero when CVI has been applied.

### 8.2.108 CUR\_SCALE\_MAX

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
212	CUR_SCALE_MAX	32bit	R/W	0 - 2047	2047	Counts	N/A

Description: Closed loop: Max running current in closed loop with "Current control" enabled. 2047 = 100 % of RUN\_CURRENT.

### 8.2.109 CUR\_SCALE\_MIN

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
213	CUR_SCALE_MIN	32bit	R	0 - 2047	1	Counts	N/A

Description: Closed loop: Minimum running current in closed loop with "Current control" enabled. 2047 = 100 % of RUN\_CURRENT. See also [Special settings](#), page 112.

## 8.2 Internal registers

### 8.2.110 CUR\_SCALE\_FACTOR

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
215	CUR_SCALE_FACTOR	32bit	R/W	1 - 10,000	500	Counts	N/A

Description: Closed loop: The slope of the velocity dependent current decrement rate. See also [Special settings](#), page 112.

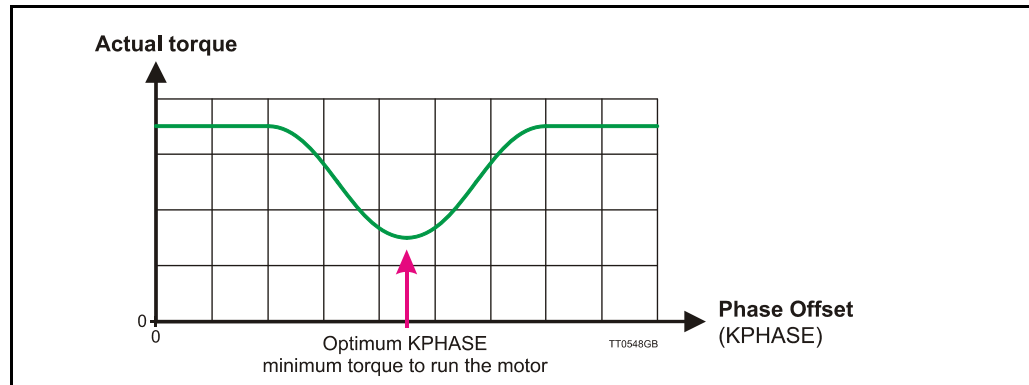
### 8.2.111 KPHASE

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
216	KPHASE	32bit	R/W	0-200	-	Counts	N/A

The KPHASE parameter is decisive for how far commutation of the motor is offset from the motor's actual position. KPHASE is velocity dependent, which means that it has increasing significance as motor velocity increases. The KPHASE parameter is factory calibrated, but can be adjusted by the user if necessary.

#### Finding the optimal KPHASE

The optimal KPHASE value is found by running the motor at high speed (2000 RPM) and observing the "Actual torque" in MacTalk. The actual torque will settle at its minimum value at the optimal KPHASE. The new KPHASE can be saved in flash and will then be used automatically after a reset.



## 8.2 Internal registers

### 8.2.112 ACTUAL\_TORQUE

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
217	ACTUAL_TORQUE	32bit	R	0-2047	-	Counts	Actual torque

Only used when the closed loop operation is active.

The register show the actual motor torque as a value from 0 to 2047 corresponding to 0-100% of the setting done in the motor "Running current" register.

- see also [Run\\_Current](#), page 198.

In passive mode = 0 %

In active mode without current control enabled = 100 %

In active mode with current control enabled = 0-100 % dependent on the load.

### 8.2.113 CUR\_SCALE\_INC

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
218	CUR_SCALE_INC	32bit	R/W	0-100000	2000	Counts	N/A

Used in closed loop operation.

The increment rate of the current determines how fast the actual torque must be increased when a rotor displacement has been measured. The rate is independent of the actual velocity. See also: [Special settings](#), page 112

### 8.2.114 CUR\_SCALE\_DEC

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
219	CUR_SCALE_DEC	32bit	R/W	0-100000	4000	Counts	N/A

Used in closed loop operation.

The increment rate of the current determines how fast the actual torque can be decreased. The rate is inverse proportional to the actual velocity, which means the it will decrease slower at higher velocities.

The slope is determined by register 215 see: [CUR\\_SCALE\\_DEC](#), page 236.

See also: [Special settings](#), page 112

### 8.2.115 XFIELD\_ADDR

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
222	XFIELD_ADDR	32bit	R/W	-	0	Special	N/A

The internal XFIELD allows the user to configure many different combinations of inputs and outputs from the motor. The RS422 interface can be used for connections like:

- External SSI encoder
- Quadrature/pulse-direction encoder input
- Internal encoder output
- Quadrature/pulse-direction step generation output

This register controls the internal addressing for this setup. It is strongly recommended to use MacTalk as interface for the setup.

## 8.2

## Internal registers

### 8.2.116 XFIELD\_DATA

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
223	XFIELD_DATA	32bit	R/W	-	0	Special	N/A

The internal XFIELD allows the user to configure many different combinations of inputs and outputs from the motor. The RS422 interface can be used for connections like:

External SSI encoder

Quadrature/pulse-direction encoder input

Internal encoder output

Quadrature/pulse-direction step generation output

This register controls the internal data for this setup. It is strongly recommended to use MacTalk as interface for the setup.

### 8.2.117 FlexRegSetup

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
224-231	FlexRegSetup	32bit	R/W	-	0	-	N/A

A set of registers that determine the address for each (0-15) bit in register 48: FlexRegister.

Each register in this range sets up 2 bits in the FlexRegister 48 = 16 bits in total.

### 8.2.118 FlexLEDSetup1

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
232	FlexLEDSetup1	32bit	R/W	-	102303769	-	N/A

Sets up LED "L2" and "L3" on the motor.

If the motor has **no** Ethernet module or CANopen interface, it will be in the default configuration and the 2 LEDs "L2" and "L3" can be configured to show various conditions. In default configuration the L2 (green) and L3 (green), can be configured to display the status of a single bit in any register.

The default settings show:

L2 = "At velocity" bit from the Status register (25).

L3 = "In position" bit from the Status register (25).

**Setup:**

Bit 0-8: Register for L3

Bit 9-13: Bit for L3

Bit 16-24: Register for L2

Bit 25-29: Bit for L2

## 8.2 Internal registers

### 8.2.119 FlexLEDSetup2

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
233	FlexLEDSetup2	32bit	R/W	-	504954880	-	N/A

Sets up LED “LI” GREEN and “LI” RED on the motor.

If the motor has **no** Ethernet module or CANopen interface, it will be in the default configuration and the LED “LI” can be configured to display the status of a single bit in any register.

The default settings show:

LI (green) = OFF

LI (red) = “Closed loop lead/lag detected” bit from the Status register (25).

**Setup:**

Bit 0-8: Register for LI (green)

Bit 9-13: Bit for LI (green)

Bit 16-24: Register for LI (red)

Bit 25-29: Bit for LI (red)

### 8.2.120 V\_SOLL\_AUTO

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
236	V_SOLL_AUTO	32bit	R/W	-3,000.00 - 3,000.00	0	0.01 RPM	Auto correction velocity

In position mode the auto correction is run with V\_SOLL, but if V\_SOLL\_AUTO != 0 it will be used instead.

### 8.2.121 V\_IST\_CALC

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
237	V_IST_CALC	32bit	R/W	-3,000.00 - 3,000.00	0	0.01 RPM	Actual velocity

The theoretical actual velocity.

### 8.2.122 MOTOR\_REV

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
238	MOTOR_REV	32bit	R/W	0-2 <sup>32</sup> -1	0	Revolutions	Event log -> Motor revolutions

Counts multiples of 409600 counts since power on. The value is added to the motor revolution counter in the Event log in order to keep the total amount of revolutions the motor has run in its entire lifetime.

## 8.2 Internal registers

### 8.2.123 EX\_CYCLIC\_SETUP

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
239	EX_CYCLIC_SETUP	32bit	R/W	-	0	Special	N/A

The actual cyclic setup from the Ethernet module.

Settings from the Ethernet module to setup the cycle period and how many percent the sync-pulse must be offset. These settings can be changed with the specific Ethernet protocol, but not directly in this register. This is read only.

Bit 0-15: Cycle period ( $\mu$ s)

Bit 16-31: Sync0 offset in percent.

### 8.2.124 EX\_CRC\_ERR

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
241	EX_CRC_ERR	32bit	R/W	0-2 <sup>32</sup> -1	0	Counts	N/A

CRC error counter of the internal communication between controller and Ethernet module.

### 8.2.125 V\_HOME\_CRAWL

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
242	V_HOME_CRAWL	32bit	R/W	0-300000 (0-3000 RPM)	0	0.01 RPM	Zero search crawl velocity

In Zero Search type 2, the “crawl” velocity is V\_HOME/64 by default.

If register 242:V\_HOME\_CRAWL is !=0, a user defined velocity is used – independent of V\_HOME. Please note that overshoot can occur if this velocity is set too high.

### 8.2.126 V\_HOME\_TIMEOUT

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
243	V_HOME_TIMEOUT	32bit	R/W	0-2 <sup>32</sup> -1	0	ms	Zero search time-out

In all Zero Search modes, the time out is by default 60 s. This delay can be changed by writing a value different from 0 to this register. The unit is milliseconds.

If 0, the Zero Search time out is 60000 ms. Else the value in this register is used.

## 8.2 Internal registers

### 8.2.127 TEMP\_LIMITS

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
244	TEMP_LIMITS	32bit	R	-	0	Special	N/A

The actual temperature limits in the motor.

A Warning will be set when the temperature exceeds this value:  
Bit 0-15: Warning limit (unit: °C)

An Error bit will be set when the temperature exceeds this value:  
Bit 16-31: Error limit (unit: °C)

The limits cannot be changed by the user. They are factory settings.

### 8.2.128 CL\_CATCH\_UP

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
245	CL_CATCH_UP	32bit	R/W	-	0	Special	Allowable over-speed. Follow error before overspeed

Bit 0-7: Allowable overspeed in percent (0-100)  
The motor can exceed the Max velocity (V\_SOLL) set in register 5 if this register is non zero. It means that the motor can catch up the follow error.  
Default 0 % = No over speed allowed.

Bit 8-31: Follow error limit before overspeed is used.  
The motor will exceed the Max velocity (V\_SOLL) if the Follow error (register 20) is higher than specified in this register. Default 5000 counts.

### 8.2.129 TEMP\_HIGHRES

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
246	TEMP_HIGHRES	32bit	R	-	-	°C/1000	Temperature

Description: Temperature measured inside the motor electronics.  
The measured temperature is presented in °C/1000 with higher resolution than register 26.

## 8.2 Internal registers

---

### 8.2.130 LOWBUSCVI\_CNT

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
252	LOWBUSCVI_CNT	32bit	R/W	-	10	Counts	N/A

Number of times in a row the voltage can be too low before error is set. Time between each measurement = 50 us.

For motors with serial numbers < 173000 this also sets the time before start saving of the internal Event log. From serial number > 173000 the Event log is saved continuously, every 1 seconds.

### 8.2.131 V\_ENCODER

Reg	Name	Size	Access	Range	Default	Unit	MacTalk name
253	V_ENCODER	32bit	R	-3,000.00 - 3,000.00	-	0.01 RPM	Internal encoder velocity

The actual velocity measured from the internal (H2/H4) encoder.



## **9 Building Sequential Programs**

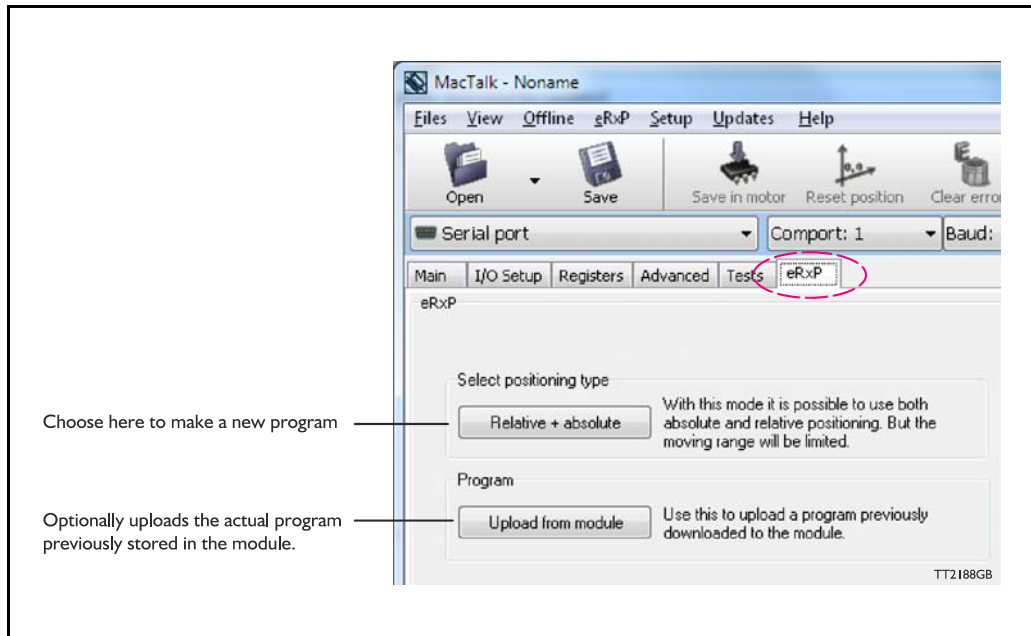
---

## 9.1 Getting started with programming

When using the MIS motors, almost any kind of program can be created using a set of user friendly icons.

Make the required choice on the eRxP Programming tab.

The name eRxP refers to the programmable module (R-module) from the MAC motor series. [e]mbedded [R]-module number [x] [P]rogramming



After making one of these 2 choices, the program window will be opened.



**Please note:** When a program is made and stored the motor will always startup in position mode. If this is not convenient insert a Mode = "passive" on first program line.

## 9.2

# Programming Main window

The main window for creating a new program or editing a program is shown below:

**RxP menu**  
Main menu for creating a new program, Verifying program size and other basic details for the motor.

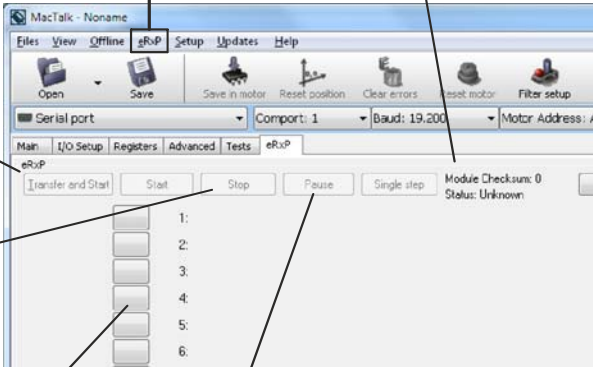
**Transfer & Start**  
Will transfer the complete program and start it. Use Stop or Pause to stop it.

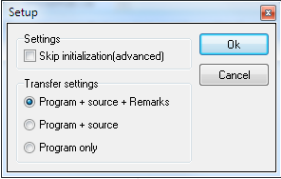
**Stop**  
Use this button if the program must be stopped.

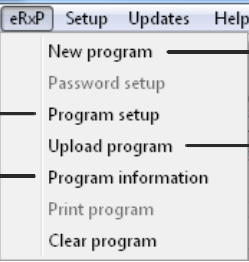
**Program lines**  
Each button represent a program line. By pushing the button a command can be entered at the program line.

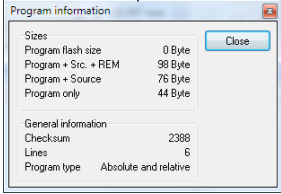
**Status texts**  
The message *Program not transferred* means that there is a difference between the program seen on the screen and the actual program in the module. This can happen if the program have been edited but not transferred. *Status: Running* (or *Stopped*) refers to the program in the module.

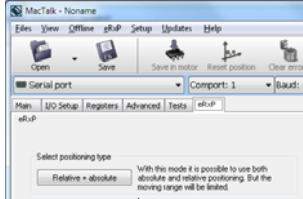
**Pause**  
Use this button if the program must be paused. By paused means that actual program line executed is temporary paused. When paused the single step feature can be used to debug the program.

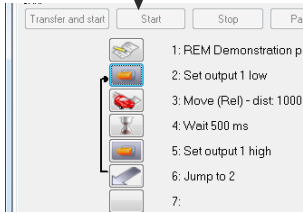








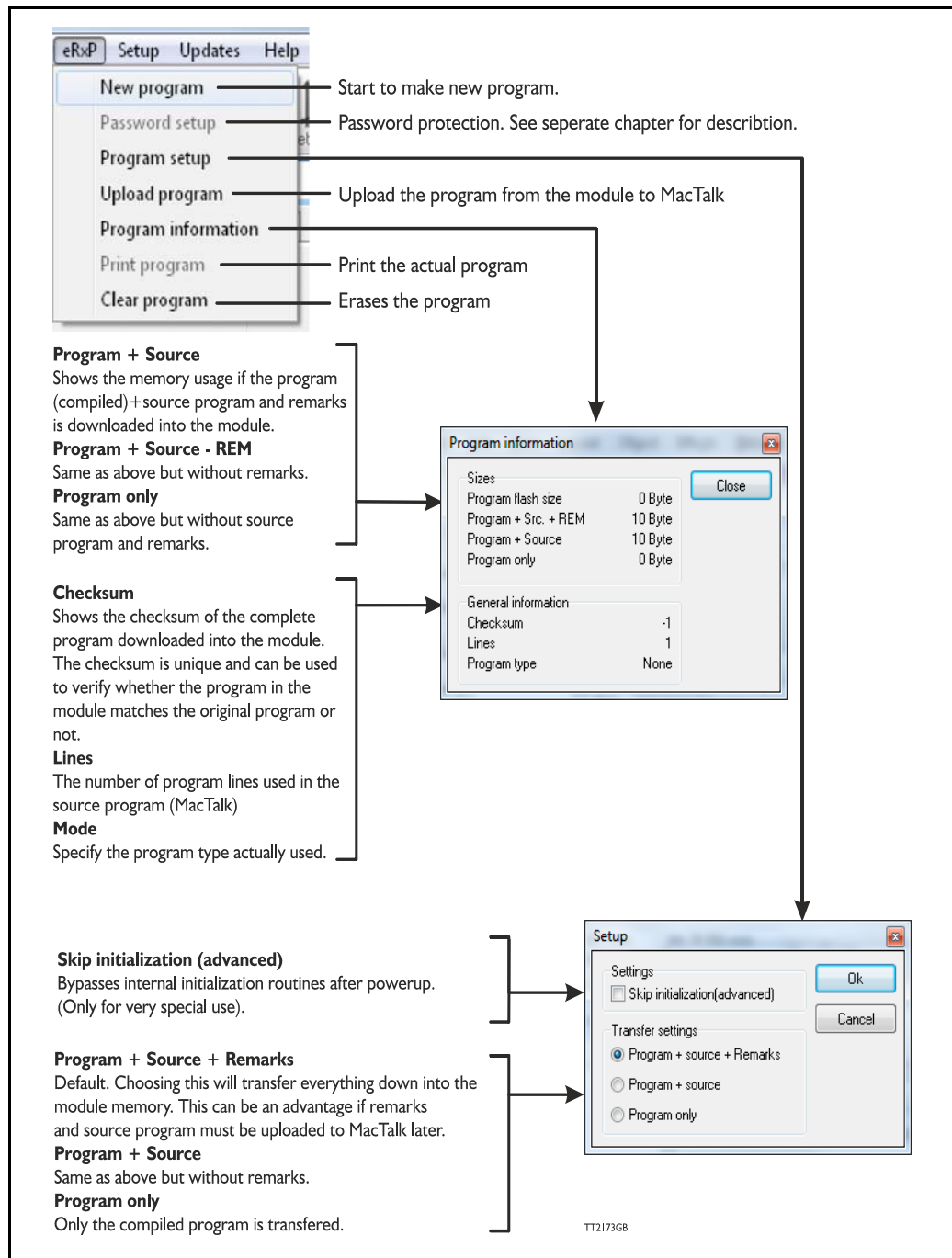




## 9.3

## Programming menu

The menu found at the top of the main window gives access to the following options:



## 9.4

# How to build a program

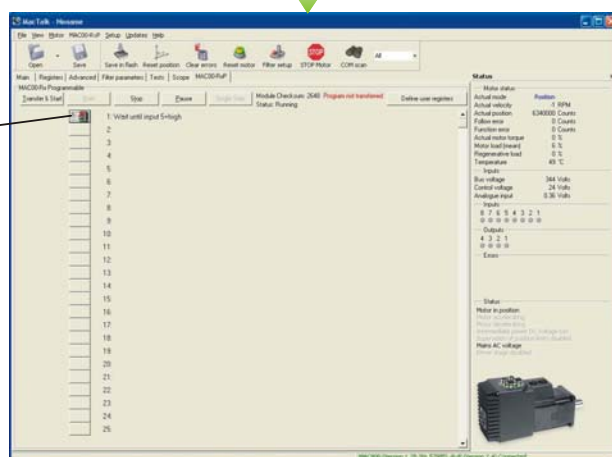
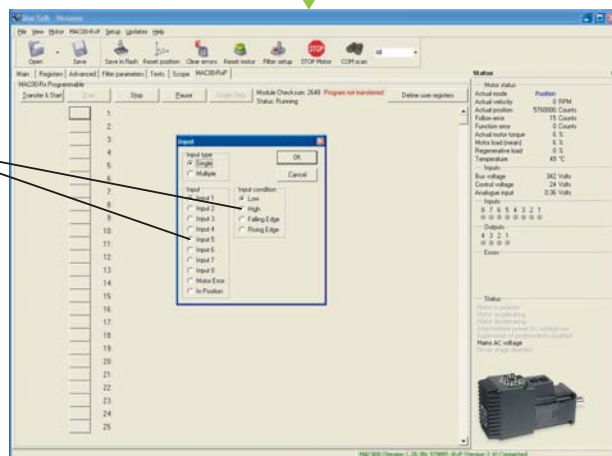
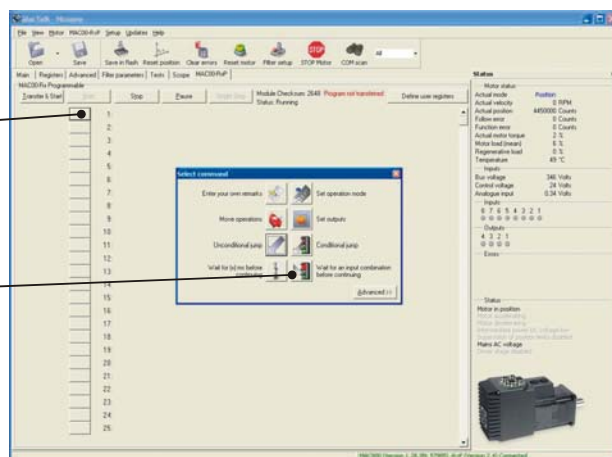
When choosing New program in the Programming menu or entering MacTalk for the first time, programming can be started.  
Press the button at line 1 and a tool box will pop up.

- ① Press the first button to create the first program line. The "Select command" box will pop up.

- ② Choose the desired command. In this example it is desired to wait for an input to be activated before further program execution.

- ③ Choose to wait until input 5 is high and press OK

- ④ The command is inserted at the previous selected program line



TT0983GB

Continued

## 9.4

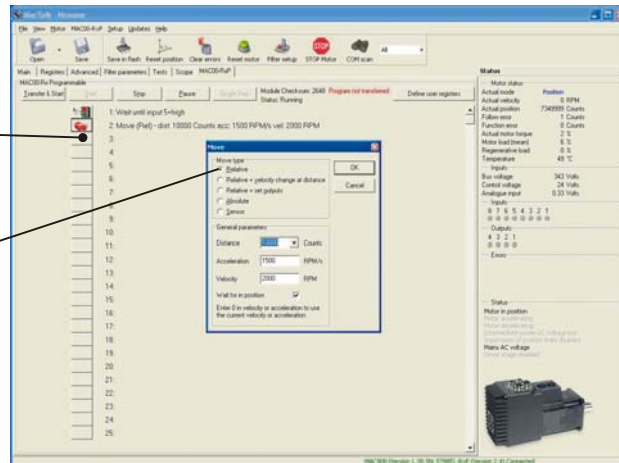
## How to build a program

⑤

Press the second button to create the second program line

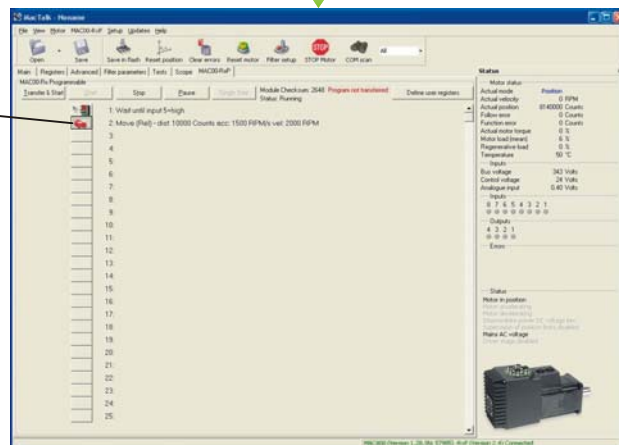
⑥

Choose the movement type needed.  
Relative: Move x counts forward with reference to the actual position.  
Absolute: Move to the x position with reference to the zero search position.



⑦

The relative move command just entered is converted into a program line.

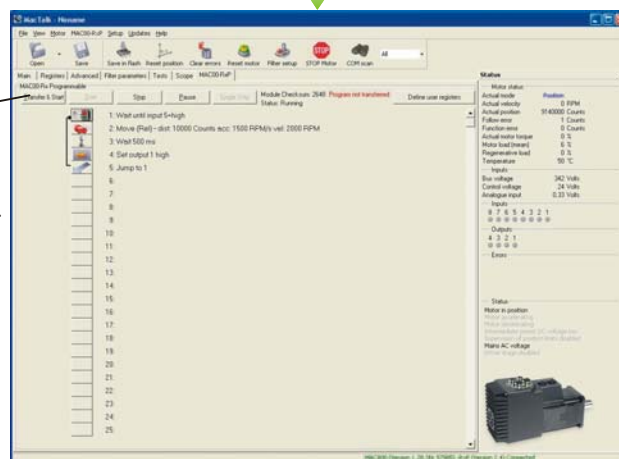


⑧

Multiple program lines are entered by the user forming the last part of the program.

⑨

Now the program is finished. Press the "Transfer & Start" button. Now the program will be transferred and stored permanently in the module. The program will be executed immediately



TT0984GB

Continued

## 9.4

## How to build a program

⑩

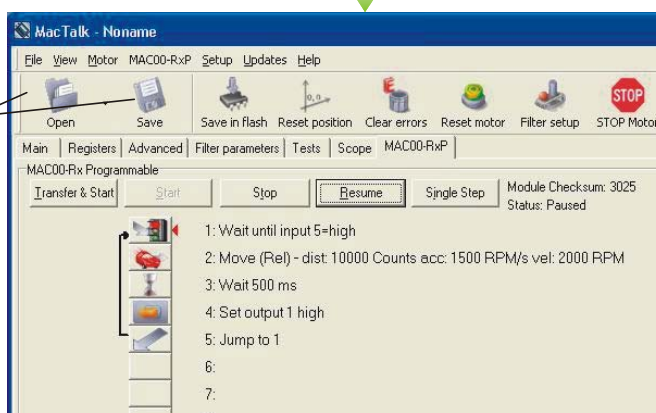
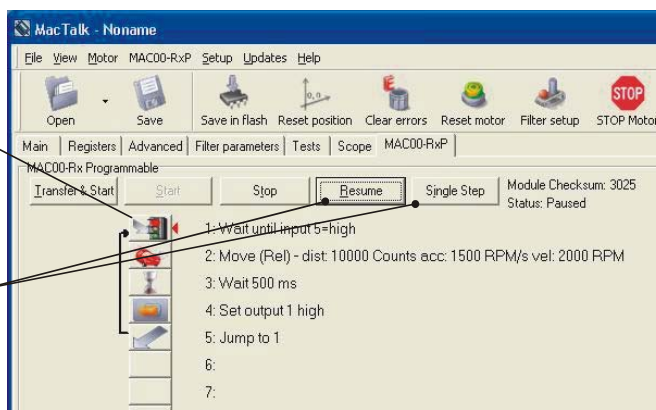
Now the program is running continuously. The actual program line which is executed is shown by the small red arrow.

⑪

By choosing the "Pause" button, the program is paused. After it is paused, it is possible to single step through each program line which can be a useful feature to debug the program since the action in each line can be closely observed.

⑫

When the program is finished, it can be saved on the harddisc or floppy disc. Please be aware that when saving the program it is the complete program including the overall setup of the motor such as servofilter, I/O setup etc. Everything is stored in a file with the extension .MAC. Later it can be opened and restored in the motor.



TT0985GB

## 9.5 General programming hints

---

When programming and saving programs the following hints may be useful to ensure that the program behaves as expected.

1. When transferring the program to the motor, it is saved permanently in memory and the program will be executed each time the motor is switched on.
2. Before beginning to program, ensure that the basic parameters for controlling acceleration, torque, safety limits, etc. are set to proper values. When saving the program to the PC, all of these basic parameter settings will be saved together with the program as a complete motor setup package.
3. A program line can be edited by double-clicking on the command text.
4. When the cursor is placed on top of the command icon, an edit menu will be shown by right-clicking.

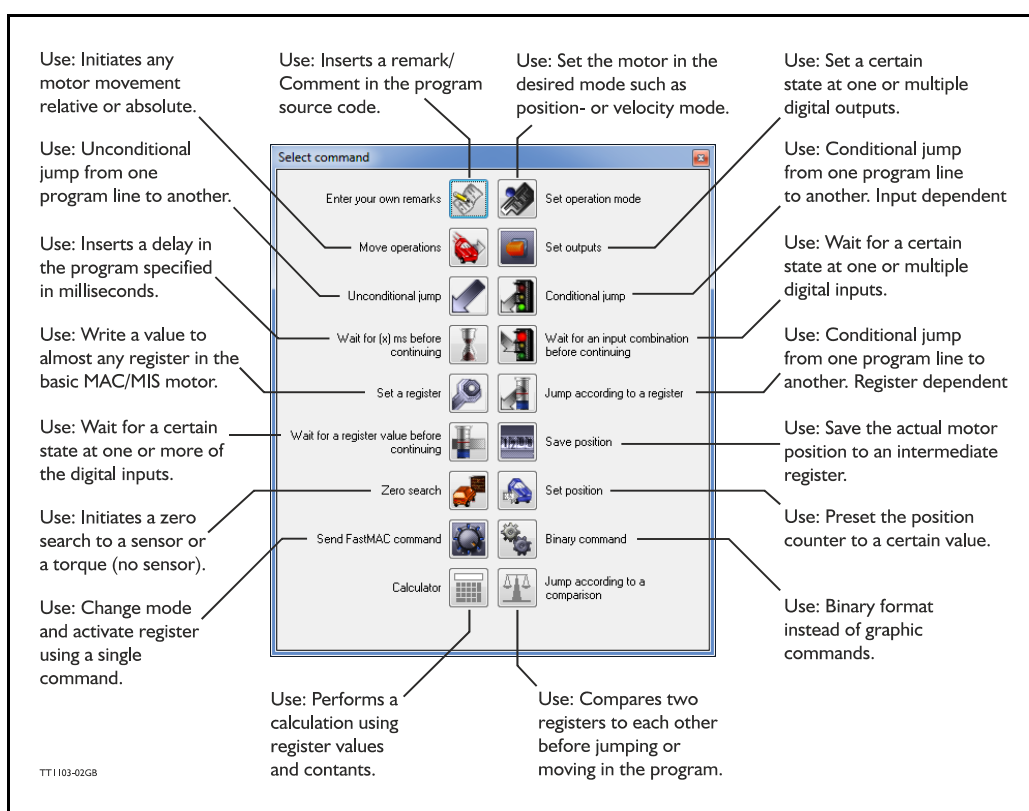
## 9.6 Command toolbox description

The toolbox used for programming covers 18 different command types.

The basic idea of the commands is to provide easy access to the most common functions of the motor. Some functions may seem to be missing at first glance, but the buttons “Set register in the QuickStep motor” or “Wait for a register value before continuing” give direct access to all the 50 registers in the basic QuickStep motor, such as the gear ratio or the actual torque register.


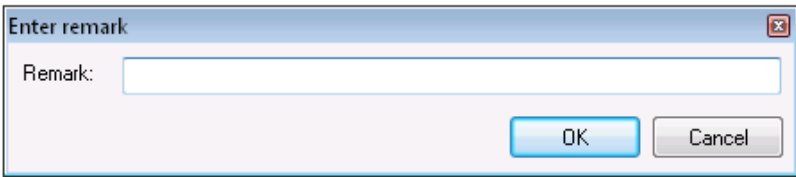
In total, this gives a very powerful programming tool since >95% of a typical program can be built using the simple command icons, while the remaining 5% is typically achieved by accessing the basic motor registers directly.

The following gives a short description of all 18 command icons.


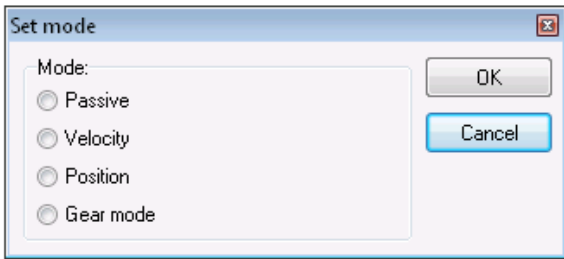


## 9.7 Graphic programming command reference


### 9.7.1 Enter your own remarks

Icon:	
Dialogue:	
Function:	Inserts a remark/comment in the source code. The program line will not do anything, but can make the source code easier to read. This can be very important if other programmers have to review or work on the code, or if the program is only worked on infrequently.

### 9.7.2 Set operation mode


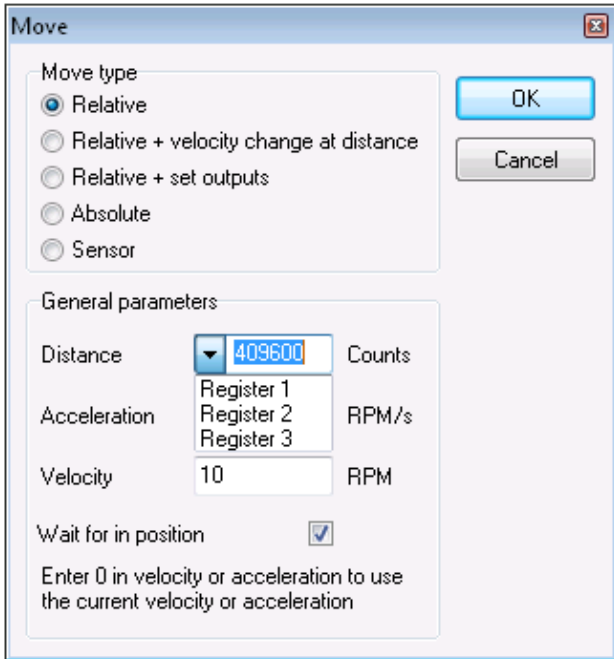
Icon:	
Dialogue:	
Function:	Sets the operating mode of the motor. When the program encounters a program line with this command, the motor's operating mode will be set to the specified mode. This allows you to use different operating modes in different parts of the program. For a detailed description of the individual operating modes, refer to section 1.3.1., <a href="#">Basic modes/functions in the QuickStep motor</a> , page 12.

### 9.7.3 Move operations

Icon:	
Function:	The Move commands are very flexible, with five different operating modes. Each mode is described in its own section below.


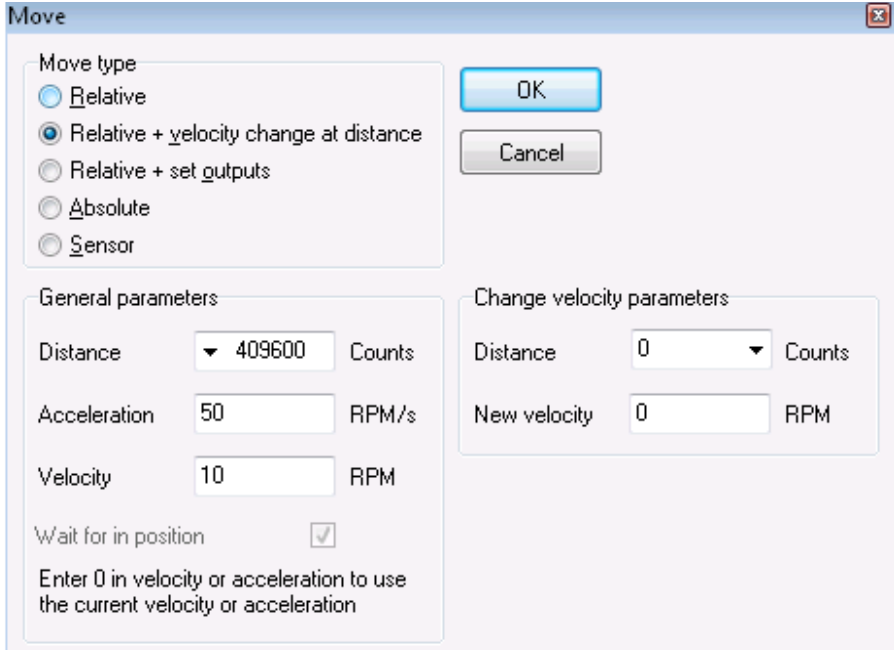
## 9.7 Graphic programming command reference

### 9.7.4 Move (Relative)

Icon:	
Dialogue:	
Function:	<p>Performs a movement relative to the current position. The distance moved is measured in encoder counts, and can either be entered directly or taken from three registers in the user memory area. For further information on using these memory registers, refer to the sections on the 'Save position' and 'Set position' commands.</p> <p>Note that if you specify a velocity, motor register no. 5 (V_SOLL) will be overwritten with this velocity value. Also, if you specify an acceleration, motor register no. 6 (A_SOLL) will be overwritten with the acceleration value specified. Register no. 49 (PI) is always overwritten by this command.</p> <p>If the 'Wait for in position' option is checked, the program will wait until the motor has finished the movement, before proceeding to the next program line. If this option is not checked, the program will start the movement, then immediately start executing the next command. The motor will finish the movement on its own, unless given other instructions by the program.</p>


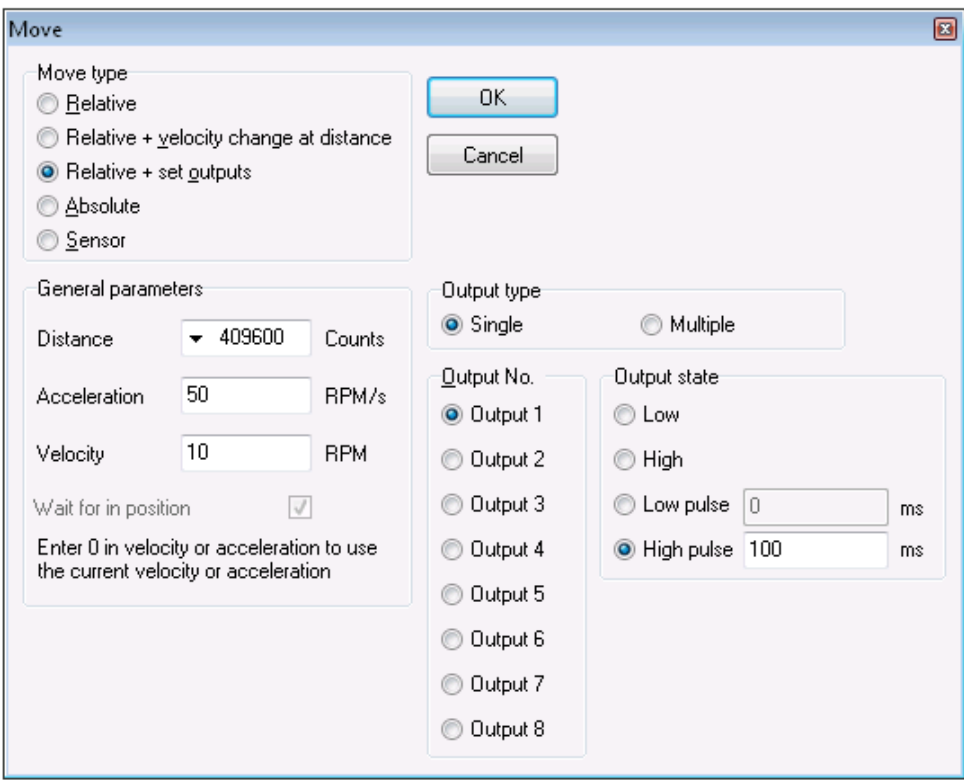
## 9.7 Graphic programming command reference

### 9.7.5 Move (Relative + velocity change at a distance)

Icon:	
Dialogue:	
Function:	<p>Performs a relative movement, and changes velocity at a specified distance before reaching the new position. The distances are measured in encoder counts and can either be entered directly, or taken from three memory registers in the RxP module. For further information on using these memory registers, refer to the sections on the 'Save position' and 'Set position' commands.</p> <p>Note that motor register no. 5 (V_SOLL) will always be overwritten with the value specified in the 'New velocity' field. Also, if you specify an acceleration, motor register no. 6 (A_SOLL) will be overwritten with the acceleration value specified. Register no. 49 (PI) is always overwritten by this command.</p> <p>This command always waits until the movement is finished, before proceeding to the next line in the program.</p> <p>In case a fatal (system) error happens such as temperature error the program execution stay at this command line until the motor is reset.</p> <p>Avoid selecting the "Wait for in position" flag and use a loop after the move command which is looking at "in position flag" and "error"</p>

## 9.7 Graphic programming command reference


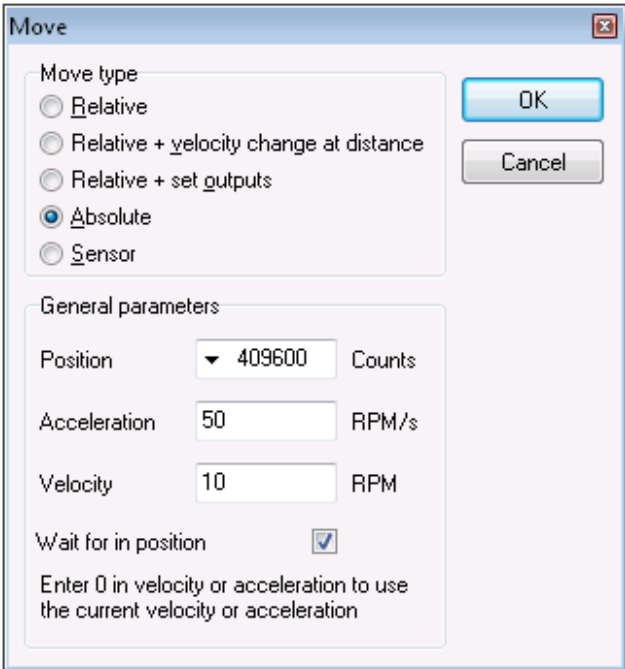
### 9.7.6 Move (Relative + set outputs)

Icon:	
Dialogue:	
Function:	<p>Performs a movement relative to the current position, and sets one or more outputs when the operation is completed. The distance moved is given in encoder counts and can either be entered directly, or can be taken from one of three memory registers in the user memory area. For further information on using these memory registers, refer to the sections on the 'Save position' and 'Set position' commands.</p> <p>Note that if you specify a velocity, motor register no. 5 (V_SOLL) will be overwritten with this velocity value. Also, if you specify an acceleration, motor register no. 6 (A_SOLL) will be overwritten with the acceleration value specified. Register no. 49 (PI) is always overwritten by this command.</p> <p>This command always waits until the movement is finished, before proceeding to the next line in the program.</p>




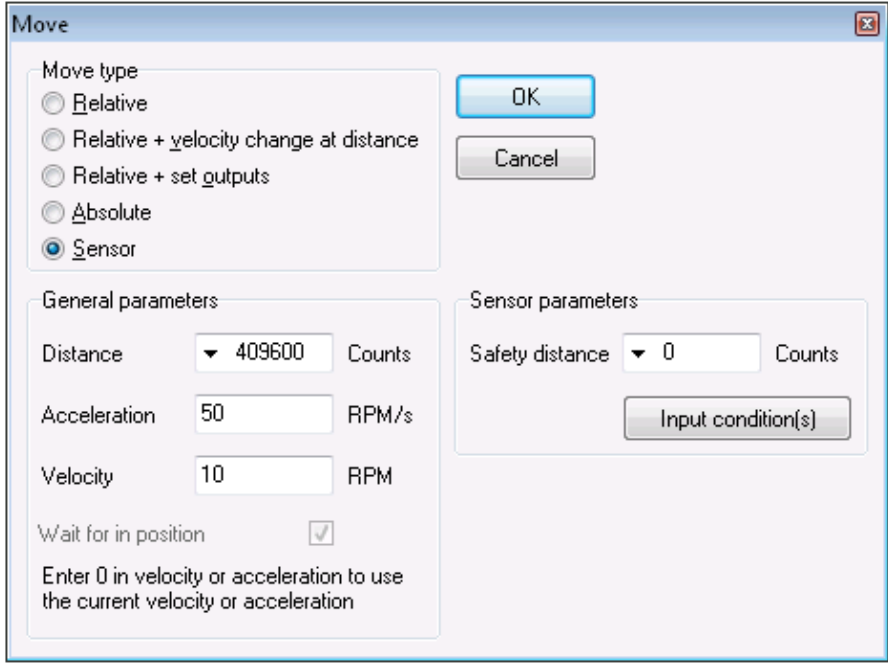
## 9.7 Graphic programming command reference

### 9.7.7 Move (Absolute)

Icon:	
Dialogue:	
Function:	<p>Moves to an absolute, non-relative position. The position is given in encoder counts and can either be entered directly, or can be taken from one of three memory registers in the user memory area. For further information on using these memory registers, refer to the sections on the 'Save position' and 'Set position' commands.</p> <p>Note that if you specify a velocity, motor register no. 5 (V_SOLL) will be overwritten with this velocity value. Also, if you specify an acceleration, motor register no. 6 (A_SOLL) will be overwritten with the acceleration value specified.</p> <p>If the 'Wait for in position' option is checked, the program will wait until the motor has finished the movement before proceeding to the next program line. If this option is not checked, the program will start the movement, then immediately start executing the next command. The motor will finish the movement on its own, unless given other instructions by the program.</p>


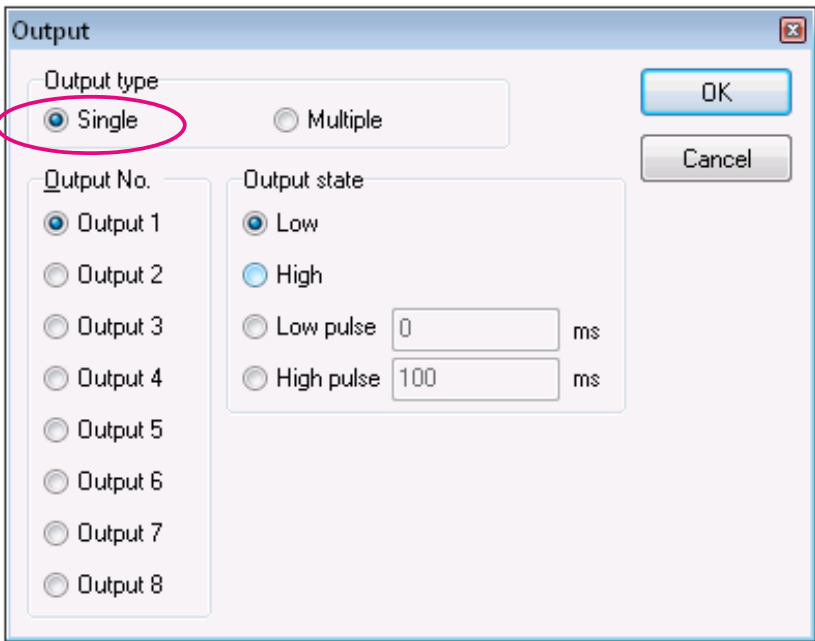
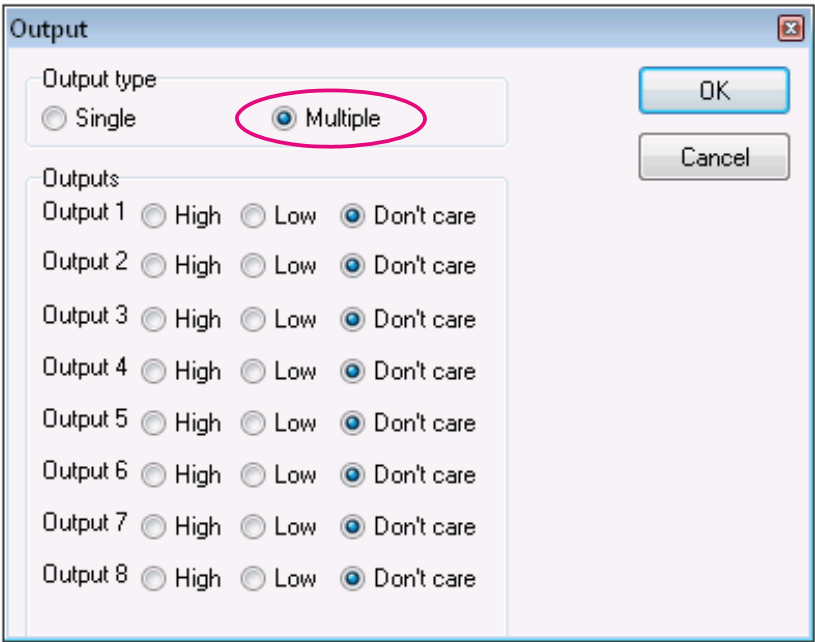
## 9.7 Graphic programming command reference

### 9.7.8 Move (Sensor)

Icon:	
Dialogue:	
Function:	<p>Performs a movement in the direction specified until an input condition is satisfied. The motor then moves the distance specified before stopping. The motor will not move farther than the Safety distance specified, regardless of whether the input condition is satisfied. The distances are measured in encoder counts and can either be entered directly, or taken from three memory registers in the user memory area. For further information on using these memory registers, refer to the sections on the 'Save position' and 'Set position' commands.</p> <p>Note that if you specify a velocity, motor register no. 5 (V_SOLL) will be overwritten with this velocity value. Also, if you specify an acceleration, motor register no. 6 (A_SOLL) will be overwritten with the acceleration value specified. Register no. 49 (PI) is always overwritten by this command.</p> <p>This command always waits until the movement is finished before proceeding to the next line in the program.</p>


## 9.7 Graphic programming command reference

### 9.7.9 Set outputs


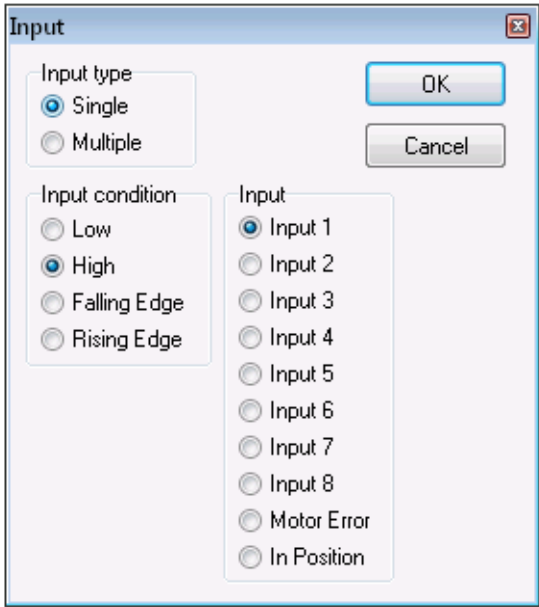
Icon:	
Dialogue:	 
Function:	Sets one or more outputs. When setting a single output, you can set it to high, low, or you can specify the length (in milliseconds) of a pulse to send out on that output. When setting multiple outputs, you can specify whether to set each output high, low, or leave it in its current state.

## 9.7 Graphic programming command reference

### 9.7.10 Unconditional jump


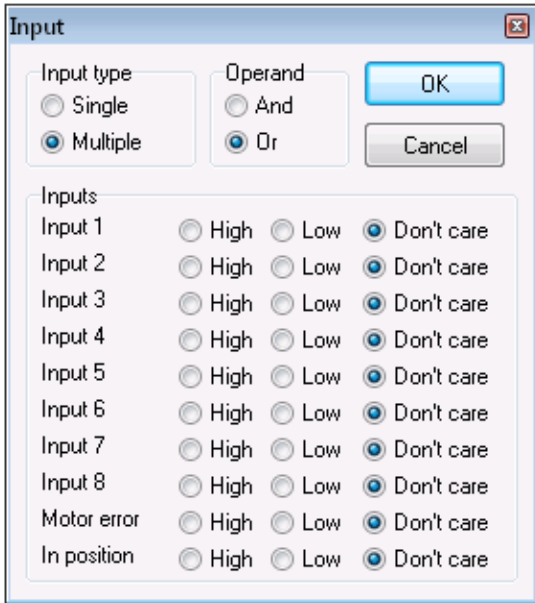
Icon:	
Dialogue:	None. After selecting this command, the mouse cursor changes. The next program line that you click on will become the destination for the jump.
Function:	Jumps to another line in the program.

### 9.7.11 Conditional jump (single input)

Icon:	
Dialogue:	
Function:	<p>Tests for an input condition before either jumping to another line in the program or moving on to the next line in the program. If the condition is met, the command jumps to the specified program line. If the condition is not met, the program proceeds to execute the next line in the program.</p> <p>When 'Input type' is set to 'Single', the command can test a single input for one of four possible conditions: the input is low, the input is high, the input has transitioned to low (Falling Edge), or the input has transitioned to high (Rising Edge). If transitions are tested for, the transition must have taken place during the last 30 microseconds.</p> <p>After pressing the OK button, the dialogue will disappear, and the mouse cursor will change. The next program line that you click on will then become the destination of the jump command.</p>



## 9.7 Graphic programming command reference

### 9.7.12 Conditional jump (multiple inputs)


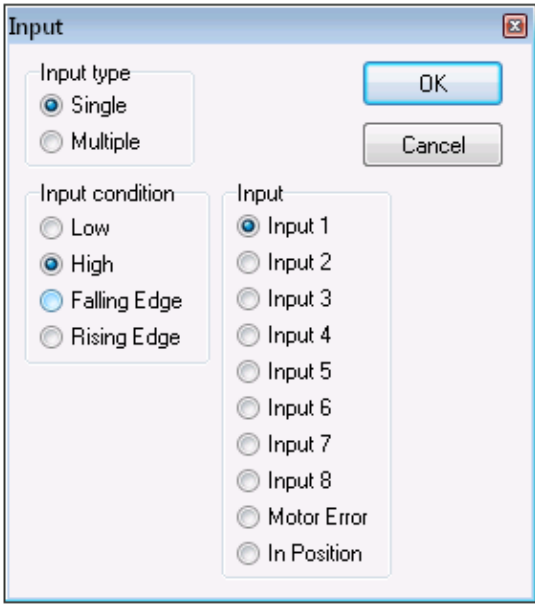
Icon:	
Dialogue:	
Function:	<p>Tests for an input condition before either jumping to another line in the program or moving on to the next line in the program. If the condition is met, the command jumps to the specified program line. If the condition is not met, the program proceeds to execute the next line in the program.</p> <p>When 'Input type' is set to 'Multiple', multiple inputs can be tested for being either high or low. The 'Operand' setting determines whether one or all of the inputs must meet their test criterion. If set to 'And', all inputs must match their test settings. If set to 'Or', only one input need match its test setting. Inputs that are set to 'Don't care' are not tested.</p> <p>After pressing the OK button, the dialogue will disappear, and the mouse cursor will change. The next program line that you click on will then become the destination of the jump command.</p>

## 9.7 Graphic programming command reference

### 9.7.13 Wait for (x) ms before continuing

Icon:	
Dialogue:	
Function:	<p>Causes the program to pause for a number of milliseconds before continuing. The maximum pause that can be specified is 32767 milliseconds. The minimum pause that can be specified is 0 milliseconds.</p> <p>Note that this command overwrites Timer 1 in the RxP module's memory.</p>


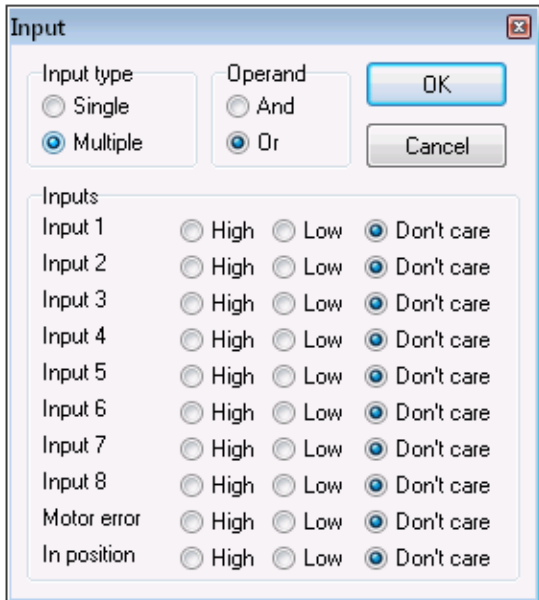
### 9.7.14 Wait for an input combination before continuing (single input)

Icon:	
Dialogue:	
Function:	<p>Waits for a specified input condition to occur. The next line in the program will not be executed until the input condition has been met.</p> <p>If 'Input type' is set to 'Single', the command will wait for one of four things to happen on the specified input: that the input tests as high, that the input tests as low, that the input transitions from high to low (Falling Edge), or that the input transitions from low to high (Rising Edge). The input is tested with 30 microsecond intervals.</p>




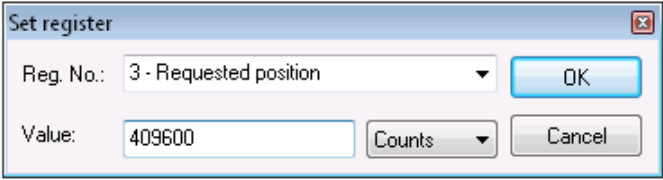
## 9.7 Graphic programming command reference

### 9.7.15 Wait for an input combination before continuing (multiple inputs)


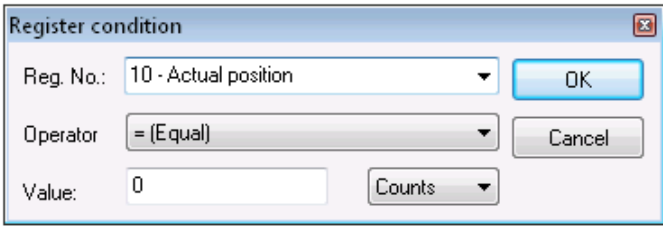
Icon:	
Dialogue:	
Function:	<p>Waits for a specified input condition to occur. The next line in the program will not be executed until the input condition has been met.</p> <p>If 'Input type' is set to 'Multiple', multiple inputs can be tested for being either high or low. The 'Operand' setting determines whether one or all of the inputs must meet their test criterion. If set to 'And', all inputs must match their test settings. If set to 'Or', only one input need match its test setting. Inputs that are set to 'Don't care' are not tested. The inputs are tested with 30 microsecond intervals.</p>

## 9.7 Graphic programming command reference

### 9.7.16 Set a register in the MIS motor


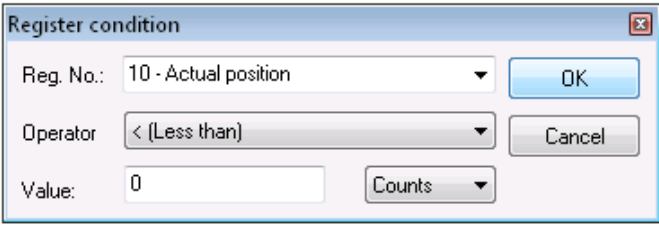
Icon:	
Dialogue:	
Function:	<p>Sets a register in the motor to a specified value. The register is selected from a list of known, user-accessible registers. The value can either be entered as native motor units or it can be entered as generic engineering units.</p> <p>The dialogue above provides an example: register no. 3 (P_SOLL, or Requested position, depending on your preference) can either be set to an integer number of encoder counts, or it can be set to a non-integer number of revolutions.</p>

### 9.7.17 Jump according to a register in the MAC motor



Icon:	
Dialogue:	
Function:	<p>Tests a register in the motor against a specified value before either jumping to another line in the program or moving on to the next line in the program. If the condition is met, the command jumps to the specified program line. If the condition is not met, the program proceeds to execute the next line in the program. The value can either be entered as native motor units, or it can be entered as generic engineering units. The dialogue above provides an example: register no. 10 (P_IST, or Actual position, depending on your preference) must be equal to 0 revolutions if the jump is to be executed. The position that the register is tested against can be specified as an integer number of encoder counts or can be specified as a non-integer number of revolutions.</p> <p>After pressing the OK button, the dialogue will disappear and the mouse cursor will change. The next program line that you click on will then become the destination of the jump command.</p>

## 9.7 Graphic programming command reference

### 9.7.18 Wait for a register value before continuing


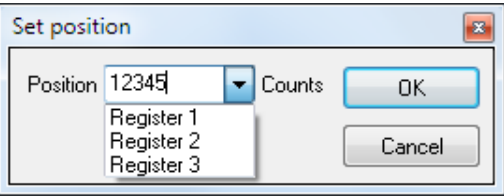
Icon:	
Dialogue:	
Function:	<p>Tests a register in the motor against a specified value and waits until the specified condition is met. The value can either be entered as native motor units or can be entered as generic engineering units.</p> <p>The dialogue above provides an example: register no. 10 (P_IST, or Actual position, depending on your preference) must be less than 0 revolutions, before the program will continue. The position that the register is tested against can be specified as an integer number of encoder counts, or can be specified as a non-integer number of revolutions.</p>

### 9.7.19 Save position


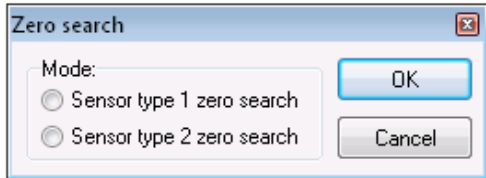
Icon:	
Dialogue:	
Function:	<p>Saves the current position from register no. 10 (P_IST) to one of three locations in the user memory area. The saved position(s) can then be used whenever a position or distance is needed in a move command.</p>

## 9.7 Graphic programming command reference

### 9.7.20 Set position


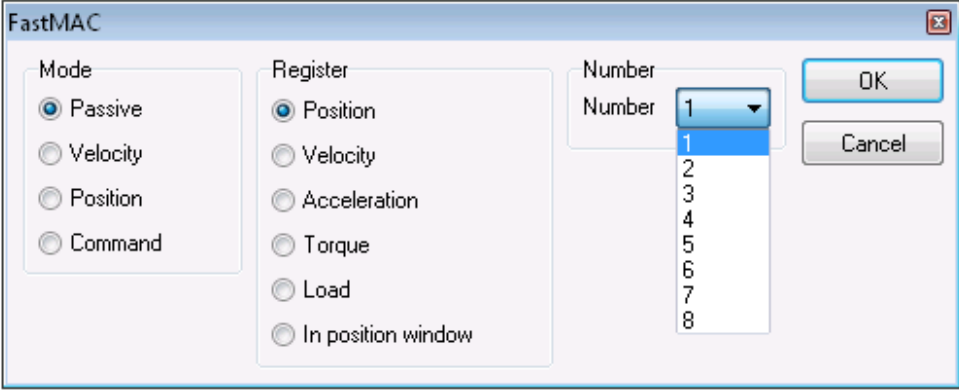
Icon:	
Dialogue:	
Function:	Change the “Actual position” (P_IST register I0) value to a new value or the value in one of three position values stored in the user memory area (register 1, 2 or 3). This is the reverse of the ‘Save position’ command.

### 9.7.21 Zero search

Icon:	
Dialogue:	
Function:	Initiates a zero search. The program waits until the zero search has completed before proceeding to the next command. For a detailed description of how to set up a zero search, refer to <a href="#">Zero search modes</a> , page 163


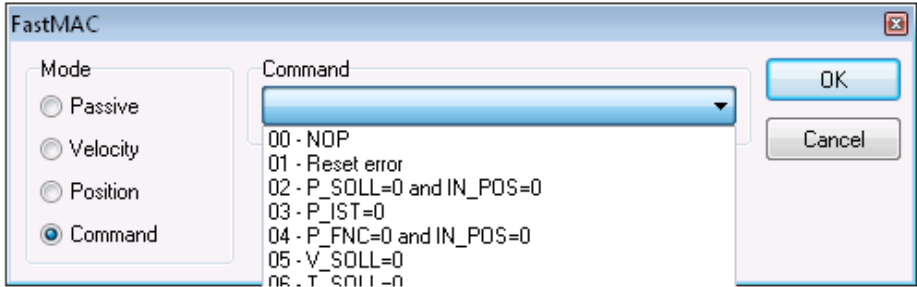
## 9.7 Graphic programming command reference

### 9.7.22 Send FastMAC command (change mode and activate register)


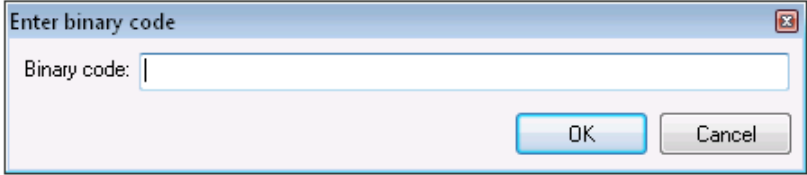
Icon:	
Dialogue:	
Function:	<p>FastMAC commands are also sometimes referred to as FlexMAC commands. The advantage of these commands is a very low communication overhead. FastMAC/FlexMAC commands are described in detail in section 4.5.7 of the MAC user manual, JVL publication no. LB0047-20GB (V2.0 or newer). However, a brief summary is in order.</p> <p>If 'Mode' is set to 'Passive', 'Velocity', or 'Position', the motor will switch to that mode. Also, one of the passive motor registers will be activated, in the sense that its value will be written to the corresponding active motor register, which actually controls motor behaviour. In the example above, the value in register no. 65 (VI) will be written to register no. 5 (V_SOLL). Move operations will then take place at that velocity.</p>

## 9.7 Graphic programming command reference

### 9.7.23 Send FastMAC command (macro command)


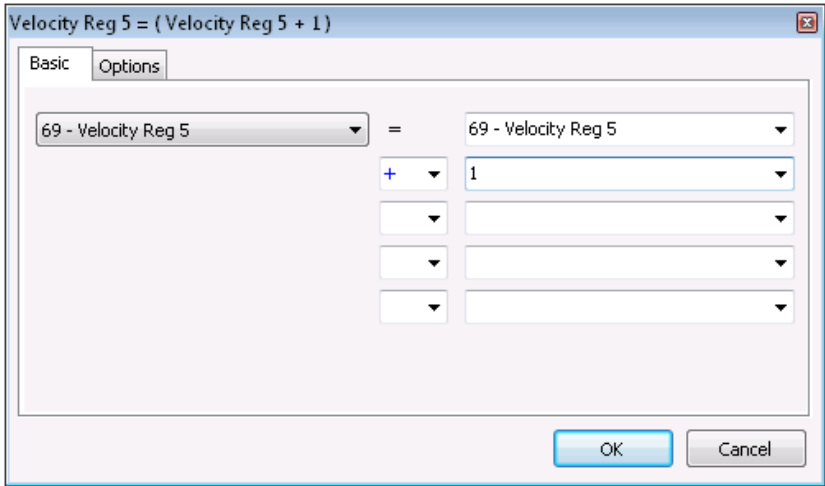
Icon:	
Dialogue:	
Function:	If 'Mode' is set to 'Command', the motor does not necessarily change mode but it can be commanded to carry out a series of predetermined operations. Describing all of the FastMAC commands is beyond the scope of this section but for example, using a single command it is possible to activate four different sets of registers, each controlling position, velocity, acceleration, torque, load factor, and in-position window. FastMAC/FlexMAC commands are described in detail in section 4.5.7 of the MAC user manual, JVL publication no. LB0047-20GB (V2.0 or newer). However, a brief summary is in order.

### 9.7.24 Binary command

Icon:	
Dialogue:	
Function:	MacTalk programs are sent to the motor in a compact, binary format, which is then interpreted by the motor's firmware. The existing set of graphic commands covers most situations, but when special needs arise, anything that can be done with programs can be done with a binary command. If special needs arise that are not covered by the other commands, contact JVL for assistance.


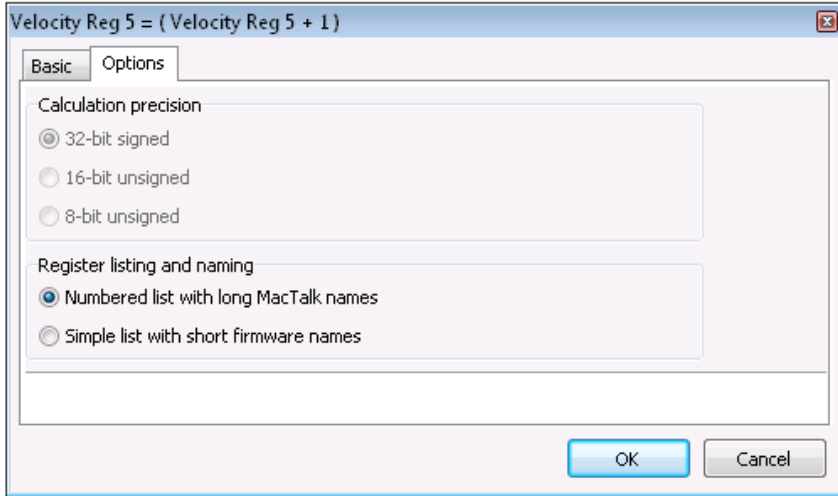
## 9.7 Graphic programming command reference

### 9.7.25 Calculator (basic)

Icon:	
Dialogue:	
Function:	<p>Performs a calculation using register values, constants, and the four basic arithmetic operations: +, -, * and /. The result is stored in a register. Arithmetic operations take place in the order that they are specified. Operands/arguments can be either integer constants or registers. The caption of the dialogue box shows the resulting expression in traditional infix format. It is continuously updated as you type in the expression.</p> <p>Note that if you write a value to a register using this command, that value is always measured in native motor units. Conversion from generic engineering units is only supported for the commands 'Set a register', 'Jump according to a register', and 'Wait for a register value before continuing'.</p> <p>If you make a calculation please be aware that most of the registers in the MIS motors operate with integers so its often needed to multiply before doing a division in order to become a precise result.</p> <p>Also be aware that all calculations are done in 32 bit format which gives the possibility to operate with values from <math>-2^{31}</math> to <math>2^{31}</math>. If the result of a calculation gives a higher value than <math>2^{31}</math> it will therefore becomes negative and similar if a calculationresult becomes lowe than <math>-2^{31}</math> the result becomes positive.</p>


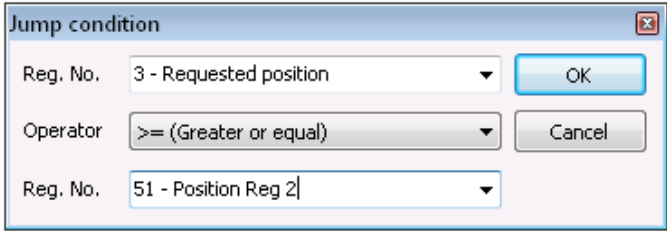
## 9.7 Graphic programming command reference

### 9.7.26 Calculator (options)

Icon:	
Dialogue:	
Function:	<p>The options tab contains various settings that affect the operation of the Calculator command. 'Calculation precision' is currently preset to 32-bit precision and cannot be changed. This is not an error, and should not be reported.</p> <p>'Register listing and naming' provides an alternative method of entering data into the dialogue by selecting 'Simple list with short firmware names'. Instead of selecting, for example, '3 – Requested position' to access register no. 3, you can simply type 'P_SOLL'. If you wish to enter a constant, you simply enter the digits – the dialogue will not mistake the constant for a register number.</p> <p>If you are in doubt about a register name, look at the expression in the caption of the dialogue box. A recognized register name will appear in the expression. An unrecognizable register name will appear as a zero. You can switch between the two methods of data entry at any time.</p>

## 9.7 Graphic programming command reference



















### 9.7.27 Jump according to a comparison

Icon:	
Dialogue:	
Function:	<p>Compares two registers with each other before either jumping to another line in the program or moving on to the next line in the program. If the condition is met, the command jumps to the specified program line. If the condition is not met, the program proceeds to execute the next line in the program.</p> <p>Any two registers can be compared with each other but the command does not do anything beyond comparing the registers numerical values measured in native motor units. To ensure that comparisons are meaningful, it is preferable to compare registers that hold the same type of information in the same binary format.</p> <p>In the example above, two position registers are compared. Both hold position information, both are 32-bit wide, and both measure position in encoder counts. Such a comparison will always yield meaningful, predictable results.</p> <p>For other types of registers, see the relevant register sections.</p>

## 9.8

## Command timing

Each command has a certain execution time. The specified execution time in the following table is the maximum execution time if not using CANopen, serial communication and the motor is disabled. The actual execution may be faster.

Icon	Name	Execution time [ $\mu$ s]
	Remarks	0
	Set operation mode	60
	Move relative (no velocity, no acceleration) <sup>1</sup>	90
	Move relative + set velocity (no acceleration) <sup>1</sup>	150
	Move relative + set velocity + set acceleration <sup>1</sup>	210
	Move absolute (no velocity, no acceleration) <sup>1</sup>	60
	Move absolute + set velocity (no acceleration) <sup>1</sup>	120
	Move absolute + set velocity + set acceleration <sup>1</sup>	180
	Set single output (high/low)	30
	Set multiple outputs	30*number of outputs
	Unconditional jump	30
	Conditional jump (inputs)	60
	Set a register	60
	Conditional jump (register)	120
	Save position	60
	Set position	90
	Send fastMAC command	30
	Binary command	30

1) The time for all move commands is shown without waiting for in position

## 9.9 More about program timing

---

The firmware is structured so that one program instruction is executed for each pass of the main loop, which takes approximately 30 microseconds ( $\mu\text{s}$ ) without CANopen, without serial communications and when the motor is not running. The Main Loop Time is termed MLT in the following text.

A single program line in MacTalk can generate more than one instruction. For example, assigning a constant value to a register uses two instructions: First load the value to the internal stack and then Store from the stack to the target register. The above table in [Motor Connections](#), page 414 reflects this operation.

The main loop time will vary depending on a number of factors: The serial communications speed and load, whether CANopen is installed, and the CANopen communications speed and load.

Serial communications on the RS-485 line can load the motor up to 1% at 19.200 baud, which is insignificant, but at the maximum baud rate of 921.600 the communications can load the motor up to 45%, which would result in an MLT of  $\sim 60 \mu\text{s}$ .

When CANopen firmware is installed, the basic MLT will change from 30 to 90  $\mu\text{s}$  with no communications.

When loading the CANbus with communications, the MLT can rise significantly. For example, when using seven transmit PDOs with an event timer value of 1 ms and a CANbus link speed of 500 kbits/s, the MLT can rise to 150-200  $\mu\text{s}$ . Also using RS-485 communications at high baud rates can result in even longer MLT values. However, this scenario is very unlikely.

**Note: In applications where program timing is critical, tests must be performed to ensure that timing is satisfactory when communication is running according to conditions used in production!**

The MIS motors offers optional 6 different Ethernet protocols.  
These are:

- **EtherCAT**
- **ModbusTCP**
- **Profinet**
- **Powerlink**
- **EthernetIP**
- **SercosIII**

This manual do only cover description of how to connect.

Concerning software and protocol setup and usage - please consult a separate manual that can be found at [www.jvl.dk](http://www.jvl.dk) using this link: [www.jvl.dk](http://www.jvl.dk)

The litterature number is LB0056-xxGB (xx=version).



This chapter covers the JVL Stepper motor controllers SMC66 and SMC85 which are used with the MIS I 7x, 23x, 34x and 43x motor series on a CANopen network.

The chapter covers the following main topics:

- General introduction: a section with general information about CANopen. See [section 11.1.1](#) to [section 11.1.5](#).
- Setting up the Baud-rate, node-id and termination of the CAN bus. Covers also the wiring of the CAN bus. See [section 11.2.1](#) to [section 11.2.5](#).
- Using CAN-explorer.  
See [section 11.3.1](#) to [section 11.3.3](#).
- Survey of Communication specific objects and manufacturer specific objects in the DS301 standard. Communication objects consist of the general information about the settings in the module, while the Manufacturer specific objects consist of the settings of input/output and the motor parameters. This section also covers the settings of the transmit and receive PDOs in the module. See [section 11.4.1](#) to [section 11.4.10](#).
- Survey of objects which are used in the DSP-402 standard. See [section 11.5.1](#) to [section 11.5.11](#).
- Section with more detailed explanations of the CANopen theory, particularly DS-301.  
See [section 11.9.1](#) to [section 11.9.7](#).

### 11.1.1 Introduction

The CANopen option is available for the MIS17x and MIS23x (SMC66) and MIS34x, MIS43x (SMC85).

When this option is installed, the controllers include a CANopen slave.

Through the CANopen slave, all the registers of the controller can be accessed.

Both implement object dictionaries that follow the CiA DS-301 standard.

The controllers and motors are designed to be used on a CANbus, CANopen DS-301 and CANopen DSP-402. Do not use the modules together with CANKingdom or DeviceNet.

#### DSP402 mode.

All the MIS motors (G2) and the SMC66 and 85 controllers supports dynamic mapping. DSP402 mode.

All profile specific objects that are PDO mappable, can be dynamically mapped and combined.

The motor comes with a factory setting for default mapping which has faster updating than the dynamic mapping. See the EDS file for default mapped objects.

#### DS301 mode

In this mode. No objects above index 59FFh are available.

Dynamic mapping of all objects in both 16- and 32-bit are supported. The data field length of a PDO can be between 1 and 8 bytes so the user can decide how to combine these objects to obtain the most efficient packed amount of information in order to decrease bus load. Please consult the section *Dynamic Mapping, page 296* to learn how to construct these PDO's. Default mapping is also supported, which makes it compatible with the old generation MIS23x motors (SMC75) if no changes in the PDO mapping have been made by the user or EDS-file.

### 11.1.2 CiA membership

CiA (CAN in Automation) is a non-profit society. The object of the society is to promote CAN (Controller-Area-Network) and to provide a path for future developments of the CAN protocol. CiA specifications cover physical layer definitions as well as application layer and device profile descriptions.

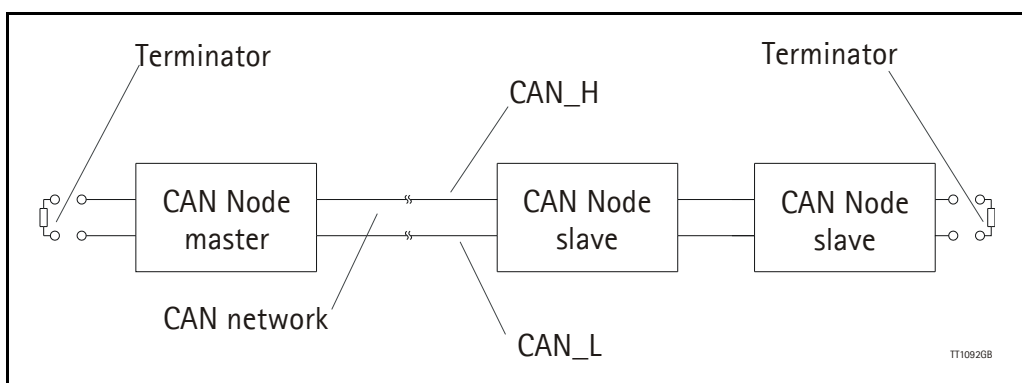
In order to receive the CAN standard, it is necessary to obtain CiA membership. The membership fee depends on a company's number of employees. Membership runs from January 1<sup>st</sup> until December 31<sup>st</sup> and is renewed automatically unless cancelled in writing by the end of a calendar year. Companies applying for membership after July 1<sup>st</sup> pay 50% of annual membership.

A PDF application form can be downloaded from <http://www.can-cia.org/cia/application.html>.

Note: Once you have received a license from CIA, standards will be sent on a CD and are downloadable via member login. All of the CiA specifications can be ordered from the following URL: <https://www.can-cia.org/groups/specifications/>

**11.1.3 CANopen network**

The CAN bus is a serial bus with multi-master capabilities where different products from different manufacturers can communicate with each other. These include, for example, devices such as PLCs, motors, sensors and actuators. Some message types have higher priority and are sent first, for time-critical applications. New devices can easily be integrated on an existing bus, without the need to reconfigure the entire network. The devices are connected through a 2-wire bus cable with ground, and data is transmitted serially.



#### 11.1.4 CANopen, general information

CANopen is a CAN-based, higher-level protocol. The purpose of CANopen is to give an understandable and unique behaviour on the CAN network. The CAN network is the hardware level of the system, and CANopen is the software level. CANopen is based on the communication profile described in CiA DS-301, and specifies all of the basic communication mechanisms.

CiA DS-301 contains message types on the lowest software level. The DSP-402 CANopen standard defines the device profile and the functional behaviour for servo drive controllers, frequency inverters and stepper motors. The DSP-402 constitutes a higher software level, and it uses the DS-301 communication, but makes the device independent of the manufacturer. Not all JVL functionality is available.

The CANbus with real-time capabilities works in accordance with the ISO 11898 standard. The major performance features and characteristic of the CANopen protocol are described below:

##### Message-oriented protocol:

The CANopen protocol does not exchange data by addressing the recipient of the message, but rather marks each transmitted message with a message identifier. All nodes in the network check the identifier when they receive a message to see whether it is relevant for them. Messages can therefore, be accepted by none, one, several or all participants.

##### Prioritisation of messages:

As the identifier in a message also determines its priority for accessing the bus, it is possible to specify a correspondingly rapid bus access for messages according to their importance. Especially important messages can thus gain access to the bus without a prolonged wait-time, regardless of the loading on the bus at any instant.

This characteristic means that important messages are transmitted with high priority even in exceptional situations, thereby ensuring proper functioning of a system even during phases of restricted transmission capacity.

**Multi-Master capability:**

Bus access rights are not issued by a mean-level control unit (bus master) per network. Instead, each network node can start to send a message with equal rights as soon as the bus has become free. If several participants access the bus at the same time, an arbitration process allocates each participant the bus access right in line with the priority of the message they want to send at that particular moment. Each participant can therefore communicate directly with every other participant. As the transmission of a message can be initiated by the message source itself, then in the case of event-controlled transmission of messages, the bus is only occupied when a new message is on-hand.

**No-loss bus arbitration:**

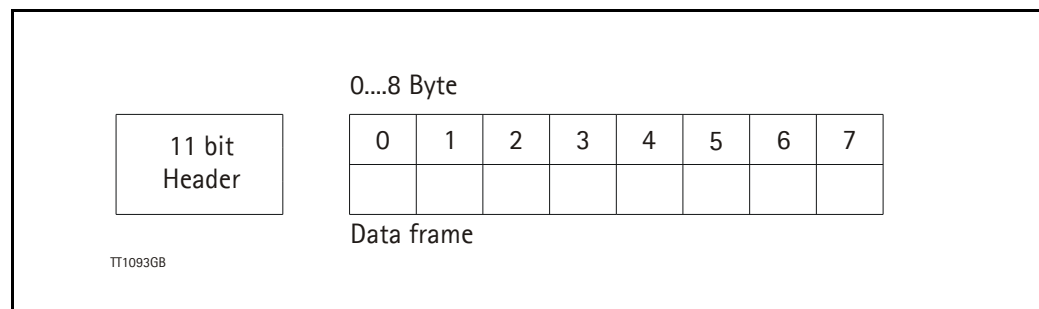
As the bus is accessed at random under the CANopen protocol, it is possible that several participants try to occupy the bus at the same time. In other random bus access routines, this causes the destruction of suppressed messages. In order to solve such a bus access conflict, a repeated occupation of the bus is required using an appropriate triggering strategy. The CANopen protocol therefore deploys a routine to ensure that the message with the highest priority at any given time is sent without any destruction of message contents.

**Short block length:**

The maximum data length of a CAN message is limited to 8 bytes. This data length is usually sufficient to transmit the information occurring in the lowest field area in a CAN message.

**11.1.5****Header**

A CAN message transmits the communications object and a variety of management and control information. The management and control information bits are used to ensure error-free data transmission, and are automatically removed from the received message and inserted before a message is sent. A simplified CANopen message could be as in the figure below:



The two bit fields "Header" and "Data" form the simplified CANopen message. The 11-bit Header is also designated as the identifier or as the COB-ID (Communication Object identifier).

JVL uses the 11-bit format type CAN A, but not the 29-bit format type CAN B.

The COB-ID carries out two tasks for the controller communications object.

- Bus arbitration: Specification of transmission priorities.
- Identification of communications objects.

The COB-ID comprises two sections:

- Function code, 4 bits in size (0....15)
- Node address (Node ID), 7 bits in size (0....127).

The function code classifies the communications objects, and controls the transmission priorities. Objects with a small function code are transmitted with high priority. For example, in the case of simultaneous bus access an object with the function code "1" is sent before an object with the function code "3".

**Node address:**

Every device is configured before network operation with a unique 7-bit long node address between 1 and 127. The device address "0" is reserved for broadcast transmissions, in which messages are sent simultaneously to all devices.

PDO, SDO, EMCY, NMT and heartbeat use the header frame for communication on the CANopen bus.

## 11.2 Connect and setup of the CAN bus

Option

### 11.2.1 Connecting the motor to the CAN bus

Before you connect the motor to the CAN-bus, the Baud-rate, the Node-ID and the termination must be selected.

On the serial bus it is possible to set a transmission speed (Baud-rate) of max. 1000 Kbit/s and a minimum of 10 Kbit/s. The Baud-rate depends on the cable length, and the wire cross-section. The table below gives some recommendations for networks with less than 64 nodes. Recommended bus cable cross-sections are according to CIA.

:

Bus Distance (m)	Cross-section (mm <sup>2</sup> )	Terminator (Ohms)	Baud-rate (Kbit/s)
25	0.25-0.34	120	1000
100	0.34-0.6	150-300	500
250	0.34-0.6	150-300	250
500	0.5-0.6	150-300	125
500	0.5-0.6	150-300	100
1000	0.75-0.8	150-300	50

The bus wires may be routed in parallel, twisted and/or shielded, depending on EMC requirements. The layout of the wiring should be as close as possible to a single line structure in order to minimize reflections. The cable stubs for connection of the bus node must be as short as possible, especially at high bit rates. The cable shielding in the housing must have a large contact area. For a drop cable, a wire cross-section of 0.25 to 0.34 mm<sup>2</sup> would be an appropriate choice in many cases.

For bus lengths greater than 1 km, a bridge or repeater device is recommended. Galvanic isolation between the bus nodes is optional.

### 11.2.2 Necessary accessories:

The EDS file for the MIS motors is available for download at JVL's web-site, <http://www.jvl.dk>, under the downloads menu, Field bus Interface Specifications Files. EDS means Electronic Data Sheet. This file contains the information about the motor settings that are required to configure the setup and program in the master.

The MIS motor is a slave module on the CAN-bus. The master can, for example, be a PLC or a PC.

If you are using a PLC as master, then make sure it is provided with a CANopen communications module, and that the correct programming tools are available. For support of the PLC master, the PLC vendor is recommended.

If you are using a PC as master, JVL provides some tools that can help when installing and using the MIS motors.

## 11.2 Connect and setup of the CAN bus

Option

The latest firmware for the MIS motors is available at JVL's web-site under the menu downloads/firmware. In the site's programs menu, the software CAN-explorer is also available. Use this link to find and download the CAN-explorer: <https://www.jvl.dk/430/free-and-trial-programs>.

Note that this is a free-ware program.

See also the chapter: [Using CAN-explorer, page 285](#)

CAN-explorer can be used to load the EDS file and operate with the motor.

The CAN-explorer software must use a special dongle for communication with the PC. For further information about the dongle, see [An overall method for communication test](#), page 285. The PC must be provided with a CANopen communications module.

### 11.2.3 EDS (Electronic data Sheet)

In order to give the user of CANopen more support, the device description is available in a standardised way, and gives the opportunity to create standardised tools for configuration of CANopen devices, designing networks with CANopen devices, and managing project information on different platforms. The EDS file are ASCII-coded.

### 11.2.4 Setting the node id and baud rate

The node id is set using MacTalk. It is located in register 166. The baud rate is also set using MacTalk and is located in register 167.

See also [Baud\\_Rate](#), page 225

### 11.2.5 Bus termination

In order to guarantee correct operation of the CAN bus, bus terminating resistors must be provided at both ends of the bus cable.

See the general connection guide for connecting CAN bus to the MIS motors.

[How to connect a MIS motor, page 41](#)

### 11.3.1 The CAN-explorer program

The CAN-explorer is a program that is very help full during installation of a CAN system. The program offers features that are very convenient and which make it very easy to start up the MIS motor when this is supplied with the CANopen option.

The program can write and send SDOs, PDOs, SYNC and heartbeat messages, and also can read EDS files.

Use this link to find and download the CAN-explorer: <https://www.jvl.dk/430/free-and-trial-programs>

Note that this is a free-ware program.

### 11.3.2 An overall method for communication test

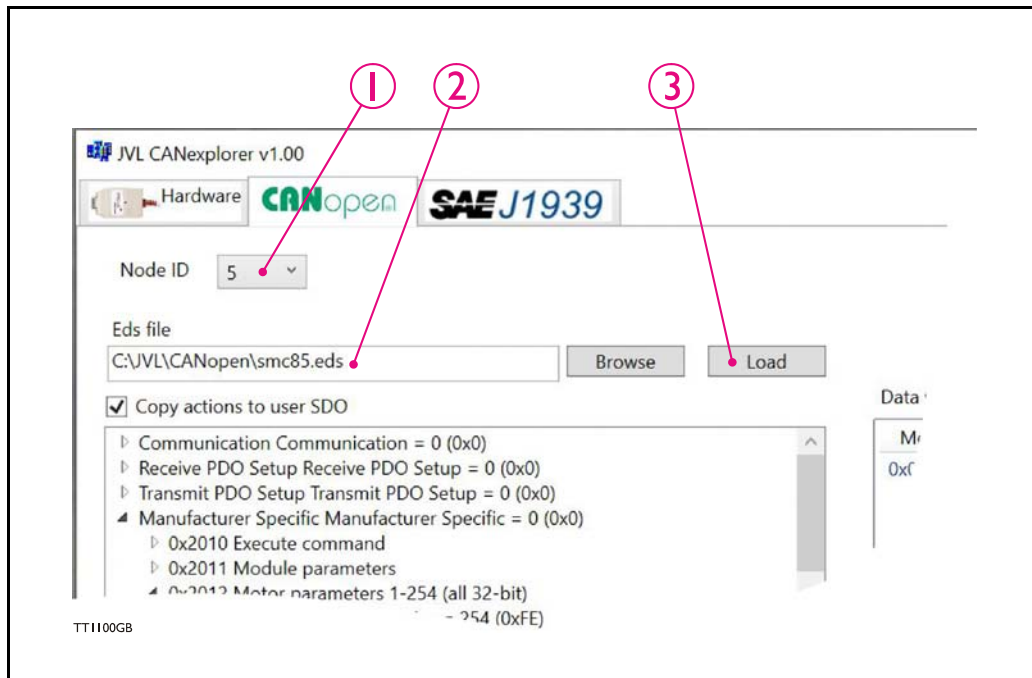
Depending on the type of master and software solution available, the following components must be available:

- PLC: PLC with a CANopen module and software that can communicate with this module.  
The CANopen module must be connected to a CAN bus, as shown in [section 11.2.5](#). To set up the master, download the EDS file from the JVL web site (see [section 11.2.2](#)). This file contains all register set-up data for the MIS motors. For details of the node-ID and the Baud-rate, see [section 11.2.4](#). The power supply must be connected to the motor as shown in [How to connect a MIS motor](#), page 41.
- PC: PC with a CAN adaptor and software that can communicate with this module, or if the CAN-explorer software is used, the PCAN-USB Dongle from Peak-system that is connected to a USB port on the PC. The Peak systems web site address is <http://www.peak-system.com>. This includes a list of distributors. To set up the master, download the EDS file from the JVL web-page, see [section 11.2.2](#). This file contains all register set-up data for the MIS motors. For details of the node-ID and the Baud-rate, see [Setting the node id and baud rate](#), page 284. The power supply must be connected to the motor as shown in [How to connect a MIS motor](#), page 41.

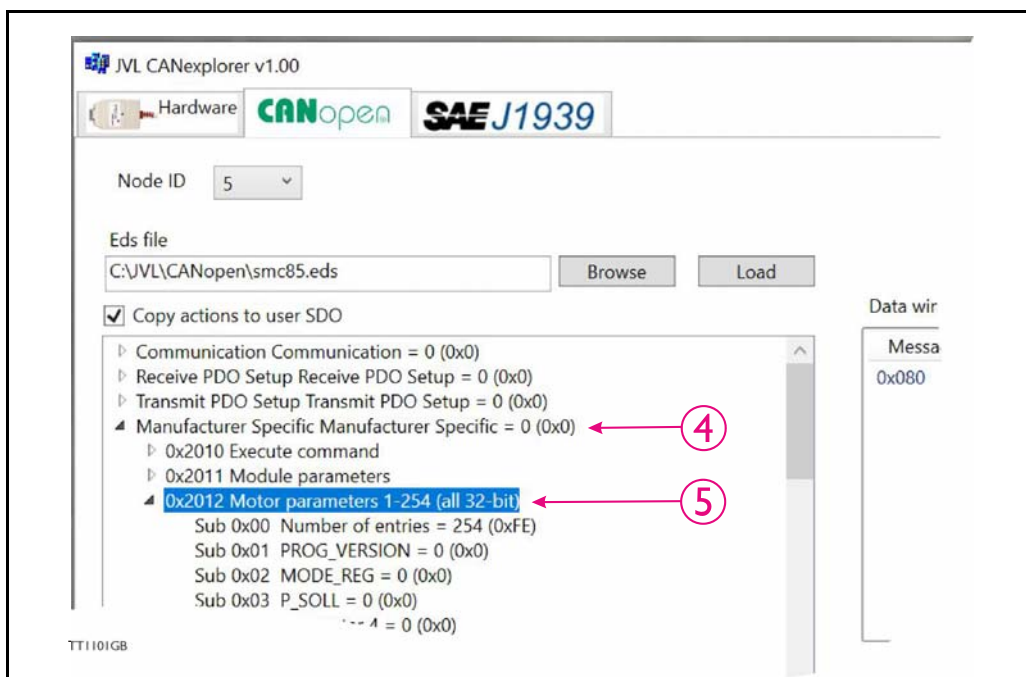
If *CAN-explorer* is used, see the following method for testing the motor communication:

- Download the program from: <https://www.jvl.dk/430/free-and-trial-programs>.
- Install CAN-explorer
- Connect the motor to the USB port via the Dongle.
- Connect power supply, see section [How to connect a MIS motor](#), page 41.
- Run the CAN-explorer program on the PC.

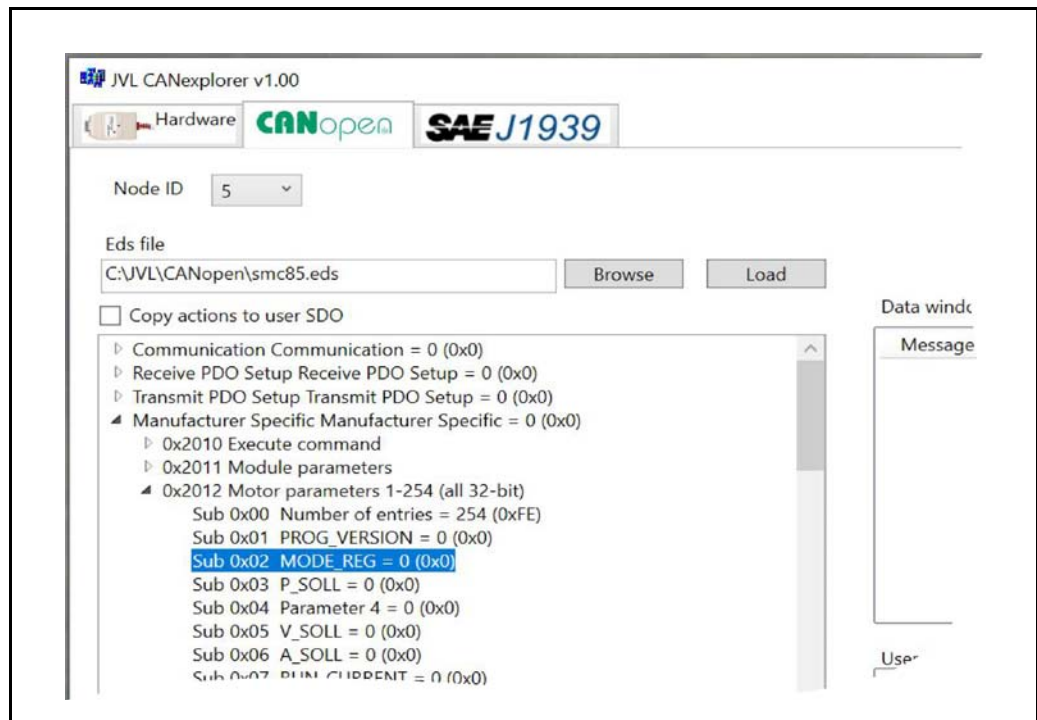
- 1: Select the correct node ID in the slave using MacTalk.  
See [Setting the node id and baud rate](#), page 284.
- 2: Select the EDS file. For all the MIS motors this file is SMC85\_V1\_00\_S.eds or newer.
- 3: Load the EDS file by pressing load.



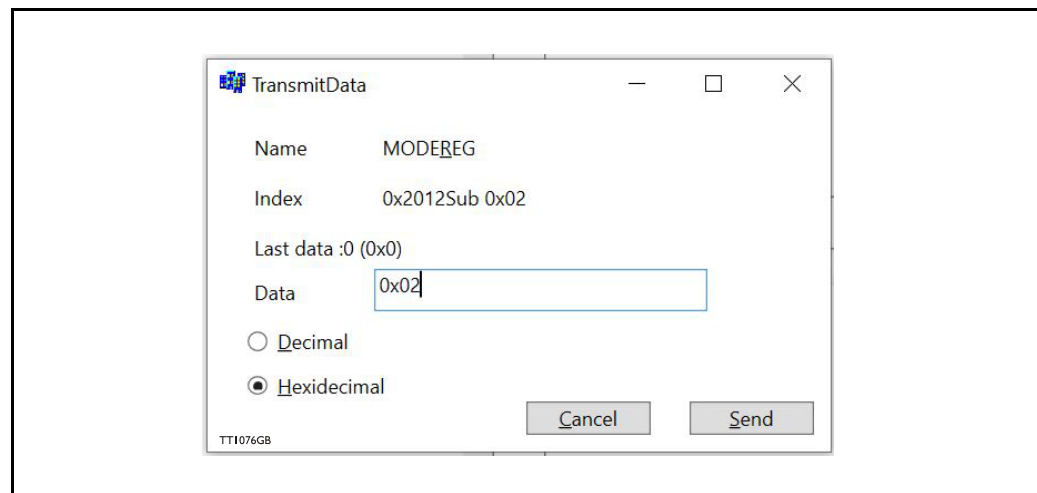
- 4: Select here on the ► the manufacturer specific register.
- 5: Select thereafter the object 0x2012. Object 0x2012 contains the motor parameters.



- 6: Point to the sub register 0x02, which is the register that determines in which mode the motor will operate.



Press W on the keyboard. The following screen appears:



- 7: Type 02 in the window, and press *Send*.
- 8: Click on the sub register 0x05, which is the register to choose the velocity the motor will use. Press W on the keyboard, type 10000 in the window, and press *Send*. The value 10000 is in RPM/100.
- 9: Click on the sub register 0x03, which is the register to choose the distance the motor will run. Press W on the keyboard, type 4096000 in the window, and type *Send*. The value 4096000 is in Steps and equivalent to 10 revolutions.

Now the motor shaft will rotate slowly, until the motor has counted 4096000 Encoder pulses. If you want to stop the motor, then click on sub register 0x02 and write 0 in the window, and the motor will switch to passive mode. If using other software, the test could be described as, (using object 2012h):

Sub-register	Name	Width	Unit	Operation	Value
02h	Mode_Reg	32 bit		Set up the motor in position mode	02h
05h	V_SOLL	32 bit	RPM	Sets up the desired velocity	10000h
03h	P_SOLL	32 bit	Steps	The motor rotates the desired numbers of encoder pulses	4096000
02h	Mode_Reg	32 bit		Sets the motor to passive mode	00h
Returning the motor with higher velocity					
02h	Mode_Reg	32 bit		Set up the motor in position mode	02h
05h	V_SOLL	32 bit	RPM	Sets up the desired velocity	20000h
03h	P_SOLL	32 bit	Steps	The motor rotates the desired numbers of Steps	-4096000
02h	Mode_Reg	32 bit		Sets the motor in passive mode	00h

### 11.3.3 How to use CAN-explorer

After start up, the name and details of the HW-interface, such as PCAN\_USB should appear upper left.

When you turn on a motor/CAN node after having started CAN-explorer, the Data Window (large centre right), will contain a message with the number 0x7xx, where xx is the node ID. For example: 0x704 will indicate node 4. Set the Node ID field top left to that value (4).

Ensure that the correct EDS\_file is loaded. The program loads a hard-coded default file - either SMC85\_VI\_00\_S.eds (or newer). It is also possible to load another EDS file by clicking on the browse button, and browse to another EDS file.

When clicking on the load button, the new eds file is loaded into the EDS window.

Normal operation will be to select an object in the EDS view pane, and press either R for read or W for write. Pressing R should read the value (successful if no error pops up). Pressing W for write will pop up a small window in which the present value is displayed in both decimal and hex. It is then possible to write a new value either in decimal or hex using a 0x prefix, such as 0x185 to enable the first TPDO on node 5 (by clearing the high bit). If the "Copy action to user SDO" checkbox is checked, the object will be added to the user SDO list as a write SDO.

The SDOs in the user SDO pane can be rearranged by dragging them with the mouse. Pressing "U" while a line in the user SDO is selected, will update the object, either by reading or writing the object.

The bus state can be changed using the NMT buttons, lower left, e.g. to Operational to enable PDOs.

The button Read User SDOs will read all of the “R” type objects in the user SDO list. This is useful for updating a large number of values in the SDO view.

The button Write User SDOs will write all of the “W” type objects in the user SDO list. This is useful for automated testing.

Entries can be deleted from the user SDO list by selecting them with the mouse and pressing the delete key.

The sync Time field (top right) sets the time in milliseconds for the SYNC, heartbeat or node guarding messages to be sent out. Cyclic transmission can be started and stopped on the start cyclic button.

The button “single sync” send one sync message for PDO testing.

The close button exits the program after saving the list of user SDOs, which will be automatically reloaded at the next program start.

The “user SDO window” will show the commands sent if the “Copy actions to user SDO” are marked.

This window can be edited so the commands can be moved up or down in the list and delays can be added using mouse – right bottom.

Just below this window there is an option for saving the user SDO list, loading an old user SDO list and Read or Write all the SDO’s in the list.

This is very useful for setting up the module for testing the PDO’s and functionality of the motor using the SDO’s in an easy way.

## 11.4 Objects in the DS301 standard

Option

### 11.4.1 DS301 specified Communications objects

The DS301 specified Communications objects are shown in the table below. To obtain the default value in CAN-explorer, press R on the keyboard, and the actual value will be shown.

Name	Index (hex)	Sub Index	Data Type	Read only	Default	Description
Device type	1000		UINT32	X	0x40192	Mandatory DS301 object
Error register	1001		UINT8			This is the mapping error register, and it is part of the emergency object. If any of the sub indices are high, an error has occurred. See also section 11.4.2. Mandatory Bit description. 0 Generic error. Mandatory 1 Current 2 Voltage 3 Temperature 4 Communication (Overrun) 5 Device profile specific 6 Reserved 7 Manufacturer
Pre-defined error field	1003	0	UINT8			Number of active errors. Write 0 to this object to reset errors
Pre-defined error field 1		1	UINT32	X		Last active error
Pre-defined error field 2		2	UINT32	X		Second last active error
COB-ID SYNC	1005		UINT32			Sync ID
Manufacturer device name	1008		String	X	JVL A/S	
Manufacturer hardware version	1009		String	X		
Manufacturer software version	100A		String	X		
Guard time	100C		UINT16			Informs about the Guard time in milliseconds. See also node guarding.
Life Time Factor	100D		UINT16			Is the factor that guard time is multiplied with to give the life time for the node guarding protocol
Store parameters	1010		UINT8			Store flash data
Store code		1	UINT32			
Restore default parameters	1011		UINT8			Restore CANopen objects to default.
Restore code		1	UINT32			

## 11.4 Objects in the DS301 standard

Option

Continued

Name	Index (hex)	Sub Index	Data Type	Read only	Default	Description
COB-ID EMCY	1014		UINT32			
Consumer Heartbeat Time	1016		UINT16			
Number of Entries		0	UINT8		3	
Consumer Heartbeat Time of Node-ID 1		1	UINT32		0	
Consumer Heartbeat Time of Node-ID 2		2	UINT32		0	
Consumer Heartbeat Time of Node-ID 3		3	UINT32		0	
Producer heartbeat time	1017		UINT16		0	If the Heartbeat timer is not 0, Heartbeat is used. See heartbeat
Identity Object	1018		UINT8			Contain general information about the motor
Number of Entries		0	UINT8		4	
Vendor ID		1	UINT32		0x117	Vendor ID, contains a unique value allocated to each manufactor. 117h is JVLs vendor ID.
Product code		2	UINT32			
Revision number		3	UINT32			
Serial number		4	UINT32			

## 11.4 Objects in the DS301 standard

Option

### 11.4.2 Emergency object

The EMCY (emergency) object is used to transfer an error message to the CANopen master, or also to another node which can process the error message.

The reaction on the emergency object is not specified. An emergency object is transmitted only once per "error event".

The COB ID of the emergency object is  $80h + \text{node}$

2 bytes CANopen Errorcode

1 byte Error register index 1001h

2 bytes ERR\_BITS register 35 (only with DSP402 enabled)

The following CANopen error codes can be generated in the first two bytes:

Supported error codes

Value	Description
1001h	Motor error
1002h	Position error
1003h	Follow error
1004h	Low bus voltage error
1010h	Homing. Unsupported profile
8000h	Communication error
8210h	wrong PDO length
8130h	Node guard fail

Errors be read from the Predefined error register 1003h

Sub index 0: Number of buffered errors.

Sub index 1 : Predefined error field 1. Last error to occur

Sub index 2 : Predefined error field 2. Previous error.

By writing the value 0 to 1003h sub index 0, a clear error event is raised in the motor, and the predefined error buffer is emptied.

Content of predefined error field:

2 bytes CANopen Errorcode

2 bytes ERR\_BITS Low word register 35 (only with DSP402 enabled)

## 11.4 Objects in the DS301 standard

Option

### 11.4.3 Node guarding

The motor support both node guarding and heartbeat.

This is two different ways to assure that nodes on the network is present.

#### Heartbeat.

Heartbeat is a signal sent from the motor on a cyclic basis.

The enabling and setting of the heartbeat is done with object 1017h

When this object has a value different then 0 the slave will transmit a heartbeat message with the time interval according to the value of object 1017h in milliseconds.

The master can monitor this signal to verify that the slave is present on the CAN bus.

#### Node guarding

With the node guarding protocol, the master and slave can monitor each other.

If enabled the slave will answer each node guard request with a response.

The slave will monitor the time between each request and go into error state if guard time is exceeded.

Node guard is configured with the objects

100Ch Guard time in milliseconds. Value 0 disables the Node Guard.

100Dh life time factor. The life time factor multiplied with the Guard time gives the maximum milliseconds allowed between two node guard poll request from the master.

### 11.4.4 Restore

Object 1011h Restore data.

By writing 64616F6Ch "LOAD" to object 1011h sub 01 the motor restores all DSP402 parameters to factory default.

No parameters are stored by this action. If factory default should be startup values, a store command should be issued after the restore command.

### 11.4.5 Store parameters

Object 1010h Store parameters

By writing 65766173h "SAVE" to object 1010h sub 01 all storable registers are stored in flash.

Some DSP402 objects are storable. See Object dictionary for DSP402 objects with store support.

## 11.4 Objects in the DS301 standard

Option

### 11.4.6 Object dictionary

Name	Index (hex)	Sub-index	Data type	Read only	Default	Description
Execute command	2010		UINT8			Execute motor command
Command		1	UINT16			
Module parameters	2011		UINT8			For backwards compatibility
Reserved		1-6				
Input setup		7				
Setup		8				
Reserved		9-16				
Error acknowledge DSP301		17				
Read/Write register 32bit	2012		UINT8			Access to the 32 bit motor register, n
Read/Write register 32bit		1-255	UINT32			
Execute fastcommand	2013		UINT8			
Command		1	UINT8			Fast Command
SubCommand		2	UINT8			Sub command
Read/Write register 16bit	2014		UINT8			Access to the motor register n, but as 16bit
Read/Write register 16bit			UINT16			
Reserved	2015		UINT8			For backwards compatibility
Reserved		1-3				
Read/Write register 256-511 32bit	2018		UINT8			Access to the 32 bit motor register, n
Read/Write register 256-511 32bit		1-255	UINT32			
Read/Write register 256-511 16bit	201A		UINT8			Access to the motor register n, but as 16bit
Read/Write register 256-511 16bit		1-255	UINT16			
Set homing torque	2100		UINT16			See homing mode

Writing to these objects in CAN-explorer is done by pressing W on the keyboard when the register in folder Manufacturer is selected. Reading is done by pressing R.

## 11.4 Objects in the DS301 standard

Option

**Object 2012h – Motor parameters 1-255**

**Object 2018h – Motor parameters 257 - 511**

With these objects, most of the registers of the MIS motor can be accessed. All the registers are accessed as 32 bit. When reading and writing to 16-bit registers, the values are automatically converted.

**Object 2014h – Motor parameters 1 - 255(16 bit)**

**Object 201Ah – Motor parameters 257 - 511 (16 bit)**

2014h and 201Ah works as 2012h and 2018, but the parameters are accessed as 16-bit. If writing to a 32bit parameter, the lsb 16-bit value will be treated as signed.

All the registers in the stepper motor is 32 bit. These object are only for backwards compatibility.

### 11.4.7 Enable and Disable PDOs

In the CANopen profile, it is recommended to limit the use of transmit and receive PDOs to max. four each, enabled at the same time.

In mode DSP301 and DSP402, all PDOs are disabled when the module is booted up. The user must choose which PDOs the application will use and enable these.

There are some DSP402 specific objects default mapped in the motor. If these are used, there is a performance advance, compared to dynamically mapped objects. See the EDS file for default mapped objects.

To enable or disable a PDO, it is necessary to write to the MSB (bit 31) in the PDO COB-ID entry in the PDO communication parameter Record. The COB-ID register is sub-index 1h, and the value range of this register is UNSIGNED32.

The PDOs are enabled when bit 31 is reset, and is disabled when bit 31 is set.

## 11.4 Objects in the DS301 standard

Option

### 11.4.8 Dynamic Mapping

All motor registers are available in the "Manufacturer Specific" objects.

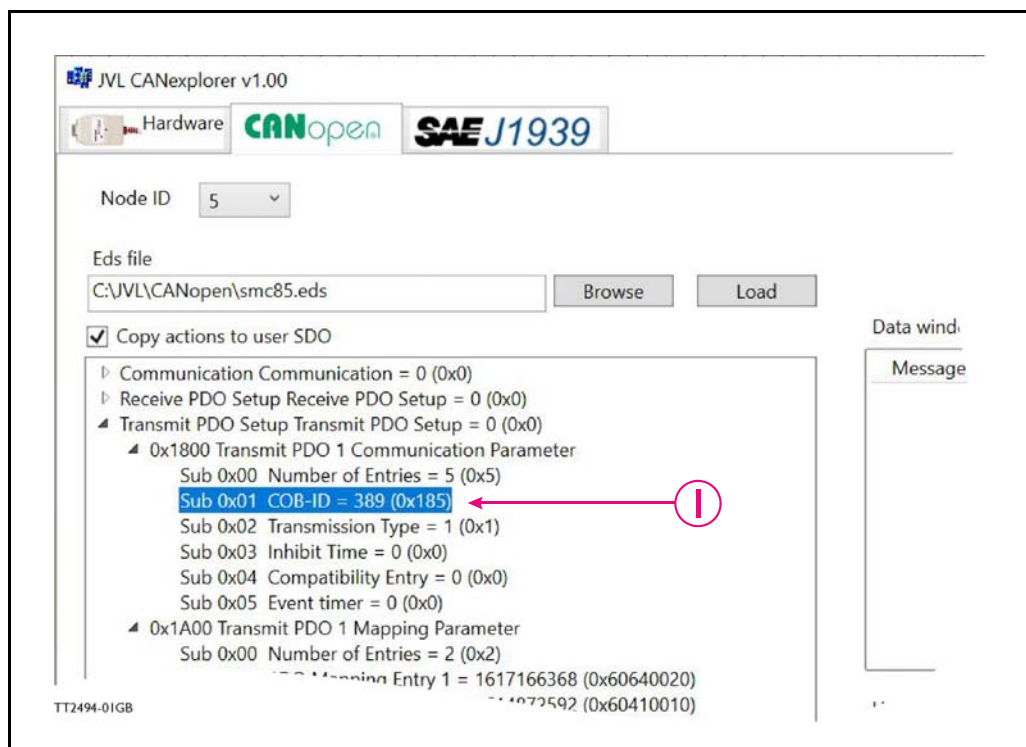
The JVL service program CAN-explorer can be used to test and verify all parameters related to the CAN interface and connection.

Example: Dynamic Mapping in CAN-explorer

CAN-explorer shows that TxPDO1 is mapped to transmit object 0x2012, sub 0x02, 32 bit.

If the user wants to map to another object, following procedure must be followed:

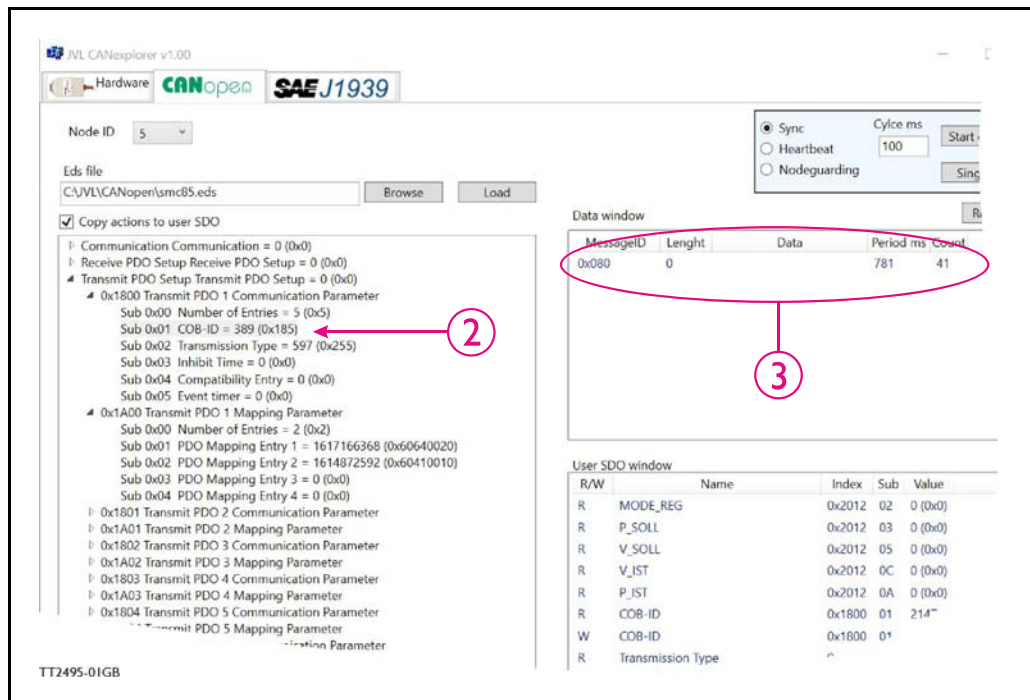
1. Set NMT state pre-operational:  
Set the motor NMT state to Pre-operational by clicking on the "Pre-oper" button under the EDS window. In this state its allowed to change the PDO mapping.
2. Disable PDO:  
The object 1800 Sub01 contains the ID of the PDO, and is used to both contain the COP-ID of the PDO and disable the PDO by setting the 31 bit in the object.  
Read the value from the motor, in this case 185h. Set the 31 bit, and write 80000185h back to the motor: - see #1 at the illustration below.
3. Enable remapping:  
Write 0 in object 1A00h Sub 00 Number of Entries, to allow to edit PDO mapping.
4. Remap:  
Remap to another object by writing to object 1A00 sub01 PDO Mapping Entry 1: 20120520h. This will map the object 2012h Sub05 to the first TX PDO. The last two digits is the number of bits in the object.  
For object 2012h Sub05 this is a 32bit object = 20h.
5. Lock mapping:  
Write 1 in object 1A00h Sub00 Number of Entries. This will lock the mapping and enable 1 entry in the PDO mapping. The number of entry should match the number of mapped objects in the list. In this case, we only have one mapped object. This step is necessary before enabling the PDO.



## 11.4 Objects in the DS301 standard

Option

6. Set transmission state:  
Set the transmission state to cyclic by writing 255 to object 1800 sub02
7. Set event time:  
Set the event timer to 100ms by writing 100 to object 1800 sub05. This combined with the transmission type 255 enables the PDO to be transmitted every 100ms.
8. Enable PDO:  
Activate the PDO by resetting the MSB bit in object 1800 Sub01, by writing the value 185h - see #2 at the illustration below
9. Set NMT state Operational:  
Click the button "Start" to set the slave to NMT state operational. The PDO is transmitted according to the transmission type object.



### 11.4.9 Receive PDOs

All the available PDO's are dynamical map-able.

The DSP401 objects (PDO 1- 19) default mapping does have a performance advance if used without remapping. The pre mapped DSP 301 object (PDO 20 - 24), are suggested mapping.

All objects in DSP402 profile support event driving transmission types.

Default mapping:

Receive PDO1

Writes the DSP402 Controlword.

Byte	0	1	2	3	4	5	6	7
Data	Controlword							
Object	6040h							

## 11.4 Objects in the DS301 standard

Option

### Receive PDO2

Writes the DSP402 Controlword, and the DSP402 mode of operation.

Byte	0	1	2	3	4	5	6	7
Data	Controlword		Mode operation					
Object	6040h		6060h					

### Receive PDO3

Set the target position and Profile velocity. e.g. Used for Profile position mode.

Byte	0	1	2	3	4	5	6	7
Data	Target Position				Profile Velocity			
Object	607Ah				6081h			

### Receive PDO4

Set the acceleration and target velocity. e.g. used for profile velocity mode.

Byte	0	1	2	3	4	5	6	7
Data	Profile Acceleration				Target Velocity			
Object	6083h				60FFh			

### Receive PDO5

Set data to the data record object. This is used in the interpolated position mode.

Byte	0	1	2	3	4	5	6	7
Data	IP mode data record							
Object	60C1h sub01							

### Receive PDO6

Set the Torque slope and Target torque. e.g. used for Profile torque mode.

Byte	0	1	2	3	4	5	6	7
Data	Torque Slope				Target Torque			
Object	6087h				6071h			

### Receive PDO2I

Update the target position (P\_SOLL), the maximum velocity (V\_SOLL) and the acceleration (A\_SOLL).

Byte	0	1	2	3	4	5	6	7
Data	P_SOLL				V_SOLL		A_SOLL	
Object	2012h, sub 3				2014h, sub 5		2014h, sub 6	

## 11.4 Objects in the DS301 standard

Option

Receive PDO22

Update the running current and operating mode.

Byte	0	1	2	3	4	5	6	7
Data	RUN_CURRENT		MODE_REG					
Object	2014h, sub 7		2014h, sub 2					

Receive PDO23

PDO23: Used to issue a Motor command.

Byte	0	1	2	3	4	5	6	7
Data	Command							
Object	2014 sub18h							

Receive PDO24

Updates the output

Byte	0	1	2	3	4	5	6	7
Data	Output							
Object	2014 Sub19h							

## 11.4 Objects in the DS301 standard

Option

### 11.4.10 Transmit PDOs

Default mapping:

Transmit PDO1:

Read actual position, DSP402 status word.

Byte	0	1	2	3	4	5	6	7
Data	Actual position				Status word			
Object	6064h				6041h			

Transmit PDO2:

Read digital inputs and Actual position.

Byte	0	1	2	3	4	5	6	7
Data	Digital Inputs				Actual position			
Object	60FDh				6064h			

Transmit PDO3:

Read the DSP402 status word, and the active mode of operation.

Byte	0	1	2	3	4	5	6	7
Data	Status word		Operation Mode Display					
Object	6041h		6061h					

Transmit PDO4:

Read the requested torque, actual torque and the actual velocity.

Byte	0	1	2	3	4	5	6	7
Data	Torque demand		Torque Actual		Velocity Actual			
Object	6074h		6077h		606Ch			

Transmit PDO21:

Read the Actual position, actual velocity (as 16 bit) and the error register (First 16 bits).

Byte	0	1	2	3	4	5	6	7
Data	P_IST				V_IST		ERR_BITS	
Object	2012h, sub0Ah				2014h, sub0Ch		2014h sub23h	

Transmit PDO22:

Read actual velocity (as 16 bit).

Byte	0	1	2	3	4	5	6	7
Data	V_IST							
Object	2014h sub0Ch							

## 11.4 Objects in the DS301 standard

Option

Transmit PDO23:

Read analog input 1,2,3 and 4.

Byte	0	1	2	3	4	5	6	7
Data	Analog Input 1		Analog Input 2		Analog Input 3		Analog Input 4	
Object	2014h, sub59h		2014h, sub5Ah		2014h, sub5Bh		2014h, sub5Ch	

Transmit PDO24:

Read analog input 5,6,7 and 8.

Byte	0	1	2	3	4	5	6	7
Data	Analog Input 5		Analog Input 6		Analog Input 7		Analog Input 8	
Object	2014h, sub5Dh		2014h, sub5Eh		2014h, sub5Fh		2014h, sub60h	

### 11.4.11 Beckhoff support

The MIS motors (or SMC66/85) supports running CAN with Beckhoff PLC.

In this mode, 4 receive and transmit PDO's are enabled from startup and are configured as PDO 1-4.

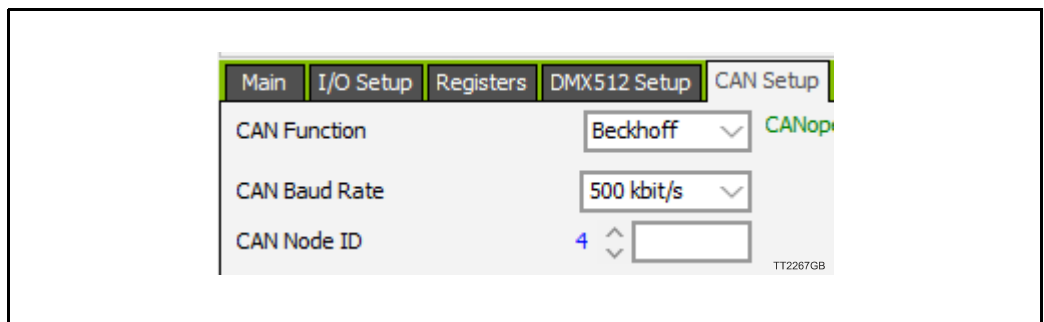
COB\_ID = 0x800000xxx: NOT ENABLED

COB\_ID = 0x000000xxx: ENABLED

### 11.4.12 PDO setup in Beckhoff mode

Normally each selected PDO needs to be enabled after power up and initialization but in Beckhoff mode PDO 1-4 is automatically enabled at power up.

To setup and use the Beckhoff mode, enable the Beckhoff support from MacTalk and press the Save in flash -button.



## 11.4 Objects in the DS301 standard

Option

### 11.4.13 Beckhoff receive PDO setup

The table below shows default values of the COB-ID:

PDO	Sub-index	Type	Description	Default	Access type
1	1	Receive	COB-ID	Nodeid+0x00000200	r/w
	1	Transmit	COB-ID	Nodeid+0x00000180	r/w
2	1	Receive	COB-ID	Nodeid+0x00000300	r/w
	1	Transmit	COB-ID	Nodeid+0x00000280	r/w
3	1	Receive	COB-ID	Nodeid+0x00000400	r/w
	1	Transmit	COB-ID	Nodeid+0x00000380	r/w
4	1	Receive	COB-ID	Nodeid+0x00000500	r/w
	1	Transmit	COB-ID	Nodeid+0x00000480	r/w

Receive PDO 1

Byte	0	1	2	3	4	5	6	7
Data	P_SOLL				V_SOLL		A_SOLL	
Object	2012h, sub 03h				2014h, sub 05h		2014h, sub 06h	

Receive PDO 2:

With this PDO it is possible to update the running current and operating mode.

Byte	0	1	2	3	4	5	6	7
Data	RUN_CURRENT		MODE_REG		Available			
Object	2014h, sub 07h		2014h, sub 02h					

Receive PDO 3:

This PDO can be used to issue a Motor command.

Byte	0	1	2	3	4	5	6	7
Data	Motor Command		Available					
Object	2014h, sub 18h							

Receive PDO 4:

This PDO updates the outputs.

Byte	0	1	2	3	4	5	6	7
Data	Output data		Available					
Object	2014h, sub 13h							

Tables shown above are default mapping.

The mapping can however also be changed dynamically. See [Dynamic Mapping](#), page 296

## 11.4 Objects in the DS301 standard

Option

### 11.4.14 Beckhoff transmit PDO setup

Tables shown below are default mapping. The mapping can however also be changed dynamically. See [Dynamic Mapping](#), page 296

Transmit PDO 1:

With this PDO the actual position can be read.

Byte	0	1	2	3	4	5	6	7
Data	P_IST				V_IST		Motor error	
Object	2012h, sub 0Ah				2014h, sub 0Ch		2014h, sub 23h	

Transmit PDO 2:

With this PDO the value of the analogue inputs 1-4 can be read.

Byte	0	1	2	3	4	5	6	7
Data	ANALOGUE1		ANALOGUE2		ANALOGUE3		ANALOGUE4	
Object	2014h, sub 59h		2014h, sub 5Ah		2014h, sub 5Bh		2014h, sub 5Ch	

Transmit PDO 3:

With this PDO the value of the analogue inputs 4-8 can be read.

Byte	0	1	2	3	4	5	6	7
Data	ANALOGUE5		ANALOGUE6		ANALOGUE7		ANALOGUE8	
Object	2014h, sub 5Dh		2014h, sub 5Eh		2014h, sub 5Fh		2014h, sub 60h	

Transmit PDO 4:

With this PDO the actual velocity can be read.

Byte	0	1	2	3	4	5	6	7
Data	V_IST		ERR_BITS		Lower word Modbus master encoder values (Not documented)			
Object	2014h, sub 0Ch		2014h, sub 23h		2014h, sub A1h			

## 11.5 Objects in the DSP-402 standard

Option

### 11.5.1 DSP-402 Support

#### Introduction

The MISxxx motors and SMC66/SMC85 supports the DSP-402 standard from CiA (<http://www.can-cia.com/>).

Please refer to this standard for details of the functions.

The DSP-402 is only a standard proposal and might be changed in the future. JVL therefore reserves the right to change future firmware versions to conform to new versions of the standard.

Not all of the functionality described in DSP-402 is supported, but all mandatory functions are supported.

The following operation modes are supported:

- Profile position mode
- Velocity mode
- Profile torque mode
- Homing mode
- Interpolated position mode
- Cyclic sync position mode
- Cyclic sync velocity mode
- Cyclic sync torque mode

#### Preconditions

The start mode of the motor must be set to passive.

No power up zero searches must be selected.

When using the DSP-402 mode, manipulating parameters with object 2012h or 2014h can corrupt the behaviour of the DSP-402 functions. Also be aware that manipulating parameters in MacTalk should be avoided when using DSP-402.

## 11.5 Objects in the DSP-402 standard

Option

### Supported objects

The following table gives the additional object dictionary defined for DSP-402 support.

Name	Index (hex)	Sub index	Data type	Read only	PDO	Units	Stored Flash
abort_connection_option_code	6007h						
error_code	603Fh						
motor_type	6402h		RO	STRING			
motor_catalogue_number	6403h		RO	STRING			
motor_manufacturer	6404h		RO	STRING			
http_motor_catalog_address	6405h		RO	STRING			
supported_drive_modes	6502h		RO	UINT32			
drive_catalogue_number	6503h		RO	STRING			
drive_manufacturer	6504h		RO	STRING			
http_drive_catalog_address	6505h		RO	STRING			
digital_inputs	60FDh		RO	UINT32	TPDO		
digital_outputs	60FEh		RO	UINT8			
physical_outputs		1	RW	UINT32	RPDO		
bitmask		2	RW	UINT32	RPDO		
controlword	6040h		RW	UINT16	PDO		
statusword	6041h		RO	UINT16	TPDO		
quick_stop_option_code	605Ah		RW	INT16			
modes_of_operation	6060h		RW	INT8	PDO		
modes_of_operation_display	6061h		RO	INT8	TPDO		
position_encoder_resolution	608Fh		RO	UINT8			
encoder_increments		1	RO	UINT32		Cts	
motor_revolutions		2	RO	UINT32		revolutions	
gear_ratio	6091h		RO	UINT8			
motor_revolutions		1	RW	UINT32		Numerator	*
shaft_revolutions		2	RW	UINT32		Divisor	*
feed_constant	6092h		RO	UINT8			
feed		1	RW	UINT32		Numerator	*
shaft_revolutions		2	RW	UINT32		Divisor	*
velocity_encoder_factor	6094h		RO	UINT8			
numerator		1	RW	UINT32		Numerator	*
divisor		2	RW	UINT32		Divisor	*
acceleration_factor	6097h		RO	UINT8			
numerator		1	RW	UINT32		Numerator	*
divisor		2	RW	UINT32		Divisor	*

## 11.5 Objects in the DSP-402 standard

Option

Name	Index (hex)	Sub index	Data type	Read only	PDO	Units	Stored Flash
polarity	607Eh						
target_position	607Ah		RW	INT32	TPDO	Pos	
software_position_limit	607Dh		RO	UINT8			
min_position_limit		1	RW	INT32		Pos	
max_position_limit		2	RW	INT32		Pos	
max_motor_speed	6080h		RW	UINT32	PDO	Vel	
profile_velocity	6081h		RW	UINT32	PDO	Vel	
profile_acceleration	6083h		RW	UINT32	PDO	Acc	
profile_deceleration	6084h		RW	UINT32	PDO	Acc	
quick_stop_deceleration	6085h		RW	UINT32		Acc	
home_offset	607Ch		RW	INT32	PDO	Pos	
homing_method	6098h		RW	INT8			
homing_speeds	6099h		RO	UINT8			
speed_during_search_for_switch		1	RW	INT32	PDO	Vel	
speed_during_search_for_zero		2	RW	INT32	PDO	Vel	
homing_acceleration	609Ah		RW	INT32	PDO	Acc	
position_actual_value	6064h		RO	INT32	TPO	Pos	
position_window	6067h		RW	UINT32		Pos	
interpolation_data_record	60C1h		RO	UINT8			
ip_data_position		1	RW	INT32	PDO	Pos	
ip_data_postype		2	RW	UINT8			
interpolation_time_period	60C2h		RO	UINT8			
ip_time_index		1	RW	INT8		Seconds x factor	
ip_time_index		2	RW	INT8		Seconds	
velocity_sensor_actual_value	6069h		RO	INT32	TPDO	Rpm/100	
velocity_demand_value	606Bh		RO	INT32		Rpm/100	
velocity_actual_value	606Ch		RO	INT32	TPDO	Vel	
target_velocity	60FFh		RW	INT32	PDO	Vel	
target_torque	6071h		RW	UINT16	PDO	Rated/1000	
max_torque	6072h		RW	UINT16	PDO	Rated/1000	
torque_demand_value	6074h		RW	UINT16	PDO	Rated/1000	
torque_actual_value	6077h		RW	UINT16	PDO	Rated/1000	
torque_slope	6087h		RW	UINT16	PDO	Rated/1000	
torque_profile_type	6088h		RW	UINT16			
DeviceType	67FFh		RO	USINT32			

## 11.5 Objects in the DSP-402 standard

Option

### 11.5.2 Factors

#### Position factor

The position factor is the relation between the user unit and the internal position unit (steps).

The position factor is automatically calculated when the feed constant (Object 6092h) and gear ratio (Object 6091h) are set.

Example:

A MIS232 Motor with a 3.5:1 gear box is connected to a belt drive.

The diameter of the drive wheel is 12.4 cm.

The unit of position is required to be in millimetres.

The perimeter of the drive wheel is 389.56mm ( $124\text{mm} \cdot \pi$ )

The parameters should be set as follows:

Object	Name	Value
6091 <sub>h</sub> subindex 1	Gear ratio - Motor revolutions	35
6091 <sub>h</sub> subindex 2	Gear ratio - Shaft revolutions	10
6092 <sub>h</sub> subindex 1	Feed constant - Feed	38956
6092 <sub>h</sub> subindex 2	Feed constant - Shaft revolutions	100

## 11.5 Objects in the DSP-402 standard Option

### Velocity encoder factor

This factor is used to convert the user unit into the internal unit (RPM).  
The factor is adjusted with the object 6094h.

Example 1:

An MIS232 has 1600 counts/revolution.

We want the user unit of velocity to be in RPM. This is the same as the internal unit.

The parameters should be set as follows:

Object	Name	Value
6094 <sub>h</sub> subindex 1	Velocity encoder factor - Numerator	1600
6094 <sub>h</sub> subindex 2	Velocity encoder factor – Divisor	1600

Example 2:

We have an MIS232 that uses RPM as the internal velocity and the same belt drive as in the above Position factor example.

We want the user unit of velocity to be in mm/s.

The parameters should be set as follows:

Object	Name	Calculated value	Value
6094 <sub>h</sub> subindex 1	Velocity encoder factor - Numerator	$(60 \times 3.5) / 389.56 = 0.53907$	53907
6094 <sub>h</sub> subindex 2	Velocity encoder factor – Divisor	1	100000

### Acceleration factor

This factor is used to convert the user unit into the internal unit (9.54 RPM/s).  
The factor is adjusted with the object 6097h.

Example 1:

We have an MIS232 with 1600 counts/revolution.

We want the user unit of acceleration to be in RPM/s.

The parameters should be set as follows:

Object	Name	Value
6097 <sub>h</sub> subindex 1	Acceleration encoder factor - Numerator	100
6097 <sub>h</sub> subindex 2	Acceleration encoder factor – Divisor	954

## 11.5 Objects in the DSP-402 standard

Option

Example 2:

We have an MIS232 with 1600 counts/revolution and the same belt drive as in the above Position factor example. We want the user unit of acceleration to be in mm/s<sup>2</sup>.

The parameters should be set as follows:

Object	Name	Calculated value	Value
6097 <sub>n</sub> subindex 1	Acceleration factor- Numerator	$(3.5 \cdot 60) / 389.56 = 0.53907$	53907
6097 <sub>n</sub> subindex 2	Acceleration factor - Divisor	9.54	954000

### 11.5.3 Changing operation mode

Change of operation mode is only possible when the operation mode is not enabled. There is one exception and that is when changing from Zero Search mode to profile position mode. This is possible when the Zero Search sequence is completed and can be done even though the operation mode is enabled.

### 11.5.4 Profile position mode

This mode can be used for positioning in which a move profile can be set up. The acceleration and maximum velocity can be programmed.

In this mode both absolute and relative movement is supported. This is selected using bit 6 (absolute/relative) in the status word. It is also possible to select different movement modes. This is done with bit 5 (change set immediately) in the status word. When this bit is 0 and a move is in progress, the new set-point is accepted, but the new set-point and profile are not activated until the previous movement is finished. When this bit is 1, the new set-point is activated instantly and the motor will move to the new position with the new profile parameters.

Objects affected.

Object	Index	Sub index	Description
Target_position	607Ah	0	Position unit (Default counts)
Profile_velocity	6081h	0	RPM/100
Profile_acceleration	6083h	0	RPM/100/sec
Profile_deceleration	6084h	0	RPM/100/sec. 0 = same as acceleration
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode. Profile position mode = 1

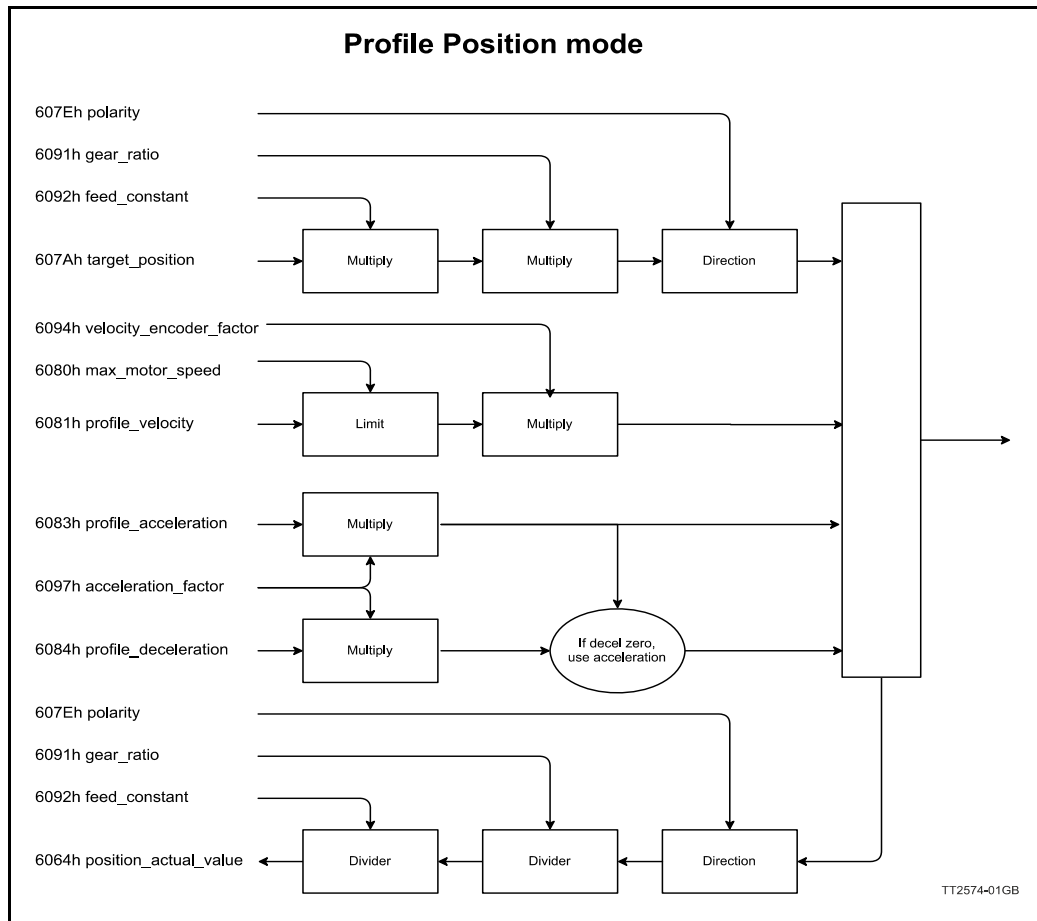
Profile specific bits

Control word bits		
Bit	Value	Name
4	10h	New setpoint
5	20h	Change Set Immediately
6	40h	Relative
8	100h	Halt
Status word		
Bit	Value	Name
10	400h	In_position
12	1000h	Setpoint acknowledge

## 11.5 Objects in the DSP-402 standard

Option

The diagram below shows how objects affect profile position mode



This mode can be used for positioning in which a move profile can be set up.

The acceleration and maximum velocity can be programmed.

In this mode both absolute and relative movement is supported.

This is selected using bit 6 (absolute/relative) in the control word.

It is also possible to select different movement modes. This is done with bit 5 (change set immediately) in the control word.

When this bit is 0 and a move is in progress, the new set-point is accepted, but the new set-point and profile are not activated until the previous movement is finished.

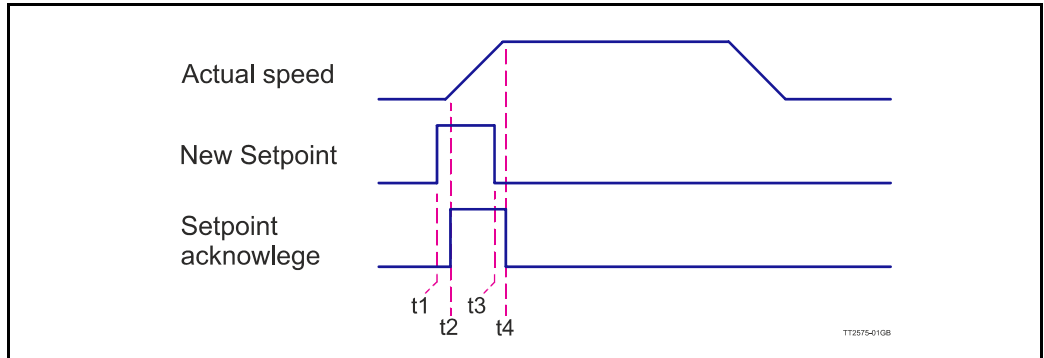
See sequential profiles. When this bit is 1, the new set-point is activated instantly and the motor will move to the new position with the new profile parameters.

See overlapping profiles.

## 11.5 Objects in the DSP-402 standard

Option

### Single profile

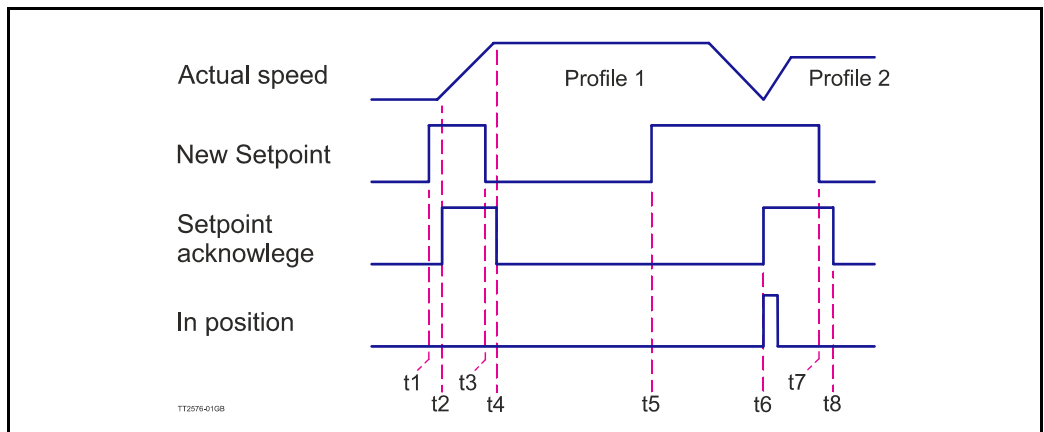


Prepare the target position, velocity, and acceleration.

- t1 Set the “setpoint” bit when ready
- t2 The motor responds with setpoint acknowledge and start the move.
- t3 Reset the “setpoint” bit when the setpoint been acknowledged.
- t4 The setpoint acknowledge is reset when setpoint is reset.

The motor is ready for next profile.

### Sequential profiles



t1 – t4 same as single profile

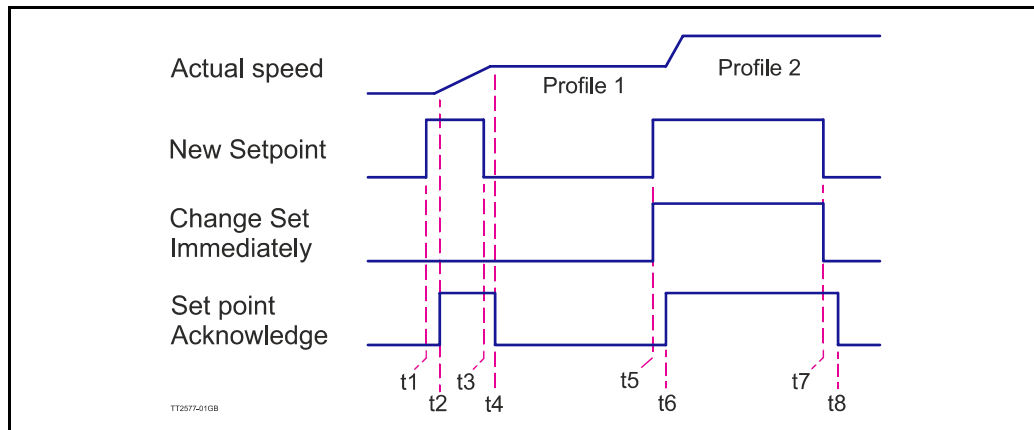
Prepare profile 2 target position, velocity, and acceleration.

- t5 Set the “setpoint” bit when done with profile settings
- t6 The motor responds with setpoint acknowledge when profile 1 position reached. Profile 2 movement is started.
- t7 Reset the “setpoint” bit when the setpoint been acknowledged.
- t8 The setpoint acknowledge is reset when setpoint is reset.

## 11.5 Objects in the DSP-402 standard

Option

### Overlapping profiles



t1 – t4 same as single profile

Prepare profile 2 target position, velocity, and acceleration.

t5 Set the “setpoint” and the “Change immediately” bit when done with profile settings.

t6 The motor responds with setpoint acknowledge. Profile 1 is aborted and Profile 2 movement is started.

t7 Reset the “setpoint” bit when the setpoint been acknowledged.

t8 The setpoint acknowledge is reset when setpoint is reset.

### 11.5.5 Velocity mode

In this mode the motor runs at a selected velocity. A new velocity can be selected and the motor will then accelerate/decelerate to this velocity.

The maximum slippage error is not supported in this mode.

Objects affected:

Object	Index	Sub index	Description
Target_velocity	60FFh	0	rpm/100
Profile_acceleration	6083h	0	rpm/100/sec
Profile_deceleration	6084h	0	rpm/100/sec. 0 = same as acceleration
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode. Velocity mode = 3

Profile specific bits

Control word bits		
Bit	Value	Name
8	100h	Halt
Status word		
Bit	Value	Name
10	400h	Velocity reached
12	1000h	Velocity zero

## 11.5 Objects in the DSP-402 standard

Option

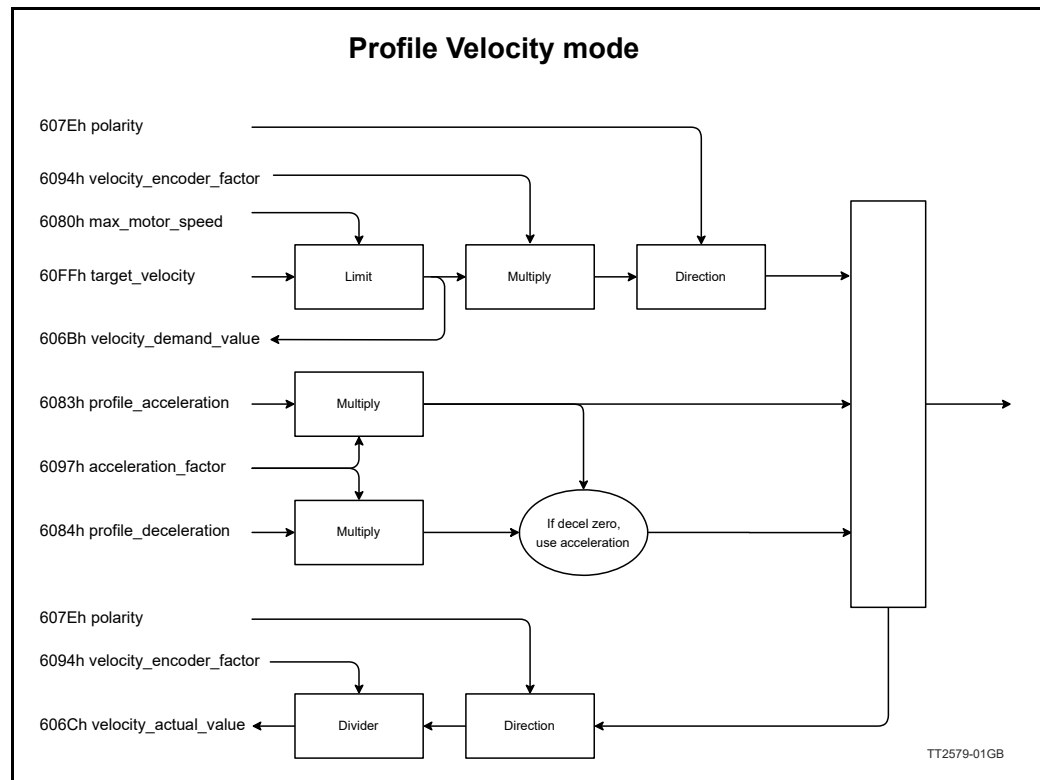
In this mode the motor runs at a selected velocity.

A new velocity can set with target velocity object and the motor will then accelerate/decelerate to this velocity with the profile acceleration.

The motor start to accelerate towards target velocity as soon as the operation state is operational, unless the Halt bit is set.

The Halt bit will decelerate to zero velocity with the profile deceleration.

The diagram below shows how objects affect profile velocity mode



### 11.5.6 Homing mode

Using this mode, different homing search sequences can be initiated. The standard Homing Search modes from 1-34 are supported. Before starting the Homing Search, the inputs must be configured properly using MacTalk or parameters 125, 129, 130, 132.

Objects affected.

Object	Index	Sub index	Description
Home_offset	607Ch	0	Position unit (Default counts)
Homing_method	6098h	0	See homing method table
Speed_during_search_for_switch	6099h	1	rpm/100
Speed_during_search_for_zero	6099h	2	rpm/100
Homing_acceleration	609Ah	0	rpm/100/sec
Homing torque	2100h	0	Max current / 2047. 2047 = same as max current
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode. Homing mode = 6

## 11.5 Objects in the DSP-402 standard

Option

### Profile specific bits

Control word bits		
Bit	Value	Name
4	10h	Start homing
8	100h	Halt
Status word		
Bit	Value	Name
12	1000h	Homing complete successfully
13	2000h	Homing complete with errors

### Supported homing modes

Mode	Index If supported	Mode
-4	Yes	Torque mode positive direction
-3	Yes	Torque mode negative direction
-2	No	Torque mode positive direction
-1	No	Torque mode negative direction
1	Yes	Negative limit switch
2	Yes	Positive limit switch
3	Yes	Rising edge homing switch. Negative direction
4	Yes	Rising edge homing switch. Positive direction
5	Yes	Falling edge homing switch. Negative direction
6	Yes	Falling edge homing switch. Positive direction
17	No	Same as 1 without index
18	No	Same as 2 without index
19	No	Same as 3 without index
20	No	Same as 4 without index
21	No	Same as 5 without index
22	No	Same as 6 without index
35	No	No movement

### Homing signals

Set the mode of operation 6060h to value 6 for homing mode.

Enable the motor, by setting the state to Operation Enable.

Set the profile for the homing sequence object 6098h e.g. 4 for Rising edge reference switch, in positive direction.

Set the speed for homing towards switch in object 609901h and the crawl speed away from the switch in object 609902h. Both objects are in RPM/100.

Acceleration is set with object 609Ah in RPM/second.

Homing offset object 607Ch, is the position of the homing switch. The motor will get this position when leaving the switch in the homing sequence.

Then the Activate homing (bit 4) is set in the Control word 6040h.

When either the Homing completed (bit 10) or the Homing Error (bit 11) is set in the Status word object 6041h, the homing sequence is done.

If homing ended with the Homing Error bit, the motor will go into error state when the Activate homing bit is cleared from the control word. It will be necessary clear this error before the motor can be activated again.

An error occur unrelated to homing, the homing mode cancelled and motor transits to Fault state.

### Reference switch settings

Before using the one of the homing modes that require a switch, the input must be configured according to the homing mode.

Homing on reference switch:

Homing mode Sensor input should be set to the input where the switch is connected.

Input 4 is default. The homing switch can be selected from Mactalk.

## 11.5 Objects in the DSP-402 standard

Option

### Homing on limit switch

Make sure that the homing switch sensor input is set to the same input as the limit switch that is used for reference.

In the figure below, the negative limit switch is set to input 4. The sensor input is also set to input 4. This setting will allow the Homing mode I to be used.

Remember to set the right active level on the limit switch input, depending on the use of a Normally Open or a Normally Closed type switch.

When using one of the limit switches as homing switch, the limit switch will not raise an error when activated during home sequence.

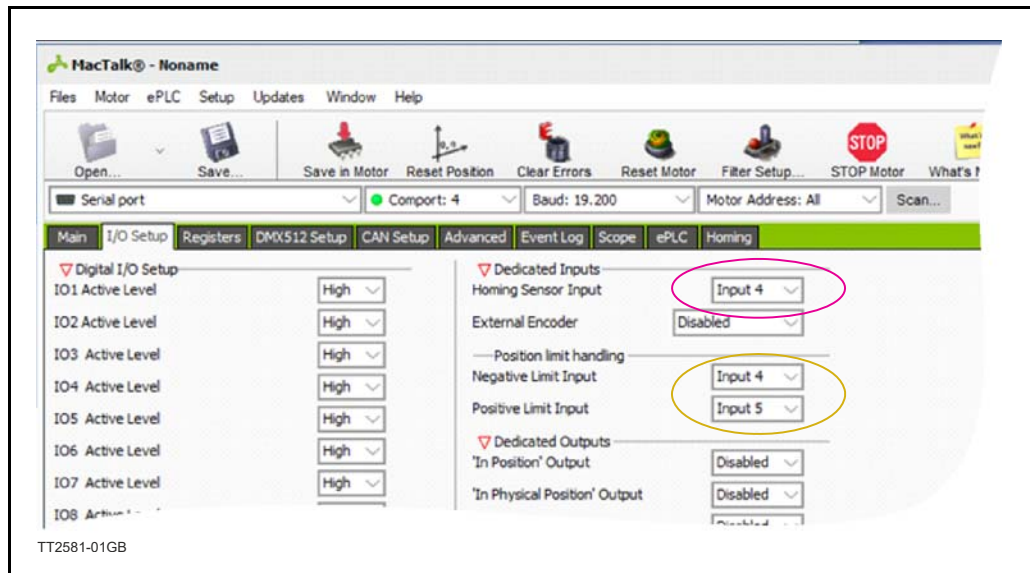
### Torque homing

Torque homing with the MIS motor is actually current triggered homing.

Set the object 2100h to current that should be considered the reference point, when reached.

Object 2100h accept the values from 0 to 2047.

The value 2047 is considered the max current in object 6072h.



### 11.5.7 Profile Torque mode

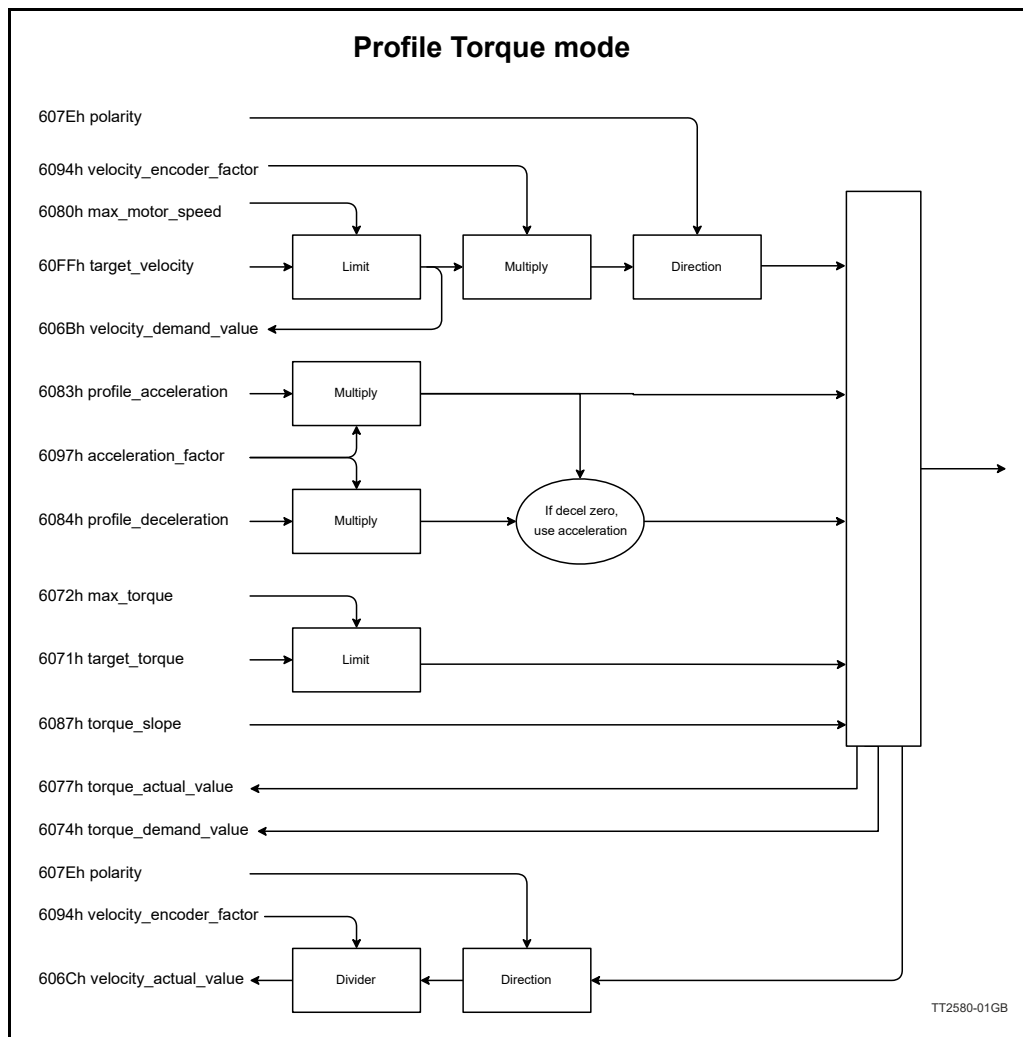
Object affected:

Object	Index	Sub index	Description
Target velocity	60FFh	0	Velocity in rpm/100
Acceleration	6083h	0	rpm/100/sec
Deceleration	6084h	0	rpm/100/sec
Target_torque	6071h	0	(Current for stepper motor)Rated current / 1000
Max_torque	6072h	0	(Current for stepper motor)Rated current / 1000
Torque slope	6087h	0	(Current for stepper motor)Rated current / 1000/ sec
Torque Profile	6088h	0	0 = Linear Profile
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode. Profile velocity mode = 3

## 11.5 Objects in the DSP-402 standard

Option

The diagram below shows how objects affect profile velocity mode



The stepper motor does not have a linear torque curve. Its is affected by the input voltage and velocity of the motor.

Therefore, the stepper motor uses current to control the Power.

The Profile Torque mode behaves like a servo motor but uses current instead of torque to regulate output power.

The torque mode is working like the velocity mode. Only in torque mode, is it possible to control the torque with a slope. This enables slowly applying force the axel of the motor, and gradually apply the force requested to drive the motor.

The Velocity and direction of the axel is controlled with the target\_velocity objects, just like the Profile velocity mode.

Torque is in rated current/1000. Rated current is different for different models of MIS motors.

If both max\_torque and target\_torque is set to 1000, the motor will allow the regulation to use maximum current to achieve target\_velocity. The maximum current for your motor can be read in the Data sheet.

## 11.5 Objects in the DSP-402 standard

Option

### 11.5.8 Interpolated position mode. IP MODE

Objects affected.

Object	Index	Sub index	Description
Interpolation position	60C1	1	Position input in position units (Default counts)
Interpolated pos type	60C1	2	Position input. 0 = Absolute, 1 = Relative
Interpolated time period	60C2	1	Time between sync in time factor units
Interpolated time index	60C2	2	Time units
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode Interpolated position mode = 7

Profile specific bits

Control word bits		
Bit	Value	Name
4	10h	ENABLE_IP_MODE
8	100h	Halt
Status word		
Bit	Value	Name
10	400h	IP_MODE_READY

This mode is used to control the motor in synchronized mode from a CANopen master. This can be used to synchronize multiple axis with an axis controller.

The position is sent to the motor in segments from 1ms to 20ms.

When the segment is sent to the motor, the next sync pulse will run the motor to the new segment position in the time set in object 60C2:01.

The segment time is controlled by the master with the sync message. A new segment position should be sent to the motor in each sync cycle.

The motor will behave like there is unlimited acceleration.

#### Controlword:

To enable Ip mode position data, set bit 4. ENABLE\_IP\_MODE

#### Status word:

When IP mode I ready for position data, bit 10 is set. (IP\_MODE\_READY)

Objects used for interpolation

#### 60C1:01 Interpolation position

Segment data.

If Interpolated data type = 0 the interpolated data is in Absolut position units.

The motor will attempt to reach that position in the time until next sync message.

If Interpolated data type = 1 the interpolated data is in relative position units.

Meaning that the motor will attempt to move that distance between two sync messages.

#### 60C1:02 Interpolated pos type

Select what format interpolated data is in

0: (default) interpolated data is in Absolut position units.

1: interpolated data is in relative position units.

#### 60C2:01 Interpolated time period

Time between sync messages from master.

Unit depends on interpolated time index object.

Default is 10ms.

## 11.5 Objects in the DSP-402 standard Option

### 60C2:02 interpolated time index

Supported value: -3

Factor for time segment. -3 means  $10^{-3}$  second = 1 milli second.

### Sample sequence

60C2:01 = Set time period

6060 = 7 (Set operation mode interpolated)

ENABLE\_IP\_MODE = 0 Controlword bit 4

Change operation state = operational

60C1:01 = 6064:00 Set interpolated data to current position.

Set Controlword bit 4 ENABLE\_IP\_MODE = 1

Wait for Statusword bit 10 IP\_MODE\_READY

60C1:01 next position

Sync message from master

60C1:01 next position

Sync message from master

Repeat position and Sync

### 11.5.9 Cyclic Synchronous position mode CSP

Objects affected.

Object	Index	Sub index	Description
Target position	607A	0	Default in counts
Interpolated time period	60C2	1	Time between sync in time factor units
Interpolated time index	60C2	2	Time units
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode. Cyclic sync position mode = 8

Profile specific bits

Control word bits		
Bit	Value	Name
4	10h	ENABLE_CSP_MODE
6	40h	Relative position
Status word		
Bit	Value	Name
10	400h	CSP_MODE_READY

CSP mode is used for synchronized control. On every write to target position object, the motor will attempt to reach the target in the time set up in the interpolation time object, with close to no ramp.

Ramp is generated from the master, by moving the motor in small steps in the beginning of a profile, and then increasing the step size in each time interval, thereby accelerating the motor.

## 11.5 Objects in the DSP-402 standard

Option

### 11.5.10 Cyclic Synchronous velocity Mode CSV

Objects affected.

Object	Index	Sub index	Description
Target_velocity	60FFh	0	rpm/100
Max_motor_speed	6080h	0	rpm/100
Polarity	607Eh	0	Bit 7 (0x40) reverse the velocity direction
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode. Cyclic sync velocity mode = 9

Profile specific bits

Control word bits		
Bit	Value	Name
8	100h	Halt
Status word		
Bit	Value	Name
10	400h	Velocity reached
12	1000h	Velocity zero

CSV mode is used for cyclic update. On every write to target velocity object, the motor will attempt to reach the target as fast as possible with close to vertical ramp. Ramp is generated from the master, by increasing the velocity in small steps, thereby controlling acceleration of the motor.

### 11.5.11 Cyclic Synchronous torque Mode CST

Objects affected.

Object	Index	Sub index	Description
target_torque	6071h	0	(Current for stepper motor)Rated current / 1000
max_torque	6072h	0	(Current for stepper motor)Rated current / 1000
max_motor_speed	6080h	0	rpm/100
polarity	607Eh	0	Bit 7 (0x40) reverse the velocity direction
Control word	6040h	0	Control Word
Status word	6041h	0	Status word
Modes of operation	6060h	0	DSP402 mode. Cyclic sync torque mode 10

The stepper motor does not have a linear torque curve.

Its is affected by the input voltage and velocity of the motor. Therefore, the stepper motor uses current to control the Power. The Profile Torque mode behaves like a servo motor but uses current instead of torque to regulate output power.

CST mode is used for cyclic update. In this mode, the motor will attempt to reach the max\_motor\_speed as fast as possible with the torque set in target torque.

The Velocity and direction of the axel is controlled with the max\_motor\_speed and ploarity objects.

Torque is in rated current/1000. Rated current is different for different models of MIS motors.

If both max\_torque and target\_torque is set to 1000, the motor will allow the regulation to use maximum current to achieve target\_velocity.

The maximum current for your motor can be read in the Data sheet.

## 11.5 Objects in the DSP-402 standard

---

Option

## 11.6 Running MIS as NC axis in TwinCAT Option

The MIS motor support the Beckhoff program TwinCAT NC axis.

The NC axis enables you to control the motor with PLC open standard.

NC axis support synchronized axis, with interpolation.

The following pages describes how to set up the MIS motor to run as NC axis in TwinCAT3 with Visual studio 2017, on a CX55 I 30 PLC with EL 675I CANopen master module.

Other variants of IDE or Hardware may variant in behaviour

This sample will guide you in the following:

- Setup the motor for CANopen
- Create a new TwincCAT project.
- Connect to your PLC hardware.
- Scan your Hardware.
- Setup the connected motor for NC axis support.

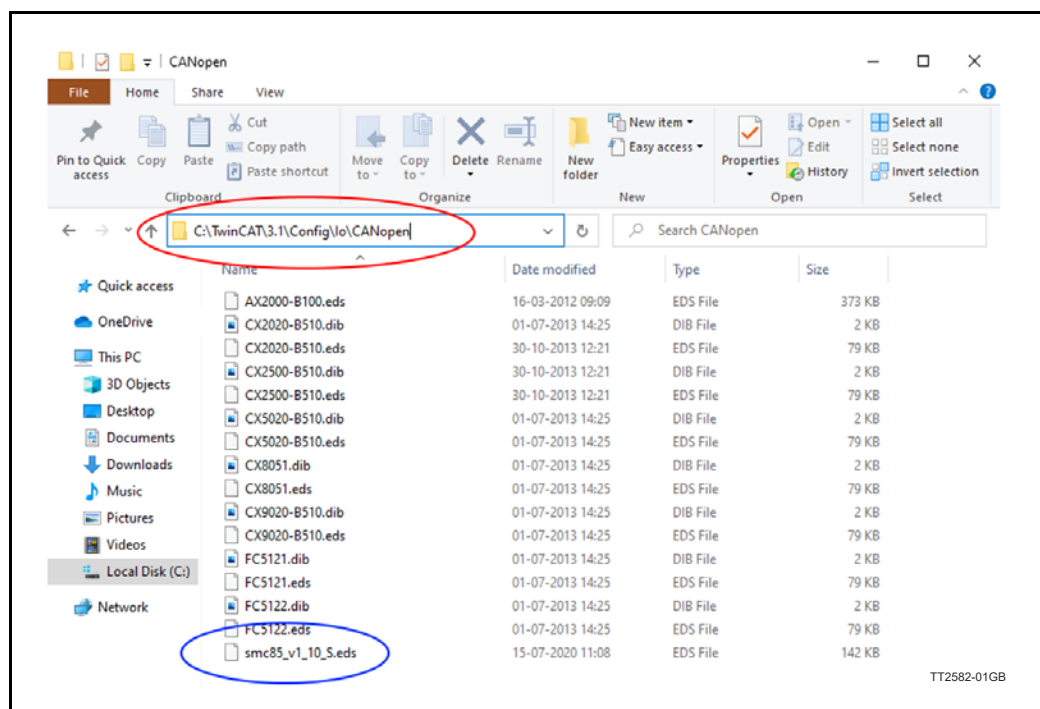
### 11.6.1 Copy EDS file to TwinCAT folder

Down load the EDS file for the MIS34 from the JVL homepage (down load menu).

The EDS file is an Electronic Data sheet, that describes the interface between the PLC and the motor. This enables the PLC to map the right registers in the motor.

# Open the TwinCAT folder.

# Copy the EDS file to this location. I am using EDS file version 1.10 in this sample.

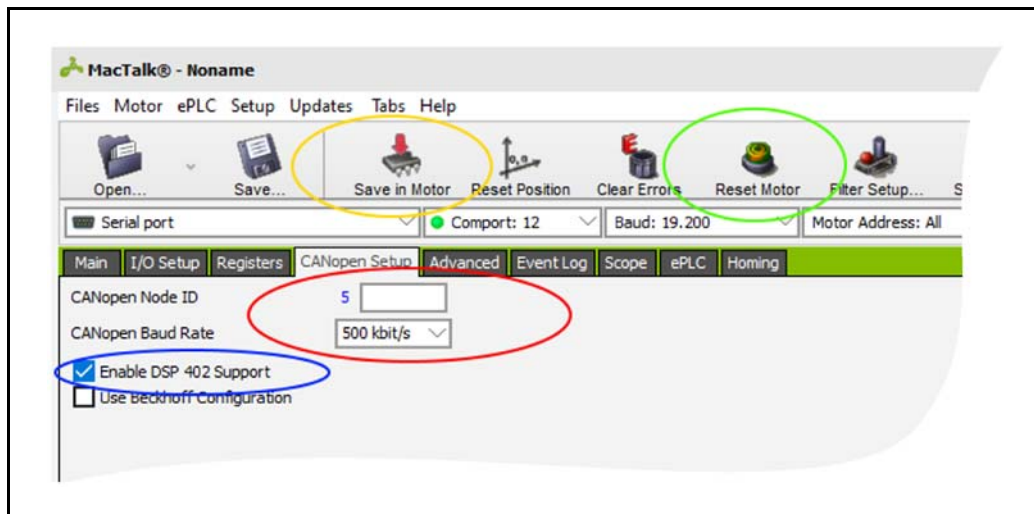


## 11.6 Running MIS as NC axis in TwinCAT

Option

### 11.6.2 Setup the motor

Connect to the MIS34x motor with Mactalk. Open the CANopen setup page.



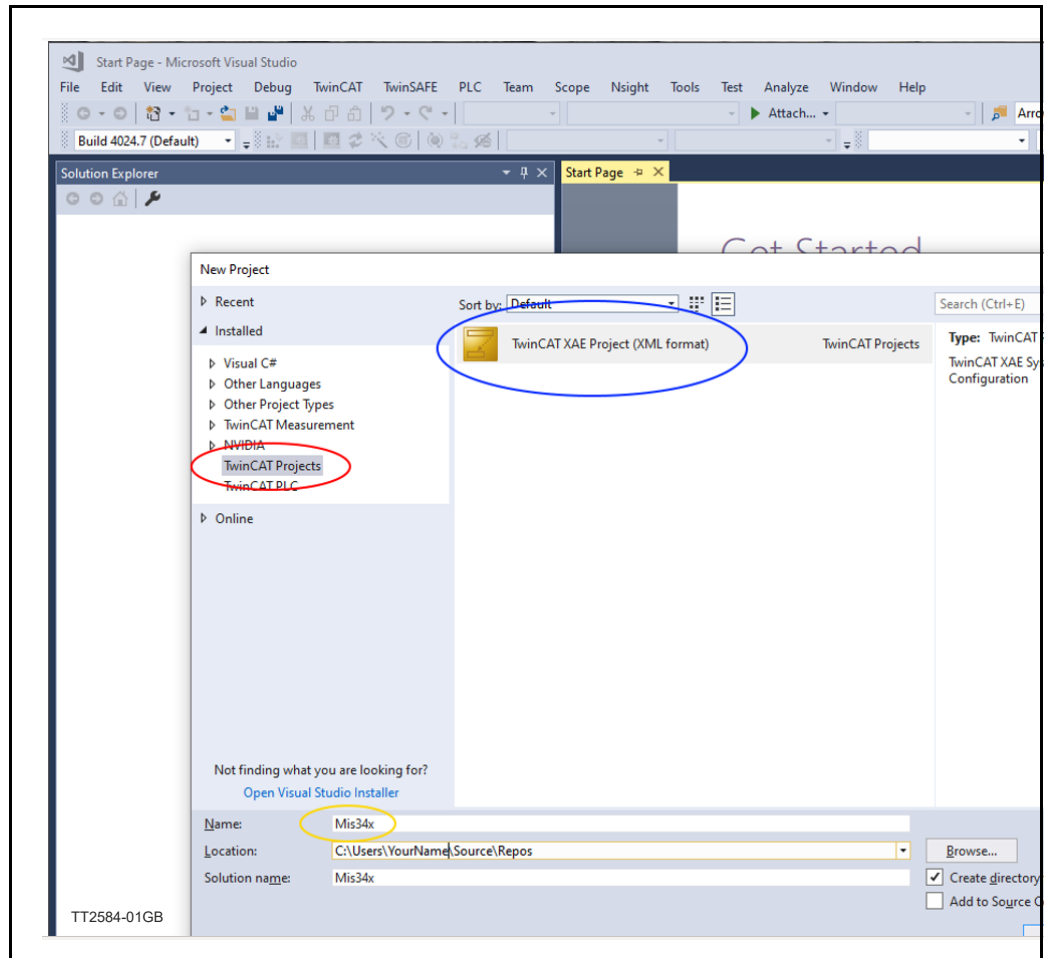
- # Choose you node ID and Baud rate. Node id 5 and 500kbits/second is default.
- # Make sure Enable DSP 402 support Is checked
- # Save the settings in the motor.
- # Reset the motor

## 11.6 Running MIS as NC axis in TwinCAT Option

### 11.6.3 Create new TwinCAT project

Open Visual studio 2017 with TwinCAT3 integration. If TwinCAT is not installed on the computer, the TwinCAT project will not be available.

The rest of this sample is done in Visual Studio. Click File->New project.

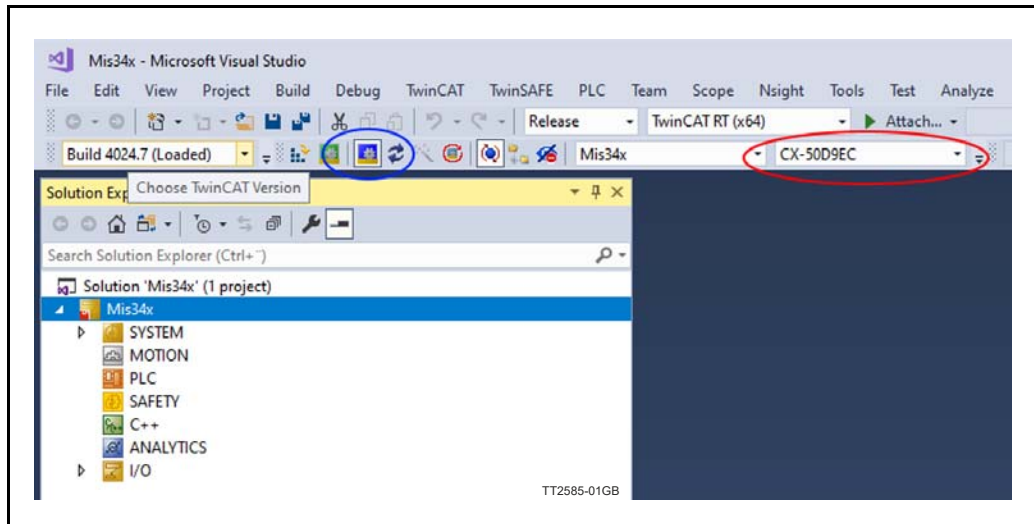


- # In the new project window, select TwinCAT project.
- # Select the twinCAT XAE project
- # Give your project a name and click “ok” button.

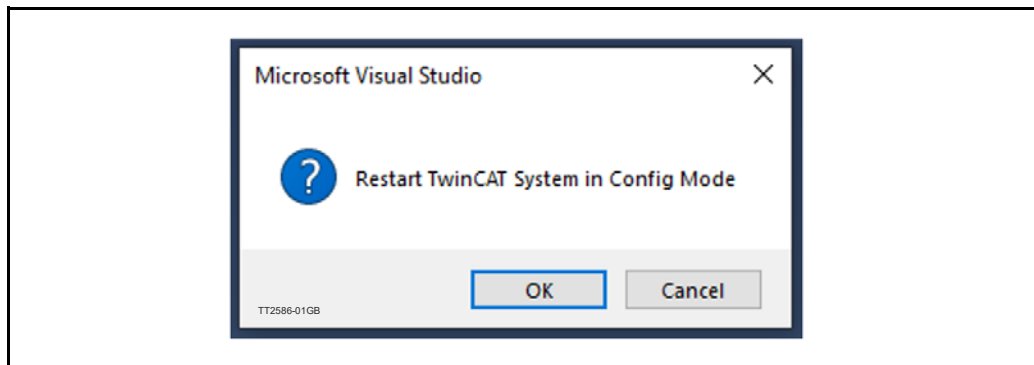
## 11.6 Running MIS as NC axis in TwinCAT Option

### 11.6.4 Connect to PLC in Config mode

Connect to PLC to scan connected hardware. This will also detect node attached to the Canbus. Make sure that all other nodes are disconnected, to get the same result as this sample.



- # Select the ADS connection for your PLC.
- # Click the Restart TwinCAT in config mode.



Click OK to allow TwinCAT to restart.

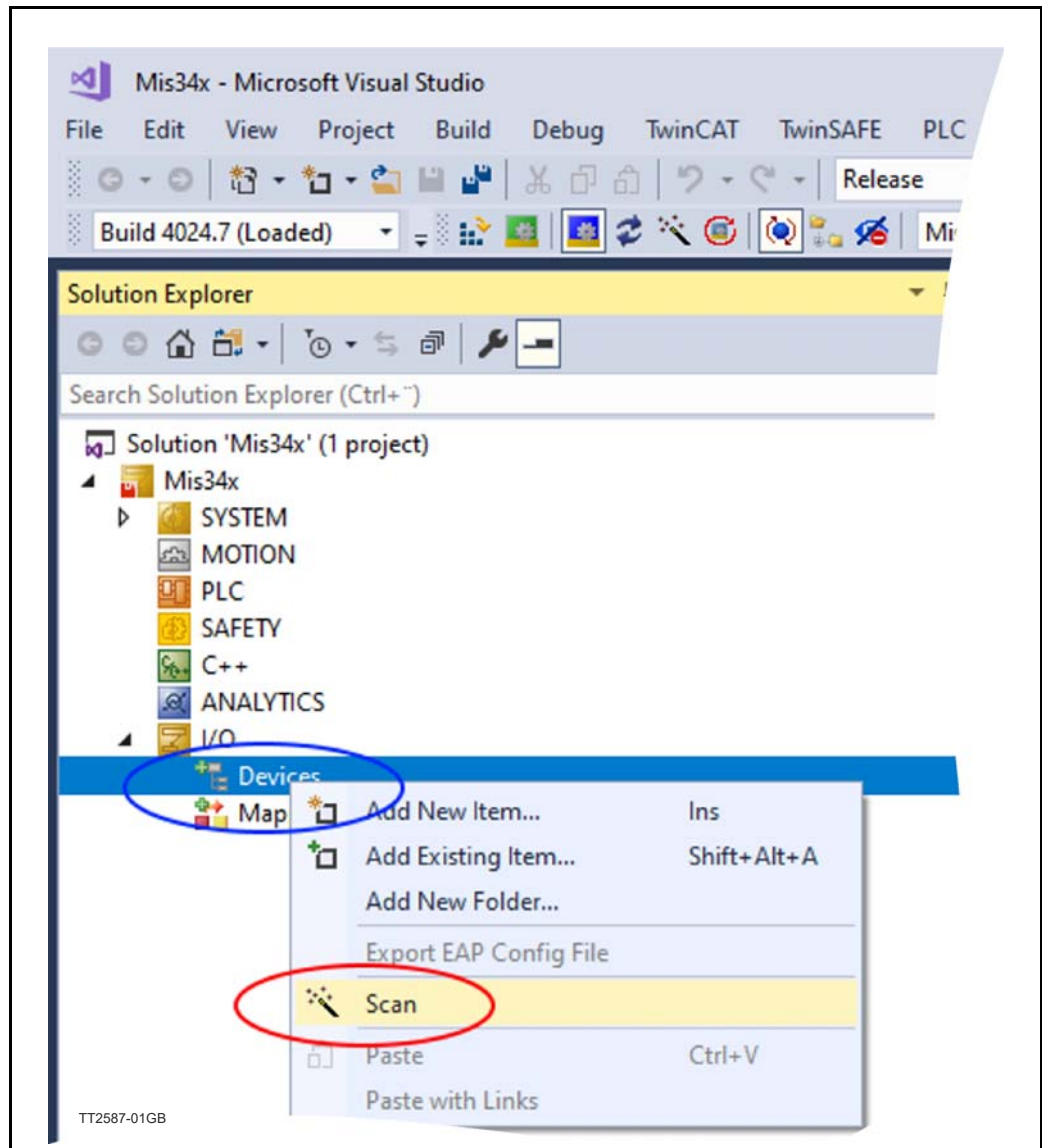
You should be connected to your PLC.

## 11.6 Running MIS as NC axis in TwinCAT

Option

### 11.6.5 Scan for hardware

Make sure your motor is connected to the Beckhoff CANopen master module and ready for communication with the master.



# Right click IO->Devices in your solution tree.

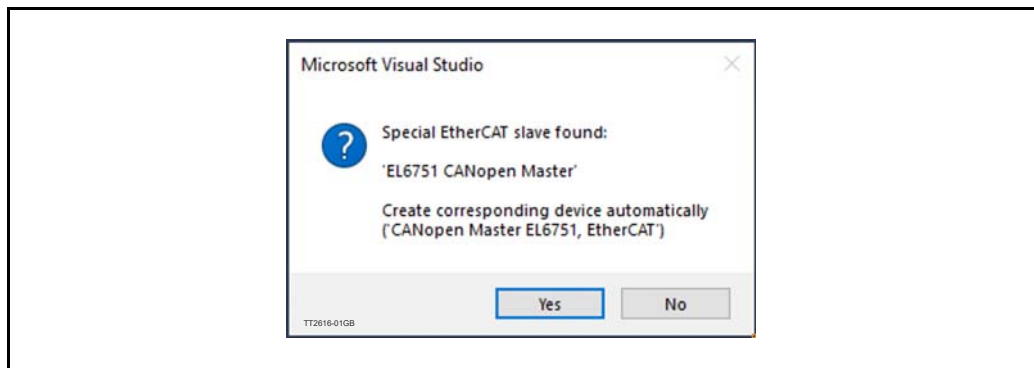
# Select Scan

This will find all hardware on your PLC. IO's, busses, etc.

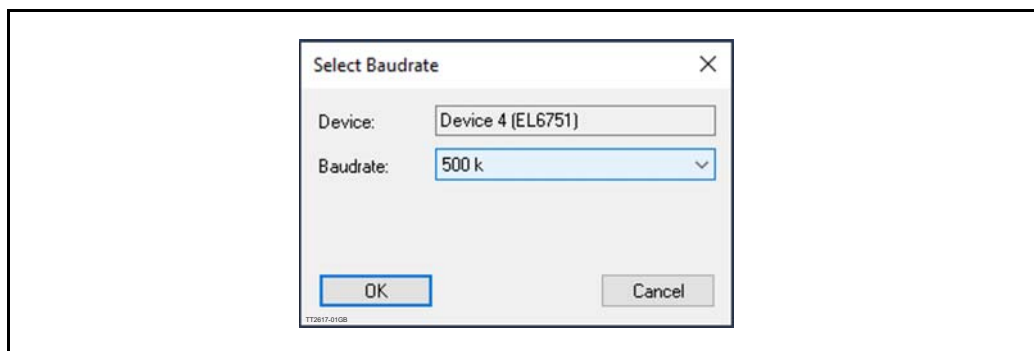
Sometime the Scan function does not find the CANopen master. In this case just scan the bus where the master is connected. In my case, the Device I Ethercat.

## 11.6 Running MIS as NC axis in TwinCAT

Option



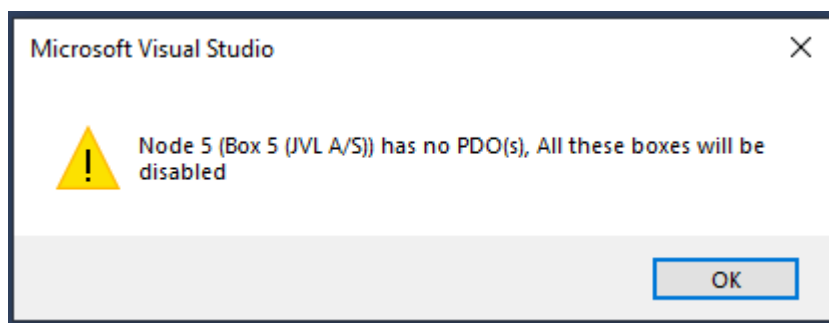
When the master is found, you will be prompted to create CANopen devices automatically. Click “Yes”. I am using the EL6751 CANopen Master module.



Select the baudrate you selected in macktalk, to scan your CANopen bus with and Click “OK”

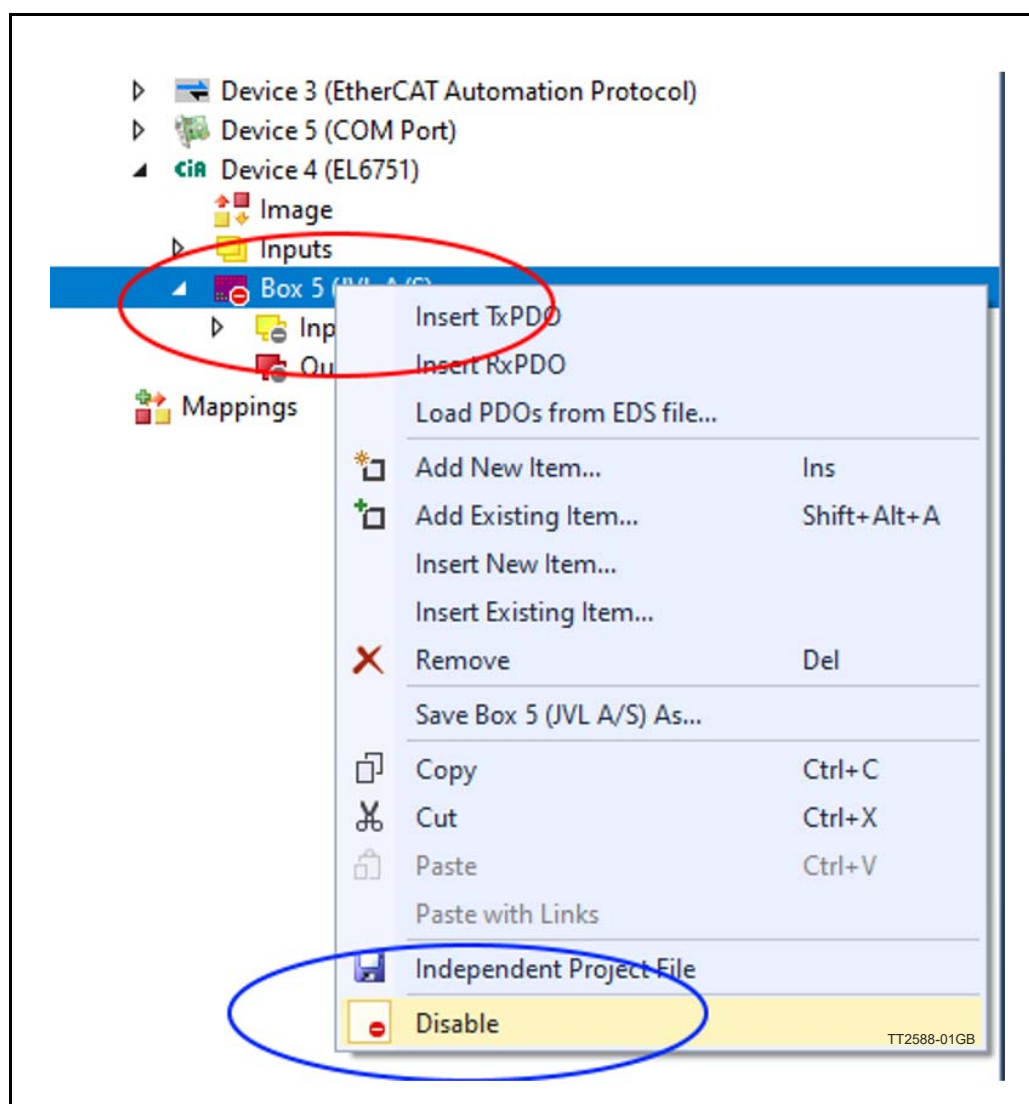
If the scan was able to detect the PDO automatically, skip to the next paragraph “Configure PDO mapping”.

If you get this message below, the PLC did not recognize the motor as the one in the EDS file.



Continue with enabling the node, and load the PDOS manually from file.

## 11.6 Running MIS as NC axis in TwinCAT Option



Find the Motor box under your CANopen master device.

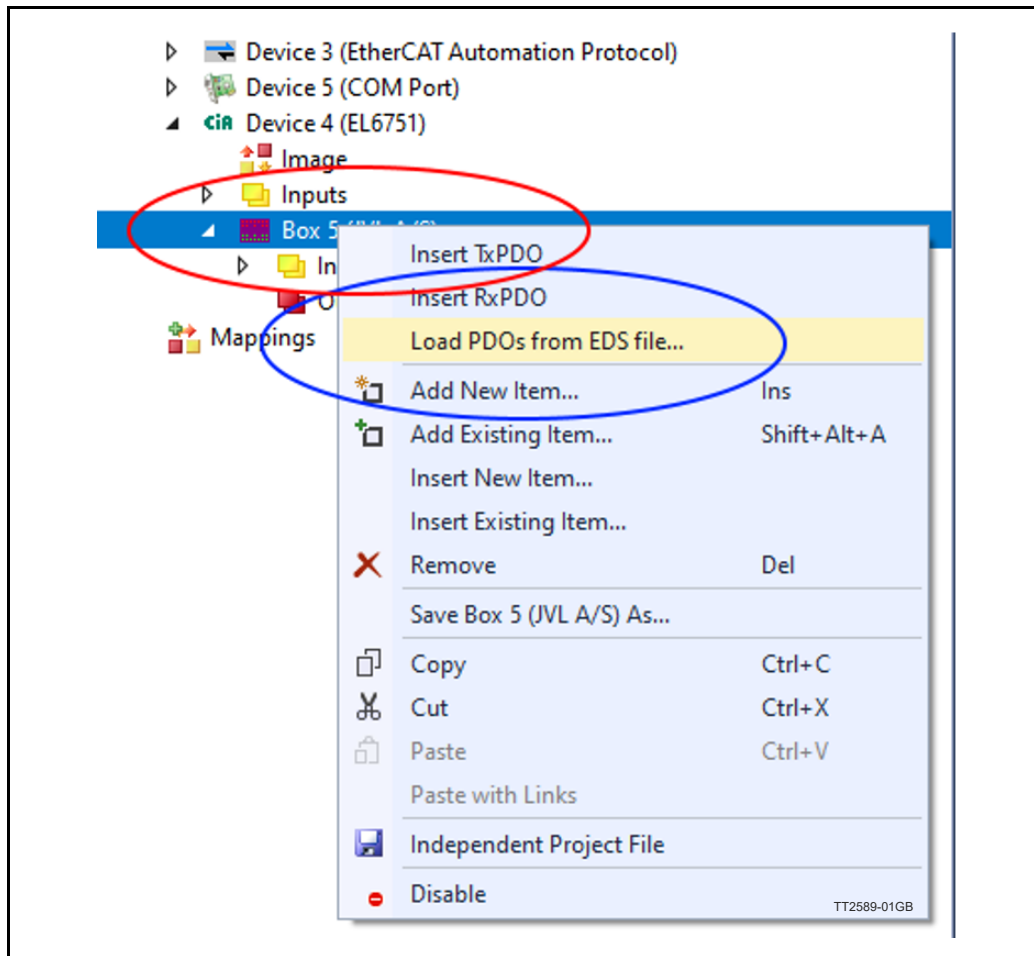
# Right click the motor Box.

# Click Disable, to re enable the node.

## 11.6 Running MIS as NC axis in TwinCAT

Option

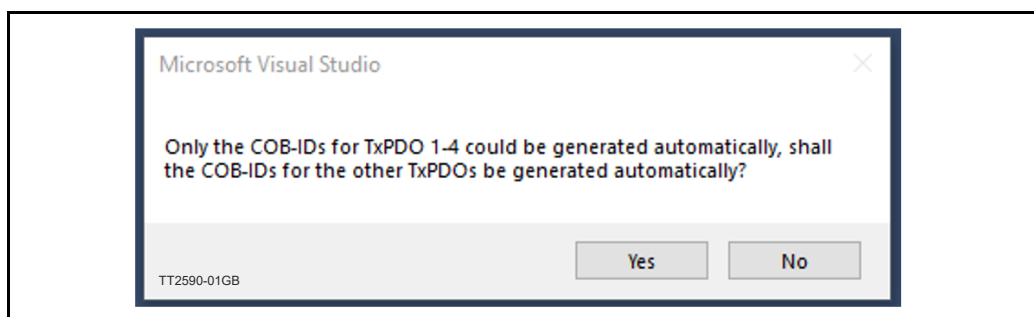
Load the PDOs from EDS file.



# Right click the motor Box.

# Select Load PDOs from EDS file.

Select the EDS file you copied to the EDS folder earlier.



TwinCAT will generate COB'ids for unused PDO's.  
Click "Yes" for both RX and TX PDO's. Unused PDO's are removed later in the guide, but allow automatically generated COBId's for now.

## 11.6 Running MIS as NC axis in TwinCAT

Option

### 11.6.6 Configure PDO mapping

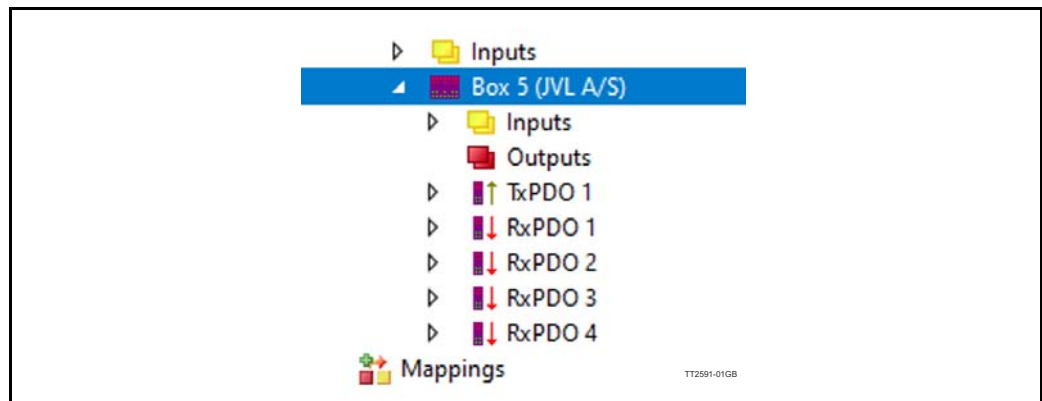
The PDO is the data send cyclic to and from the node.

We need to send (RX PDO) Controlword and requested position to the motor, and we need to receive (TX PDO) Status word and actual position from the motor. This is easy to setup when the EDS file does most of the work.

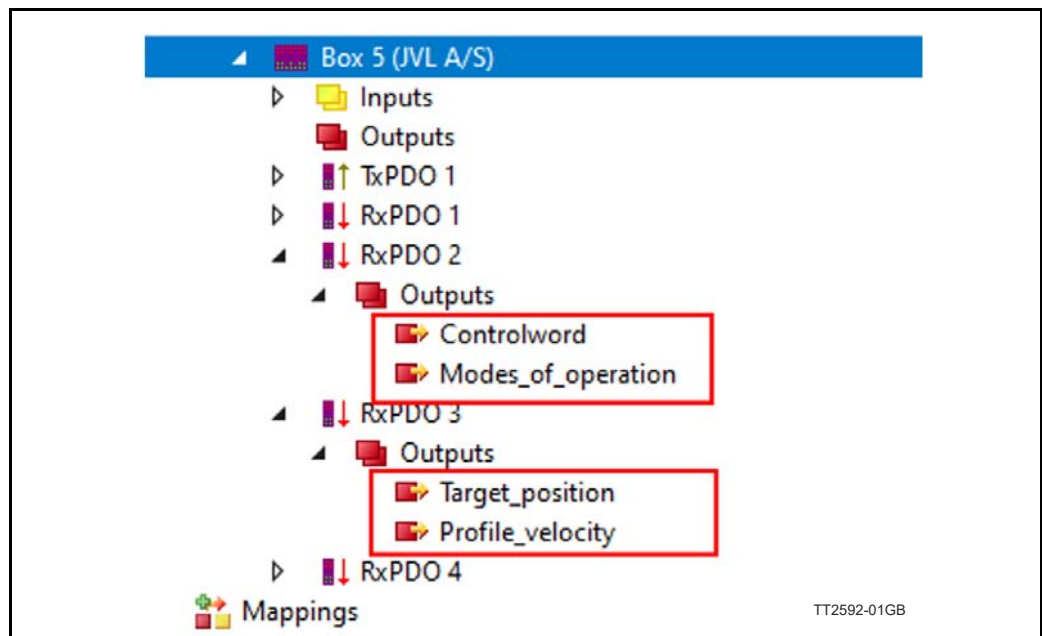
The scan can generate up to 24 RX PDOs, and 24 TX PDO's

We only need TXPDO1 and RXPDO1 and RXPDO4

To limit the communication to what is essential for NC axis, all PDO's larger than the ones used, can be deleted. Delete TXPDO2 to TXPDO24 and RXPDO5 to RXPDO24. You should end up with something like this:



We do not need RxPDO2 and RxPDO3, but we cannot delete them without destroying the mapping of RxPDO4, but we can remove the mapping inside them. Expand RxPDO2 and RxPDO3 by clicking on the white triangle.



Delete the mapped object marked with red squares.

Select them with mouse and use the "Delete" key to remove them.

## 11.6 Running MIS as NC axis in TwinCAT

Option

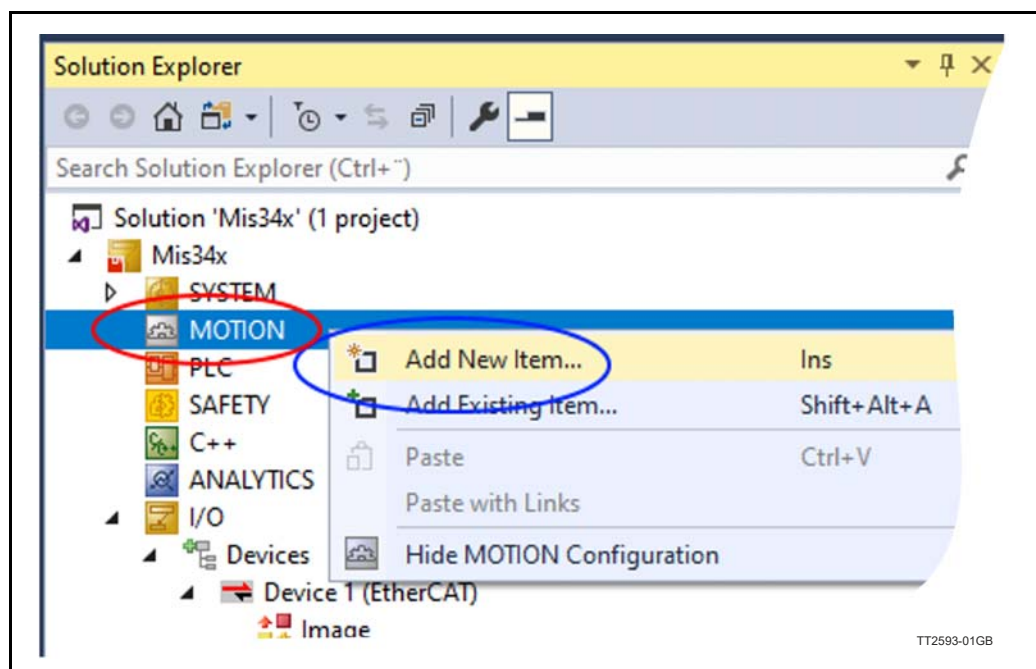
### 11.6.7 Create NC axis

The NC axis, is a PLC motion component that represent a virtual axis. You can use a NC axis without any hardware, and use it to simulate a moving axis.

In this case, we map the virtual axis to a physical motor. The motor will be a slave of the virtual axis. The NC axis will handle all control and monitoring of the physical motor, when mapped correctly to the axis.

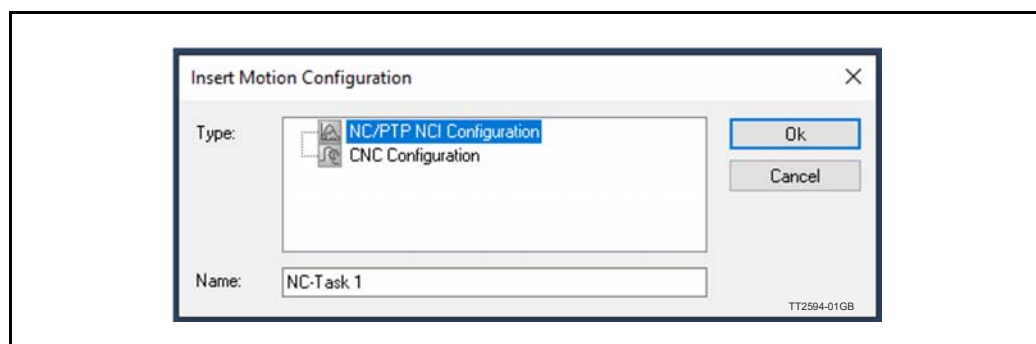
This way the PLC program does not need any information about what kind of hardware that is mapped to the axis. All motor behave the same when mapped to a NC axis.

Start by creating a NC task.



- # Right click on the Motion icon,
- # select "Add new Item"

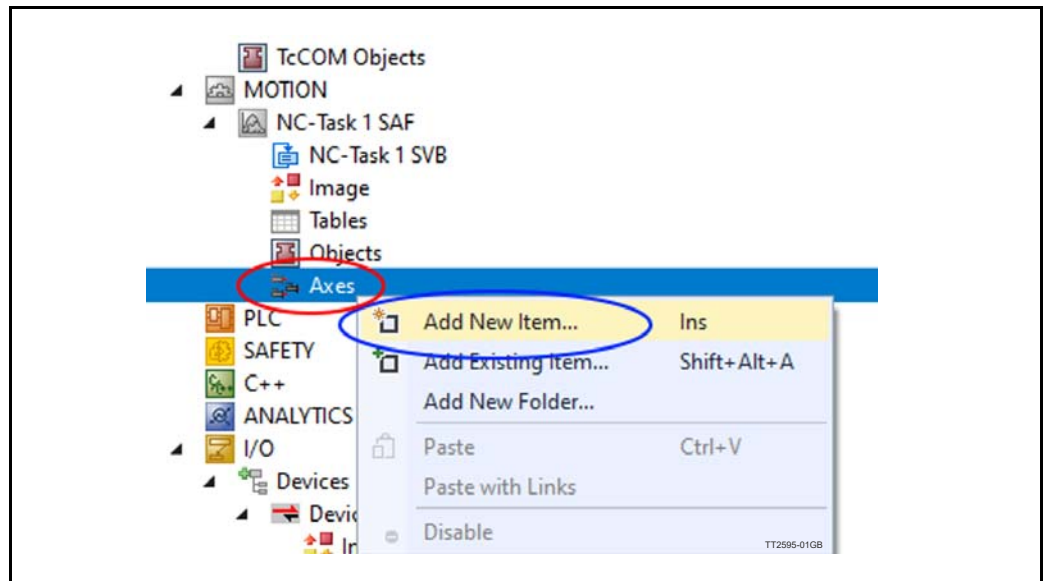
Select "NC/PTP NCI configuration" and give the motion configuration a name



## 11.6 Running MIS as NC axis in TwinCAT

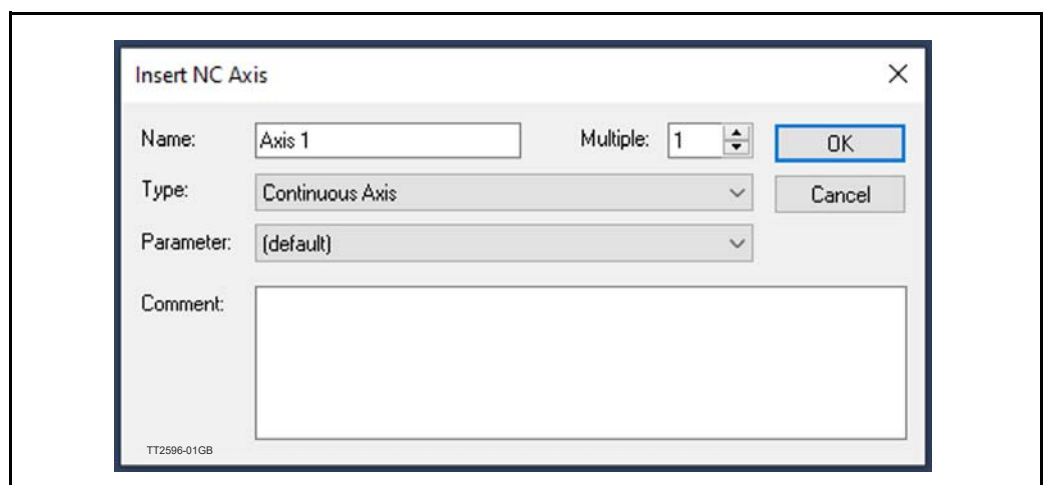
Option

Add the axis to the new task.



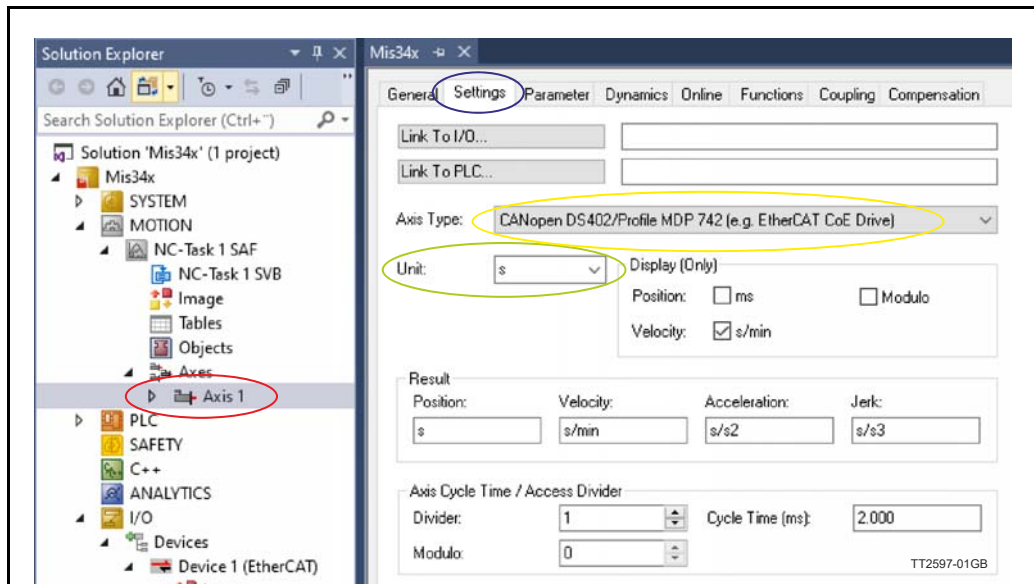
- # Right click "Axes"
- # click "Add new item"

Give the axis a name, and click "OK".



## 11.6 Running MIS as NC axis in TwinCAT Option

Set the axis to the right type and with the right units.

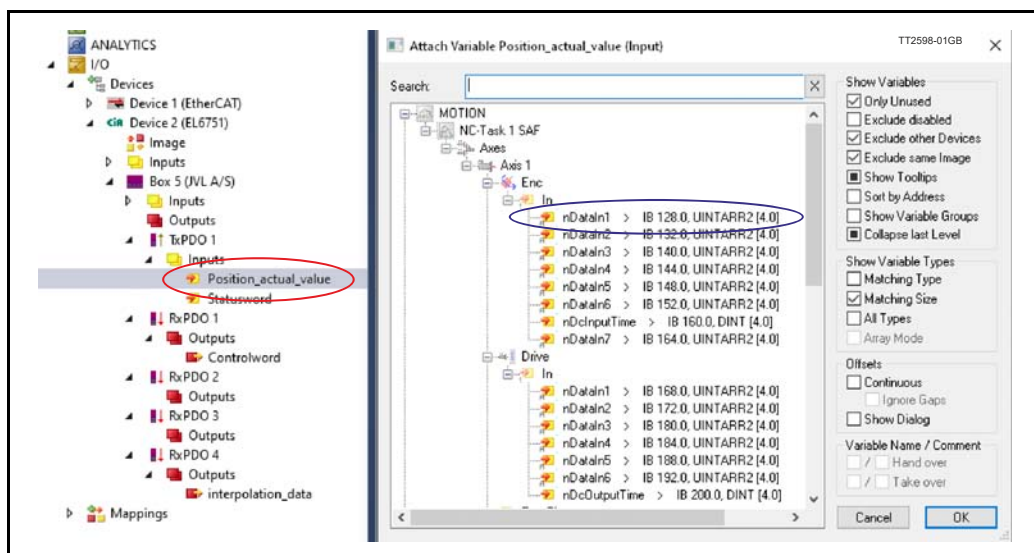


- # Double click on axis.
- # Select the settings tab.
- # Under axis type, select CANopen DS402.
- # Units: This example are using angle as units. This could also be a linear axis and could use millimeters instead.

Map PDO data.

Mapping is where data is mapped from the virtual axis to the physical motor.

### 11.6.8 Link position from motor to axis



Double-click Position\_actual\_value to attach PDO to nDataIn1 of the Enc device and press OK.

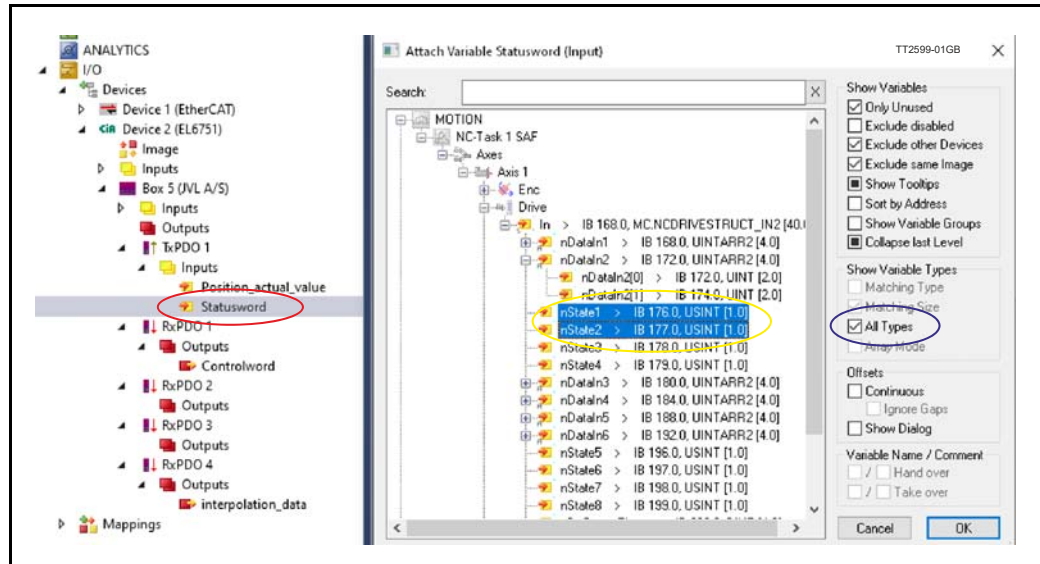
## 11.6 Running MIS as NC axis in TwinCAT

Option

### 11.6.9 Link status word from motor to axis

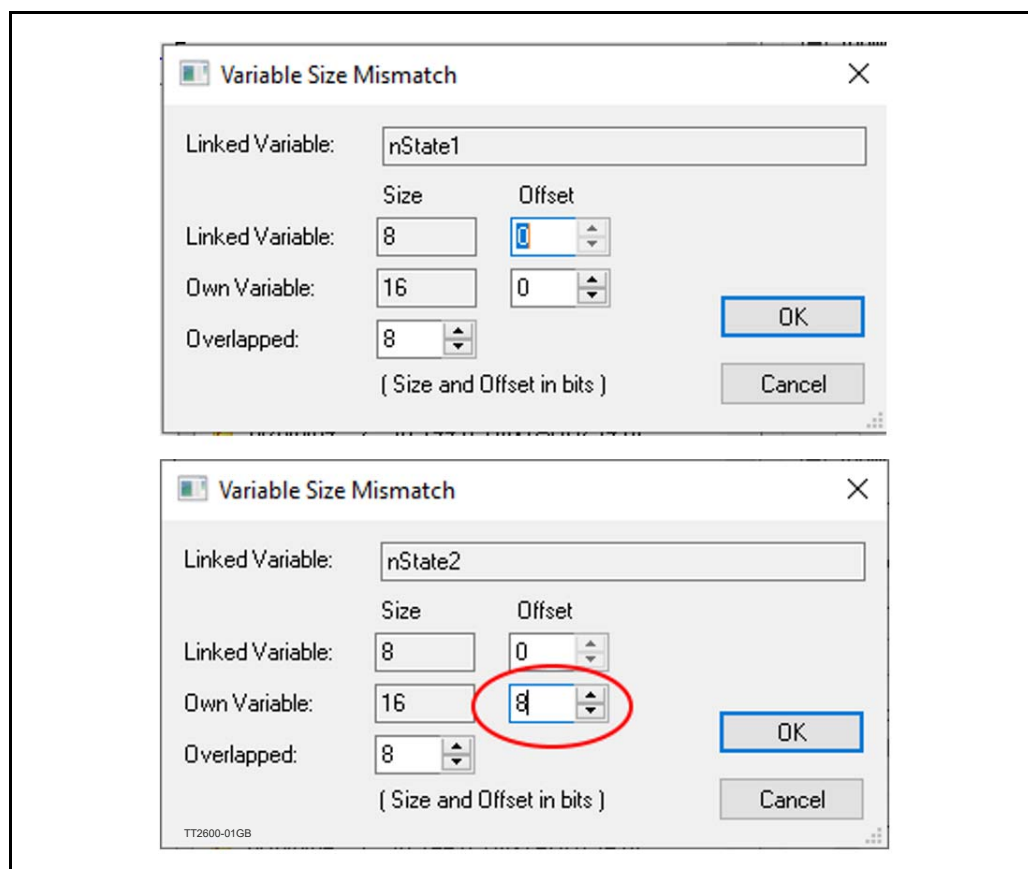
Double-click Statusword to attach PDO to nState1 and nState2 and press OK

If nState1 and nState2 is not visible, enable "All types" shown in blue circle



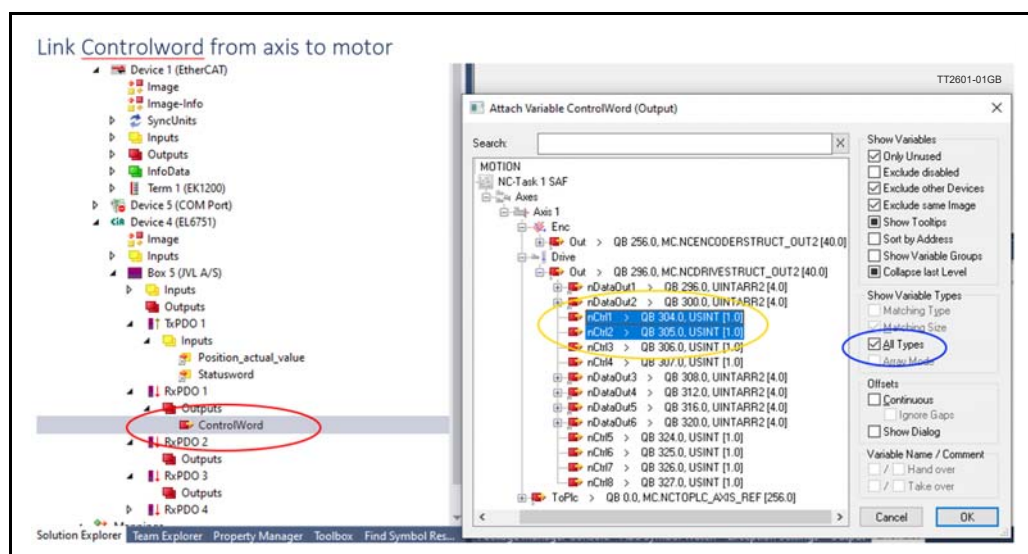
## 11.6 Running MIS as NC axis in TwinCAT Option

For nState1 and nState2 to get the right data, they need to be offset as shown below. The "Variable Size Mismatch" window can be opened by double-click the nState variable.



### 11.6.10 Link Controlword from axis to motor

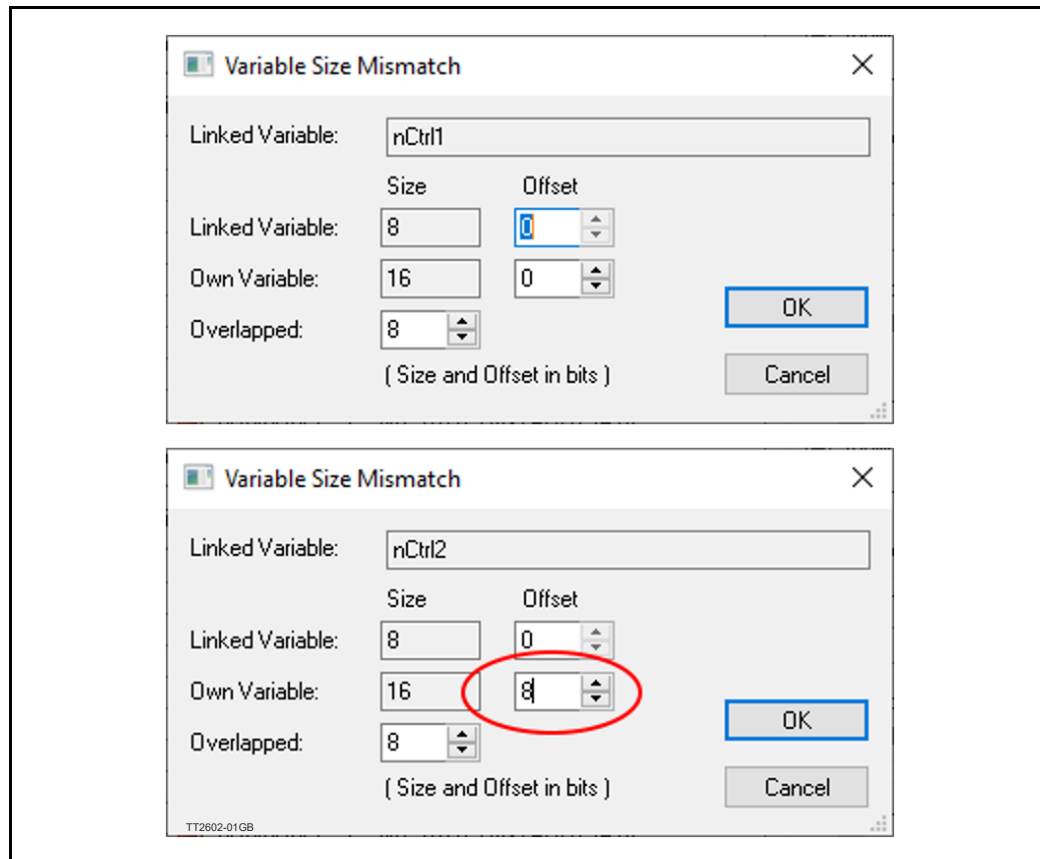
Double-click Controlword to attach PDO to nCtrl1 and nCtrl2 and press OK  
If nCtrl1 and nCtrl2 is not visible, enable "All types" shown in blue circle



## 11.6 Running MIS as NC axis in TwinCAT

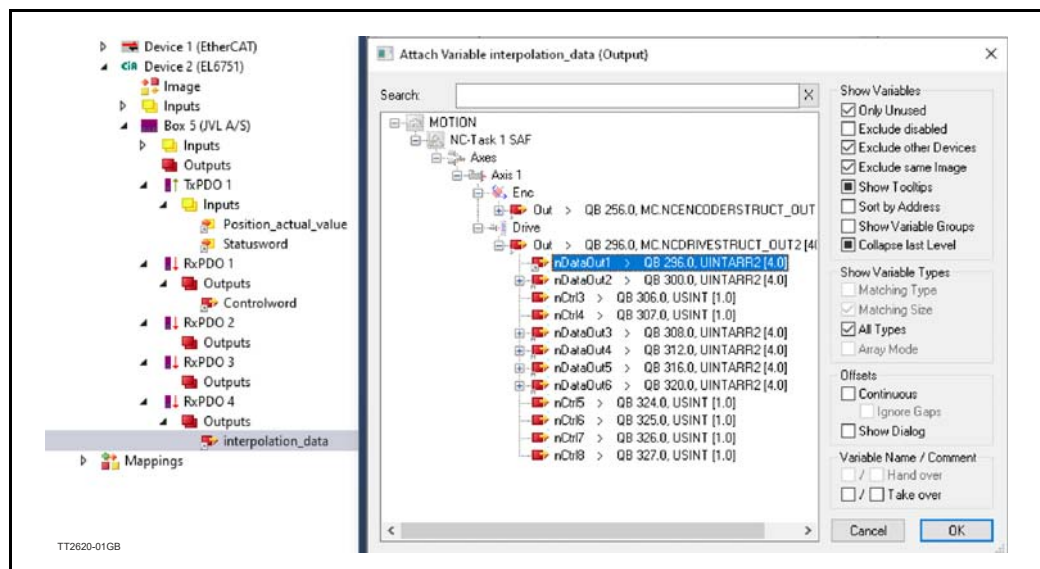
Option

For nCtrl1 and nCtrl2 to get the right data, they need to be offset as show below.  
The "Variable Size Mismatch" window can be opened by double-click the nCtrl variable.



### 11.6.11 Link position from axis to motor

Double-click Interpolation\_data to attach PDO to nDataOut1 of the Drive device and press OK



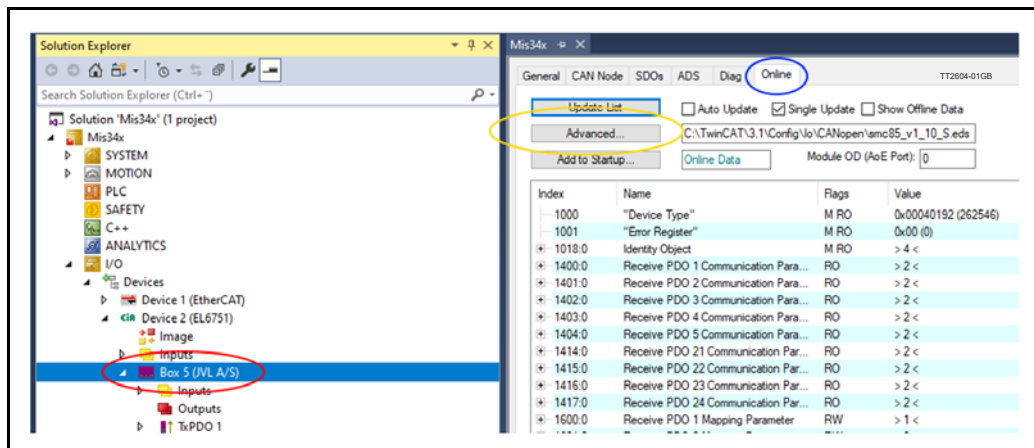
## 11.6 Running MIS as NC axis in TwinCAT Option

### 11.6.12 Set SDO data

SDO is data send at startup to do some pre use configuration, to enable the motor to respond correctly to the PDO data.

A few parameters in the motor need to be setup:

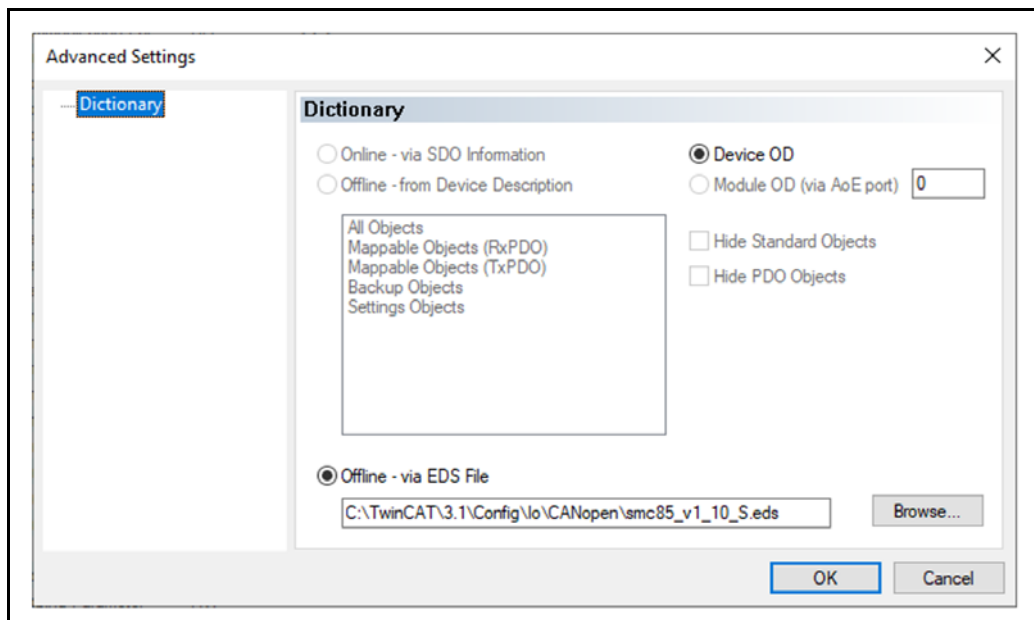
Set motor “mode of operation” to 7 : Interpolated position mode. This is the operation mode the NX axis is using to control the motor. The NC axis expect the motor to be in this operation mode when its available and ready.



# Click on Motor Box.

# Select the online tab

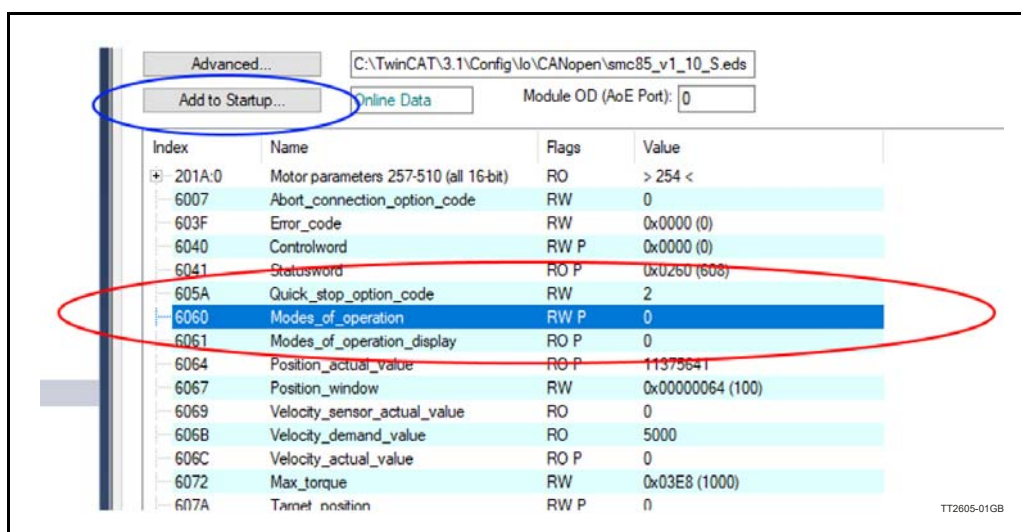
# If list is empty. Click the Advanced button. And select the EDS file manually. See the window below.



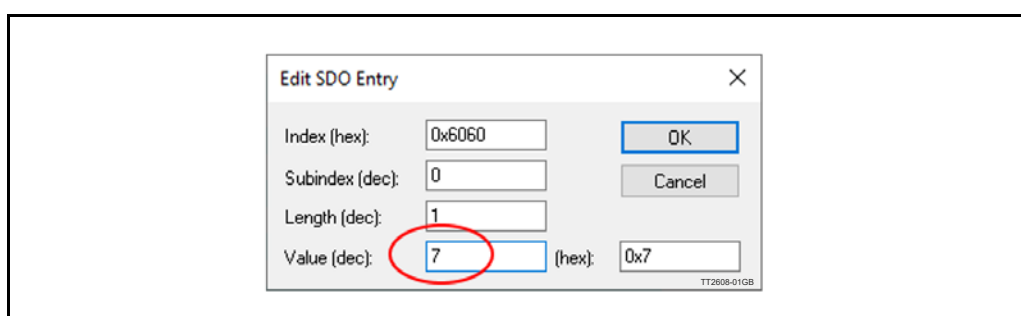
If necessary, browse to your EDS file and press OK to get a off-line object list.

## 11.6 Running MIS as NC axis in TwinCAT

Option



- # Select object 6060 Modes of operation from the list of objects.
- # Click the button “Add to startup”.



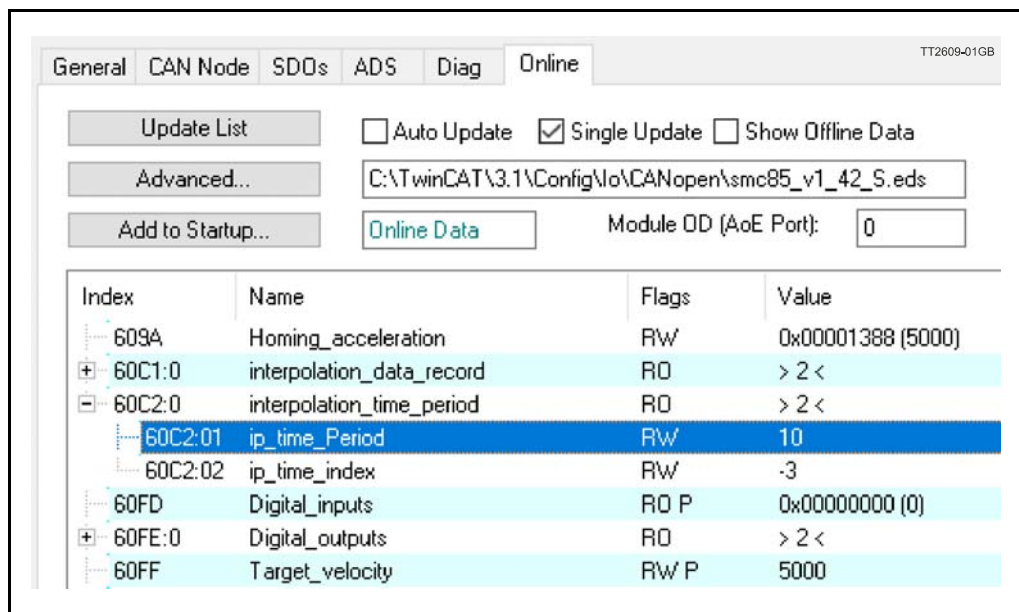
Change value to 7 and press OK.

## 11.6 Running MIS as NC axis in TwinCAT Option

### 11.6.13 Set the synchronous timing with SDO

The motor and the PLC need to be synchronized. The PLC used the timing to transmit Sync messages to the motor, and the motor uses the timing to calculate when to expect the next sync message. It is important that the motor and the PLC is set up to use the same timing.

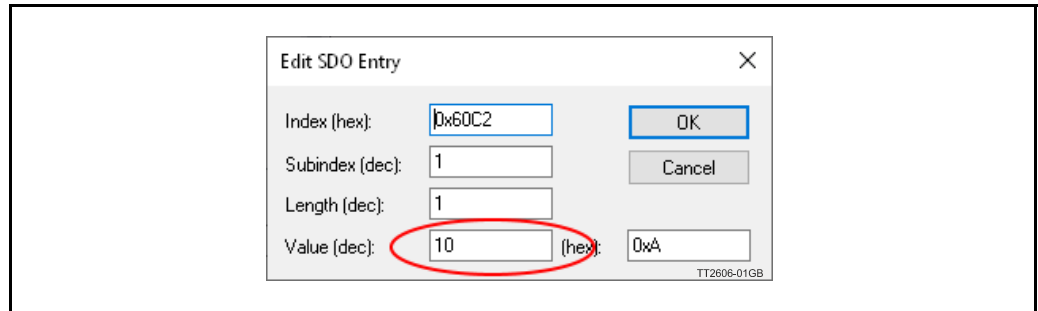
The object 60C2 sub 01 is used to tell the motor what timing is used in the PLC. Select the object 60C2:01 ip\_time\_period.



Click the button "Add to startup".

## 11.6 Running MIS as NC axis in TwinCAT

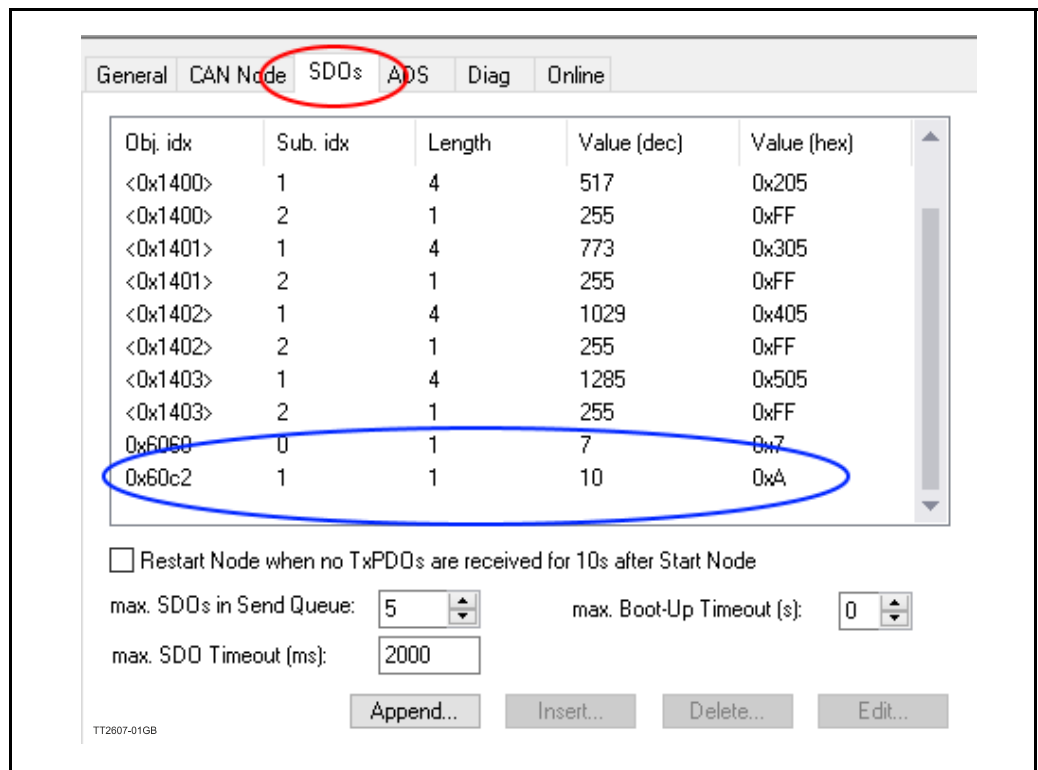
Option



Set value for the sync time in milliseconds. The default is 10ms. Lower synchronization can be used for better precision.

Make sure this value is the same set by the CANopen master.

Both custom SDO values can be viewed and edited under the SDO tab.



## 11.6 Running MIS as NC axis in TwinCAT

Option

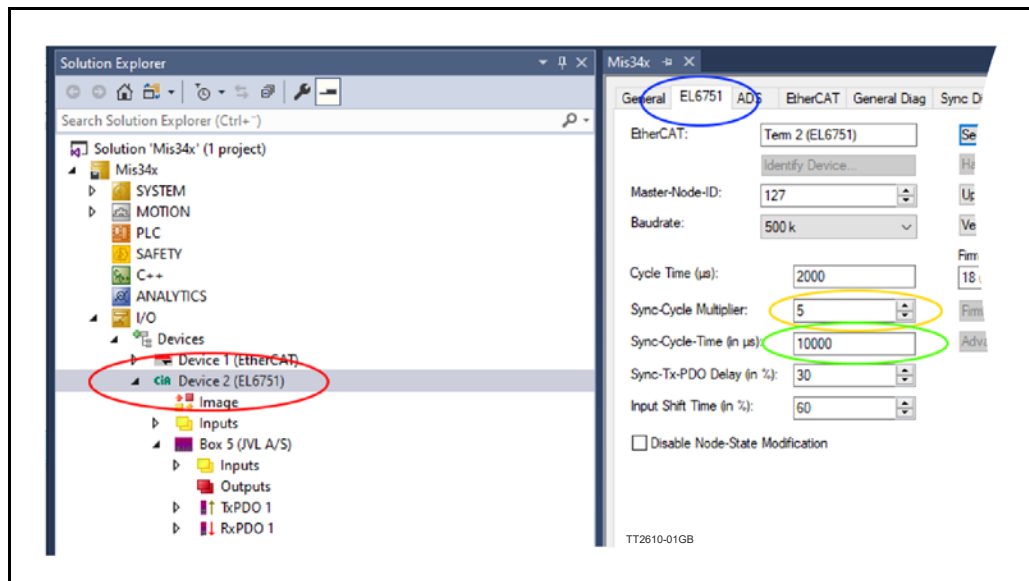
### 11.6.14 Sync interval

This is where the CANopen master timing is configured.

The cycle time is setup automatically to the PLC task that uses the fastest timing. In this case we only have a NC task, which by default uses 2ms cycle time. We can adjust the Sync time by multiplying the cycle time.

The default cycle time for JVL motor is 10ms, which need a multiplier of 5.

Set the CANopen master Sync interval. The sync interval should match the object 60C2:01.



- # Select the Cia device
- # Click the EL6751 (Beckhoff CANopen master)
- # Adjust the multiplier so that Cycle time times multiplier is the Sync interval.
- # Se sync interval updates on next build.

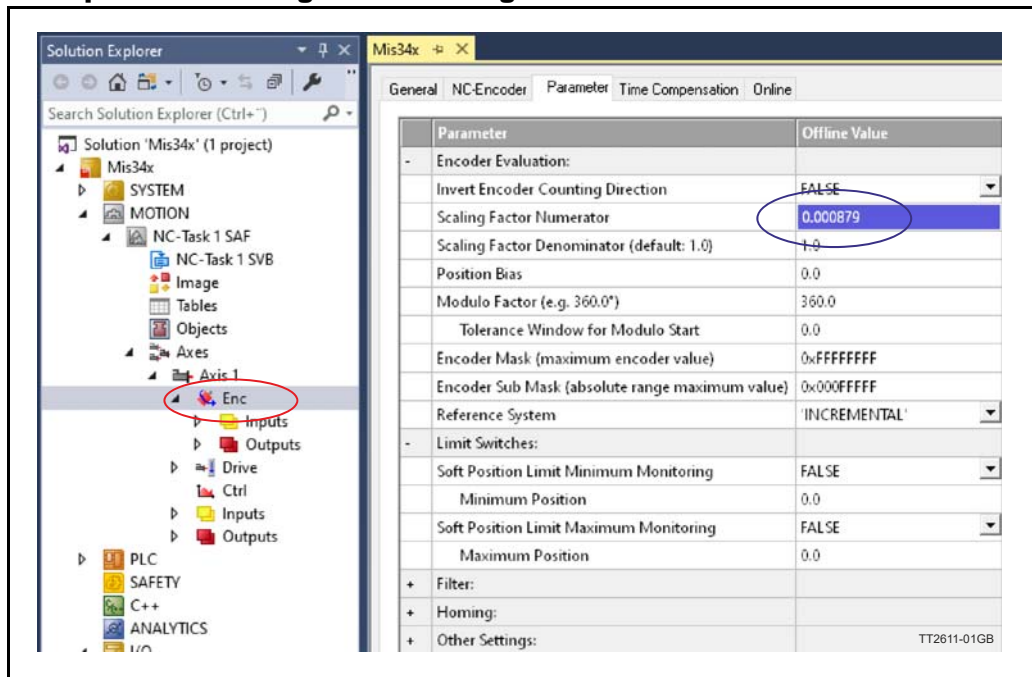
### 11.6.15 Set gear factor

The gear factor converts the virtual axis units to a physical unit.

In this sample we use mechanical angles. Thus 360 units is 1 revolution on the motor.

## 11.6 Running MIS as NC axis in TwinCAT Option

### 11.6.16 Example: Set scaling factor in degrees.



# Select the encoder for the NC axis

# Enter the calculated unit for the motor. See sample calculation below.

Sample calculation:

Count per revolution = 409600

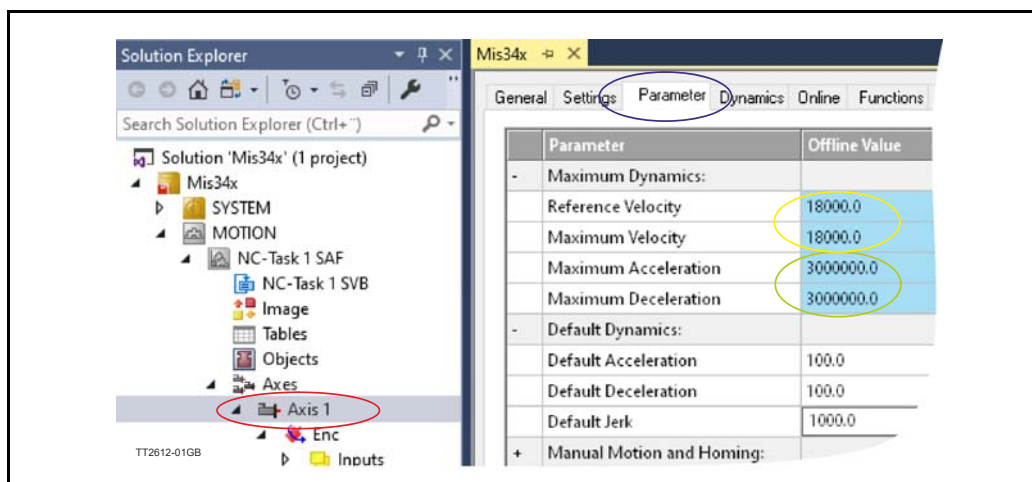
Degrees per revolution = 360

Scaling Factor Numerator:

Degrees per count = Degrees per revolution / Count per revolution =  $360/409600 = 0.000879$

### 11.6.17 Set motor parameters

Maximum velocity and maximum acceleration should also be entered to get the best performance from the motor.



# Select the motor axis # select parameters

## 11.6 Running MIS as NC axis in TwinCAT

Option

### Maximum Velocity #

Enter the calculated values. Sample calculation below.

Sample calculation:

Max RPM = 3000

Seconds per minute = 60

Degrees per revolution = 360

Degrees per second = Max RPM \* degrees per revolution / seconds per minute = 3000 \* 360 / 60 = 18000

### Max acceleration #

Enter the calculated values. Sample calculation below.

Sample calculation:

Max acceleration RPM/second = 500000

Seconds per minute = 60

Degrees per revolution = 360

Degrees in seconds<sup>2</sup> = Max acceleration RPM \* degrees per revolution / seconds per minute = 500000 \* 360 / 60 = 3000000

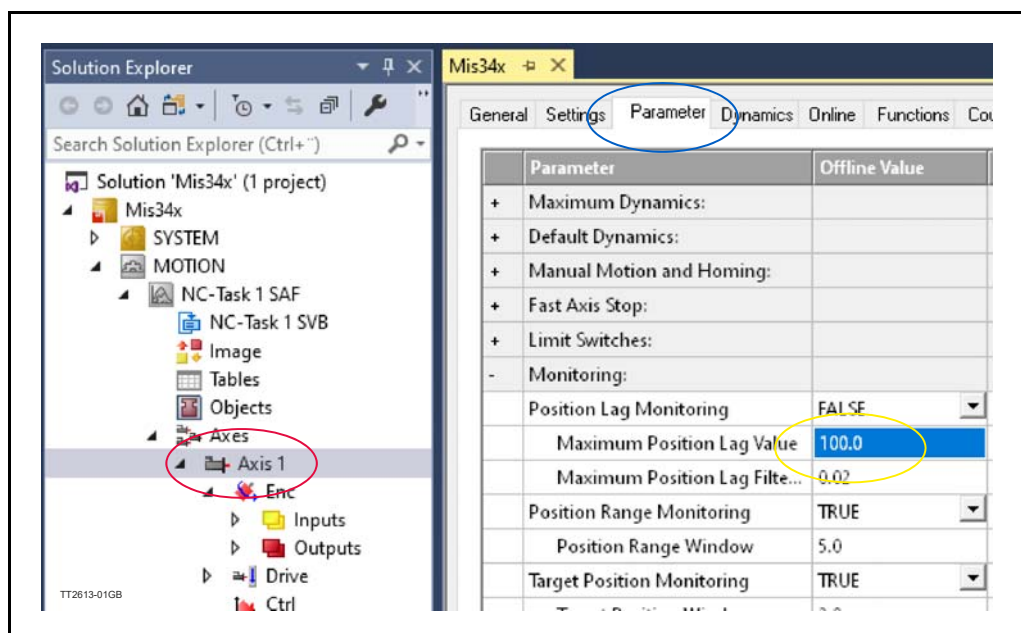
### 11.6.18 Follow error monitoring

The NC axis monitors the following error. This is how far behind the motor is at any given time. The motor will always be behind at least the time it takes to move between two sync cycles. With full speed that is significant.

It is recommended to adjust the following error limit to your suit the application.

TwinCAT call following error "Position Lag Value" monitoring.

In this sample it is set to 100 degrees.



# Select the motor axis

# Select parameters

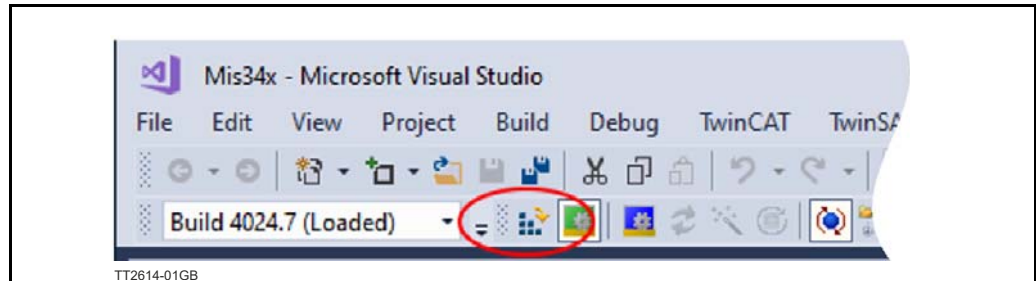
# Enter the follow error limit in the "Maximum Position Lag Value" field.

## 11.6 Running MIS as NC axis in TwinCAT

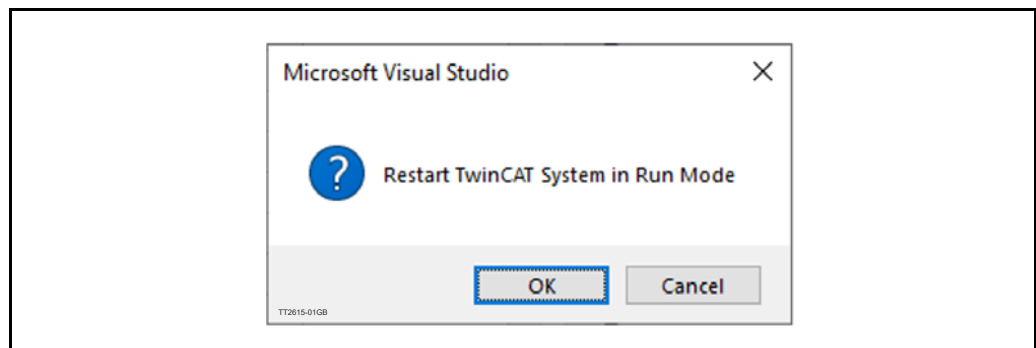
Option

### 11.6.19 Store settings in PLC

To use the new settings in the PLC project, the configuration need to be activated. This is done with the “Activate configuration” button.



# click the Activate configuration



Allow TwinCAT to restart in Run mode by clicking “OK”.  
The NC axis is ready to control your JVL motor.

This product supports the LSS functionality, that enables change of node id and Baud rate over CAN.

Make sure to transmit LSS message with the Baud rate the slave is currently configured to use.

LSS messages is transmitted from the master with COB ID 0x7E5 and is always with 8 bytes data. Non used data bytes are set to 0

The answer from the slave in Configuration mode is with COB ID 0x7E4

### 11.7.1 Switch Mode Global

To make contact with a device to be configured, the "Switch Mode Global" command is transmitted:

Data Byte		
Byte 0	Byte1	Byte remaining
0x04	0x01	0

This command sets the device to LSS configuration mode.

This is an unacknowledged LSS service, to which the device will therefore not respond, even if it has carried it out.

When the slave is in LSS configuration mode, the Error and LI LED is blinking rapidly.

### 11.7.2 Inquire

The current node-ID is requested via the "Inquire Node-ID" service:

This can be used to detect if the motor is in LSS mode.

Data Byte	
Byte 0	Byte remaining
0x5E	0

If successful, the device responds with:

Data Byte		
Byte 0	Byte1-5	Byte remaining
0x5E	Node ID	0

The inquire message also have other variants.

Byte 0 can have the following values:

- 0x5A : Responds with Vendor ID
- 0x5B : Responds with Product code
- 0x5C : Responds with Revision number
- 0x5D : Responds with Serial number
- 0x5E : Responds with Node ID

**11.7.3 Configure Node-ID**

The "Configure Node-ID" service is used to configure the new node-ID:

Data Byte		
Byte 0	Byte1	Byte remaining
0x11	Node ID	0

The error code is included in the device response:

Data Byte			
Byte 0	Byte1	Byte 2	Byte remaining
0x11	Error code	Error extension	0

Error code 0 means success;

error code 1 means inadmissible node-ID;

Other error codes are reserved.

The error extension contains vendor-specific information but is only valid for error code 0xFF.

**11.7.4 Configure Bit Timing Parameters**

The baud rate is configured with the "Configure Bit Timing Parameters" service:

Data Byte			
Byte 0	Byte1	Byte 2	Byte remaining
0x13	Timing table	Baudrate	0

The standardized CANopen baud rates are listed in the following table:

**Timing table**

0: Use standard CIA timing table

1 - 255: reserved

**Baudrate table 0x00**

0 = 1000 kBit/s

1 = 800 kBit/s

2 = 500 kBit/s

3 = 250 kBit/s

4 = 125 kBit/s

5 = reserved

6 = 50 kBit/s

7 = 20 kBit/s

8 = 10 kBit/s

The device response is:

Data Byte			
Byte 0	Byte1	Byte 2	Byte remaining
0x13	Error code	Error extension	0

Error code 0 means success;

Error code 1 means inadmissible baudrate;

Other error codes are reserved.

The error extension contains vendor specific information, but is only valid for error code 0xFF.

**11.7.5 Store Configuration**

Now that the node-ID and the baudrate are configured, and settings should be saved in flash with the "Store Configuration" service:

Data Byte	
Byte 0	Byte remaining
0x17	0

Whereupon the device acknowledges:

Data Byte			
Byte 0	Byte1	Byte 2	Byte remaining
0x17	Error code	Error extension	0

Error code 0 means success;

Error code 1 means that the device does not support saving;

Error code 2 means that there is a problem with access to the storage medium;

Other error codes are reserved.

The error extension contains vendor specific information, but is only valid for error code 0xFF.

**11.7.6 Switch Mode Global**

The device is switched back from configuration mode to normal mode via "Switch Mode Global":

Data Byte		
Byte 0	Byte1	Byte remaining
0x04	0x00	0

After a reset, the device now works with the new settings.

The SAE J1939 protocol is a CAN 2.0B based protocol, that takes advantage of the extended 29 bit identifier of CAN.

It is primarily used for the heavy equipment segment of automotive industry.

The protocol is used in an environment where all nodes support is based on their capabilities.

A majority of the communication is broadcast based, and every node will only respond to messages containing commands or request for operations supported by the node.

The protocol is not ideal for a multi-purpose devices like a servo/step motor in its native format.

The JVL motor take advantage of some of the later developed protocol features, with support for more custom specific peer to peer communication.

Its recommended to have some knowledge of SAE j1939 standard before using this protocol.

JVL J1939 I is primarily based on the following standards.

- SAE J1939/21 Data link Layer
- SAE J1939/31 Network Layer
- SAE J1939/71 Application layer
- SAE J1939/73 Application layer diagnostic
- Memory Access Request (DM14)
- Memory Access Response (DM15)
- Binary Data Transfer (DM16)

JVL has a tool called CAN-explore for helping debugging J1939 applications.

The CAN Explorer is a program that was developed for internal use only, especially in production, but the program offers features that helps understand how to read and write registers in the JVL motor over the j1939 Bus. The CAN-explorer can be downloaded from here: <https://www.jvl.dk/430/free-and-trial-programs>

### **11.8.1 How to Enable the J1939 protocol.**

To enable the J1939 protocol, make sure your motor has CAN bus support.

The j1939 can either be activated from the CAN tab in Macktalk, or by setting bit 8 in register 123, called setup register.

Save the setting in the motor and reset the motor.

This will disable all other CAN features in the motor.

### **11.8.2 Baud rate**

Supported Baud rates

10,20,50,100,125,250,500,1000 kbit/sec.

Baudrate/2 is setup with register 167

### **11.8.3 Node address**

Support addresses from 1 – 252

Recommended addresses 128-247

Negotiable address support 129 – 247

Node is 5 after factory default. setup node id with register 166

#### 11.8.4 Name field

When the motor is initialized with J1939 enabled, the Name field is transmitted. The name field is a 64bit message containing 10 fields of information about the node. The name field is transmitted with PGN 60928 as broadcast, and with the node address as source address.

Byte	Bit	
0		Identity number bit 0-7
1		Identity number bit 8-15
2	0-4	Identity number bit 16-19
	5-7	Manufacturer code bit 0-3
3		Manufacturer code bit 4-11
4	0-2	ECU instance
	3-7	Function instance
5		Function
6	0	Reserved (not assigned)
	1-7	Vehicle system
7	0-3	Vehicle system instance
	4-6	Industry group
	7	Arbitrary address capable

MIS motor name field contains the following data:

- Identity number: FFFFFh
- Manufacturer code: FFFh
- ECU instance: 1
- Function instance: 0
- Function: FFh
- Vehicle system: 7Fh
- Vehicle system instance: 0
- Industry group: 0
- Arbitrary address capable: 1

This is what the Name field looks like in the explorer with node ID 5:

The screenshot shows a CAN bus configuration interface. It includes fields for CanID dec (418316037), CanID Hex (0x18EEFF05), Priority (6), PGN Dec (61183), PGN Hex (0x00EEFF), and SourceAddr. (5). There are checkboxes for memory request read and write. The Data len. is set to 8, and the Data field contains the hex value 0xFF 0xFF 0xFF 0xFF 0x01 0xFF 0xFE 0x80. A Send button is present. The bottom section shows a message log with the following details:

MessageID	Dir	Length	Data	Period ms	Count
0x18EEFF05	RX	8	FF FF FF FF 01 FF FE 80	0	1

**11.8.5 Mac register access**

All mac register can be accessed with the J1939 protocol, by using the memory access application layer.

Length: Size of register (1,2, or 4 bytes)  
 Pointer type: Direct spatial addressing is used to access Mac registers  
 Command: Read (1) / Write (2) / Operation Complete (4)  
 Pointer: 1 – 512 (Mac register)  
 Pointer extension: Identifier for Mac registers (80h)  
 Key/user level: NA

**11.8.6 Package format****Memory access request (DM14)**

Parameter group 55552

Data length: 8

Byte	Bit	
0		Length LSB
1	5-7	Length MSB
	4	Pointer type
	1-3	Command
	0	Reserved (always 1)
2-4		Pointer
5		Pointer extension
6-7		Key/user level

**Memory access response (DM15)**

Parameter group 55296

Data length: 8

Byte	Bit	
0		Length LSB
1	5-7	Length MSB
	4	Reserved
	1-3	Status 0: Proceed 4: Operation Completed 5: Operation Failed
	0	Reserved
2-4		Error indicator
5		EDCP extension
6-7		Seed

**Binary data Transfer (DM16)**

Parameter group 55040

Data length: 8

Byte	Bit	
0		Data length (1-4)
1-7		Data (unused should be set to FFh)

**11.8.7 Transfer table****Normal read operation**

Tool	Direction	Device
Ma request Read memory	>	
	<	MA Response Proceed
	<	Ma Binary data
	<	Ma Response Complete
Ma request Operation Complete	>	

**Multiple read request**

Tool	Direction	Device
Ma request Read memory	>	
	<	MA Response Proceed
	<	Ma Binary data
	<	Ma Response Complete
Ma request Read memory	>	
	<	MA Response Proceed
	<	Ma Binary data
	<	Ma Response Complete
Ma request Operation Complete	>	

**Failed read operation**

Tool	Direction	Device
Ma request Read memory	>	
	<	MA Response operation Failed
Ma request Operation Complete	>	

**Memory request Write**

Tool	Direction	Device
Ma request write memory	>	
	<	MA Response Proceed
Timeout 1000 milliseconds.		
Ma Binary data	>	
	<	Ma Response Complete
Ma request Operation Complete	>	



## 11.9 More details of CANopen Theory

Option

### 11.9.1 CANopen DS-301 device profiles

Standardized devices in CANopen have their characteristics described in a device profile. For each device profile, particular data and parameters are strictly defined. Data and parameters are known as objects in CANopen. Objects perform all processes in CANopen; they can perform various tasks, either as communications objects or as device-specific objects where they are directly related to the device. A communication object can transport data to the bus control and establish connection, or supervise the network devices.

The application layer makes it possible to exchange meaningful real-time-data across the CAN network. The format of this data and its meaning must be known by the producer and the consumer(s). There are encoding rules that define the representation of values of data types and the CAN network transfer syntax for the representations. Values are represented as bit sequences. Bit sequences are transferred in sequences of octets (byte). For numerical data types, the encoding is with the lowest byte first.

Every object is described and classified in the object dictionary (or index) and is accessible via the network. Objects are addressed using a 16-bit index so that the object dictionary may contain a maximum of 65536 entries.

Index (Hex)	Object	Supported
0000-	Not used	
0001-001F	Static data types	
0020-003F	Complex data types	
0040-005F	Manufacturer specific Data Types	
0060-0FFF	Reserved for further use	
1000-1FFF	Communication Profile area DS301	Yes
2000-5FFF	Manufacturer specific profile area	Yes
6000-9FFF	Standardised Device Profile area (DSP-402)	Yes
A000-FFFF	Reserved for further use	

Index 0001-001F:

Static data types contain type definitions for standard data types like boolean, integer, floating point, etc. These entries are included for reference only, they cannot be read or written.

Index 0020-003F:

Complex data types are predefined structures that are composed out of standard data types and are common to all devices.

Index 0040-005F:

Manufacturer-specific data types are also structures composed of standard data types but are specific to a particular device.

Index 1000-1FFF:

The communication Profile area contains the parameters for the communication profile on the CAN network. These entries are common to all devices.

Index 2000-5FFF:

The manufacturer-specific profile area, for truly manufacturer-specific functionality.

## 11.9 More details of CANopen Theory Option

Index 6000-9FFF:

The standardised device profile area contains all data objects common to a class of devices that can be read or written via the network. The drives profile uses entries from 6000h to 9FFFh to describe the drive parameters and the drive functionality. Within this range, up to 8 devices can be described. In such a case, the devices are denominated Multi Device Modules. Multi Device Modules are composed of up to 8 device profile segments. Using this feature it is possible to build devices with multiple functionality. The different device profile entries are shifted with 800h.

A 16-bit index is used to address all entries within the object dictionary. In the case of a simple variable, this index references the value of the variable directly. In the case of records and arrays however, the index addresses the whole data structure. To allow individual elements of structures of data to be accessed via the network, a sub-index has been defined. For single object dictionary entries such as Unsigned8, Boolean, Integer32, the value of the sub-index is always zero. For complex object dictionary entries such as arrays or records with multiple data fields, the sub-index refers to fields within a data-structure pointed to by the main index. Index counting starts with one.

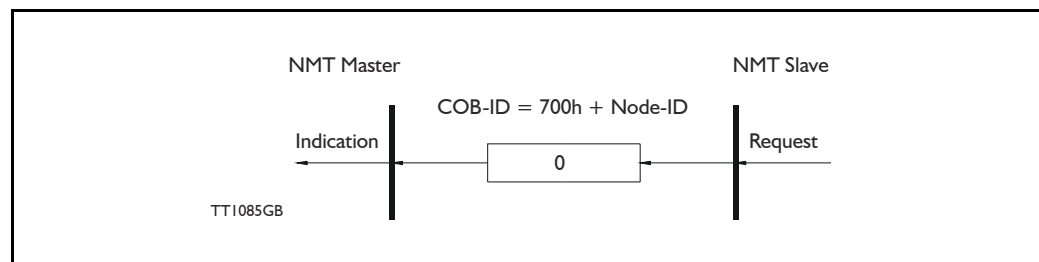
The DS-301 standard constitutes the application and the communications profile for a CANopen bus, and is the interface between the devices and the CAN bus. It defines the standard for common data and parameter exchange between other bus devices, and it controls and monitors the devices in the network. The table below lists some of the communications profile objects:

Data Transfer	Parameter Transfer	Special functions	
PDO			Process Data Objects
	SDO		Service Data Objects
		SYNC	Synchronisation
		EMCY	Emergency

The access from the CAN network is done through data objects PDO (Process Data Object) and SDO (Service Data Object).

### 11.9.2 Boot up telegram

After the initialization phase, a CANopen slave logs on with a boot up message. The node address of the slave is contained in this. This allows a CANopen master to know which slaves are connected to the network. The protocol uses the same identifier as the error control protocols. See the figure below:



One data byte is transmitted with value 0.

**11.9.3 PDO (Process Data Object)**

PDO: Performs real-time transfers, and the transfer of PDOs is performed without a protocol. PDOs are used in two ways: for data transmission and for data reception. PDOs can bundle all objects from the object data directory, and a PDO can handle max 8 bytes of data in the same PDO. The PDO can consist of multiple objects. Another PDO characteristic is that it does not reply when it is receiving data, in order to make data transfer fast. It has a high priority identifier.

PDO connections follow the Producer/Consumer model, whereby a normal PDO connection follows the Push model and an RTR connection the Pull model.

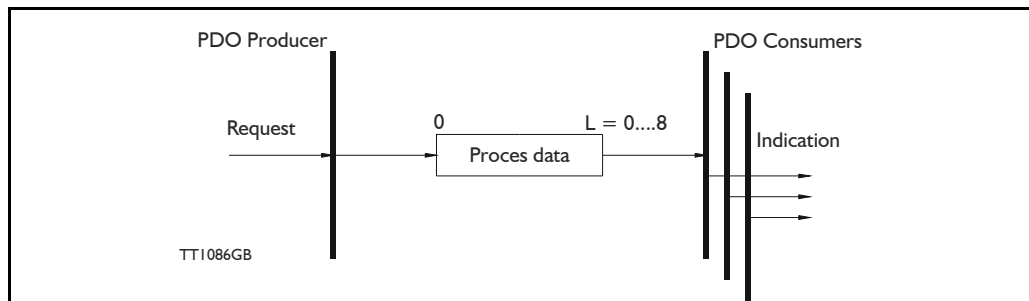
Objects are mapped in a PDO. This mapping is an agreement between the sender and receiver regarding which object is located at which position in the PDO. This means that the sender knows at which position in the PDO it should write data and the receiver knows where it should transfer the data to that is received.

The PDOs correspond to entries in the Device Object Dictionary and provide the interface to the application objects. Data type and mapping of application objects into a PDO are determined by a corresponding PDO mapping structure within the Device object Dictionary. The number and length of PDOs of a device are application specific and must be specified within the device profile

Write PDO service:

The Write PDO service is unacknowledged. A PDO producer sends its PDO to the PDO consumer. There can be 0 or more consumers in the network. For receive PDOs the MIS23x (SMC75) is the consumer and for Transmit PDOs, the producer.

The following figure shows a Write PDO service:

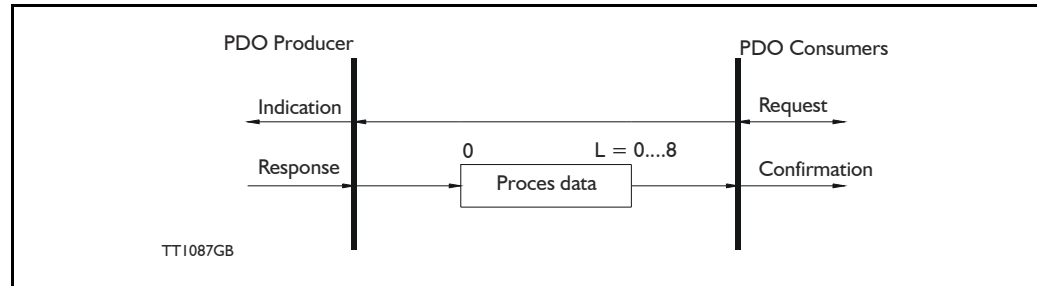


## 11.9 More details of CANopen Theory

Option

Read PDO service:

The read PDO service is an acknowledged service. One of the several PDO consumers send an RTR message to the network. After it has received the RTR message, the PDO producer sends the requested PDO. This service is used for RTR queries. Using this service, an actual value can be interrogated independently of the selected cycle time. The following figure shows a read PDO service:



PDO identifier:

In the CANopen profile, it is only possible to have four transmit and four receive PDOs enabled at the same time. In the MIS motors (or SMC66/85), all PDOs are disabled when the module is booted up. The user must choose which PDOs the application will use and enable these.

The PDO configuration can be seen either in the EDS-file or in the CAN-explorer program, where the communication and the mapping parameters are shown.

There are two standard methods to map the PDOs in CANopen: static mapping and dynamic mapping. In static PDO mapping all PDOs are mapped in accordance with some fixed, non-modifiable setting in the relevant PDO. In dynamic PDO mapping, the setting of a PDO can be modified. It is also allowable to have a flexible combination of different process data during operation.

### 11.9.4 SDO (Service Data Objects)

SDO: can access all entries in the object directory but they are normally used in the initialization during the boot up procedure. Some SDOs characteristics are:

- Confirmed transfer of objects
- Data transfer/exchange is always non-synchronous
- Values greater than 4 bytes are transferred (Normal transfer)
- Values not more than 4 bytes are transferred (Expedited transfer)

Basically an SDO is transferred as a sequence of segments. Prior to transferring the segment, there is an initialization phase where client and server prepare themselves for transferring the segment. For SDOs, it is also possible to transfer a dataset of up to four bytes during the initialization phase. This mechanism is called an expedited transfer.

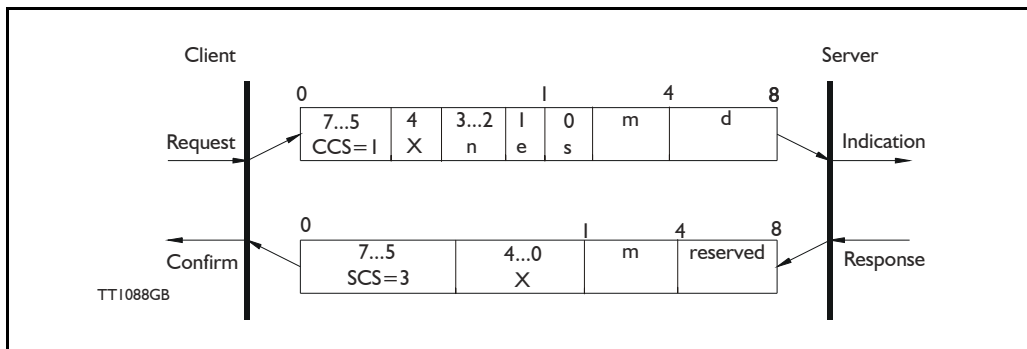
Download SDO protocol:

The download SDO protocol is used to write the values of the object directory into the drive.

## 11.9

## More details of CANopen Theory

Option



Upload SDO protocol:

The upload SDO protocol is used to read the values in the object directory of the drive.

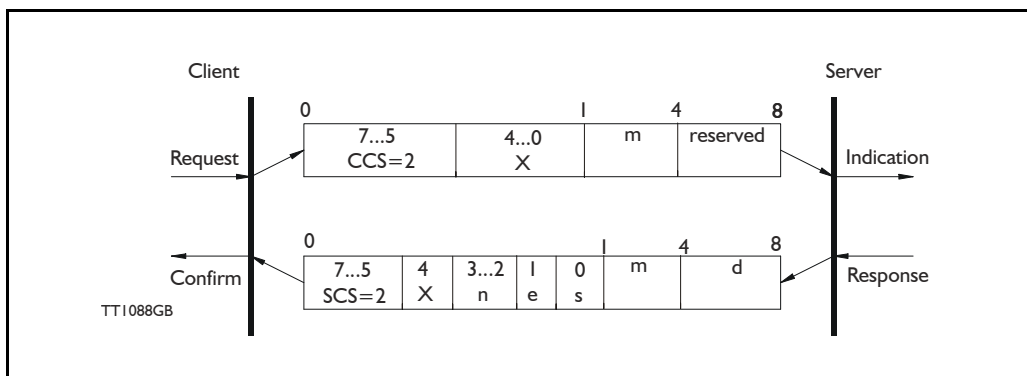


Table for upload and download SDO protocol.

	CCS:	SCS:	n:	e:	s:	m:
Down-load	1: Initiate down-load request	3: Initiate download response	Only valid if e=1 and s=1 otherwise 0. If valid it indicates the number of bytes in d that do not contain data. Bytes [8-n,7] do not contain data	Transfer type: 0= normal transfer 1= expedited transfer	Size indicator: 0=data set size is not indicated 1=data set size is indicated	Multiplexer. It represents the index/sub-index of the data to be transfer by the SDO
Upload	2: Initiate upload request	2: Initiate upload response	Only valid if e=1 and s=1 otherwise 0. If valid it indicates the number of bytes in d that do not contain data. Bytes [8-n,7] do not contain data	Transfer type: 0= normal transfer 1= expedited transfer	Size indicator: 0=data set size is not indicated 1=data set size is indicated	Multiplexer. It represents the index/sub-index of the data to be transfer by the SDO

CCS: Client command specified.

SCS: Server commander specified.

## 11.9 More details of CANopen Theory

Option

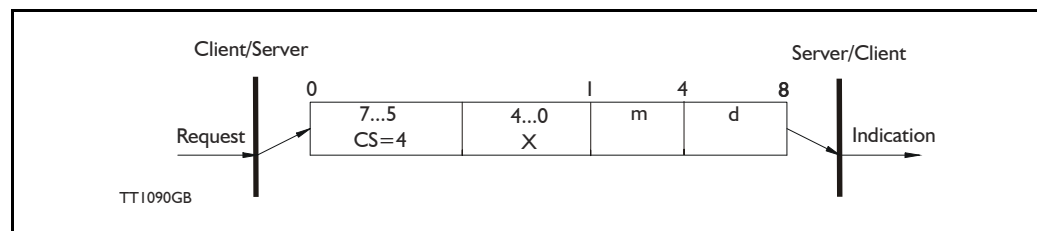
Table for upload and download SDO protocol (continued)

	d:	X:	Reserved:
Download	<b>e=0, s=0:</b> d is reserved for further use <b>e=0, s=1:</b> d contains the number of bytes to be downloaded. Byte 4 contains the lsb and byte 7 contains the msb <b>e=1, s=1:</b> d contains the data of length 4-n to be downloaded, the encoding depends on the type of the data referenced by index and sub-index.	not used, always 0	Reserved for further use, always 0
Upload	<b>e=0, s=0:</b> d is reserved for further use <b>e=0, s=1:</b> d contains the number of bytes to be uploaded. Byte 4 contains the lsb and byte 7 contains the msb <b>e=1, s=1:</b> d contains the data of length 4-n to be uploaded, the encoding depends on the type of the data referenced by index and sub-index.	not used, always 0	Reserved for further use, always 0

Abort SDO transfer protocol:

SDO tasks which the MIS motors (or SMC66/85) cannot process are responded to using an abort SDO protocol. If the module does not respond in the expected time, the CANopen master also sends an abort SDO.

The following figure shows an abort SDO transfer protocol:



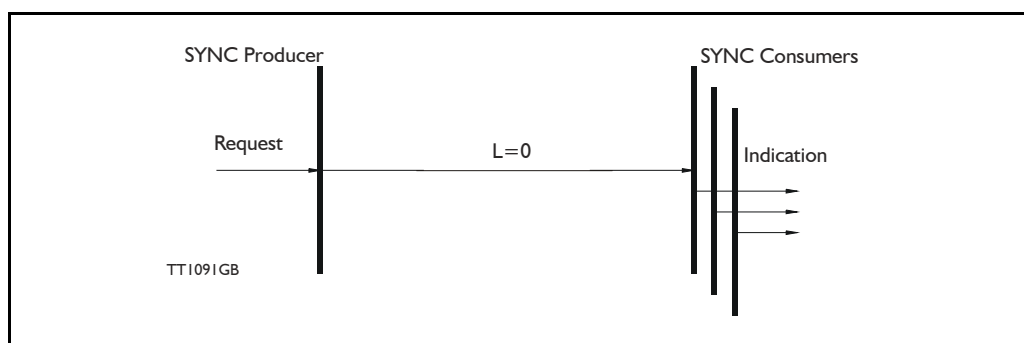
There are various abort codes in CANopen. These are listed in the table below:

Abort code	Description
0503 0000h	Toggle bit not alternated
0504 0000h	SDO Protocol timed out
0504 0001h	Client/server command specified not valid or unknown
0504 0002h	Invalid block size (block mode only)
0504 0003h	Invalid sequence number (block mode only)
0504 0004h	CRC error (block mode only)
0504 0005h	Out of memory
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write-only object
0601 0002h	Attempt to write a read-only object
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO

Abort code	Description
0604 0042h	The number and length of the objects to be mapped would exceed PDO length
0604 0043h	General parameter incompatibility reason
0606 0000h	Access failed due to a hardware error
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist
0609 0030h	Value range of parameter exceeded (only for write access)
0609 0031h	Value of parameter written too high
0609 0032h	Value of parameter written too low
0609 0036h	Maximum value is less than minimum value
0800 0000h	General error
0800 0020h	Data cannot be transferred or stored to the application
0800 0021h	Data cannot be transferred or stored to the application because of local control
0800 0022h	Data cannot be transferred or stored to the application because of the present device state
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of a file error).

### 11.9.5 SYNC (Synchronisation Object)

A SYNC producer sends the synchronization object cyclically a broadcast telegram. The SYNC telegram defines the basic clock cycle of the network. The time interval of the SYNC telegram is set using the object Communication Cycle period (1006h). In order to obtain a precise (accurate) cycle between the SYNC signals, the SYNC telegram is sent with a high-priority identifier. This can be modified using the object (1005h). The SYNC transfer applies the producer/consumer push model and is non-confirmed.



The SYNC does not carry any data ( $L=0$ ). The identifier of the SYNC object is located at object 1005h.

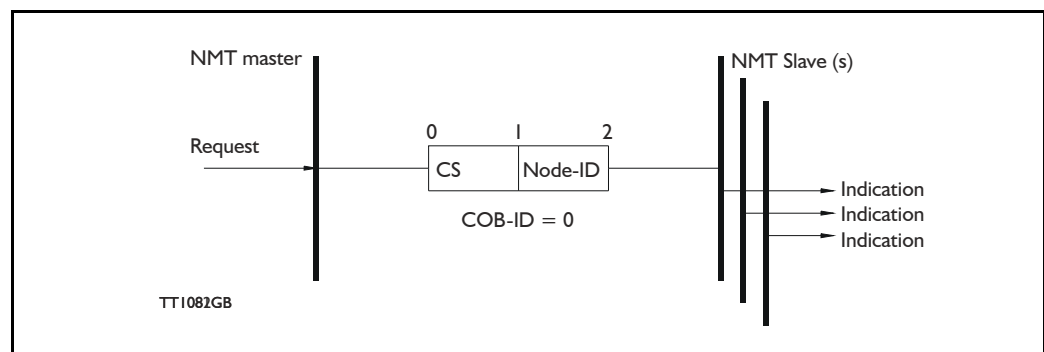
## 11.9 More details of CANopen Theory

Option

### 11.9.6 NMT (Network Management services)

The Network Management is structured according to nodes and follows a master-slave structure. NMT objects are used for executing NMT services. Through NMT services, nodes are initialised, started, monitored, reset or stopped. All nodes are regarded as NMT slaves. An NMT slave is uniquely identified in the network by its Node-ID. NMT requires that one device in the network fulfils the function of the NMT master. The NMT master controls the state of the NMT slaves. The state attribute is one of the values (Stopped, Pre-operational, Operational, Initialising). The module control services can be performed with a certain node or with all nodes simultaneously. The NMT master controls its own NMT state machine via local services which are implementation dependent. The Module Control Service, except Start Remote Node, can be initiated by the local application.

A general NMT protocol:



Where **CS** is the NMT command specified. The Node-ID of the NMT slave is assigned by the NMT master in the Node Connect protocol, or 0. If 0, the protocol addresses all NMT slaves.

CS =	Operation
1	Start Remote Node
2	Stop Remote Node
128	Enter Pre Operational
129	Reset Node
130	Reset Communication

**Start Remote Node:**

This is an instruction for transition from the Pre-Operational to Operational communications state. The drive can only send and receive process data when it is in the Operational state.

**Stop Remote Node:**

This is an instruction for transition from either Pre-Operational to stopped or from Operational to Stopped. In the stopped state, the nodes can only process NMT instructions.

**Enter Pre Operational:**

This is an instruction for transition from either Operational or Stopped state to Pre-Operational. In the Pre-Operational state, the node cannot process any PDOs. However, it can be parameterized or operated via SDO. This means set point can also be entered.

## 11.9 More details of CANopen Theory

Option

### Reset Node:

This is an instruction for transition from the Operational, Pre-Operational or Stopped states to Initialization. After the Reset Node instruction, all objects (1000h-9FFFh) are re-set to the Voltage On stage.

### Reset Communication:

This is an instruction for transition from Operational or Stopped to Initialization. After the Reset Communication instruction, all communication objects (1000h-1FFFh) are re-set to the initial state.

In the various communication states, nodes can only be accessed via CANopen using specific communication services. Further, the nodes in the various states only send specific telegrams. This is clearly shown in the following table:

	Initializing	Pre-Operational	Operational	Stopped
PDO			X	
SDO		X	X	
Synchronization Object		X	X	
Emergency Object		X	X	
Boot-Up Object	X			
Network Management object		X	X	X

### 11.9.7 Error Control Services

Two possibilities exist for performing Error Control:

- Node Guarding/Life Guarding
- Heartbeat

#### Node Guarding/Life Guarding

With Node Guarding, the CANopen master sends each slave an RTR telegram (Remote Transmit request) with the COB-ID 1792 (700h) + node-ID.

Using the same COB-ID, the slave responds with its communications state, i.e. either Pre-Operational, Operational or stopped.

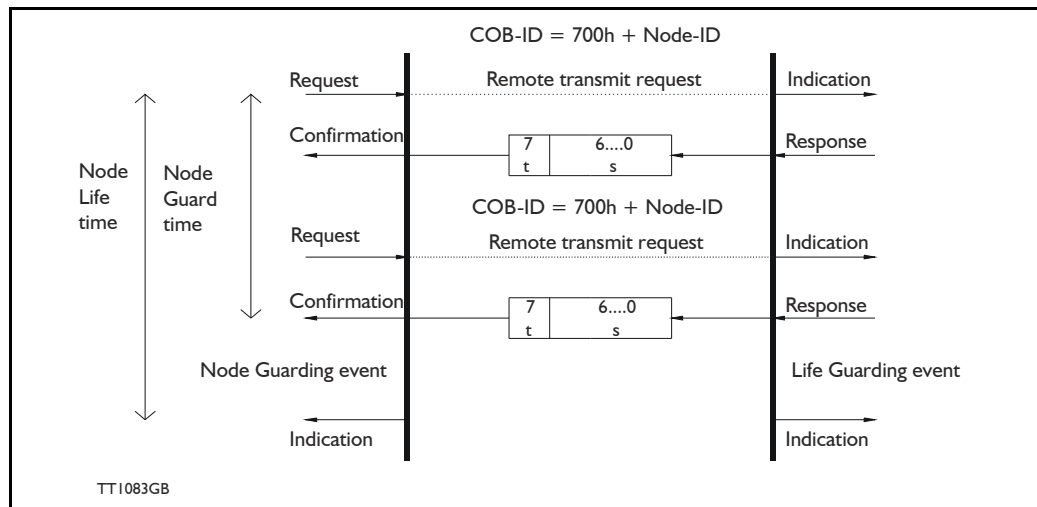
The CANopen slave also monitors the incoming RTR telegram from the master.

The cycle of the incoming RTR telegrams is set using the Guard Time Object.

The number of RTR telegrams which can fail (at a maximum) before the slave initiates a Life Guarding event is defined using the Life time factor object.

The Node Life Time is calculated from the product of the Guard Time and Life Time Factor. This is the maximum time that the slave waits for an RTR telegram.

The figure below shows a Node Guarding/Life Guarding protocol.



Where s is the state of the NMT slave:

s	NMT state
4	Stopped
5	Operational
7	Pre-operational

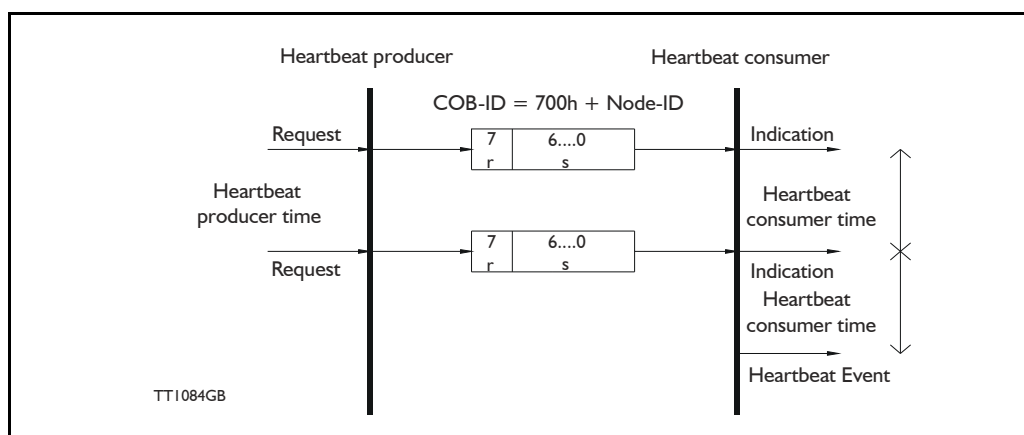
**t:** is the toggle bit. It alternates between 2 consecutive responses from the NMT Slave. The value of the toggle-bit of the first response after the guarding protocol becomes active is 0.

The Toggle Bit in the guarding protocol is only reset to 0 when the NMT message Reset Communication is passed (no other change of state resets the toggle bit).

If a response is received with the same value of the toggle-bit as in the preceding response, then the new response is handled as if it was not received.

#### Heartbeat:

With the Heartbeat protocol, a Heartbeat Producer cyclically sends its communications state to the CAN bus. One or more Heartbeat Consumers receive the indication. The relationship between producer and consumer is configurable via the object dictionary. The Heartbeat Consumer guards the reception of the Heartbeat within the Heartbeat Consumer time. If the Heartbeat is not received within the Heartbeat Consumer Time, a Heartbeat Event will be generated.



Where r is reserved (always 0).

s: is the state of the Heartbeat producer:

s	NMT state
0	Boot up
4	Stopped
5	Operational
7	Pre-operational

Only one communication monitoring service may be activated. This is either Node Guarding/Life Guarding or Heartbeat. If the Heartbeat Producer Time is configured on a device, the Heartbeat Protocol begins immediately. If a device starts with a value of the Heartbeat Producer Time different from 0, the Heartbeat Protocol starts with the state transition from Initialising to Pre-operational. In this case the Bootup Message is regarded as the first heartbeat message. If the Heartbeat producer time is not 0, the heartbeat protocol is used.

In the MIS motors (or SMC66/85), none of the error control mechanisms is enabled when the modules are started up, because if there is any fault in the system it is impossible to contact the module. After the module has started up and there is communication between the master and the slave, activate the required error control mechanism in the object Dictionary. See [DS301 specified Communications objects](#), page 290.



## 12.1

## Modbus Interface

---

### 12.1.1 Modbus introduction

The MIS/SMC can be interfaced using the Modbus RTU protocol. Modbus RTU communication is available on the main RS-485 channel, normally used for MacTalk communication, and on the secondary RS-422 channel as well.

Note: Modbus on the RS-485 channel has been available since version 4.00 and Modbus on the RS-422 channel was introduced in firmware version 5.04. It is not possible to use Modbus on the RS-485 channel if the motor is equipped with an ethernet module.

The Modbus implementation is a subset of the Modbus Specification V1.1b. This standard can be downloaded free of charge from the website [www.modbus.org](http://www.modbus.org). You may want to download and read the [Modbus Serial Line Protocol and Implementation Guide V1.02](#), that describes many aspects of the signals, and the details of using and inter-connecting two-wire RS-485.

The MIS firmware supports two types of Modbus frames:  
“Read Holding Registers” (0x03) and  
“Write Multiple Registers” (0x10).  
Furthermore, a custom JVL frame can be used to read/write PDO registers using a single request. See also [Read/write PDO register command](#), page 373.  
All other commands will result in Exception replies (exception type 1, Illegal Function). It is possible to access all registers using Modbus.

## 12.2

## Modbus Configuration

The slave address of the motor can be configured using register 150. This address is the same as the MacTalk motor address. Note that the default is 254, which is not a valid according to the Modbus address.

Change it if your Modbus Master does not accept this slave address.

The serial format used for Modbus can be configured using register 121.

There is a single setup register for Modbus on RS485 and Modbus on RS422. It is possible to enable Modbus on both channels, but both channels use the same configuration.

Bits	Values	Description
0	0 = Disabled 1 = Enabled on RS485 channel	Enable Modbus RTU on the main RS-485 channel. If bit is set, Modbus will be activated once MacTalk communication times out.
1	Reserved	Reserved for future use. Set to zero.
3:2	0 = No parity 1 = Odd parity 2 = Even parity 3 = Reserved	Parity selection. Refer to the manual of the PLC/HMI to determine what parity is required.
4	Reserved	Reserved for future use. Set to zero.
5	0 = 1 stop bit 1 = 2 stop bits	Select the number of stop bits. Select the same format as the Modbus client PLC, HMI or PC uses
7:6	Reserved	Reserved for future use. Set to zero.
8	0 = Disabled 1 = Enabled on RS422 channel	Enabled Modbus RTU on the RS-422 channel. Modbus is activated without delay, unlike Modbus on the RS485 channel.
31:9	Reserved	Reserved for future use. Set to zero.

There is a separate baud rate register for the main RS485-channel (register 146), and for the secondary RS-422-channel (register 286).

The value of the register is an integer in the range [0; 6]

Value	Baud rate	Notes
0	9600	
1	19200	Default value for the RS-485 channel
2	38400	
3	57600	
4	115200	
5	230400	
6	460800	Default value for the RS-422 channel

Additional configuration registers:

Register #	Description
199	Modbus slave time out: Controls the max time between two messages before timeout response is initiated. The value is given in milliseconds. If 0 is written to this register the time out function will be disabled (time out will never occur)
200	Time out response: Set the action activated by time out. If time out is set to 0, no action or event will occur. 0: Time out disabled (no action, only event). 1: Max velocity is set to zero 2: Motor is set to passive mode
201	Frame delay: The response to the master request is delayed with this value. If the value is zero, the default frame delay of 2.5 bytes is used. This can be necessary to adjust if the master needs time to change direction on the serial driver.

## 12.2 Modbus Configuration

---

### 12.2.1 Activation of Modbus on RS-485 channel

Note: Modbus on the main RS485 channel is only possible if no ethernet module is installed.

The main RS-485 is normally reserved for communication with MacTalk. MacTalk is not able to use Modbus, so it cannot be used to configure the motor or update the firmware once Modbus is enabled. MacTalk communication is always the active protocol after the motor has booted to ensure that is possible to perform a firmware update.

The motor will switch to Modbus after 1 second of inactivity on the serial line (if Modbus has been enabled, bit 0 in register 121 is set).

The recommended procedure to configure Modbus on the RS-485 channel:

1. Configure the baud rate using register 146 (The baud rate will change as soon as the setting has been written).
2. Enable Modbus (bit 0), and apply the correct serial settings in register 121
3. Write to flash

Modbus on the RS-485 channel will be activated once MacTalk-communication has timed out. The settings will persist after a power-cycle.

### 12.2.2 Activation of Modbus on RS-422 channel

Note: There are two differential receivers on the PCB, and they be configured for various applications such as SSI communication, quadrature output, quadrature input, and RS-422 communication. Make sure that you do not use the transceivers already if you wish to use Modbus on RS-422. Unlike the RS-485 channel, it is possible to use Modbus on the RS-422 channel if an ethernet module is installed.

The recommended procedure to configure Modbus on the RS-422 channel:

1. Configure the baud rate using register 286 (The baud rate will change as soon as the setting has been written).
2. Enable Modbus (bit 8), and apply the correct serial settings in register 121
3. Write to flash

Modbus on the RS-422 will be active immediately after following theses steps, and the settings will persist after a power-cycle.

## 12.3 Reading and Writing Registers

---

All registers in the MIS are 32-bits, while Modbus use 16-bit registers. To comply with the 16-bit Modbus standard, a 32-bit register must be read or written as two consecutive 16-bit registers. The motor registers are therefore also 16-bit addressable, so the ordinary register numbers must be multiplied by 2.

**Example:**

Actual position has register number 10. Modbus register index 20 addresses the lower 16-bits of the actual position. Register index 21 addresses the upper 16-bits of the actual position.

Modbus uses big-endian byte-order for 16-bit values. Pay special attention to this, when dealing with 32-bit registers. The two 16-bit parts of the 32-bit register are each big-endian encoded. Reading two consecutive 16-bits registers will **NOT** be 32-bit big-endian, since the lower part of the 32-bit register has a lower register index than the upper 16-bits. Each 16-bit parts must be handled separately, converted to the correct endianness, and then combined to retrieve the correct 32-bit value.

**Example:**

The actual position, register 10, has the value 123456789 (0x075BCD15). The user wishes to read the actual position, so the Modbus-registers 20 and register 21 are read consecutively. The response will contain the values {0xCD, 0x15, 0x07, 0x5B}.

When writing to registers, you it is only permitted to write whole 32-bit value (2 Modbus registers), even though the Modbus standard allows for single register writes. Additionally, the register index must be even (addressing the beginning of a motor register). These restrictions are to prevent corrupting a register value by unaligned access, and to prevent that the two 16-bit parts are updated separately.

All registers have an alternative register index mapping at offset 40 000 e.g. the motor register  $n$  can be read at Modbus register index  $2n$ ,  $2n + 1$  or at register index  $40000 + n$ ,  $40000 + n + 1$ .

## 12.4 Read Holding Registers (0x03)

Motor registers can be read using the Modbus command “Read Holding Registers” (0x03). The number of registers that can be read in a single request is restricted to 16 16-bits registers (or lower). It is recommended to keep this number as low as possible, and to split the reads into multiple requests.

The request is structured as

Offset	Value	Description
0	Slave address	The slave address of the motor. Address 0 is a broadcast, and not valid for reads.
1	Command = 0x03	The type of Modbus request.
2	Index (high)	Upper and lower byte of the register index (2 x motor register address).
3	Index (low)	
4	Count (High)	Upper and lower byte of the number of 16-bit registers. Must be 16 or lower.
5	Count (Low)	
6	CRC1	16-bit CRC
7	CRC2	

The structure of the reply is

Offset	Value	Description
0	Slave address	The slave address of the motor, repeated.
1	Command = 0x03	The type of Modbus request, repeated.
2	Number of bytes	The number of bytes of data. This should correspond to 2 x the number of 16-bit registers requested.
3	Reg 0 (high)	Lower 16-bits of the 32-bit register
4	Reg 0 (low)	
5	Reg 1 (high)	Upper 16-bits of the 32-bit register value
6	Reg 1 (low)	
...		
3 + N	CRC1	16-bit CRC
3 + N + 1	CRC2	

**Example:** Read actual position (register 10) from motor with slave address 42. The number of 16-bits registers is 2. The request frame is

{0x2A, 0x03, 0x00, 0x14, 0x00, 0x02, NN, MM}

Where NN and MM are the CRC-16 bytes. If the value of the actual position is 123456789, then the reply frame will be

{0x2A, 0x03, 0x04, 0xCD, 0x15, 0x07, 0x5B, NN, MM}

## 12.5 Write Multiple Registers (0x10)

Registers can be written to by using the Modbus command “Write Multiple Registers” (0x10). The structure of the request is

Offset	Value	Description
0	Slave address	The slave address of the motor. Address 0 is the broadcast address and can be used for writes.
1	Command = 0x10	The type of Modbus request.
2	Index (high)	Upper and lower byte of the register index (2 x motor register address).
3	Index (low)	
4	Count (High)	Upper and lower byte of the number of 16-bit registers. Must be 16 or lower.
5	Count (Low)	
6	Number of bytes	The number of bytes of data. This should correspond to 2 x the number of 16-bit registers requested.
7	Reg 0 (high)	Lower 16-bits of the first 32-bit register value, encoded in big-endian byte order.
8	Reg 0 (low)	
9	Reg 1 (high)	Upper 16-bits of the first 32-bit register value, encoded in big-endian byte order.
10	Reg 1 (low)	
...		
7 + N	CRC1	16-bit CRC
7 + N + 1	CRC2	

The reply structure is

Offset	Value	Description
0	Slave address	The slave address of the motor. Address 0 is the broadcast address and can be used for writes.
1	Command = 0x10	The type of Modbus request.
2	Index (high)	Upper and lower byte of the register index (2 x motor register address).
3	Index (low)	
4	Count (High)	Upper and lower byte of the number of 16-bit registers. Must be 16 or lower.
5	Count (Low)	
6	CRC1	16-bit CRC
7	CRC2	16-bit CRC

**Example:** We wish to set Velocity 1 (register 65) to 409600 and Velocity 2 (register 66) to 13.

Since the register numbers are consecutive, it can be done in a single frame.

The motor has slave address 60. The request is

{0x3C, 0x10, 0x00, 0x82, 0x00, 0x04, 0x08, 0x40, 0x00, 0x00, 0x06, 0x00, 0x0D, 0x00, x00, NN, MM}

If everything is in order, we will receive the response

{0x3C, 0x10, 0x00, 0x82, 0x00, 0x04, NN, MM}

## 12.6

## PDO Registers

---

PDO registers can be used to remap the internal motor register addresses. This is practical if you frequently want to read multiple registers that are not in consecutive order. The PDO registers map the non-consecutive registers addresses into a consecutive address space. This way, all the mapped registers can be read from or written to in a single request, rather than using a request for each register.

There are two separate PDO register sets available, PDO1, and PDO2, each consisting of 8 read-setup registers, 8 write-setup registers and 8 value-registers (PDO1\_TX\_SETUP[0..7], PDO1\_RX\_SETUP[0..7] and PDO1\_VALUE[0..7]). The two PDO-sets can be configured independently.

### Example:

In an application you want to read actual position (10), actual velocity (12), and status (25) registers cyclically. Since these registers are not in consecutive order, 3 separate Modbus requests must be sent, which has a lot of overhead.

To be able to read the registers in a single Modbus frame, the registers are remapped using PDO registers. The setup registers are configured as

```
PDO1_TX_SETUP[0] = 10
PDO1_TX_SETUP[1] = 12
PDO1_TX_SETUP[2] = 25
```

The three register values can be retrieved in a single frame by reading 3 32-bit register values from register PDO1\_VALUE[0..2]. You can't write to PDO1\_VALUE[0..2] since PDO1\_RX\_SETUP hasn't been configured yet. Let's say you wish to write to requested position (3) and requested velocity (5), so the write registers are setup as

```
PDO1_RX_SETUP[0] = 3
PDO1_RX_SETUP[1] = 5
```

Now you can read actual position, actual velocity, and status by reading PDO1\_VALUE[0], PDO1\_VALUE[1], PDO1\_VALUE[2], and write to requested position, and requested velocity by writing to PDO1\_VALUE[0] and PDO1\_VALUE[1] respectively.

### 12.6.1 Setup of PDO registers

**Note:** PDO register setup is not saved to flash. They must be configured at every boot.

The 8 readable PDO registers of PDO1 or PDO2 are configured by writing a list of register indices to register PDO1\_TX\_SETUP[0] (see the register mapping for PDO1 and PDO2). The first index in the list maps reads from PDO1\_VALUE[0] to the motor register of that index, the second index maps reads from PDO1\_VALUE[1] and so forth. The write to the setup register is done using a single "Write Multiple Registers" command containing the 32-bit register index for each motor register to be mapped. The number of 32-bit values written to PDO1\_TX\_SETUP[0] determines the number of activated PDO TX registers. It is possible to configure less than 8 registers.

**Note:** It is not allowed to write to register PDO1\_TX\_SETUP[1..7], you have to start with register PDO1\_TX\_SETUP[0], and write all values in one request.

Setup of writable registers are done in a similar fashion by writing a list of 32-bit motor register indices to PDO1\_RX\_SETUP[0].

## 12.6

## PDO Registers

### 12.6.2 Reading/Writing PDO registers

A PDO register can be read by reading the PDO<sub>n</sub>\_VALUE[k] register if PDO<sub>n</sub>\_TX\_SETUP[k] has been configured. The request will return the value of the register [PDO<sub>n</sub>\_TX\_SETUP[k]]. The read is done using a normal Modbus read request.

If the given PDO register has not been configured for reading, an exception will be returned for invalid address. It is possible to read from one or more PDO registers in a single frame, and the start address does not have to be PDO<sub>n</sub>\_VALUE[0].

A PDO register can be written to by writing to PDO<sub>n</sub>\_VALUE[k] register if PDO<sub>n</sub>\_RX\_SETUP[k] has been configured. The value of the request is written to register [PDO<sub>n</sub>\_RX\_SETUP[k]]. The write is performed using a normal Modbus write-request. If the given PDO register has not been configured for writing, an exception will be returned for invalid address. It is possible to write to one or more PDO registers in a single frame, and the start address does not have to be PDO<sub>n</sub>\_VALUE[0].

### 12.6.3 PDO1 Register Mapping

RX Setup Registers for PDO1:

MB address	Register	Description
0xF200	PDO1_RX_SETUP[0]	PDO1 RX setup registers. Each register is 32-bit wide. The register holds a register number (normal motor register number, not Modbus register index). The value maps write-requests to PDO1_VALUE[k] register to writes to the motor register with index PDO1_RX_SETUP[k].
...		
0xF20E	PDO1_RX_SETUP[7]	

TX Setup Registers for PDO1:

MB address	Register	Description
0xF300	PDO1_TX_SETUP[0]	PDO1 TX setup registers. Each register is 32-bit wide. The register holds a register number (normal motor register number, not Modbus register index). The value maps read requests from PDO1_VALUE[k] register to the motor register with index PDO1_TX_SETUP[k].
...		
0xF30E	PDO1_TX_SETUP[7]	

Value Registers for PDO1:

MB address	Register	Description
0xF400	PDO1_VALUE[0]	PDO1 value registers. Each register is 32-bit wide. When you write a 32-bit value to PDO1_VALUE[k], a value is written to the motor register which PDO1_RX_SETUP[k] points to.  When you read a 32-bit value from PDO1_VALUE[k], the register number stored in PDO1_TX_SETUP[k] determines which actual motor register you read from.
...		
0xF40E	PDO1_VALUE[7]	

## 12.6

## PDO Registers

### 12.6.4 PDO2 Register Mapping

RX Setup Registers for PDO2:

MB address	Register	Description
0xF500	PDO2_RX_SETUP[0]	PDO2 RX setup registers. Each register is 32-bit wide. The register holds a register number (normal motor register number, not Modbus register index). The value maps write-requests to PDO2_VALUE[k] register to writes to the motor register with index PDO2_RX_SETUP[k].
...		
0xF50E	PDO2_RX_SETUP[7]	

TX Setup Registers for PDO2:

MB address	Register	Description
0xF600	PDO2_TX_SETUP[0]	PDO2 TX setup registers. Each register is 32-bit wide. The register holds a register number (normal motor register number, not Modbus register index). The value maps read requests from PDO2_VALUE[k] register to the motor register with index PDO2_TX_SETUP[k].
...		
0xF60E	PDO2_TX_SETUP[7]	

Value Registers for PDO2:

MB address	Register	Description
0xF700	PDO2_VALUE[0]	PDO2 value registers. Each register is 32-bit wide. When you write a 32-bit value to PDO2_VALUE[k], a value is written to the motor register which PDO2_RX_SETUP[k] points to.  When you read a 32-bit value from PDO2_VALUE[k], the register number stored in PDO2_TX_SETUP[k] determines which actual motor register you read from.
...		
0xF70E	PDO2_VALUE[7]	

## 12.7 Read/write PDO register command

The MIS/SMC has support for a custom (non-standard) Modbus frame that allows you to both read and write PDO registers in a single frame. The command 0x4A is for reading/writing to PDO1 registers while 0x4B is for reading/writing to PDO2 registers.

Request structure:

Offset	Value	Description
0	Slave address	The slave address of the motor.
1	Command = 0x4A or 0x4B	The type of Modbus request. 0x4A is PDO1 0x4B is PDO2.
2	Number of bytes	The number of bytes of data. This should correspond to 4 x the number of PDO registers that have been configured for write.
3	Write-value[0] (b1)	First value being written to the register that RX_PDO_SETUP[0] points to.
4	Write-value[0] (b0)	
5	Write-value[0] (b3)	
6	Write-value[0] (b2)	
7	Write-value[1] (b1)	Second value being written to the register that RX_PDO_SETUP[1] points to.
8	Write-value[1] (b0)	
9	Write-value[1] (b3)	
10	Write-value[1] (b2)	
...		
3 + N	CRC1	16-bit CRC
3 + N + 1	CRC2	16-bit CRC

Response structure

Offset	Value	Description
0	Slave address	The slave address of the motor.
1	Command = 0x4A or 0x4B	The type of Modbus request. 0x4A is PDO1 0x4B is PDO2.
2	Number of bytes	The number of bytes of data. This should correspond to 4 x the number of PDO registers that have been configured for read.
3	Read-value[0] (b1)	First value that contains the register value of the register that TX_PDO_SETUP[0] points to.
4	Read-value[0] (b0)	
5	Read-value[0] (b3)	
6	Read-value[0] (b2)	
7	Read-value[1] (b1)	Second value that contains the register value of the register that TX_PDO_SETUP[1] points to.
8	Read-value[1] (b0)	
9	Read-value[1] (b3)	
10	Read-value[1] (b2)	
...		
3 + N	CRC1	16-bit CRC
3 + N + 1	CRC2	16-bit CRC



## **13        DMX512 Protocol and interface**

---

## 13.1 DMX512 Description

### 13.1.1 Introduction

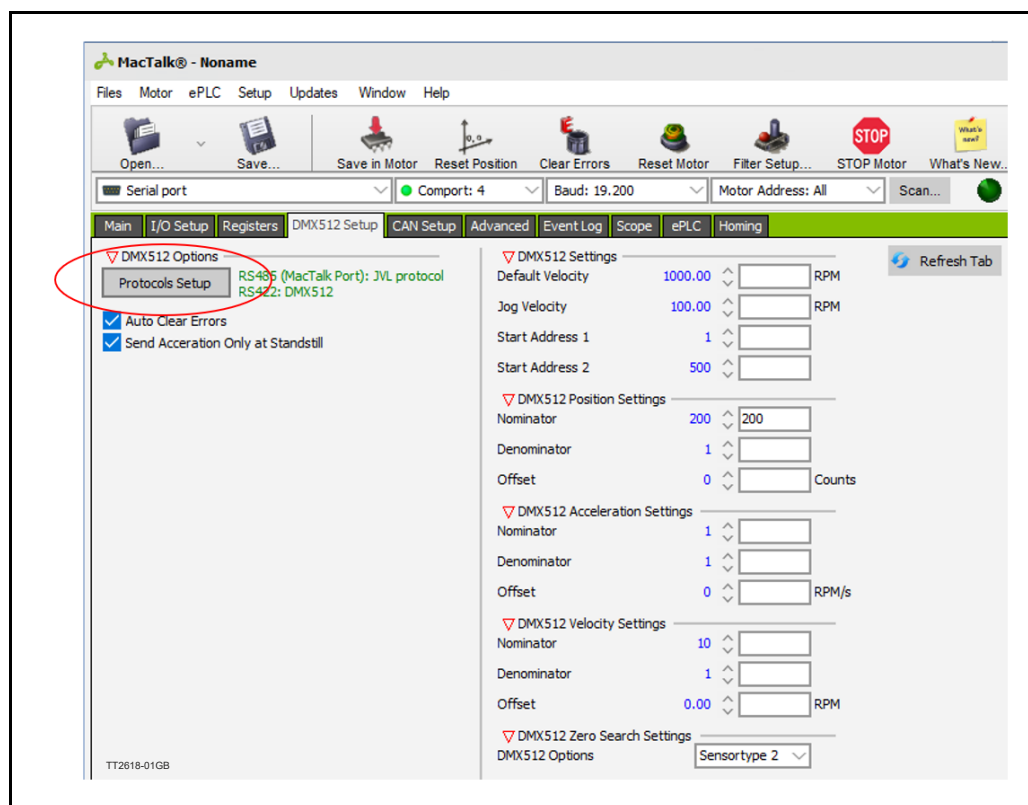
This document describes how to use the DMX512 protocol, widely used for theatre equipment, with the stepper motor. The DMX512 protocol is a standard feature in all MIS and MIL motors. (Working from Firmware version 5.03)

### 13.1.2 General description

The DMX interface supports control of Position, Mode, Acceleration and Velocity of the motor. It is intended to be used with a set of motors, sharing the same Acceleration and Velocity, but with different positions and separate mode control.

### 13.1.3 How to enable DMX 512

The DMX functionality is enabled by selecting the DMX 512 setup tab. Click on the protocol setup button in the red circle in the figure. Select the interface that should be used for DMX. Either RS485 or RS422. Set the parameters requested for the application of the motor. Save the settings in the motor.



### 13.1.4 DMX settings

To set up several motors to the same values, except the DMX start address, save the motor parameters to a .MAC file from MacTalk via the Save button, and then load that file into the next motor using the Open button.

**13.1.5 Homing**

The DMX homing uses the build in homing.

The following parameters need to be set up before using DMX homing.

- Torque if mechanical/Torque homing is used.
- Velocity search.
- Velocity Crawl if sensor homing is used.
- Homing sensor input if sensor homing is used.

See homing commands in the Control channel section [Control channel, page 378](#).

**13.1.6 DMX Address names**

The DMX can address four values: Position value, Control channel, Acceleration and maximum Velocity.

Using the MacTalk program, two separate DMX start addresses must be configured.

Start Address 1 (called SAI below) points to the three bytes holding Position (coarse + fine) and the Control Channel, which is used to select the motors operating mode to either Homing mode or Position mode.

Start Address 2 (SA2) points to the two bytes holding Acceleration and Velocity.

It is intended that each motor has separate values for SAI, but all motors share the value of SA2 and thus will use the same Acceleration and Velocity.

SAI + 0 Position value (coarse)

SAI + 1 Position value (fine)

SAI + 2 Control channel

SA2 + 0 Acceleration

SA2 + 1 Maximum velocity

**13.1.7 Scaling of parameters**

The 16-bit position value and the 8-bit Acceleration and Velocity values must be scaled and possibly offset to fit their working range in motor counts.

This is done by multiplying each value by a separate fraction and then adding an offset. All of these scaling parameters are configured in the MacTalk application with the DMX512 Setup Tab.

The formulas used are:

$P\_SOLL$  (Target position) =

$DMX\ Position\ (16\text{-bit}) * DmxPosNom / DmxPosDen + DmxPosOffset\ (32\text{-bit})$

$A\_SOLL$  (Commanded acceleration/deceleration) =

$DMX\ Acceleration\ (8\text{-bit}) * DmxAccNom / DmxAccDen + DmxAccOffset\ (16\text{-bit})$

$V\_SOLL$  (Commanded velocity) =

$DMX\ Velocity\ (8\text{-bit}) * DmxVelNom / DmxVelDen + DmxVelOffset\ (16\text{-bit})$

Note that all of the three Offsets and the Nominators for Position and Velocity are signed values, while all Denominators and the Acceleration offset are unsigned. MacTalk will limit the ranges to prevent invalid entries, such as division by zero.

# 13 DMX512 Protocol and interface

## 13.1.8 Control channel

Setting the MODE\_REG is done using the DMX Control Channel in a special way.

00-09	Set Motor in Passive mode
60-69	Set motor in Position mode
180-189	Run homing
200-209	Reset error

The motor is kept in its configured start-up mode until 10 identical values in the range 180..189 have been received on the DMX Control Channel. Then the motor starts a homing operation, which is terminated when the homing is completed.

Homing will not be aborted if the value of the Control Channel is changed away from the 180..189 range after Homing was started. Homing can be repeated at any time.

The motor will then stay in Position mode until the next Homing operation is requested (or until an error occurs). The homing procedure can be disabled with the DMX512 tab in MacTalk. If the motor is with absolute encoder the homing should be disabled.

When the value of the Control Channel is anywhere in the range 60..69, the scaled values for Position, Acceleration and Velocity are transferred from the DMX bus to the basic motor every time a new value is detected on the DMX bus. When the Control Channel values are neither in the range 60..69 or 180..189, the motor is kept in Position mode, but no new values for Position/Acceleration/Velocity are transferred.

IF the setting "auto clear errors" is set, the motor will clear any errors before any DMX command is carried out. Note that if the source(s) of the error(s) are still present, the errors will be set again immediately.

## 13.1.9 Jogging using digital inputs

When input 1 is switched from Off to On, it will set the motor into Jogging (Velocity) mode at an initial speed of zero RPM and ignore any data from the DMX bus. It can be selected to also send a Reset Errors command at this time. While in Jogging mode, setting input 2 On will make the motor run at the velocity configured in MacTalk as Jog Velocity. Setting input 2 Off in jogging mode will set the velocity to zero. While in Jogging mode, setting input 3 On will make the motor run at the velocity configured in MacTalk as Jog Velocity, but in the opposite direction of input 2. Setting input 3 Off in jogging mode will set the velocity to zero.

IN1	IN2	IN3	Mode/Function
OFF	x	x	Passive mode
ON	OFF	OFF	Velocity mode - Velocity = 0
ON	ON	OFF	Velocity mode - Velocity (+) set for jogging in the DMX tab
ON	OFF	ON	Velocity mode - Velocity (-) set for jogging in the DMX tab

DMX512 can be configured to also send a Reset Errors command every time one of inputs 2 and 3 is switch On/Off state.

When input 1 is switched from On to Off, Jogging mode is ended, the maximum velocity is set to the value configured in MacTalk as Default Velocity.

The motors target position is set [close] to the actual position resulting from the jogging. The DMX must perform a Homing operation after jogging. This is done to prevent that the motor will start to move to its last received DMX target position as soon as the jogging enable, input 1, is set back to Off.

**13.1.10 Other considerations**

The DMX serial channel will accept either one or two stop bits per byte to be compatible also with DMX masters that send only one stop bit.

Only DMX packets with type zero will be used. Other packet types, like SIP (System Information Packets) will be ignored.

The firmware was tested during development using the freeware Windows application Q Light Controller Version 3.2.0-3 with a USB RS-485 adapter.

DMX512 can be configured so that the module will send new values for Velocity and Acceleration to the motor only when a new value for Position is received.

In other words, a new Position value will cause the values for Position, Velocity and Acceleration in the same DMX telegram to be sent to the motor as a set.

This is so a complete movement to one or more motors can be started and the motor will complete the movement using the same Velocity and Acceleration, even if other values are sent on the DMX bus to configure other motors to perform movements using other values for Velocity and/or Acceleration.





## 14.1 Step motor controllers (SMCxx)

Sorry - Pictures soon coming

The compact step motor controller SMC66 and SMC85 are designed for positioning and speed control of stepper motors. SMC66 is a PCB with dimensions 34x65 mm and SMC85 is 78x86mm.

Both PCB's are used in the MIS motors, forming a complete integrated step motor. It may also be used with other types of step motors according to customers requirements. The basic features of the controller are overall similar to the MIS motors feature list:

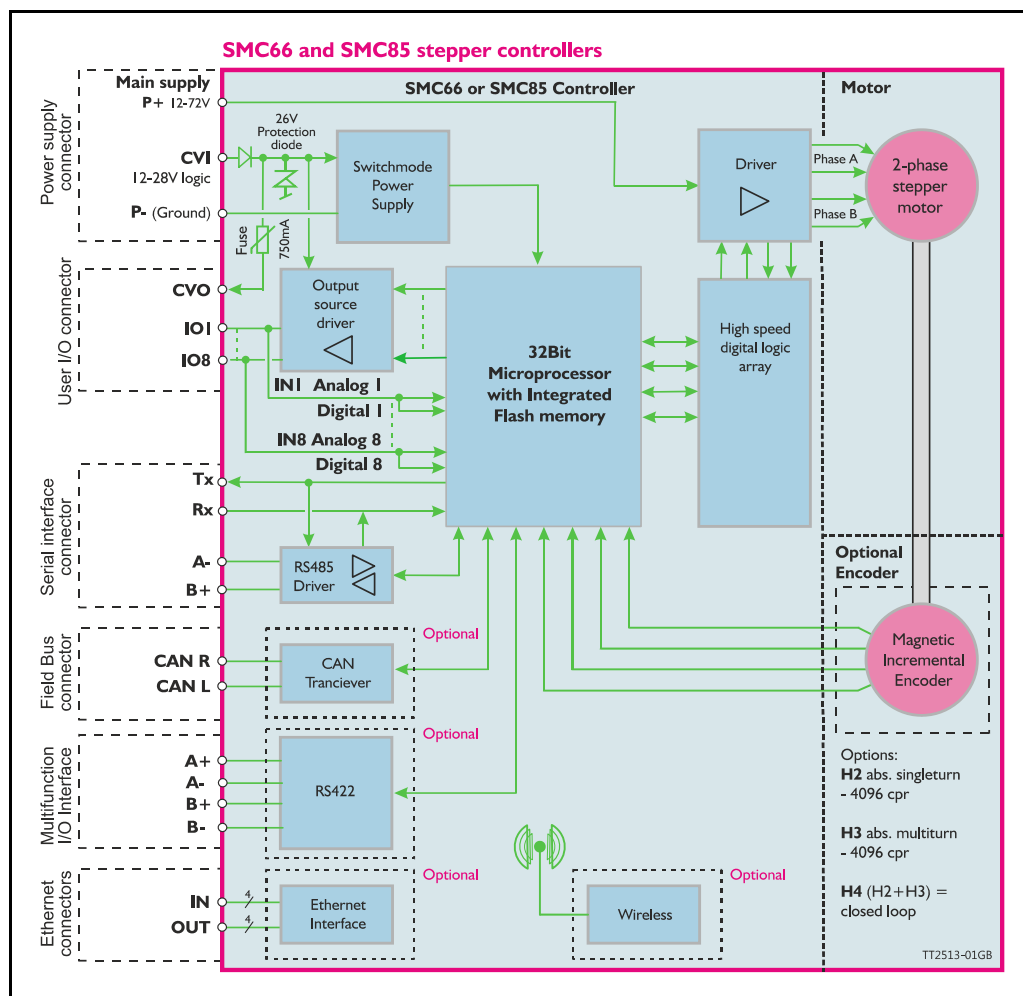
- Serial RS485 or 5V serial position controller.
- Build-in mini PLC with graphic programming.
- Option for CANbus, CANopen DS-301. Fully ISO 11898-2:2016 compliant/(DSP-402 in development).
- Options for EthernetIP, Profinet, Powerlink, ModbusTCP, SercosIII and EtherCAT.
- A dual supply facility is available so that position and parameters are maintained at emergency stop.
- Electronic Gear mode.
- MACmotor protocol so MAC servomotors and MIS stepper motors can be connected on the same RS485 bus.
- Command for easy PLC/PC setup and communication.
- Power supply 12-72 VDC.
- Extremely high torque vs speed - up to 3000 RPM with good performance.
- Fixed 409600 steps per revolution
- Built-in 32Bit  $\mu$ processor with 8 In/Out that can be configured as inputs, PNP outputs or analogue inputs. 5V serial and RS485 interface for set up and programming.
- MODBUS interface.
- 9.6kbit/sec. to 1Mb/sec. communication.

Benefits when using the SMC66 and SMC85 controllers:

- De-central intelligence.
- Simple installation. No cables between motor and controller/driver.
- EMC safe. Switching noise remains within motor.
- Compact. Does not take space in the control cabinet.
- Option: Closed loop feature by means of magnetic encoder with resolution of up to 4096 pulses/rev. (H2 or H4 option)
- Option: Absolute multi turn encoder for keeping the position permanent also during power down. (H3 or H4).
- Vibration tested at 4G in 3 axis and shock tested at 15G in 3 axis according to IEC60068.
- Interface possibilities:
- From PC/PLC with serial commands via 5V serial or RS485.
- Pulse/direction input. Encoder output.
- CANopen.
- 8 I/O, 5-30VDC that can be configured as Inputs, Outputs or analogue inputs.
- Wireless options: WiFi, Bluetooth and Zigbee.

# 14.1 Step motor controllers (SMCxx)

## 14.1.1 Block diagram, Positioning/Speed Control



# 14.1 Step motor controllers (SMCxx)

Only SMC85

## 14.1.2 SMC85 Connector overview

The connections to the various connectors of the SMC85 PCB board is shown below. Note that GND and P- are connected together internally.

### Connector J1 - pin description

GND (ground for I/O's same as P-)	1	2	Do not connect !
GND (ground for I/O's same as P-)	3	4	Do not connect !
GND (ground for I/O's same as P-)	5	6	Do not connect !
GND (ground for I/O's same as P-)	7	8	Do not connect !
GND (ground for I/O's same as P-)	9	10	Do not connect !
Not used - do not connect	11	12	(Conn. internally to CVI) CVO
+5V out - max. 50mA !	13	14	(I/O channel 1) IO1
RX (RS232 receive - 3.3V !)	15	16	(I/O channel 2) IO2
TX (RS232 transmit - 3.3V !)	17	18	(I/O channel 3) IO3
CAN_H (optional)	19	20	(I/O channel 4) IO4
CAN_L (optional)	21	22	(I/O channel 5) IO5
RS485 A- (RS485 setup interface)	23	24	(I/O channel 6) IO6
RS485 B+ (RS485 setup interface)	25	26	(I/O channel 7) IO7
A1+ (RS422)	27	28	(I/O channel 8) IO8
A1- (RS422)	29	30	Do not connect !
B1+ (RS422)	31	32	(H2 enc. output optional) EA
B1- (RS422)	33	34	(H2 enc. output optional) EB
GND (ground for I/O's same as P-)	35	36	(H2 enc. output optional) EI

### Power in - J3

J3 pin 3 - P- (power GND)  
J3 pin 2 - CVI (control supply 12-30VDC)  
J3 pin 1 - P+ (power 12-72VDC) max 90V

Mating conn.  
Molex series  
SPOX™ and KK®

Pin 1

I/O - J1

Phase B-  
Phase B+  
Phase A-  
Phase A+

### Motor output - J2

Optional conn.  
Molex series  
Mini-Fit® Jr.  
fits into the board

### SMC85C1AA - Bottom side

Mounting holes (x4) Ø3.0mm  
All have solid contact with  
GND (ground)

TT2350-02GB - 20171017

JVL have a range of stepper motors with an integrated stepper driver (none programmable) that are cost effective and easy to use.

JVL also offer a range of stand alone stepper drivers in the working range 3 to 9 ARMS output current and 24 to 150VDC supply.

**Please contact your JVL distributor for further information.**





Main Supply Voltage (P+)	Voltage Range	Nominal +7-72VDC (absolute max. = 90VDC ripple free). Min. voltage must be without ripple			
	Current consumption	Power supply current requirements = minimum 3A recommended. Please refer to the power supply chapter. The actual power supply currents will depend on voltage and load. Please consult the power supply chapter in this manual.			
Control Voltage (CVI)	Range	+7 to +30VDC. Supply current 50-130mA@24VDC (depends which options installed) Supply for the internal control circuitry, the output driver (IO1-8), and feed-back circuits (if present). If the motor connected or passive mode: 100mA. Note: Battery supply 12VDC is also possible. The function of the motor is not affected before supply is below 7VDC. Please make sure that no voltages below this point is present since the processor will reset/restart if "dips" exist at the supply.			
Mechanical	Motor type:	MIS171S/Q	MIS173S/Q	MIS176S/Q	
	Holding torque Nm [Oz-In]	0.36 [50.9]	0.56 [79.3]	0.80 [113.3]	
	Inertia kgcm² [Oz-In-Sec²]	0.020 / [0.000283]	0.054 / [0.000793]	0.102 / [0.00144]	
	Weight - kg [lb]	0.54 [1.19]	0.68 [1.5]	0.9 [1.99]	
	Max. axial shaft force N	-	-	-	
	Max. radial shaft force (N) (applied 20 mm from flange)	-	-	-	
Analogue Input	Number/Resolution	8 inputs available / 12 Bit. The analogue inputs are shared with general I/O's			
	Voltage Range	0 to +5.00 VDC			
General Purpose I/O	Number/Type	8 I/O's available. Can be setup as source output or digital input. All +24V tolerant.			
	Input levels / bandwidth	An I/O when used as input is activated when Vin >2.8V. The bandwidth is 100 kHz max.			
	Output Source Current	Up to 300 mA per output. All 8 outputs can deliver this at the same time.			
	Protection	Over Temp. Short Circuit. Transient. Over Voltage. Inductive Clamp.			
	Input Filter	0.1 or 1 to 100 ms			
Multifunction I/O's	General info	The multifunction I/O ports are based on the RS422/RS485 standard which means its a differential bidirectional line interface. The bandwidth is 10 Mbit/sec (10 MHz). Min./max voltage applied to either line is -7V to +12V.			
	Output performance	Min. 2.0V differential between the outputs @ 100 Ohm load between the lines.			
	Input performance	Logic 0 or 1 when voltage difference between lines is >0.2V or >-0.2V. Hyst. = 50 mV			
Communication	Type (Standard)	RS485			
	Baud Rate	9.6 to 921.6 kbps			
	Type (Optional)	CANopen DS301 (VS3.0), 2.0A Active. Ethernet: Powerlink, EthernetIP, Profinet, EtherCAT, SercosIII, ModbusTCP			
	Isolation	RS485/RS422/CANopen : None / Ethernet : Yes - withstand up to 500VDC in potential difference.			
	Features	Node Guarding, heartbeat, SDOs, PDOs (Dynamic mapping)			
Motion	Open Loop operation	Operation modes	Passive, Position, Gear, Velocity		
		Resolution per rev.	409600 counts		
	Internal Encoder (option: H2)	Type	Internal, magnetic, absolute 1 rev. Closed loop ready.		
		Resolution per rev.	4096 counts / 1024 lines (quadrature output)		
	Internal Encoder (option: H3)	Type	Internal, magnetic, absolute multiturn.		
		Resolution per rev.	Displayed: 409600 counts - internal: 1024 counts		
	Internal Encoder (option: H4)	Type	Internal, magnetic, absolute multiturn Closed loop ready.		
		Resolution per rev.	Displayed: 409600 counts - internal: 4096 counts		
	Counters	Type	Position, Encoder / 32 Bit		
		Edge Rate (Max.)	12.0 MHz		
	Velocity	Range	-3000.00 to +3000.00 RPM		
		Resolution	0.01 RPM		
		Precision	±50 ppm		
	Accel./Deceleration.	Range	1 - 500000 RPM/s		
		Resolution	1 RPM/s		
	Electronic Gearing	Range/Resolution (External Clock/encoder In)	Input (ext.) / Output (int.) = 1/409600 up to 409600/1		
Software	Program Storage	Type/Size	Flash 3072 Bytes		
	User Registers	2248 Bytes/32 bits			
	User Program Variables	Up to 224			
	Math Functions	+, -, x, /, >, <, =, <=, >=, AND, OR, XOR, NOT, I, &, ^.			
	Branch Functions	Branch & Call			
	General Purpose I/O Functions	Inputs	Home, Limit Plus, Limit Minus, Analogue In, General Purpose		
		Outputs	Moving, Fault, general Purpose		
	Party Mode Addresses	254			
Encoder Functions (options)	Stall Detection, Position maintenance, Find Index, Closed loop, Absolute Multiturn encoder				
Thermal	Operating/storage temp.	Ambient 0 to +40°C (32-104°F)/ -20 to +85°C. (-4 to 185 °F) (Humidity 90%). A warning message is generated if the internal temperature passes 80°C The motor is set in passive mode if the temperature passes 90°C and an error message is generated.			

# 16.2

# MIS23x Technical Data

Only MIS23x

Main Supply Voltage (P+)	Voltage Range	Nominal +7-72VDC (absolute max. = 90VDC ripple free). Min. voltage must be without ripple				
	Current consumption	Power supply current requirements = minimum 3A recommended. Please refer to the power supply chapter. The actual power supply currents will depend on voltage and load.				
Control Voltage (CVI)	Range	+7 to + 30VDC. Supply current 50-130mA@24VDC (depends which options installed) Supply for the internal control circuitry, the output driver (IO1-8), and feed-back circuits (if present). If the motor connected or passive mode: 100mA. Note: Battery supply 12VDC is also possible. The function of the motor is not affected before supply is below 7VDC. Please make sure that no voltages below this point is present since the processor will reset/restart if "dips" exist at the supply.				
Mechanical	Motor type:	MIS231S/Q	MIS231T/R	MIS232S/Q	MIS232T/R	MIS234S/Q
	Holding torque Nm [Oz-In]	0.97 [137.4]	1.16 [164.3]	1.97 [279.0]	2.53 [358.3]	3.08 [436.2]
	Inertia kgcm² [Oz-In-Sec²]	0.3 [0.00423]	0.3 [0.00423]	0.48 [0.00677]	0.48 [0.00677]	0.65 [0.0092]
	Weight - kg [lb]	1.1 [2.43]	1.1 [2.43]	1.4 [3.09]	1.4 [3.09]	2.0 [4.41]
	Max. axial shaft force N	15	15	15	15	15
	Max. radial shaft force (N) (applied 20 mm from flange)	75	75	75	75	75
Analogue Input	Number/Resolution	8 inputs available / 12 Bit. The analogue inputs are shared with general I/O's				
	Voltage Range	0 to +5.00 VDC				
General Purpose I/O	Number/Type	8 I/O's available. Can be setup as source output or digital input. All +24V tolerant.				
	Input levels / bandwidth	An I/O when used as input is activated when Vin >2.8V. The bandwidth is 100 kHz max.				
	Output Source Current	Up to 300 mA per output. All 8 outputs can deliver this at the same time.				
	Protection	Over Temp. Short Circuit. Transient. Over Voltage. Inductive Clamp.				
	Input Filter	0.1 or 1 to 100 ms				
Multifunction I/O's	General info	The multifunction I/O ports are based on the RS422/RS485 standard which means its a differential bidirectional line interface. The bandwidth is 10 Mbit/sec (10 MHz). Min./max voltage applied to either line is -7V to +12V.				
	Output performance	Min. 2.0V differential between the outputs @ 100 Ohm load between the lines.				
	Input performance	Logic 0 or 1 when voltage difference between lines is >0.2V or >-0.2V. Hyst. = 50 mV				
Communication	Type (Standard)	RS485				
	Baud Rate	9.6 to 921.6 kbps				
	Type (Optional)	CANopen DS301 (VS3.0), 2.0A Active. Ethernet: Powerlink, EthernetIP, Profinet, EtherCAT, SercosIII, ModbusTCP				
	Isolation	RS485/RS422/CANopen : None / Ethernet : Yes - withstand up to 500VDC in potential difference.				
	Features	Node Guarding, heartbeat, SDOs, PDOs (Dynamic mapping)				
Motion	Open Loop operation	Operation modes	Passive, Position, Gear, Velocity			
		Resolution per rev.	409600 counts			
	Internal Encoder (option: H2)	Type	Internal, magnetic, absolute 1 rev. Closed loop ready.			
		Resolution per rev.	4096 counts / 1024 lines (quadrature output)			
	Internal Encoder (option: H3)	Type	Internal, magnetic, absolute multiturn.			
		Resolution per rev.	Displayed: 409600 counts - internal: 1024 counts			
	Internal Encoder (option: H4)	Type	Internal, magnetic, absolute multiturn Closed loop ready.			
		Resolution per rev.	Displayed: 409600 counts - internal: 4096 counts			
	Counters	Type	Position, Encoder / 32 Bit			
		Edge Rate (Max.)	12.0 MHz			
	Velocity	Range	-3000.00 to +3000.00 RPM			
		Resolution	0.01 RPM			
		Precision	±50 ppm			
	Accel./Deceleration.	Range	1 - 500000 RPM/s			
		Resolution	1 RPM/s			
Software	Electronic Gearing	Range/Resolution (External Clock/encoder In)	Input (ext.) / Output (int.) = 1/409600 up to 409600/1			
	Program Storage	Type/Size	Flash 3072 Bytes			
	User Registers	2248 Bytes/32 bits				
	User Program Variables	Up to 224				
	Math Functions	+, -, x, /, >, <, =, <=, >=, AND, OR, XOR, NOT, I, &, ^.				
	Branch Functions	Branch & Call				
	General Purpose I/O Functions	Inputs	Home, Limit Plus, Limit Minus, Analogue In, General Purpose			
		Outputs	Moving, Fault, general Purpose			
	Party Mode Addresses	254				
Encoder Functions (options)	Stall Detection, Position maintenance, Find Index, Closed loop, Absolute Multiturn encoder					
Thermal	Operating/storage temp.	Ambient 0 to +40°C (32-104°F)/ -20 to +85°C. (-4 to 185 °F) (Humidity 90%). A warning message is generated if the internal temperature passes 80°C The motor is set in passive mode if the temperature passes 90°C and an error message is generated.				

# 16.3

# MIS34x Technical Data

Only MIS34x

Main Supply Voltage (P+ terminal)	Voltage Range	Nominal +7-72VDC (absolute max. = 90VDC ripple free). Min voltage must be without ripple.			
	Current consumption	Power supply current requirements = 10ARMS (max.). Actual power supply currents will depend on voltage and load. When motor is in "Passive" mode the current consumption is < 10mARMS@24VDC supply			
Control Voltage (CVI terminal)	Voltage Range	Control voltage input to maintain power to the internal control circuitry including output driver and feed-back circuits and interface (all except the motor power for turning the motor). Nominal voltage range 7-30VDC. Maximum 32VDC. Min voltage must be without ripple.			
	Current consumption	Typical 45mA@24.0VDC when motor is in passive mode and none of the I/O's are connected.			
Mechanical	Motor type:	MIS340	MIS341	MIS342	MIS343
	Holding torque - Nm [Oz-In]	3.0 [424]	6.1 [863]	9.0 [1274]	12.0 [1727]
	Inertia - kgcm² [Oz-In-Sec²]	1.4 [0.0198]	2.7 [0.0381]	4.0 [0.0564]	4.0 [0.0564]
	Weight - kg [lb]	2.05 [4.52]	3.13 [6.9]	4.2 [9.26]	4.5 [10.71]
	Max. axial shaft force	115N			
	Max. radial shaft force	180N applied 12.5mm from shaft end			
Analogue Input	Number/Resolution	8 inputs available / 12 Bit. The analogue inputs are shared with general I/O's			
	Voltage Range	0 to +5.00 VDC			
General Purpose I/O	Number/Type	8 I/O's available. Can be setup as source output or digital input. All +24V tolerant.			
	Input levels / bandwidth	An I/O when used as input is activated when Vin >2.8V. The bandwidth is 100 kHz max.			
	Output Source Current	Up to 300 mA per output. All 8 outputs can deliver this at the same time.			
	Protection	Over Temp. Short Circuit. Transient. Over Voltage. Inductive Clamp.			
	Input Filter	0.1 or 1 to 100 ms			
Multifunction I/O's	General info	The multifunction I/O ports are based on the RS422/RS485 standard which means its a differential bidirectional line interface. The bandwidth is 10 Mbit/sec (10 MHz). Min./max voltage applied to either line is -7V to +12V.			
	Output performance	Min. 2.0V differential between the outputs @ 100 Ohm load between the lines.			
	Input performance	Logic 0 or 1 when voltage difference between lines is >0.2V or >-0.2V. Hyst. = 50 mV			
Communication	Type (Standard)	RS485			
	Baud Rate	9.6 to 921.6 kbps			
	Type (Optional)	CANopen DS301 (VS3.0), 2.0A Active. Ethernet: Powerlink, EthernetIP, Profinet, EtherCAT, SercosIII, ModbusTCP			
	Isolation	RS485/RS422/CANopen : None / Ethernet : Yes - withstand up to 500VDC in potential difference.			
	Features	Node Guarding, heartbeat, SDOs, PDOs (Dynamic mapping)			
	Internal Encoder (option: <b>H2</b> )	Type	Internal, magnetic, absolute 1 rev. Closed loop ready.		
		Resolution per rev.	4096 counts / 1024 lines (quadrature output)		
	Internal Encoder (option: <b>H3</b> )	Type	Internal, magnetic, absolute multiturn.		
		Resolution per rev.	Displayed: 409600 counts - internal: 1024 counts		
	Internal Encoder (option: <b>H4</b> )	Type	Internal, magnetic, absolute multiturn Closed loop ready.		
		Resolution per rev.	Displayed: 409600 counts - internal: 4096 counts		
	Counters	Type	Position, Encoder/32 Bit		
		Edge Rate (Max.)	12.0 MHz		
	Velocity	Range	0.01 to 3000.00 RPM		
		Resolution	0.01 RPM		
		Precision	±50ppm		
	Accel./Deceleration	Range	1 - 500000 RPM/s		
		Resolution	1 RPM/s		
	Electronic Gearing	Range/Resolution (External Clock In)	Input (ext.) / Output (int.) = 1/409600 up to 409600/1		
Software	Program Storage	Type/Size	Flash 3072 Bytes		
	User Registers	2248 Bytes/32 bits			
	User Program Variables	Up to 224			
	Math Functions	+, -, x, /, >, <, =, <=, >=, AND, OR, XOR, NOT, I, &, ^.			
	Branch Functions	Branch & Call			
	General Purpose I/O Functions	Inputs	Home, Limit Plus, Limit Minus, Analogue In, General Purpose		
		Outputs	Moving, Fault, general Purpose		
	Party Mode Addresses	254			
	Encoder Functions (options)	Stall Detection, Position maintenance, Find Index, Closed loop, Absolute Multiturn encoder			
Thermal	Operating/storage temp.	Ambient 0 to +40°C (32-104°F)/ -20 to +85°C. (-4 to 185 °F) (Humidity 90%). A warning message is generated if the internal temperature passes 80°C The motor is set in passive mode if the temperature passes 90°C and an error message is generated.			

# 16.4

# MIS43x Technical Data

Only MIS43x

Main Supply Voltage (P+ terminal)	Voltage Range	Nominal +7-72VDC (absolute max. = 90VDC ripple free). Min voltage must be without ripple.		
	Current consumption	Power supply current requirements = 10ARMS (max.). Actual power supply currents will depend on voltage and load. When motor is in "Passive" mode the current consumption is < 10mARMS@24VDC supply		
Control Voltage (CVI terminal)	Voltage Range	Control voltage input to maintain power to the internal control circuitry including output driver and feed-back circuits and interface (all except the motor power for turning the motor). Nominal voltage range 7-30VDC. Maximum 32VDC. Min voltage must be without ripple.		
	Current consumption	Typical 45mA@24.0VDC when motor is in passive mode and none of the I/O's are connected.		
Mechanical	Motor type:	MIS430	MIS432	
	Holding torque - Nm [Oz-In]	10.0 [1416]	25.0 [3540]	
	Inertia - kgcm <sup>2</sup> [Oz-In-Sec <sup>2</sup> ]	5.5 [0.0779]	16.2 [0.229]	
	Weight - kg [lb]	5.5 [12.13]	12.2 [26.9]	
	Max. axial shaft force	80 N		
	Max. radial shaft force	At 5mm max 640N, At 10mm max 425 N, At 15mm max 320N, At 20mm max 240N		
Analogue Input	Number/Resolution	8 inputs available / 12 Bit. The analogue inputs are shared with general I/O's		
	Voltage Range	0 to +5.00 VDC		
General Purpose I/O	Number/Type	8 I/O's available. Can be setup as source output or digital input. All +24V tolerant.		
	Input levels / bandwidth	An I/O when used as input is activated when Vin >2.8V. The bandwidth is 100 kHz max.		
	Output Source Current	Up to 300 mA per output. All 8 outputs can deliver this at the same time.		
	Protection	Over Temp. Short Circuit. Transient. Over Voltage. Inductive Clamp.		
	Input Filter	0.1 or 1 to 100 ms		
Multifunction I/O's	General info	The multifunction I/O ports are based on the RS422/RS485 standard which means its a differential bidirectional line interface. The bandwidth is 10 Mbit/sec (10 MHz). Min./max voltage applied to either line is -7V to +12V.		
	Output performance	Min. 2.0V differential between the outputs @ 100 Ohm load between the lines.		
	Input performance	Logic 0 or 1 when voltage difference between lines is >0.2V or >-0.2V. Hyst. = 50 mV		
Communication	Type (Standard)	RS485		
	Baud Rate	9.6 to 921.6 kbps		
	Type (Optional)	CANopen DS301 (VS3.0), 2,0A Active. Ethernet: Powerlink, EthernetIP, Profinet, EtherCAT, SercosIII, ModbusTCP		
	Isolation	RS485/RS422/CANopen : None / Ethernet : Yes - withstand up to 500VDC in potential difference.		
	Features	Node Guarding, heartbeat, SDOs, PDOs (Dynamic mapping)		
		Internal Encoder (option: <b>H2</b> )	Type	Internal, magnetic, absolute 1 rev. Closed loop ready.
Resolution per rev.			4096 counts / 1024 lines (quadrature output)	
Internal Encoder (option: <b>H3</b> )		Type	Internal, magnetic, absolute multiturn.	
		Resolution per rev.	Displayed: 409600 counts - internal: 1024 counts	
Internal Encoder (option: <b>H4</b> )		Type	Internal, magnetic, absolute multiturn Closed loop ready.	
		Resolution per rev.	Displayed: 409600 counts - internal: 4096 counts	
Counters		Type	Position, Encoder/32 Bit	
		Edge Rate (Max.)	12.0 MHz	
Velocity		Range	0.01 to 3000.00 RPM	
		Resolution	0.01 RPM	
		Precision	± 50ppm	
Accel./Deceleration		Range	1 - 500000 RPM/s	
		Resolution	1 RPM/s	
Electronic Gearing		Range/Resolution (External Clock In)	Input (ext.) / Output (int.) = 1/409600 up to 409600/1	
Software	Program Storage	Type/Size	Flash 3072 Bytes	
	User Registers	2248 Bytes/32 bits		
	User Program Variables	Up to 224		
	Math Functions	+, -, x, /, >, <, =, <=, >=, AND, OR, XOR, NOT, !, &, ^.		
	Branch Functions	Branch & Call		
	General Purpose I/O Functions	Inputs	Home, Limit Plus, Limit Minus, Analogue In, General Purpose	
		Outputs	Moving, Fault, general Purpose	
	Party Mode Addresses	254		
Encoder Functions (options)	Stall Detection, Position maintenance, Find Index, Closed loop, Absolute Multiturn encoder			
Thermal	Operating/storage temp.	Ambient 0 to +40°C (32-104°F)/ -20 to +85°C. (-4 to 185 °F) (Humidity 90%). A warning message is generated if the internal temperature passes 80°C The motor is set in passive mode if the temperature passes 90°C and an error message is generated.		

## 16.5

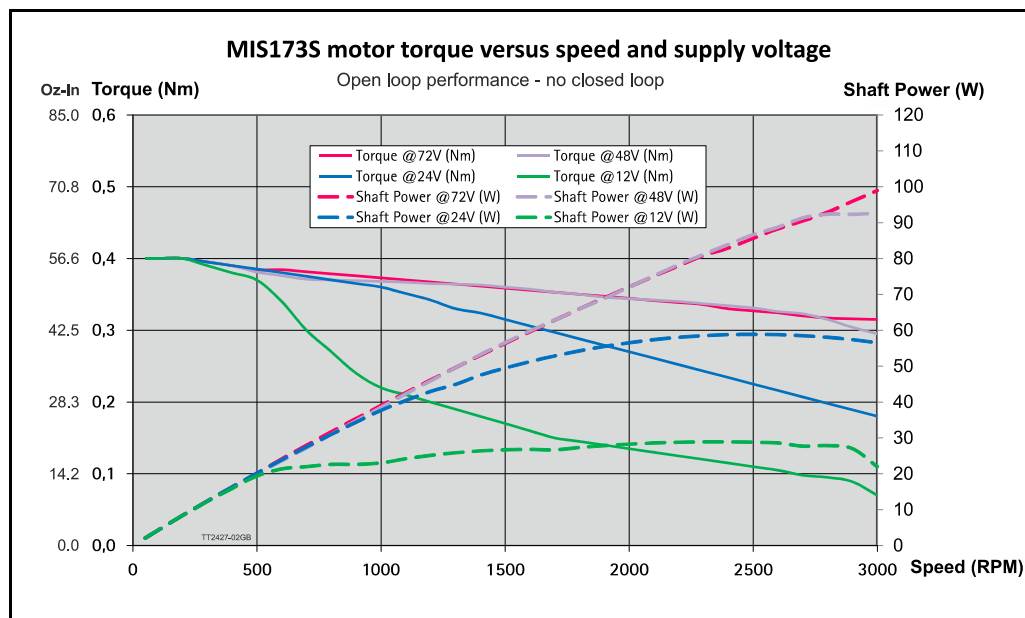
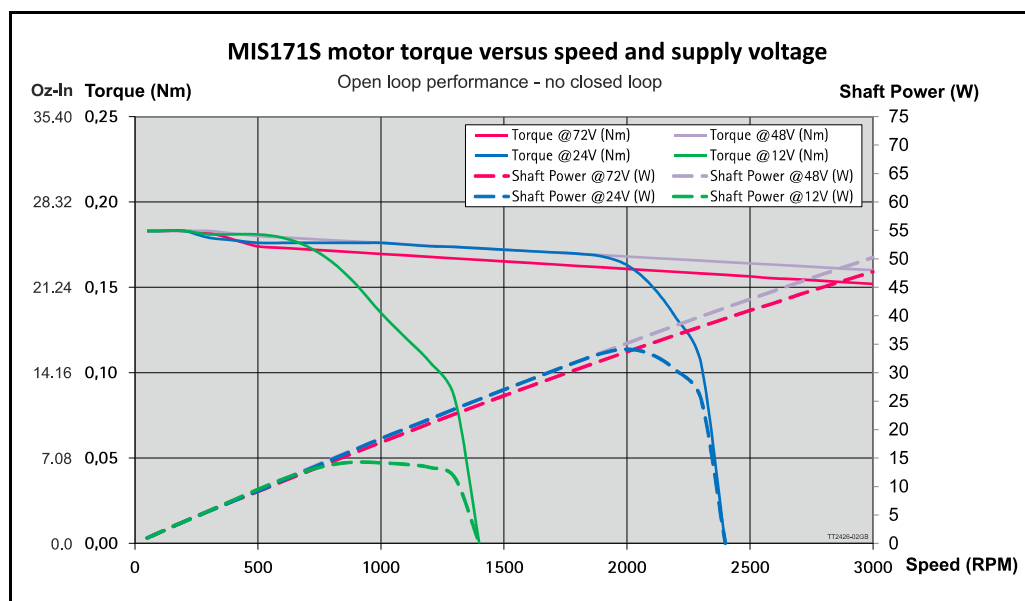
## Torque Curves

### 16.5.1 MIS171S, 173S and 176S Torque and power curves

Below the torque performance for the MIS17x motor family is shown. As it can be seen the supply voltage have a significant influence at the torque performance at higher speeds.

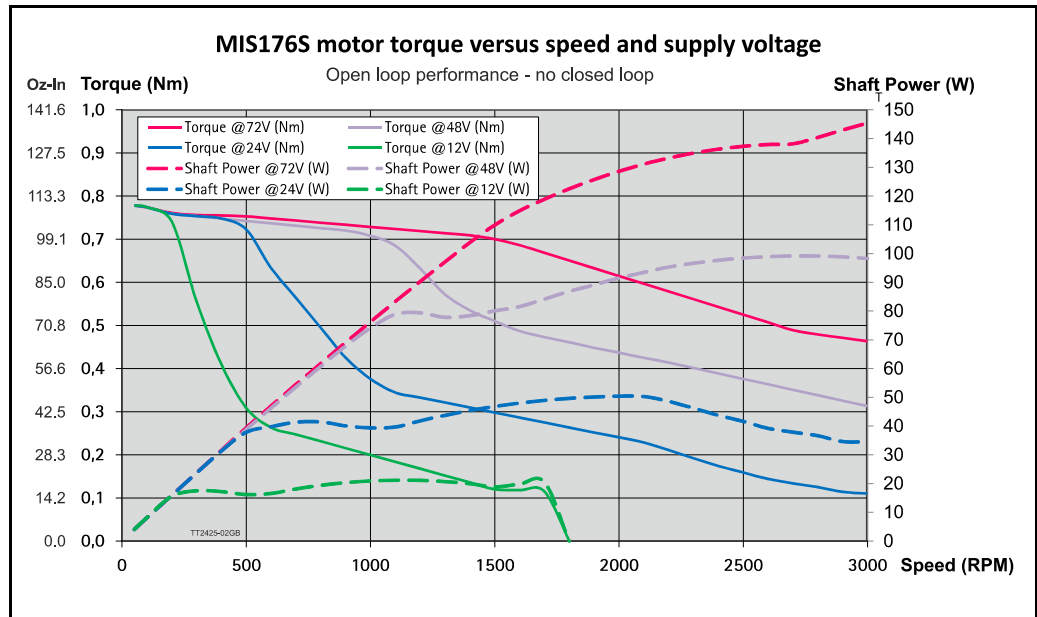
Please make sure to use a supply voltage which is appropriate for the actual application. Also make sure that the supply voltage is stable without too much ripple since voltage dips can cause the motor to stall and loose position.

Also the shaft power (mechanical power at motor shaft) is shown.



## 16.5

## Torque Curves

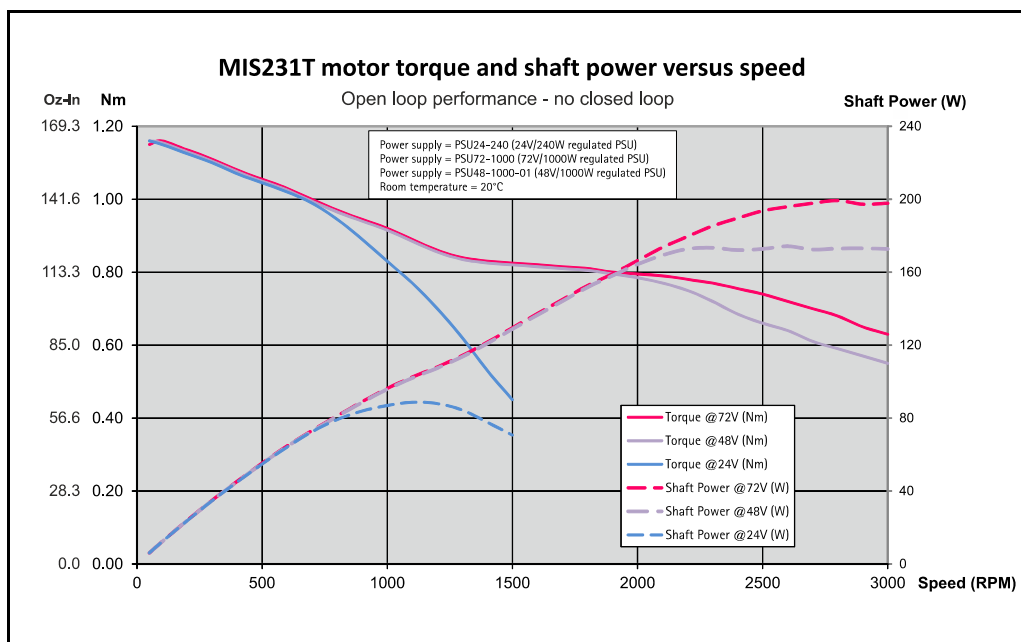
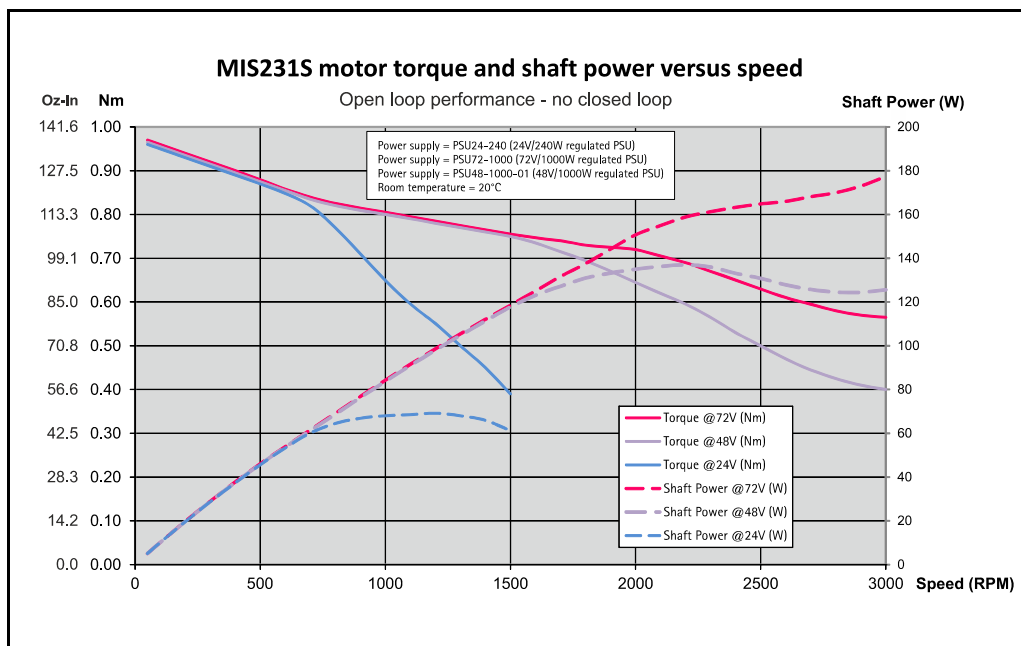


## 16.5

## Torque Curves

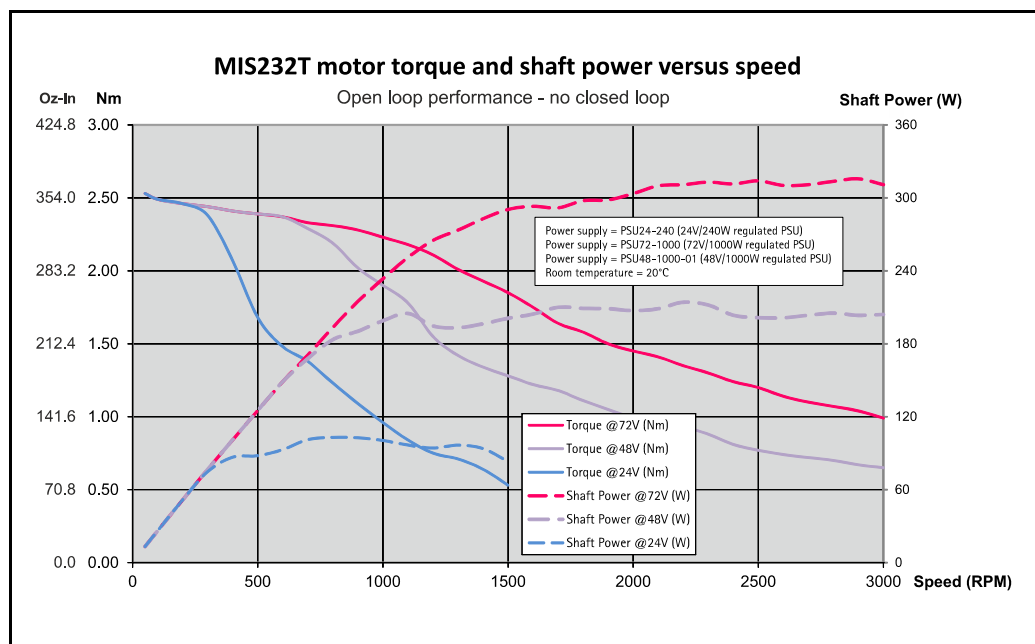
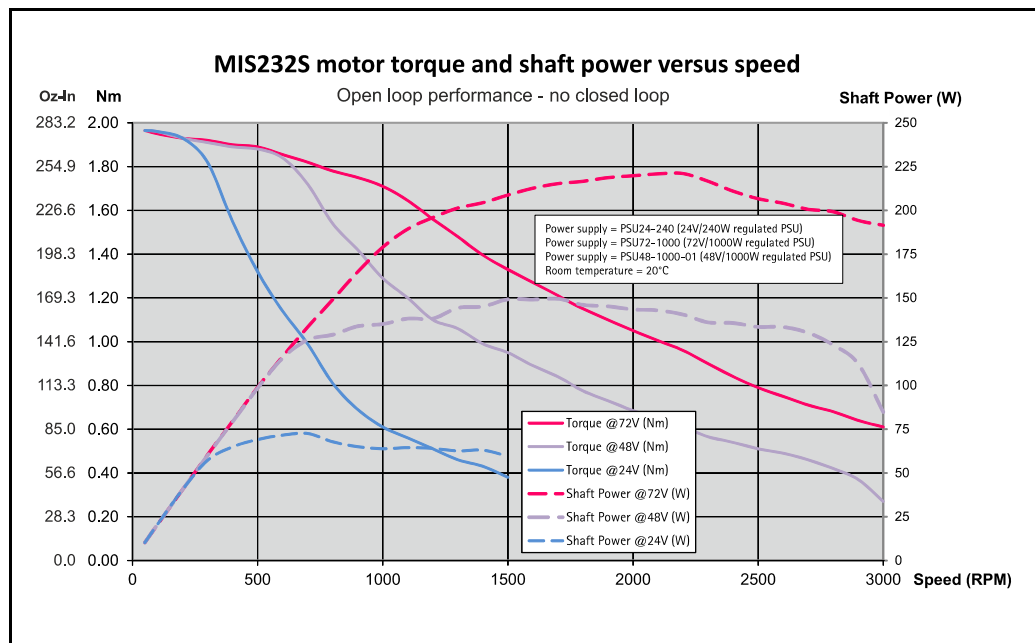
### 16.5.2 MIS231S and MIS231T Torque and power curves

Below the torque performance for both motor families is shown. As it can be seen the supply voltage have a significant influence at the torque performance at higher speeds. Please make sure to use a supply voltage which is appropriate for the actual application. Also make sure that the supply voltage is stable without too much ripple since voltage dips can cause the motor to stall and loose position. Also shaft power (mechanical power at motor shaft) is shown.



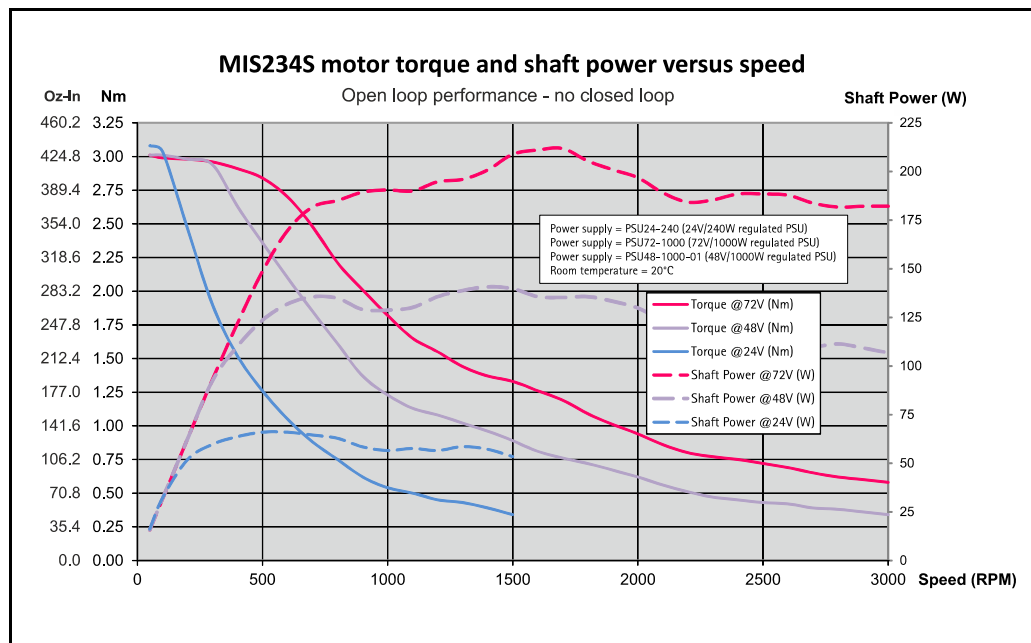
## 16.5 Torque Curves

### 16.5.3 MIS232S and MIS232T Torque and power curves



## 16.5 Torque Curves

### 16.5.4 MIS234S Torque and power curves

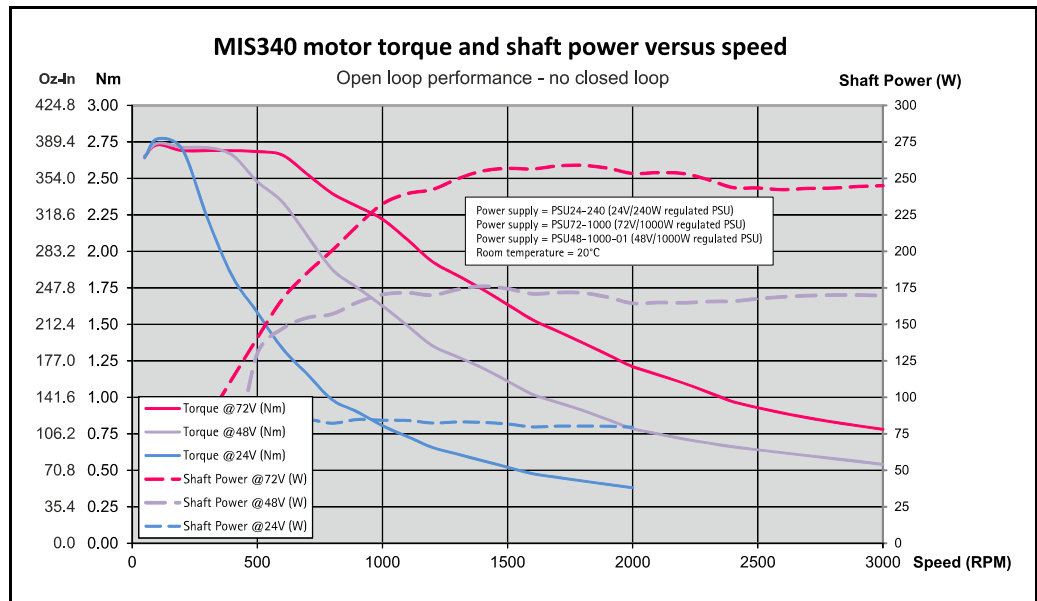


## 16.5 Torque Curves

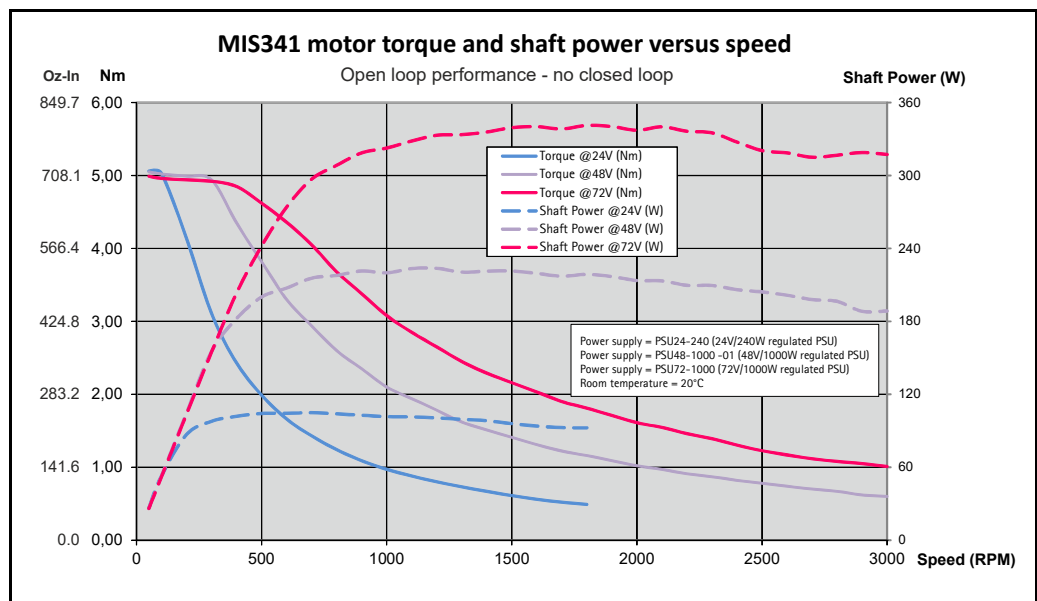
### 16.5.5 MIS34x Torque curves

Below the torque performance for both motor families is shown. As it can be seen the supply voltage have a significant influence at the torque performance at higher speeds. Please make sure to use a supply voltage which is appropriate for the actual application. Also make sure that the supply voltage is stable without too much ripple since voltage dips can cause the motor to stall and loose position.

### 16.5.6 MIS340 Torque curve

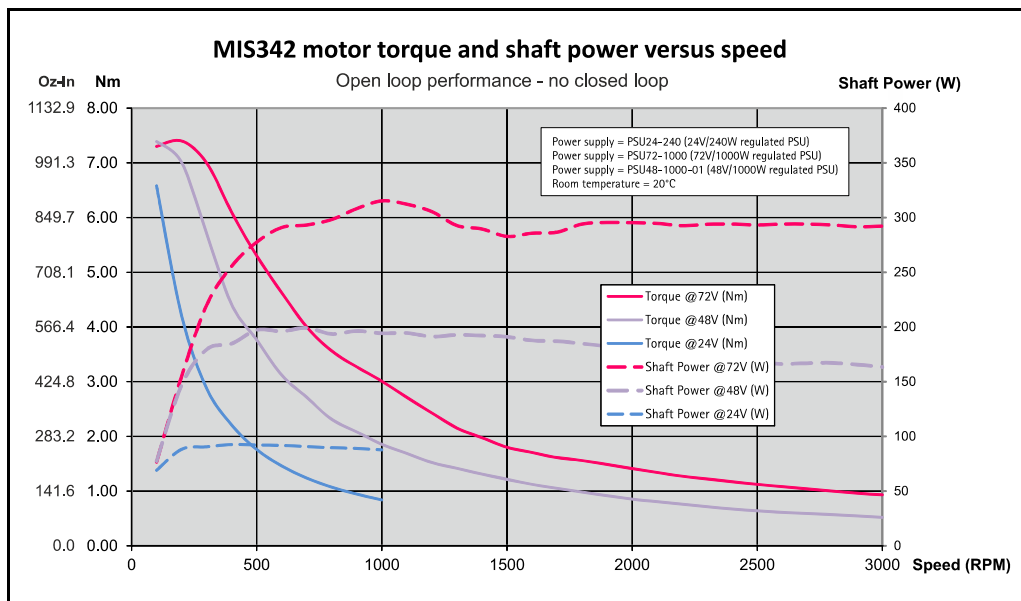


### 16.5.7 MIS341 Torque curve

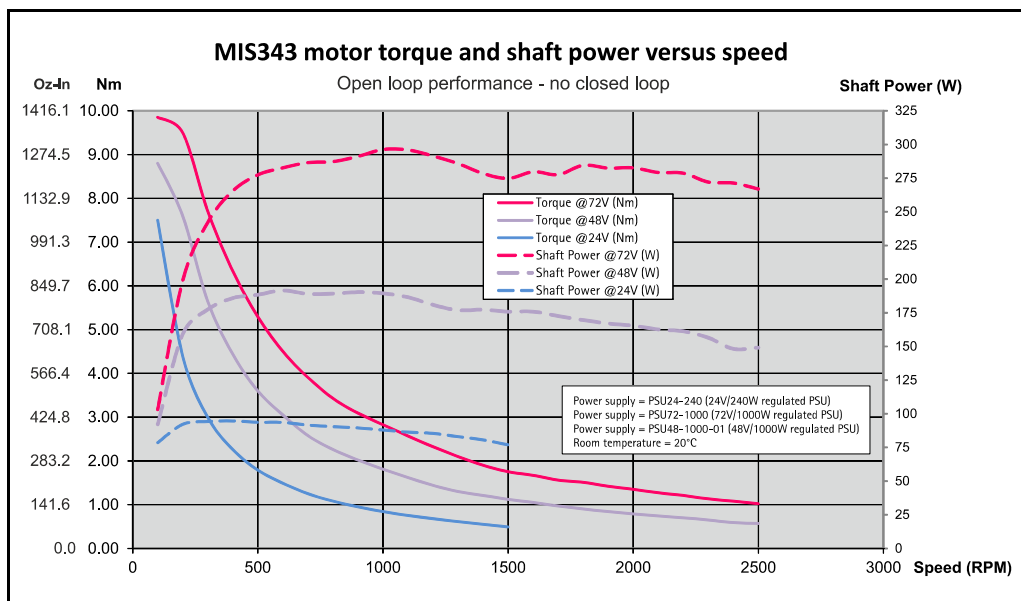


## 16.5 Torque Curves

### 16.5.8 MIS342 Torque curve



### 16.5.9 MIS343 Torque curve



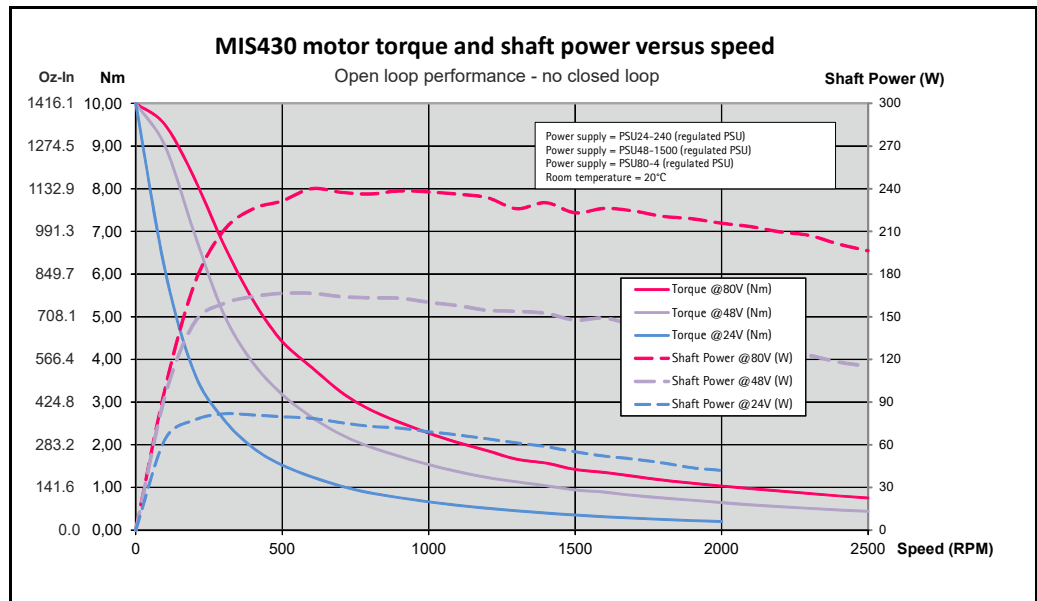
## 16.5

## Torque Curves

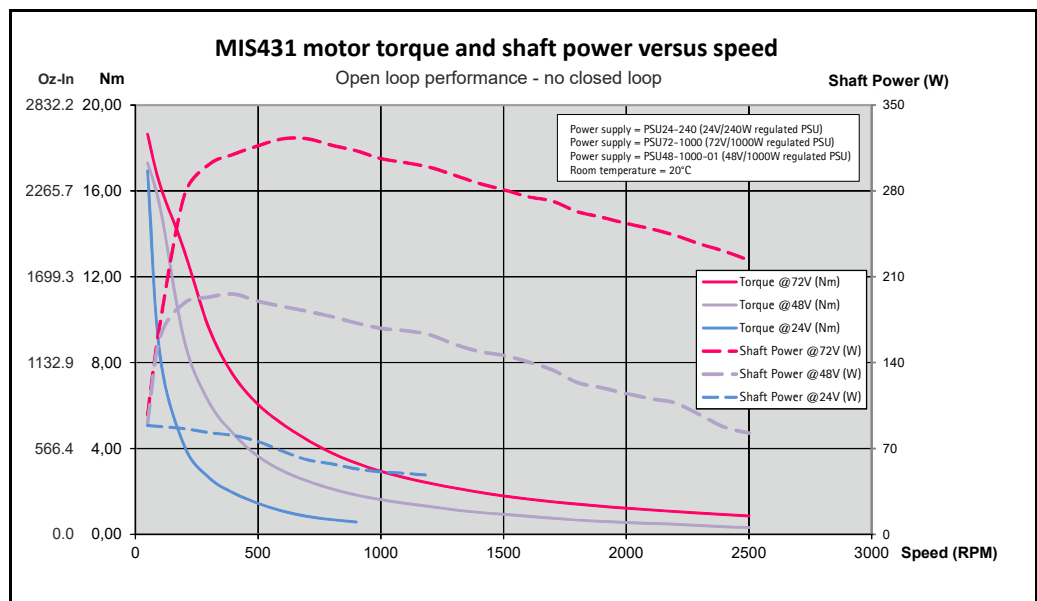
### 16.5.10 MIS430 and MIS432 Torque curves

Below the torque performance for the MIS43x families is shown. As it can be seen the supply voltage has a significant influence at the torque performance at higher speeds. Please make sure to use a supply voltage which is appropriate for the actual application. Also make sure that the supply voltage is stable without too much ripple since voltage dips can cause the motor to stall and loose position.

### 16.5.11 MIS430 Torque curve

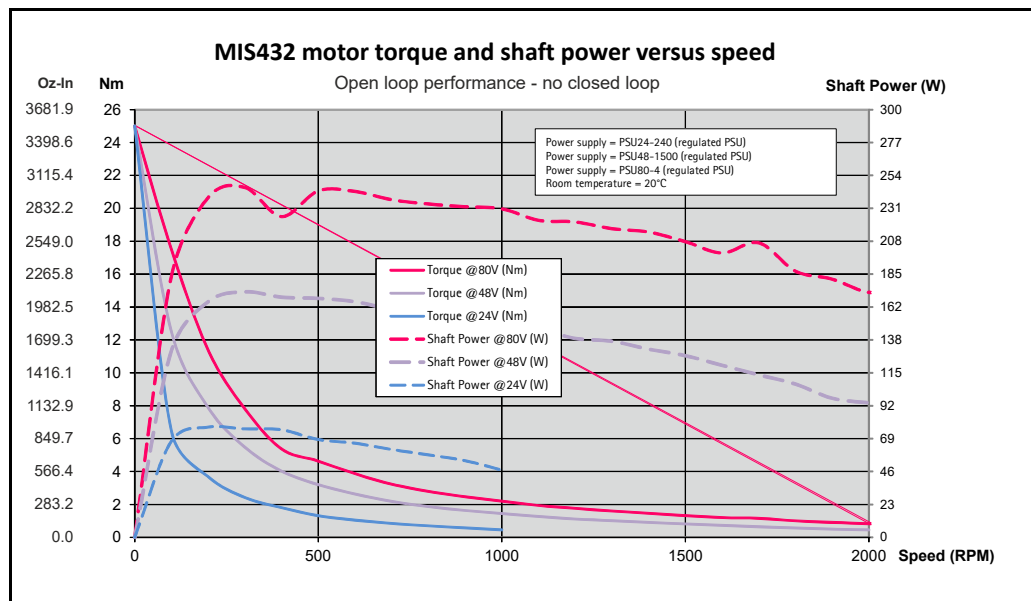


### 16.5.12 MIS431 Torque curve



## 16.5 Torque Curves

### 16.5.13 MIS432 Torque curve



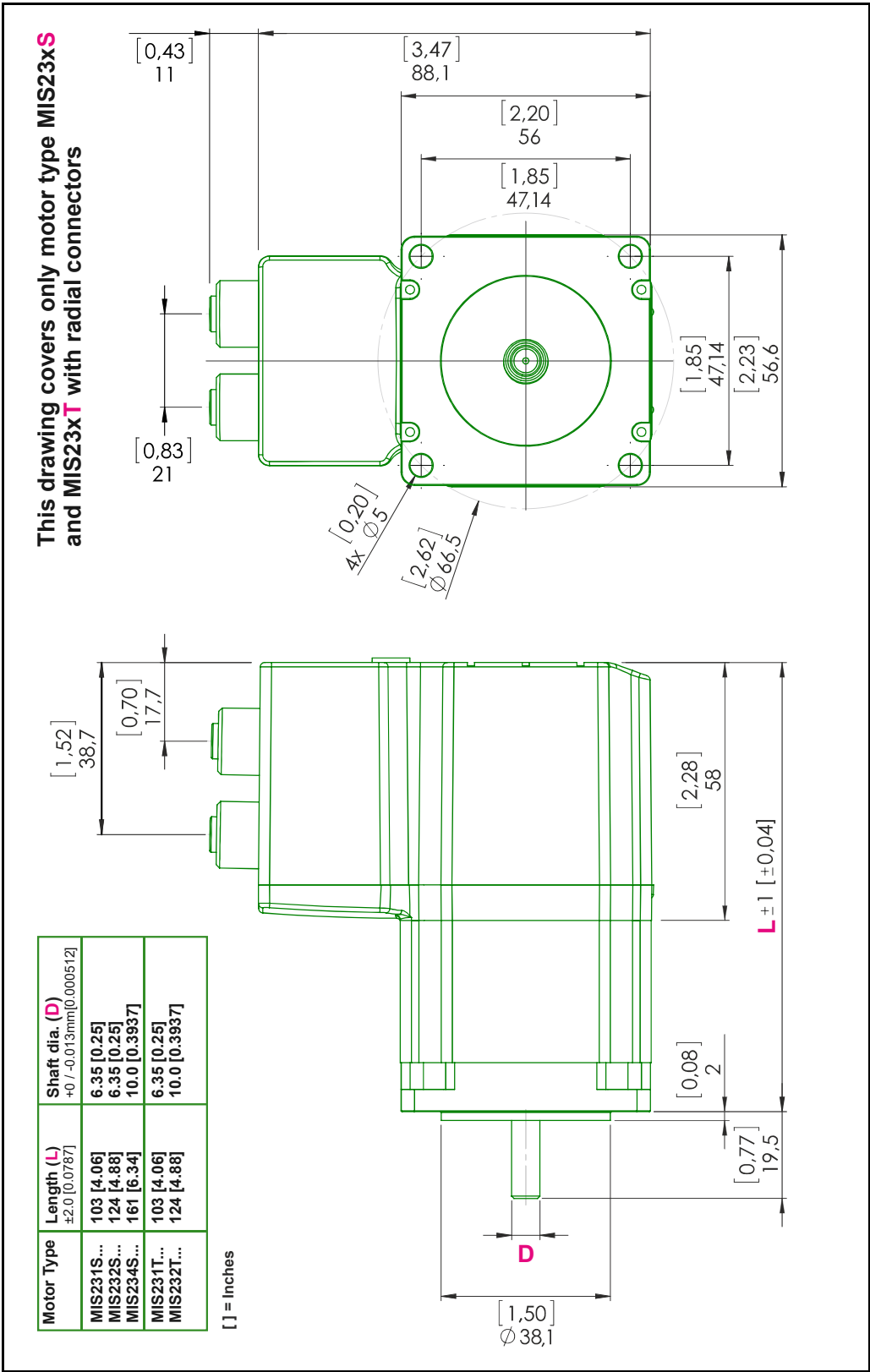
# 16.6

# Physical Dimensions

Only MIS23x

## 16.6.1 Physical dimensions MIS231S/T, MIS232S/T and MIS234S/T

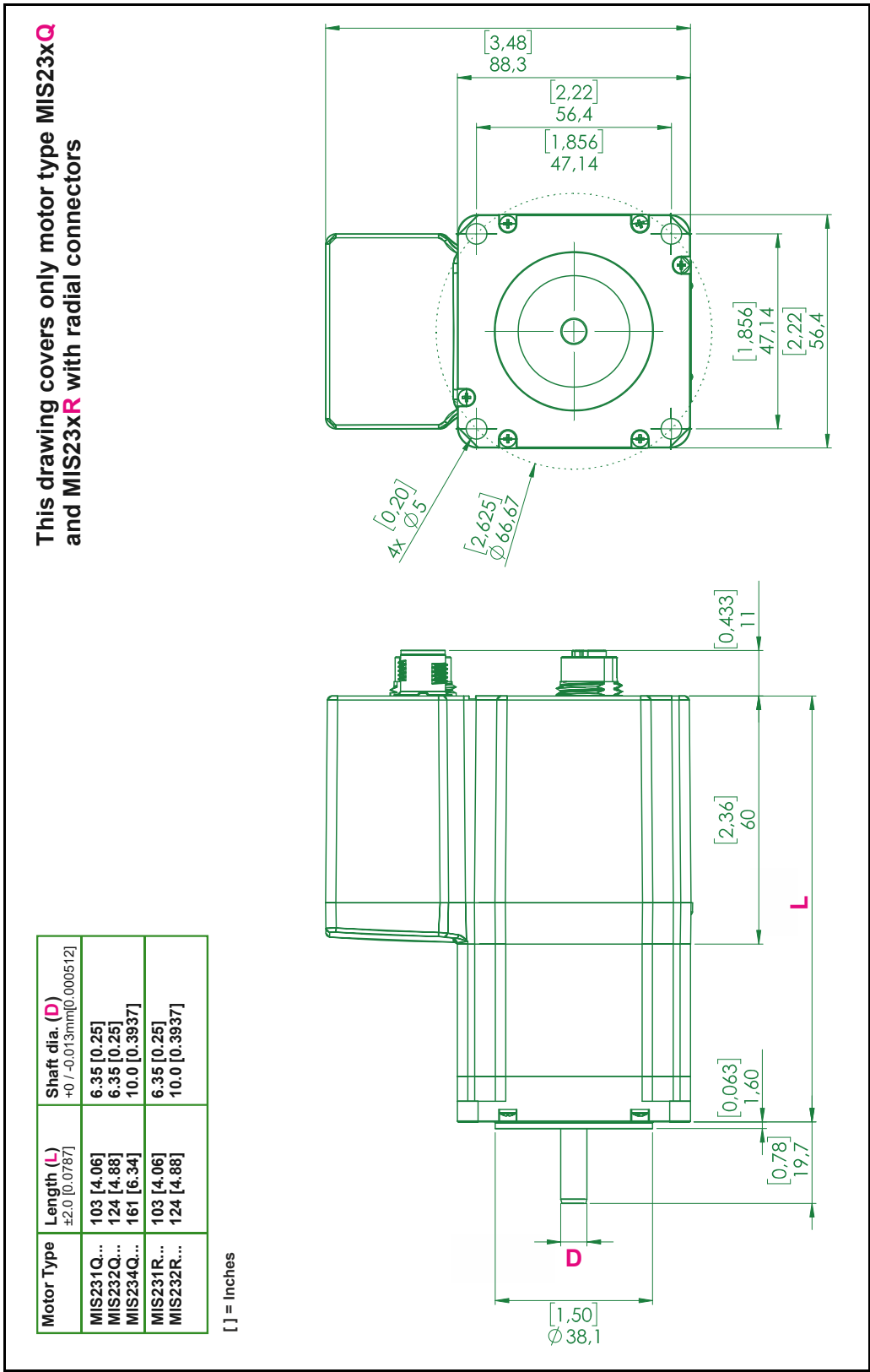
Notice that this chapter only covers the MIS23x family generation 2.



# 16.6 Physical Dimensions

## 16.6.2 Physical dimensions MIS231Q/R, MIS232Q/R and MIS234Q/R

Notice that this chapter only covers the MIS23x family generation 2.

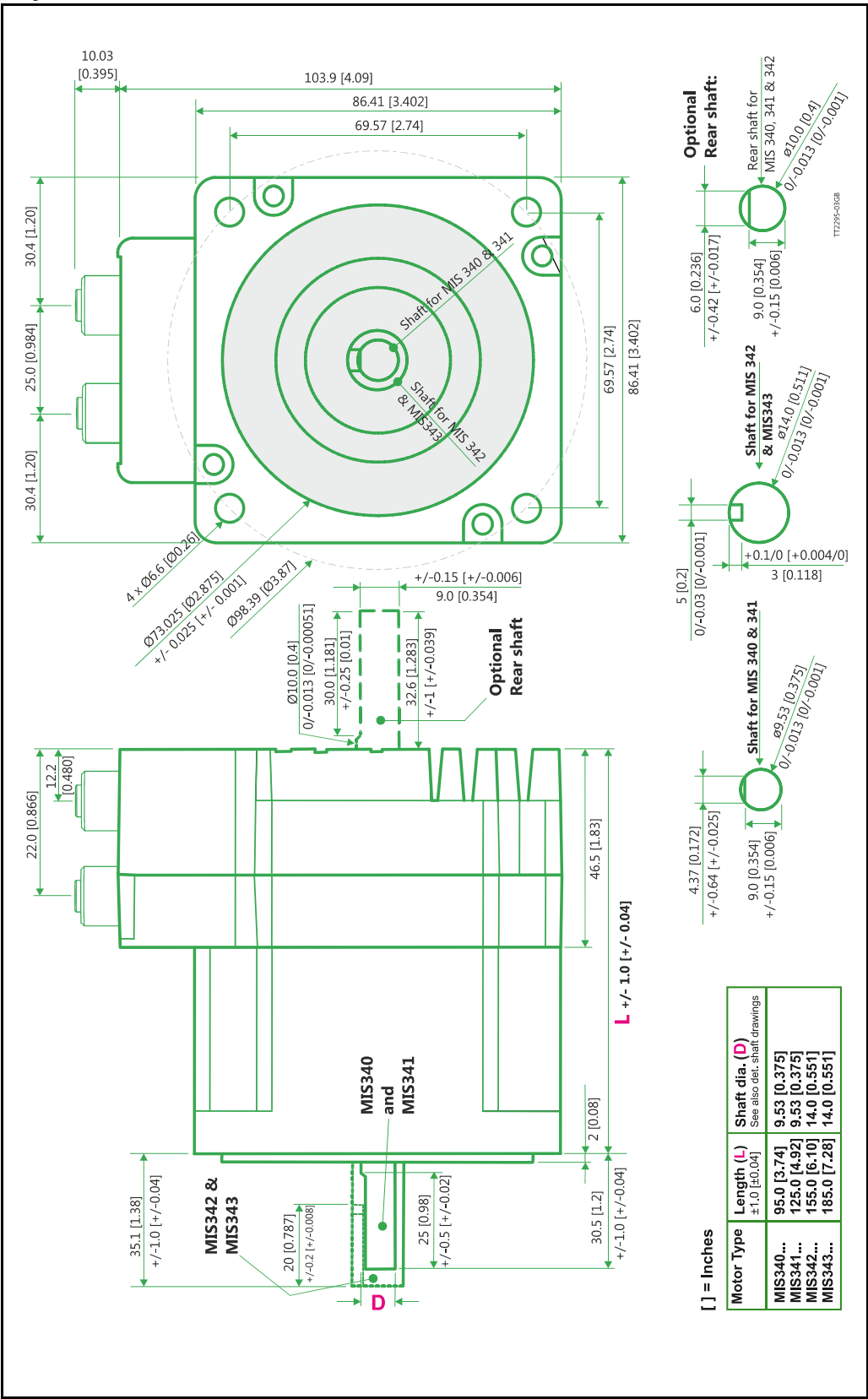


# 16.6

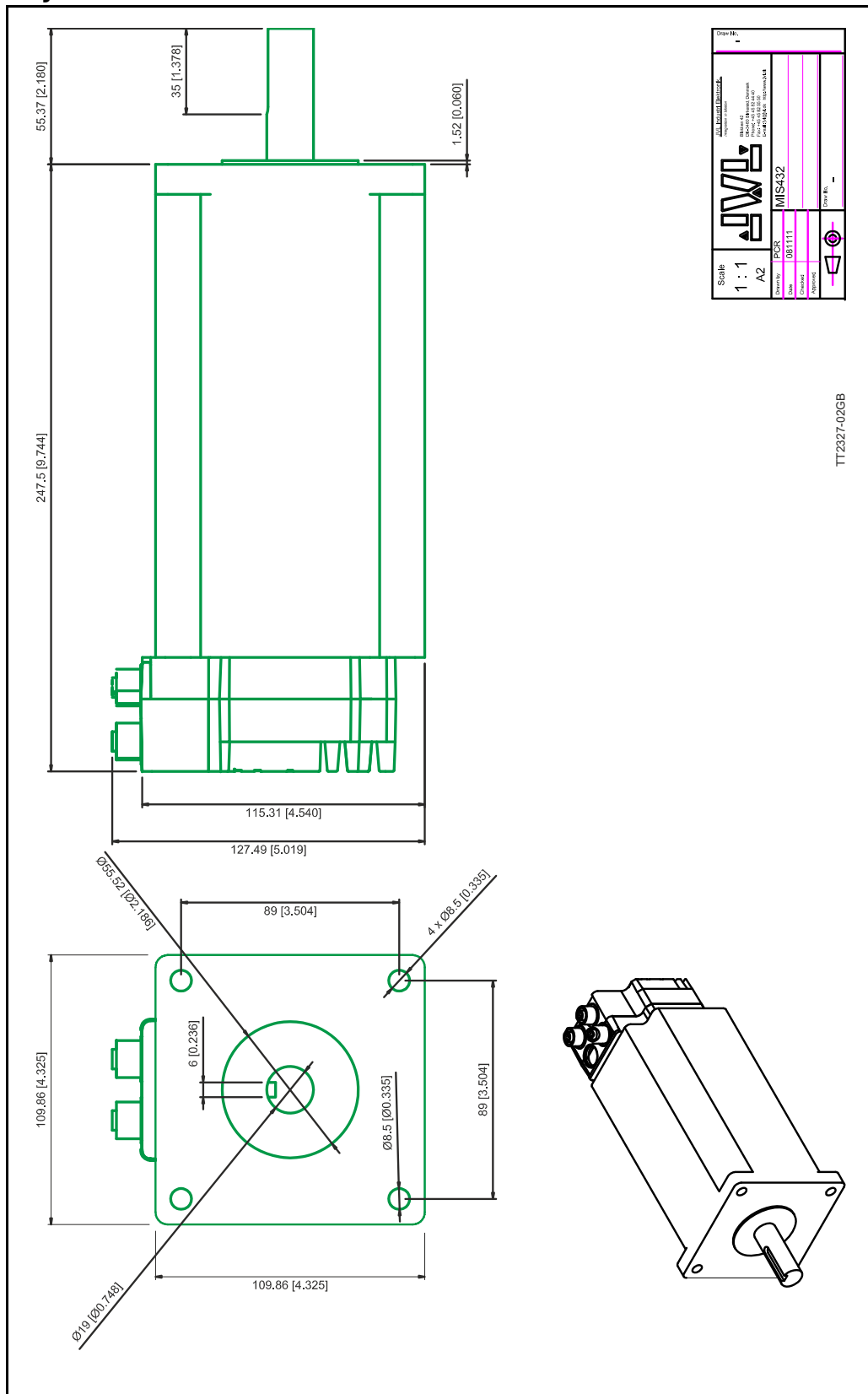
# Physical Dimensions

Only MIS34x

## 16.6.3 Physical dimensions MIS340 - MIS343



#### 16.6.4 Physical dimensions MIS432



**16.7.1 Life time of ball bearings in MIS34x**

The curve below can be used to determine the relation between the radial load at the motor output shaft and where the load is placed at the shaft with reference to the flange of the motor.

The curves are based on a continuous speed of 3000 RPM.

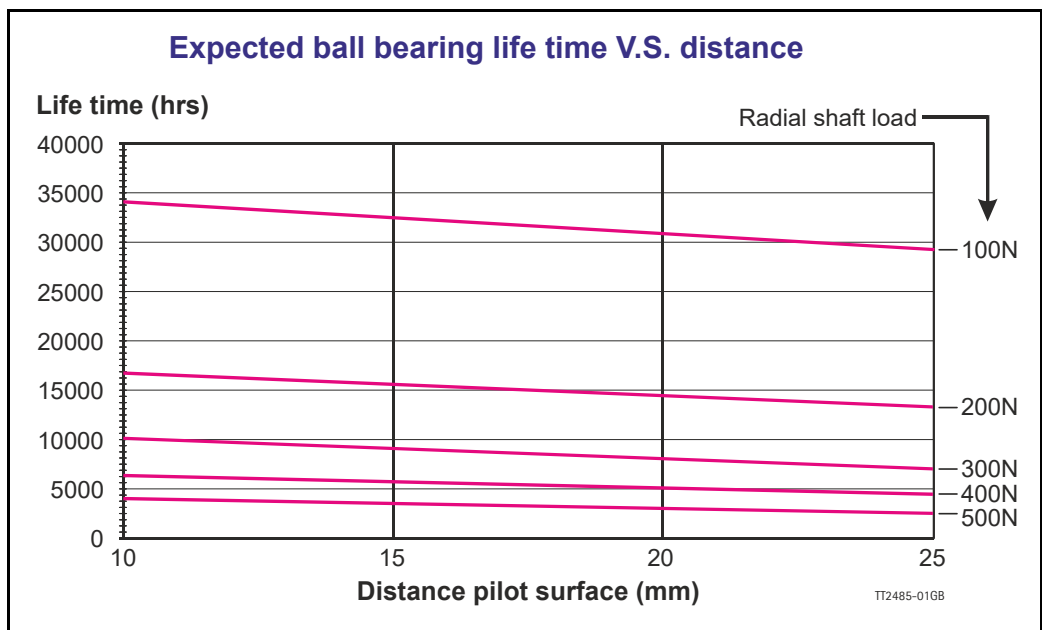
If the speed is lowered the lifetime will increase inversely proportional.

**Example:**

A motor is having a radial load of 200N placed with centre 15 mm from the flange.

According to the curve the lifetime will be 15.050 hours at 3000 RPM.

If the speed is lowered to 300 RPM (10 times lower than the curve is specified at) the lifetime will increase 10 times giving a total of 150.500 hours of operation.



## **16.8                      Trouble-shooting guide**

---

The following accessories are available for the MIS motor series.



## 17.2

## Power Supplies

---

### 17.2.1 PSU00-PD1

Combined power dump, resistor, and capacitor unit. For a complete power supply system, only a transformer with a secondary winding supplying 32VAC is required.

For systems with up to 5-8 QuickStep motors, this unit can serve as a central power dump unit.

The capacitor offers an efficient and economical way of storing the energy returned from the motors during deceleration of high inertias. See also [www.jvl.dk](http://www.jvl.dk)

### 17.2.2 PSU48-240

A compact switch-mode power supply with 240W output power at 48VDC.

The power supply is UL and CSA approved. It is protected against overvoltage, overtemperature and short-circuit or overload of the output. The power supply can either be mounted on a DIN rail or “wall” mounted. See also the data-sheet LD0047 which can be downloaded from [www.jvl.dk](http://www.jvl.dk)

### 17.2.3 Other power supplies

JVL offers a wide range of power supplies in the power range 45W to 1.5kW with output voltages 24 and 48VDC. They all use switch-mode technology in order to minimize physical dimensions and for easy adaptation to mains voltages in the range 90 to 240VAC.

The product range covers the following types: PSU05-045, PSU24-075, PSU24-240, PSU48-240, PSU48-800, PSU48-1000, PSU48-1500.

See also the data-sheet LD0058 (overview) which can be downloaded from [www.jvl.dk](http://www.jvl.dk) or LD0053 (detailed) which can be downloaded from [www.jvl.dk](http://www.jvl.dk).

## 17.3 Brakes and shaft reinforcement

---

A family of electromechanical brakes for external mounting is available for the MIS motors.

All brake types can be mounted directly on all the MIS motors and require 24VDC applied to release the motor

Further data for adding a brake to the MIS motors can be found using following links:

MIS23x: [www.jvl.dk](http://www.jvl.dk).

MIS34x: [www.jvl.dk](http://www.jvl.dk).

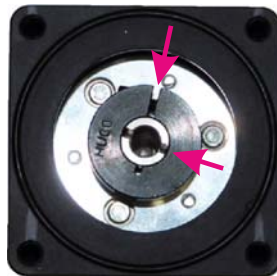
## 17.4 Gear and brake mounting instruction

### 17.4.1 Mounting a gear or a brake at the motor

When a gear or a brake is to be mounted on the front end of a motor it is very important that this is done in the right way since a wrong way of mounting may have fatal influence at lifetime of the motor or gear/brake and performance.

Please follow this instruction step by step to make sure that the mounting is done with a good result.

- ① Step 1 - Make sure that the shaft collar is oriented correctly in order to assure that the right tension around the motor shaft is possible.  
Hint: Tighten the shaft collar gently just to keep it in the right position.

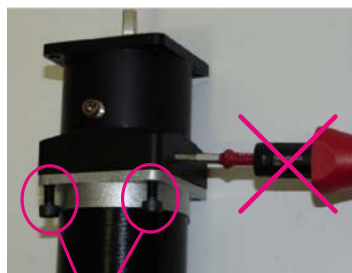


The inner and outer slit is NOT aligned. Make sure they are aligned as shown at right illustration



The inner and outer slit is aligned as they should.

- ② Step 2 - Mount the gear or brake at the motor but make sure to fasten the 4 shaft bolt first before fastening the shaft collar.  
Its recommended to use Locktite 278 in the threads to make sure that the bolts stay in place.



Do NOT tighten the shaft collar before the flange bolts are tightend



Flange bolts properly mounted and tightend.

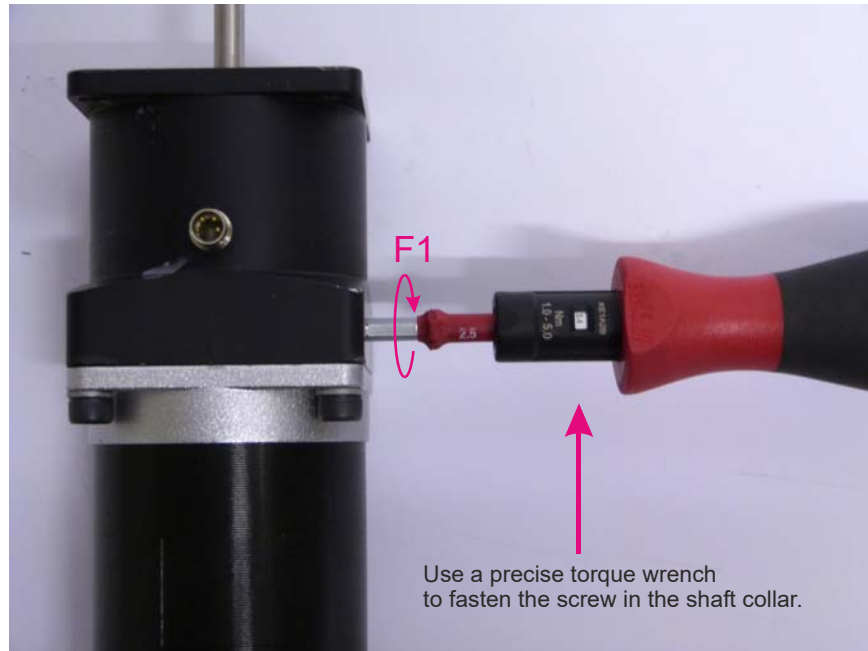
TT1536-01GB

## 17.4 Gear and brake mounting instruction

3

Step 3 - Final stage. Fasten the shaft collar with a torque of according to the scheme below.

Please notice that it can be fatal not to use the specified torque since the shaft may slip over time and cause a position offset.



### Gears (Product type to be mounted)

Series	Used with motor type	Tool	Torque (F1)
HTRG05	MAC050 to MAC141 (Ø6.35 shaft)	Hex size 3	5Nm
HTRG05	MIS230-233 (Ø6.35 shaft)	Hex size 3	5Nm
HTRG06	MAC050 to MAC141 (Ø6.35 shaft)	Hex size 3	5Nm
HTRG06	MAC400-402 (Ø14 shaft)	Hex size 3	11Nm
HTRG08	MIS340-341 (Ø9.53 shaft)	Hex size 4	5Nm
HTRG08	MIS342 (Ø14 shaft)	Hex size 5	8Nm
HTRG08	MAC800 (Ø19 shaft)	Hex size 5	11Nm
HTRG10	MAC800 (Ø19 shaft)	Hex size 5	11Nm

### Brakes (Product type to be mounted)

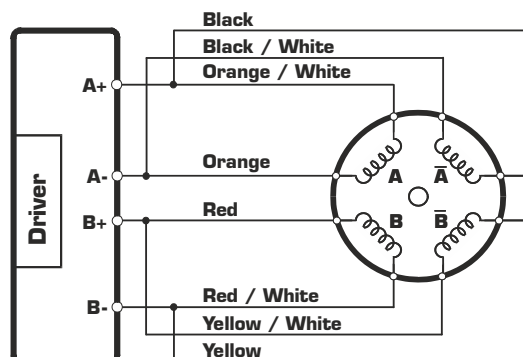
Series	Used with motor type	Tool	Torque (F1)
MAB23x	MAC050 to MAC141 (Ø6.35 shaft)	Hex size 2.5	2Nm
MAB23x	MIS230-233 (Ø6.35 shaft)	Hex size 2.5	2Nm
MAB34x	MIS340-341 (Ø9.53 shaft)	Hex size 3	5Nm
			TT1537-01GB



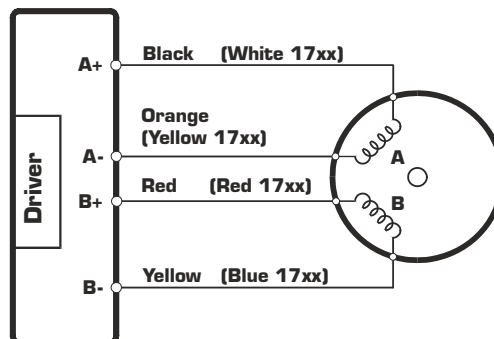
# 18.1

# Motor Connections

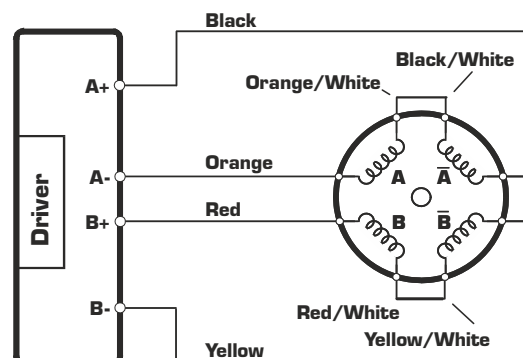
## Connection of JVL and MAE motors (parallel). Type MST23x/ MST34x and HY200-xxxx-xxx-x8



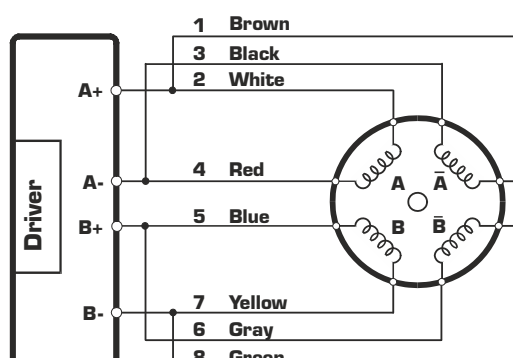
## Connection of JVL and MAE 4 wire motors. Type MST17x and HY200-xxxx-xxx-x4



## Connection of JVL and MAE motors (serial). Type MST23x/ MST34x and HY200-xxxx-xxx-x8



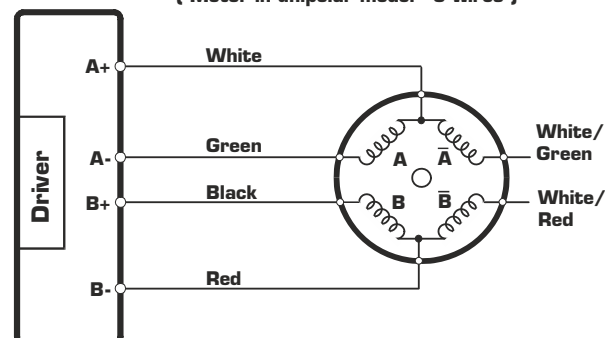
## Connection of Zebotronics motor Type : SMxxx.x.xx.x (8 terminals)



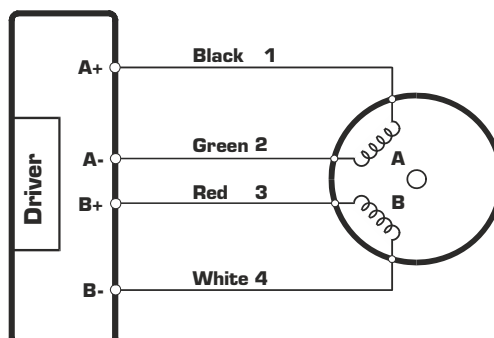
SM87/SM107/168.x.xx ↑ ↑ SM56.x.xx

## Connection of MAE motor (unipol.) Type HY200-1xxx-xxxxx6

( Motor in unipolar model - 6 wires )



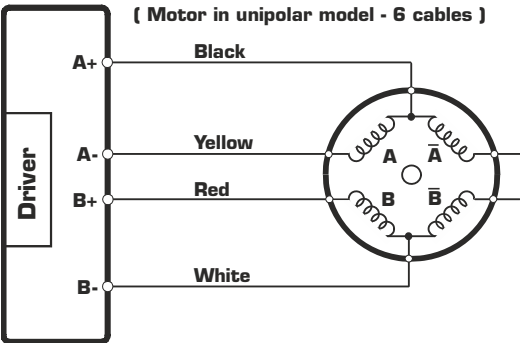
## Connection of Zebotronics motor Type : SMxxx.x.xx.x (4 terminals)



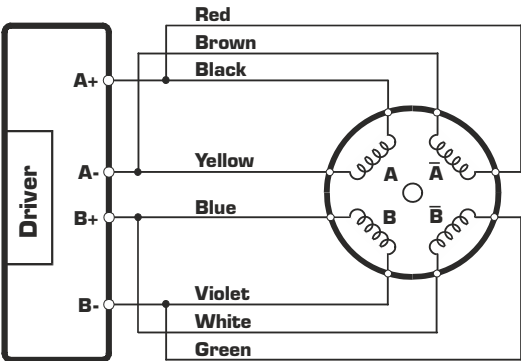
TT0005

# 18.1 Motor Connections

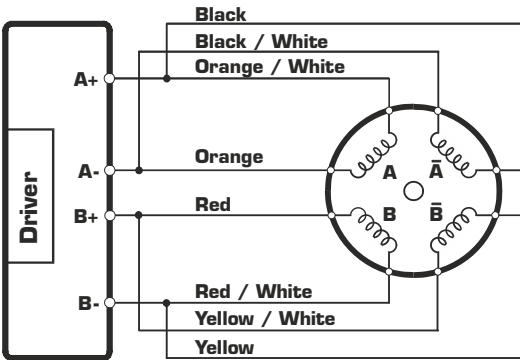
## Connection of Vexta motor Type PH2xx.xxx



## Connection of Phytron motor Type ZSx.xxx.x,x



## Connection of Vexta stepper motor Type : PH2xx-xxx



TT0006

## 18.2 Serial communication

This section describes control of the MIS motor (or SMC66/85) via the serial interface (RS485).

The interface is RS485 compatible and uses 8 data bits, 1 stop bit and no parity.

The MIS motors (or SMC66/85) are completely controlled by reading and writing to registers.

The registers are numbered 1-255. The width of the registers is 32 bits.

To protect communication from errors, the data is transmitted twice.

First the data byte is transmitted and then an inverted version (255-x) is transmitted.

The easiest way to become familiar with the registers and MacTalk communication is to use the MacRegIO program. This program lists all of the registers, and the serial commands sent and received can be monitored.

### 18.2.1 Supported commands

Sync	Response Sync	Description
0x50	0x52	Read register
0x51	0x52	Read register block
0x52	0x11 (Acknowledge)	Write register
0x54	0x11 (Acknowledge)	Enter safe mode
0x55	0x11 (Acknowledge)	Exit safe mode
0x56	0x11 (Acknowledge)	Write to flash
0x57	None	Reset controller
0x59	None	Group write register
0x61	0x61	Program status and command
0x62	0x11 (Acknowledge)	Write program flash
0x63	0x63	Read program flash

### 18.2.2 Read register

This command can read a register. All registers are read as 32-bit.

Master sends	MIS/SMC Response
<Read><Address><RegNum><End>	<Write><MAddress><RegNum><Len><Data><End>

### Block description

Block name	Protected	Example	Description
<Read>	No	50h,50h,50h	Read command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS or SMC
<RegNum>	Yes	05h,FAh (RegNum 5)	The register number to read
<End>	No	AAh, AAh	Command termination
<Write>	No	52h,52h,52h	Write command
<MAddress>	Yes	00h,FFh (Address 0)	This will always be 0, because this is the address of the master
<RegNum>	Yes	05h,FAh (RegNum 5)	This will always be the same as requested
<Len>	Yes	04h,FBh (Len = 4)	The length will always be 4
<Data>	Yes	E8h,17h, 03h,FFh, 00h, FFh, 00h,FFh (Data = 1000)	The data read from the register
<End>	No	AAh, AAh	Command termination

## 18.2 Serial communication

### 18.2.3 Read register block

Using this command it is possible to read 64 consecutive registers at once.

Master sends	MIS/SMC Response
<ReadB><Address><RegNum><End>	<Write><MAddress><RegNum><Len><Data><End>

#### Block description

Block name	Protected	Example	Description
<ReadB>	No	51h,51h,51h	Read block command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS or SMC
<RegNum>	Yes	05h,FAh (RegNum 5)	The first register to read
<End>	No	AAh, AAh	Command termination
<Write>	No	52h,52h,52h	Write command
<MAddress>	Yes	00h,FFh (Address 0)	This will always be 0, because this is the Address of the master
<RegNum>	Yes	05h,FAh (RegNum 5)	This will always be the same as requested
<Len>	Yes	80h,7Fh (Len = 128)	The length will always be 128, so 64 registers is read in each block.
<Data>	Yes	E8h,17h, ..., 03h,FCh	The data read from the registers

### 18.2.4 Write Register

Using this command, a register can be written.

Controller sends	MIS/SMCResponse
<Write><Address><RegNum><Len><Data><End>	<Accept>

#### Block description

Block Name	Protected	Example	Description
<Write>	No	52h,52h,52h	Write command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS/SMC
<RegNum>	Yes	05h,FAh (RegNum 5)	The register number to write to
<Len>	Yes	02h,FDh (Len = 2)	The number of data bytes
<Data>	Yes	E8h,17h, 03h,FCh (Data = 1000)	The data to write to the register
<End>	No	AAh, AAh	Command termination
<Accept>	No	11h, 11h,11h	Accept from MIS/SMC

### 18.2.5 Enter safe mode

When this command is sent, the MIS/SMC switches to safe mode. In safe mode, no program or commands can enable the motor. The mode can only be exited using either an "Exit safe mode" or "Reset" command.

Controller sends	MIS/SMC response
<EntSafe><Address><End>	<Accept>

#### Block description

Block Name	Protected	Example	Description
<EntSafe>	No	54h,54h,54h	Enter safe mode command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS/SMC
<End>	No	AAh, AAh	Command termination
<Accept>	No	11h, 11h,11h	Accept from MIS/SMC

## 18.2 Serial communication

### 18.2.6 Exit safe mode

When this command is sent, the MIS/SMC switches back to normal mode.

Controller sends	MIS/SMC response
<ExitSafe><Address><End>	<Accept>

#### Block description

Block Name	Protected	Example	Description
<ExitSafe>	No	55h,55h,55h	Exit safe mode command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS/SMC
<End>	No	AAh, AAh	Command termination
<Accept>	No	11h, 11h,11h	Accept from MIS/SMC

### 18.2.7 Write to flash

This command writes the register values to flash memory. The values will then be retained after a power down. The command will only work if the motor is in “Safe mode” After the command is executed, the motor will reset. The response will only be transmitted if the command failed, e.g. if the motor is not in safe mode.

Controller sends	MIS/SMC response
<WriteFlash><Address><End>	<Accept>

#### Block description

Block Name	Protected	Example	Description
<WriteFlash>	No	56h,56h,56h	Write to flash command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS/SMC
<End>	No	AAh, AAh	Command termination
<Accept>	No	11h, 11h,11h	Accept from MIS/SMC

### 18.2.8 Reset controller

This command resets the MIS/SMC. No response will be transmitted from the MIS/SMC.

Controller sends	MIS/SMC response
<Reset><Address><End>	None

#### Block description

Block Name	Protected	Example	Description
<Reset>	No	57h,57h,57h	Reset command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS/SMC
<End>	No	AAh, AAh	Command termination

## 18.2 Serial communication

### 18.2.9 Group write register

Using this command it is possible to write a register in several MIS/SMCs with one command.

The command includes a sequence number which must be changed for each write. This is used so that the same command can be written several times, to ensure that all controllers received it. The last received sequence id can be read in register I48.

Controller sends	MIS/SMC Response
<GWrite><Group><Sequence><RegNum><Len><Data><End>	None

#### Block description

Block Name	Protected	Example	Description
<GWrite>	No	59h,59h,59h	Group write command
<Group>	Yes	07h,F8h (Address 7)	The group id of the MIS/SMCs to write to.
<Sequence>	Yes	04h,FBh (Sequence 4)	The sequence number of the write.
<RegNum>	Yes	05h,FAh (RegNum 5)	The register number to write to
<Len>	Yes	02h,FDh (Len = 2)	The number of data bytes
<Data>	Yes	E8h,17h, 03h,FCCh (Data = 1000)	The data to write to the register
<End>	No	AAh, AAh	Command termination

### 18.2.10 Program status and command

Using this command, different actions can be executed. The command also returns some information about the program state.

The table below shows the possible commands:

Com- mand	Data 1	Data 2	Description
0	-	-	No operation
1	-	-	Start program execution
2	-	-	Stop program execution
3	-	-	Pause program execution
4	Start Address (16bit)	End Address (16bit)	Run the program until the program pointer is outside the area [Start Address End Address] Then the program is paused
5	Set outputs (8bit)	Clear outputs (8bit)	Modifies the outputs. The bits set in the "Set outputs" data will be set and cleared for "Clear outputs". Example: The data 0x06,0x01 sets output 2+3 and clears output 1
6			Reserved
7	Size (16 bit)		Prepare the flash for a new program. Data 1 specifies the size of the program in bytes.

The command number is placed in the first command data byte. Data 1 + Data 2 are placed in the following command data bytes.

Controller sends	MIS/SMC Response
<PStat><Address><Len1><Data1><End>	<PStat><MAddress><Len2><Data2><End>

## 18.2 Serial communication

### Block description

Block Name	Protected	Example	Description
<PStat>	No	61h,61h,61h	Program status command
<Address>	Yes	07h,F8h (Address 7)	The address of the MIS/SMC's to write to.
<Len1>	Yes	01h,FEh (Len = 1)	Length of the command data
<Data1>	Yes	01h,FEh (Start)	Command data
<MAddress>	Yes	00h,FFh (Address 0)	This will always be 0, because this is the address of the master
<Len2>	Yes	08h,F7h (Len = 8)	The length of the return data
<Data2>	Yes	09h,F6h, (Program state) 00h,FFh, 00h,FFh, (Program pointer) 00h,FFh, (Stack pointer) 00h,FFh, 00h,FFh, (Program checksum) 80h,7Fh, (Inputs) 00h,FFh (Outputs)	Data returned from MIS/SMC
<End>	No	AAh, AAh	Command termination

The returned data has the following format:

Data offset	Size	Description
0	8 bit	Program state. See table below for states.
1	16 bit	Program pointer. The current location of the program pointer.
3	8 bit	Stack pointer
4	16 bit	Program checksum. This checksum is calculated when the program is started.
6	8 bit	Input status.
7	8 bit	Output status

Program states:

Program state	Name	Description
0	Passive	The program execution is stopped. This state is only entered shortly at power-up.
1	Running	The program execution is running
2	Single Step	A single step is in progress. The program will run until the selected program position is reached.
3	Paused	The program execution is paused, but can be resumed again.
4	Stack Overflow	The stack pointer has overflowed
5	Program Overflow	The program pointer has overflowed.
6	Invalid Ins.	An invalid instruction is encountered in the program.
7	Stopped	The program execution is stopped.
8	Com. Error	Internal communication error has occurred. This cannot happen on MIS/SMC.
9	Starting Program	Program execution is being prepared. After this is completed the state will change to running.
10	Flash Error	The program data is corrupted.
11	Flash Checksum Error	The program data checksum is incorrect.

## 18.3

## Ordering information

---

See JVL's product part number builder using the following link:  
<http://www.jvl.dk/ppnb/ppnb.htm>

## 18.4

## Encoder calibration

### 18.4.1 Procedure to calibrate encoder

Normally the encoder (H2, H3 or H4 option) is calibrated from the factory but it can be required to recalibrate for example if the encoder firmware have been updated from an older version.

If the error message “Encoder lost position” is shown its required that the following calibration procedure takes place.

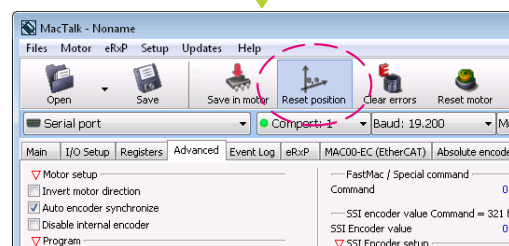
#### Step 1

Make sure the motor is free from any load and can rotate without any mechanical resistance.

Also make sure to set the motor settings to default (under the «Motor» menu).

#### Step 2

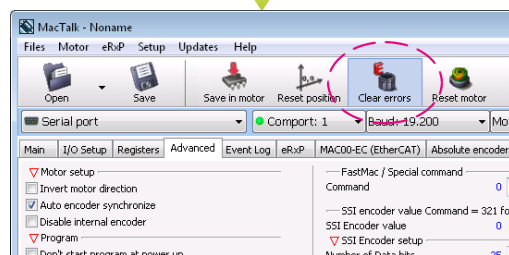
Press «Reset position»



#### Step 3

Press «Clear errors»

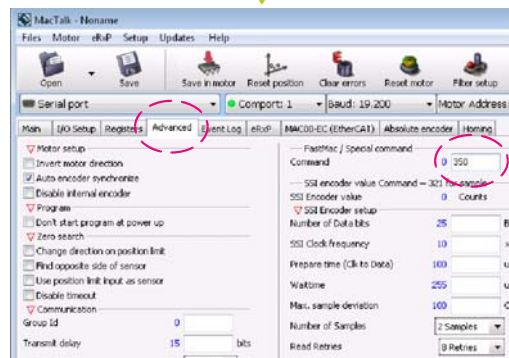
Any errors shown should now disappear.



#### Step 4

Now go to the «Advanced» tab and write **999** (enter) and **350** (enter) in the Command field and hit enter.

The motor will now move 10 fast revolutions and 1 slow. This takes approx. 2 minutes and when finished the motor stands still and the actual position is approximately 409500. Now the calibration is finished and the position can be reset if needed.



Finished !

TT2554-01GB





***EU - Declaration of Conformity*****Manufacturer**

Company Name: JVL Industri Elektronik A/S  
Address: Bregnerødvej 127  
DK-3460 Birkerød  
Denmark  
Telephone: +45 45 82 44 40  
E-mail: jvl@jvl.dk  
Web: www.jvl.dk

**Hereby declares that**

**Product**

No.: MIS231, 232 and 234  
Name: Integrated Hybrid stepper motor  
Type: Main no. followed by R, S, T or Q incl. subversions

- is in conformity with:

DIRECTIVE 2014/30/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 26 February 2014 on the harmonisation of the laws of the Member States relating to electromagnetic compatibility

and

DIRECTIVE 2014/35/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of electrical equipment designed for use within certain voltage limits

- is manufactured in accordance with the following standards:

*EN 61800-3 Adjustable speed electrical power drives systems - part 3:  
EMC product standard including specific test methods..*

July 2017



Bo V. Jessen  
Technical Director  
JVL Industri Elektronik A/S

LX0027-01GB

## 19.1 CE Declaration of Conformity

### ***EU - Declaration of Conformity***

#### **Manufacturer**

Company Name: JVL A/S  
Address: Bregnerødvej 127  
DK-3460 Birkerød  
Denmark  
Telephone: +45 45 82 44 40  
E-mail: jvl@jvl.dk  
Web: www.jvl.dk

#### **Hereby declare that:**

#### **Product**

No.: MIS340, 341, 342, 343, 430 and 432  
and MIL340, 341, 342 and 343  
Name: Integrated Stepper Motor  
Sub-types: -Czznnyxw5  
zz=shaft option  
nn=internal option module,  
yx=optional encoder options,  
w=STO option

- is in conformity with:

DIRECTIVE 2014/30/EU (EMC directive) OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 26 February 2014 on the harmonisation of the laws of the Member States relating to electromagnetic compatibility

and

DIRECTIVE 2014/35/EU (LVD directive) OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of electrical equipment designed for use within certain voltage limits.

- is manufactured in accordance with the following standards:

EN 61800-3 Adjustable speed electrical power drives systems - part 3:

EMC product standard including specific test methods.

and

EN/IEC 62368-1:2020/A11:2020 Safety of electrical and electronic equipment.

June 2021



Bo V. Jessen  
Technical Director - JVL A/S



**JVL**  
intelligent motors

## 19.2

## Vibrationtest certificates

---

All MIS motor sizes have passed a complete vibration and shock test performed by a independent accredited test institute.

The tests have been performed as follows

### **Vibration**

Standard: EN/IEC 60068-2-6

Test: 5 - 25 Hz:  $\pm 1.6\text{mm}$ , 25 - 500Hz: 4 g, 1.0 oct./min., 3 x 10 sweep cycles.

### **Shock**

Standard: EN/IEC 60068-2-27


Test: Shock; 15 g, 30ms, 6 x 1000 shocks.

All tested motors did function fully during and after the test.

At next page a test assessment for the MIS34x motors are shown. Additional testdata for the other motor sizes can be requested from your nearest JVL distributor.



## Mechanical assessment sheet no. 1336

<b>DELTA client</b> JVL Industri Elektronik A/S Blokken 42 3460 Birkerød Denmark	<b>DELTA project no.</b> T207608
<b>Product identification</b> Series MIS34x: MIS340 MIS341 MIS342	
<b>DELTA report(s)</b> DELTA project no. T207608, DANAK-19/13991 Revision 1	
<b>Other document(s)</b> "The MIS34x integrated stepper motor family", 20140214, JVL Industri Elektronik A/S – BVJ 12.5.2 Physical dimensions MIS340, MIS341 and MIS342, JVL Industri Elektronik A/S – User Manual – Integrated Stepper Motors MIS23x, 34x, 43x 12.2 MIS34x Technical Data, JVL Industri Elektronik A/S – User Manual – Integrated Stepper Motors MIS23x, 34x, 43x	
<b>Conclusion</b> The MIS342C14EPH385 has been tested according to the below listed standards. The test results are given in the DELTA report listed above. The tests were carried out as specified and neither malfunctions nor mechanical damages were detected.  IEC 60068-2-6: 2007, Test Fc; 5 – 25 Hz: $\pm 1.6$ mm, 25 - 500 Hz: 4 g, 1.0 oct./min., 3 x 10 sweep cycles IEC 60068-2-27, Test Ea, Shock; 15 g, 30 ms, 6 x 1000 shocks  The MIS34x integrated stepper motor family covers a number of family members ie. MIS340, MIS341, and MIS342. The tested MIS342C14EPH385 is worst case since the motor length of this family member is larger than the rest of the family and it is equipped with all extra options available and relevant for the vibration and shock tests i.e. ethernet and absolute multiturn encoder option.  Based on the documentation mentioned above, the test results are considered to be representative for smaller family members, MIS340, MIS341, and MIS342 models without options.	
<b>Date</b>  Hørsholm, 7 October 2014	<b>Assessor</b>   Susanne Otto B.Sc.E.E., B.Com (Org)

## CERTIFICATE OF COMPLIANCE

Certificate Number 20180711-E254947  
Report Reference E254947-20180427  
Issue Date 2018-JULY-11

Issued to: JVL INDUSTRI ELEKTRONIK A/S  
Bregnerødvej 127  
3460 Birkerød DENMARK


This is to certify that COMPONENT - POWER CONVERSION EQUIPMENT  
representative samples of "See Addendum Page"

Have been investigated by UL in accordance with the Standard(s) indicated on this Certificate.

Standard(s) for Safety: UL 61800-5-1 - Adjustable Speed Electrical Power Drive Systems - Part 5-1: Safety Requirements - Electrical, Thermal and Energy  
CSA 22.2 No. 274 - Adjustable Speed Drives

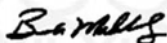
Additional Information: See the UL Online Certifications Directory at [www.ul.com/database](http://www.ul.com/database) for additional information

Only those products bearing the UL Certification Mark should be considered as being covered by UL's Certification and Follow-Up Service.

The UL Recognized Component Mark generally consists of the manufacturer's identification and catalog number, model number or other product designation as specified under "Marking" for the particular Recognition as published in the appropriate UL Directory. As a supplementary means of identifying products that have been produced under UL's Component Recognition Program, UL's Recognized Component Mark: , may be used in conjunction with the required Recognized Marks. The Recognized Component Mark is required when specified in the UL Directory preceding the recognitions or under "Markings" for the individual recognitions.

Recognized components are incomplete in certain constructional features or restricted in performance capabilities and are intended for use as components of complete equipment submitted for investigation rather than for direct separate installation in the field. The final acceptance of the component is dependent upon its installation and use in complete equipment submitted to UL LLC.

Look for the UL Certification Mark on the product.



Bruce Mahrenholz, Director North American Certification Program  
UL LLC

Any information and documentation involving UL Mark services are provided on behalf of UL LLC (UL) or any authorized licensee of UL. For questions, please contact a local UL Customer Service Representative at <http://ul.com/aboutul/locations/>

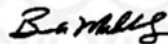


## CERTIFICATE OF COMPLIANCE

Certificate Number 20180711-E254947  
Report Reference E254947-20180427  
Issue Date 2018-JULY-11

This is to certify that representative samples of the product as specified on this certificate were tested according to the current UL requirements.

USR/CNR – Power Conversion Equipment, Open Type, Cat. Nos. MIS340C, MIS341C, MIS342C, MIS343C followed by additional letters or numbers



Bruce Mahrenholz, Director North American Certification Program  
UL LLC

Any information and documentation involving UL Mark services are provided on behalf of UL LLC (UL) or any authorized licensee of UL. For questions, please contact a local UL Customer Service Representative at <http://ul.com/aboutul/locations>





# Certificate

No. SEBS-A.153157/17 V1.0



TÜV NORD Systems GmbH & Co. KG hereby certifies to

## JVL Industri Elektronik A/S

Bregnerødvej 127  
3460 Birkerød, Denmark

that the Safe Torque Off (STO) function used in

## Mlxxyz Integrated Stepper Series

meets the requirements listed in the below mentioned standards

- |                                |                                       |
|--------------------------------|---------------------------------------|
| • IEC 61508-1:2010; SIL 3; SC3 | • ISO 13849-1:2015; PL e              |
| • IEC 61508-2:2010; SIL 3; SC3 | • ISO 13849-2:2012                    |
| • IEC 61800-5-2:2016; SIL 3    | • ISO 62061:2015; SIL <sub>CL</sub> 3 |

Certification program Leittechnik (SEB-ZE-SEECERT-VA-320-20, Rev. 3/9.15)

Base of certification is the report  
SEBS-A-153157/17TB and the tracking  
list in the valid version.

This certificate entitles the holder to  
use the pictured Safety Approved  
conformity mark.

Valid until: 07.11.2023  
File reference: 8115308330

Hamburg, 07.11.2018

*B. Pfuff*

Bianca Pfuff



Certification Body SEECERT  
TÜV NORD Systems GmbH & Co. KG  
Große Bahnstraße 31, 22525 Hamburg, Germany



**A**

A\_SOLL 198, 253–255, 257–258, 298, 302  
 Abort SDO 357  
 Acc\_Emerg 207  
 Acceleration factor 308  
 Address, CANopen 282  
 Address, MacTalk 85  
 Afzup\_ConfMax 215  
 Afzup\_ConfMin 214  
 Afzup\_Filter 215  
 Afzup\_MaxSlope 215  
 Afzup\_ReadIndex 214  
 Afzup\_WriteBits 214  
 An 212  
 Analog input filters 30  
 AnalogFiltered 213  
 AnalogIn 213  
 Analogue inputs 29  
 Auto correction 100  
 Available\_IO 229

**B**

Baud rate 202, 225, 274, 284, 388–391  
 Binary command 269  
 Boot up telegram 353  
 Bootloader\_Ver 229  
 Brakes and shaft reinforcement 410  
 Busvol 213

**C**

Cables 44, 47, 50, 53, 408  
 Cabling 283  
 Calculator (basic) 270  
 Calculator (options) 271  
 CAN A 282  
 CAN B 282  
 CAN bus connectors 284  
 CANopen 37, 184, 229, 274, 277–278, 280–292, 294, 298, 300, 303–305, 307–309, 319, 352–362  
   CAN bus connectors 284  
   CanOpen Explorer 284–289  
   Communication test 285  
   Connecting the SMC75 Controller to the CAN bus 283  
   DS-301 device profiles 352  
   Node id and baud rate 284  
 CANopen network 279  
 Capacitor 17  
 CE requirements 424  
 Checksum 228

CiA membership 278  
 Clear errors 85  
 COB-ID 282, 302  
 Command 202  
 Command timing 414  
 Conditional jump (multiple inputs) 261  
 Conditional jump (single input) 260  
 Confidence alarms 33  
 Confidence check 32  
 Connecting the SMC75 Controller to the CAN bus 283  
 Connections  
   MIS23x 41  
   SMC75 384  
 Connectors 42–53  
   M12 42–53  
 Control voltage 19  
 CVI control voltage 19

**D**

Declaration of Conformity 424–425  
 Digital inputs 27  
 Dimensions 401  
 Direction inputs 156  
 Download SDO 355  
 DS-301 290  
 DS301 specified Communications objects 290  
 DSP-402 Support 304

**E**

EDS file 283  
 Emergency object 292  
 Enable and Disable PDOs 295  
 Encoder\_Pos 200  
 Encoder\_Type 214  
 End-of-travel inputs 134  
 Enter safe mode 417  
 Err\_Bits 36, 208  
 Error acceleration 172  
 Error Control Services 360  
 Error handling 171  
 Error output 121  
 Error\_Mask 223  
 Errors, clearing 85  
 Exit safe mode 418  
 Expansion modules  
   MAC00-B1/B2/B4 43–44, 46–47, 49–50, 52–53  
 Ext\_Encoder 230  
 Ext\_Encoder\_Vel 231–241

**F**

Factors 307

- 
- Fbus\_Baud 230
  - Fbus\_Node Id 230
  - Filtering 32
  - Filters 39, 85
  - Filters, analog input 30
  - FilterStatus 215
  - Flash 85
  - Flwerr 201
  - Flwerrmax 201
  - Function description 162
  - Fuse dimensioning 20–22
  - G**
  - Galvanic isolation 26, 29, 36
  - Gear mode 155, 162
  - GEAR1 12, 151, 197, 200
  - GEAR2 12, 197, 200
  - Ground 26
  - Grounding 42, 45, 48, 51
  - Grounding, power supply 18
  - Group write register 419
  - Group\_Id 226
  - Group\_Seq 226
  - H**
  - Hardware\_Rev 228
  - Heartbeat 360–361
  - Home sensor 166
  - Home\_Bits 219
  - Homemode 210
  - Homing mode 313
  - I**
  - In physical position output 121
  - In position output 121
  - Index\_Offset 218–219
  - Inpos\_Mask 223
  - Input\_Filter\_Cnt 223
  - Input\_Filter\_Mask 223
  - Inputs 201
    - Analogue 29
    - Digital 27
    - End-of-travel 134
    - Multifunction I/O 162
    - Quadrature input 162
    - SMC75 25
    - Step pulse and direction 156
  - Interface
    - RS485 38
  - losetup 201, 220
  - IP67 44, 47, 50, 53
  - J**
  - Jump 260
  - Jump according to a comparison 272
  - Jump according to a register in the MAC motor 265
  - Jumps 260–261, 265, 272
  - L**
  - Life Guarding 360
  - M**
  - M12 42–53
  - MAB23x-01 410
  - MAB23x-02 410
  - MAC00-B1/B2/B4 Expansion Modules 43–44, 46–47, 49–50, 52–53
  - MAC00-B4 cables 44, 47, 50, 53
  - MacTalk 83–84, 88–91
  - Main Loop Time 274
  - Max\_P\_Ist 207
  - Max\_Voltage 228
  - Min bus voltage 172
  - Min\_Busvol 213
  - Min\_P\_Ist 206
  - MIS23x connections 41
  - MLT 274
  - MODE\_REG 302
  - Mode\_Reg 196, 288, 302
  - Modes of operation 12, 151, 252, 309
    - Gear mode 155
    - Passive mode 152
    - Positioning mode 154
    - Velocity mode 153
    - Zero search mode 163, 165–166, 168–169
  - Motor Connections 414
  - Motortype 227
  - Move (Absolute) 257
  - Move (Relative + set outputs) 255
  - Move (Relative + velocity change at a distance) 254
  - Move (Relative) 253
  - Move (Sensor) 258
  - Move current 98
  - Move operations 252
  - Multi-Master capability 281
  - My\_Addr 226
  - N**
  - Negative limit 134
  - NL\_Mask 221
  - NL, negative limit 134
  - NMT (Network Management services) 359
  - Node address 282
  - Node Guarding/Life Guarding 360
-

- 
- Node id 284
  - No-loss bus arbitration 281
  - Notsaved 229
  - NPN output 26
  - O**
  - Object dictionary 294
  - Object dictionary defined for DSP-402 support 305
  - Opening a file 86
  - Operating modes 12, 151–155, 163, 165–166, 168–169, 252, 309
    - Gear mode 162
  - Optical isolation 26, 29, 36
  - Option\_Bits 229
  - Outputs 201
    - Error output 121
    - In position 121
    - In pyhsical position 121
    - SMC75 user outputs 35
  - P**
  - P- terminal 17
  - P\_Home 210
  - P\_Ist 199, 201, 225, 300–301, 303
  - P\_New 212, 225
  - P\_Soll 12, 100, 151, 271, 298, 302
  - P+ terminal 17
  - Passive mode 152
  - PDOs 295, 297, 300, 354–355
  - PL, positive limit 134
  - PLC systems 121
  - Pn 212
  - PNP output 26
  - Position factor 307
  - Position limit min and max 172
  - Position mode 12
  - Positioning mode 154
  - Positioning-Speed Control 8, 10, 382–383
  - Positive limit 134
  - Power Supplies 409
  - Power Supply
    - Capacitor 17
  - Power supply
    - Grounding 18
  - Power supply, SMC75 16
  - Profile position mode 309
  - Prog\_Vers 195, 243
  - Program comments 252
  - Program jumps 260–261, 265, 272
  - Program status and command 419
  - Programming 243–255, 257–262, 264–272
    - PSU05-045 409
    - PSU24-075 409
    - PSU24-240 409
    - PSU48-1000 409
    - PSU48-1500 409
    - PSU48-240 409
    - PSU48-800 409
  - Pull-up resistor 26
  - Q**
  - QuickStep motors 11
  - R**
  - Read register 416
  - Read register block 417
  - Receive PDOs 297
  - Register overview 184
  - Registers 183–184, 197–202, 207–208, 214, 217, 224
    - A\_Soll 198, 253–255, 257–258, 298, 302
    - Acc\_Emerg 207
    - Afzup\_ConfMax 215
    - Afzup\_ConfMin 214
    - Afzup\_MaxSlope 215
    - Afzup\_ReadIndex 214
    - Afzup\_WriteBits 214
    - An 212
    - AnalogFiltered 213
    - AnalogIn 213
    - Available\_IO 229
    - Bootloader\_Ver 229
    - Busvol 213
    - Checksum 228
    - Command 202
    - Encoder\_Pos 200
    - Encoder\_Type 214
    - Err\_Bits 36, 208
    - Error\_Mask 223
    - Ext\_Encoder 230
    - Ext\_Encoder\_Vel 231–241
    - Fbus\_Baud 230
    - Fbus\_NodeId 230
    - FilterStatus 215
    - Flwerr 201
    - Flwerrmax 201
    - GEAR1 12, 151, 197, 200
    - GEAR2 12, 197, 200
-

- 
- Group\_Id 226
  - Group\_Seq 226
  - Hardware\_Rev 228
  - Home\_Bits 219
  - Homemode 210
  - Index\_Offset 218–219
  - Inpos\_Mask 223
  - Input\_Filter\_Cnt 223
  - Input\_Filter\_Mask 223
  - Inputs 201
  - losetup 201, 220
  - Max\_P\_Ist 207
  - Max\_Voltage 228
  - Min\_Busvol 213
  - Min\_P\_Ist 206
  - Mode\_Reg 196, 288, 302
  - Motortype 227
  - My\_Addr 226
  - NL\_Mask 221
  - Notsaved 229
  - Option\_Bits 229
  - Outputs 201
  - P\_Home 210
  - P\_Ist 199, 201, 225, 300–301, 303
  - P\_New 212, 225
  - P\_Soll 12, 100, 151, 271, 298, 302
  - Pn 212
  - Prog\_vers 195, 243
  - Register descriptions 197–202, 207–208, 214, 217, 224
  - Run\_Current 198, 299, 302
  - Serial\_Number 228
  - Setup\_Bits 220
  - Standby\_Current 199
  - Standby\_Time 199
  - Startmode 210
  - Statusbits 205
  - Temp 206, 240
  - Tn 212
  - Turntable\_Mode 221
  - V\_Home 210
  - V\_Ist 199, 300, 303
  - V\_Soll 12, 151, 198, 253–255, 257–258, 268, 288, 298, 302
  - V\_Start 196–197, 200
  - Vn 212
  - Remarks 252
  - Reset controller 418
  - Reset motor 85
  - Reset position 85
  - Resistors, termination 39
  - RS232/RS485 416
  - RS485 interface 38
  - Run\_Current 198, 299, 302
  - S**
  - Save in flash 85
  - Save position 266
  - Saving a file 86
  - Scope function 91
  - SDO (Service Data Objects) 355
  - Send FastMAC command 268–269
  - Serial communication 416
  - Serial\_Number 228
  - Set a register in the MIS motor 265
  - Set operation mode 252
  - Set outputs 259
  - Set position 267
  - Setup\_Bits 220
  - Short block length 281
  - Slope alarms 33
  - Slope limitation 32
  - SMC75 8, 10, 382–383
    - Inputs 25
  - SMC75 analogue inputs 29
  - SMC75 connector 384
  - SMC75 Power Supply 16
  - SMC75 user outputs 35
  - Specifications 388–389, 401
  - Standby current 98
  - Standby time 98
  - Standby\_Current 199
  - Standby\_Time 199
  - Startmode 210
  - Statusbits 205
  - Step pulse and direction inputs 156
  - Step pulse inputs 156
  - SYNC (Synchronisation Object) 358
  - T**
  - Technical Data 388–389, 401
  - Temp 206, 240
  - Temperature protection 121
  - Termination 283–284
  - Termination resistors 39
  - Tn 212
  - Torque 99
  - Torque zero search 167
  - Transmit PDOs 300
  - Trouble-shooting 406
-

Turntable\_Mode 221

## U

Unconditional jump 260

Upload SDO protocol 356

User outputs 35

## V

V\_Home 210

V\_Ist 199, 300, 303

V\_SOLL 268

V\_Soll 12, 151, 198, 253–255, 257–  
258, 268, 288, 298, 302

V\_Start 196–197, 200

Velocity accuracy 414

Velocity encoder factor 308

Velocity mode 12, 153, 312

Vn 212

Voltage Overload 29

## W

Wait for (x) ms before continuing 262

Wait for a register value before  
continuing 266

Wait for an input combination before continuing  
(multiple inputs) 264

Wait for an input combination before continuing  
(single input) 262

Write Register 417

Write to flash 418

## Z

Zero search 267

Torque 167

Zero search mode 163, 165–166, 168–  
169