INSTALLATION, USE AND MAINTENANCE MANUAL

FC Integrated Drive Technology

Riccardo Piccinini
Mini Motor S.P.A.

FC-GEN-EN
Firmware Revision: 4.056

Version 19, 09-2026

Table of Contents

1. Safety 1
- 1.1. Safety Warning 1
- 1.2. Storage and Installation Environment 1
- 1.3. Transport, installation and wiring 1
- 1.4. Operation and adjustments 2
- 1.5. Maintenance and Inspection 2

2. General Information 4
- 2.1. Introduction 4
- 2.2. Product Checks 4
- 2.3. Label check 5
- 2.4. Product Technical Specification 6
- 2.5. S3 Service 8

3. Installation 9
- 3.1. Electric Connections 9
- 3.2. Power Supply Connector CN1 - FC 10
- 3.3. USB service connector CN4 11
- 3.4. Fieldbus connection 12
- 3.5. Protective Device Installation 15
- 3.6. Mechanical Installation 16

4. Parametrization interface 18
- 4.1. Software Installation 18
- 4.2. Software 19

5. Parameters 25
- 5.1. System 25
- 5.2. Appl 25
- 5.3. Motor 33
- 5.4. Alarms 34

6. LED status 35
- 6.1. Fieldbus Board CanOpen\Modbus 35
- 6.2. Fieldbus Board EtherCAT 37
- 6.3. Fieldbus Board EthernetIP 39
- 6.4. Fieldbus Board Ethernet Powerlink 42
- 6.5. Fieldbus Board Ethernet ProfiNet IO 44

7. Alarms 46
- 7.1. Events 47
- 7.2. Troubleshooting 47

8. Modbus Registers 50
- 8.1. Read and Write Registers 50
- 8.2. Read only registers 51
- 8.3. Homing Registers 52

9. CiA402 - Servodrive application profile 53
- 9.1. DS402 overview 53
- 9.2. State Machine 55
- 9.3. position/speed/torque scaling 58
- 9.4. Homing 63
- 9.5. Touch Probe (optional) 63
- 9.6. Interpolated mode (optional) 65
- 9.7. Cyclic Synchronous Position mode (CSP) (optional) - Only EtherCAT/Powerlink 65
- 9.8. Default PDO Mapping 66

9.9. EMCY codes 66
9.10. DBS touch-probe extension (optional) 67
10. CanOpen/EtherCAT/Powerlink 70
10.1. Overview 70
10.2. DS301 Communication Layer 70
10.3. Object Dictionary 72
11. Ethernet IP 79
11.1. CIP Objects 79
11.2. Process Data 83
12. ProfiNET 92
12.1. Device identity 92
12.2. Process Data 92
12.3. Saving parameters 99
13. Homing Types 100
13.1. Type 1: Homing on anticlockwise limit switch and index pulse (resolver zero). 100
13.2. Type 2: Homing on clockwise limit switch and index pulse (resolver zero) 100
13.3. Type 3 and 4: homing on the positive Home Switch and Index Pulse (resolver zero) 100
13.4. Type 5 and 6: homing on the negative Home Switch and Index Pulse (resolver zero) 101
13.5. Type 7, 8, 9, 10, 11, 12, 13, 14: homing on the Home Switch and Index Pulse (resolver zero) 102
13.6. Types 33 and 34: Homing on Index Pulse (resolver zero). 103
13.7. Type 37: homing in current position. 103
14. Warranty Conditions 104
15. Disclaimer 105
Appendix A: Gains tuning 106
A.1. Tuning parameters 106
A.2. Control scheme 107
A.3. Preparation 108
A.4. Tuning the speed loop 108
A.5. Tuning the position loop 109
A.6. Fine adjustments 109
A.7. Diagnostics 110
Appendix B: Accelerometer 111
B.1. Axes and available data 111
B.2. Parameters 112
B.3. Sampling 112
B.4. Tuning the RMS filter 113
B.5. Over-acceleration events 113
B.6. Reading over the fieldbus 114
B.7. Vibration analysis 114
Appendix C: Encoder 117
C.1. Resolution 117
C.2. The backup battery 117
C.3. Storage management 118
C.4. Loss of the multiturn position 118
Appendix D: Prewired Cables 119
D.1. CBL023 119
D.2. CBL024 120
D.3. CBL031 121
D.4. CBL032 122
D.5. CBL033 123
D.6. CBL034 124

D.7. CBL035 125
D.8. CBL036 125
D.9. CBL037 126
D.10. CBL038 127
D.11. CBL039 128
D.12. CBL040 129
Appendix E: Regenerative Braking 131
E.1. When braking energy becomes a problem 131
E.2. First check: lengthen the deceleration 131
E.3. Sizing 132
E.4. Capacitors or braking resistor 132
E.5. The braking boards 133
E.6. Choosing the switch-on threshold 134
E.7. Connection 134

1. Safety

1.1. Safety Warning

Closely follow the warnings given in this chapter, as they are crucial for safety. The warning levels used in this manual are the following:

General hazard: risk of harm to people, the environment or the product.

Dangerous electrical voltage: risk of electric shock.

1.2. Storage and Installation Environment

Do **not** store or use the product in the following environments:

- Exposed to direct sunlight
- Exposed to higher levels of temperature, condensation and relative humidity than ones indicated in the specifications table
- Exposed to dust, salinity, water, oil, chemicals, corrosive or inflammable gases, fuels
- Pollution degree higher than 3
- When subject to impact or vibrations

1.3. Transport, installation and wiring

- Do not drop the product
- Grasp the product by the motor, always checking first to ensure it is not hot – burn danger!
- Do not install the product if there are damage signs on the packaging or on the product itself
- Wiring must be performed by qualified personnel, with the power supply disconnected
- During the installation and wiring operations, protect any product openings with appropriate guards in order to avoid that any kind of conductive metallic object is able to get in
- Ensure the connectors are firmly tightened
- Install appropriate safety components (fused disconnect switch) up line of the power supply
- Use shielded cables in locations subject to static electricity, strong magnetic fields or in the presence of electrical lines
- Use suitable and appropriately sized cables for connection of the power supply, motor and control signals connections;

- For control signal connections longer than 1m, it is recommended to use shielded cables with the shielding connected to an available earth signal;
- Make sure that the power supply voltage complies with the one on the rating plate;
- Ensure that there is sufficient ventilation to dissipate the heat produced by the product.
- Connect service usb only with the drive powered.

1.4. Operation and adjustments

- At first power on or after a period of more than two months of inactivity, it's recommended to keep fed the logical part for at least 24 hours;
- Activate the product only after checking the compatibility of the load and of the machine in which it will be installed;
- Stay away from the machine when the product is in reboot phase from an error (if the automatic reboot is enabled), because the machine could restart suddenly;
- Do not place body parts near the rotating parts;
- Use appropriate personal protective equipment during the work near to the axis ends (presence of sharp edges in the tongue seat);
- Before performing the machine test, provide adequate protections around the rotating parts (joints, etc.);
- Never activate the device or its switches with wet hands

1.5. Maintenance and Inspection

- The product is equipped with a backup battery used for the multi turn position retain. To guarantee the life performance and expected life it is necessary to perform a complete charge cycle for a minimum time of 24 consecutive hours every 6 months.
- Ensure that all safety precautions have been taken before performing maintenance or inspection operations
- Do not touch the heatsink and the motor as these components heat up when the product is operational and remain hot even after it is switched off
- Do not uninstall the product when it is powered, switch it off then wait at least two minutes before doing so
- Do not change the wiring, the supply voltages, etc. while the product is powered
- Do not repair or modify the product
- The product does not require any particular preventive maintenance. However, we recommend performing the following checks regularly:
  - Check the condition and tightening of the connections
  - Check that the heatsink is clean and ventilated

Do not treat the product as normal domestic waste: it must be taken to an appropriate

> collection point for electrical and electronic waste. Failure to observe this requirement can lead to civil and/or criminal penalties, and could have damaging consequences for the environment and human health.

2. General Information

2.1. Introduction

Thank you for choosing this product. This document is an integral part of the product, as it contains the procedures and parameters necessary for the assembly, installation, wiring, commissioning and support of the product. The buyer and/or qualified person with knowledge of electrical and/or electronic systems who will use the product need the use and maintenance manual and must keep it available for reference at any moment.

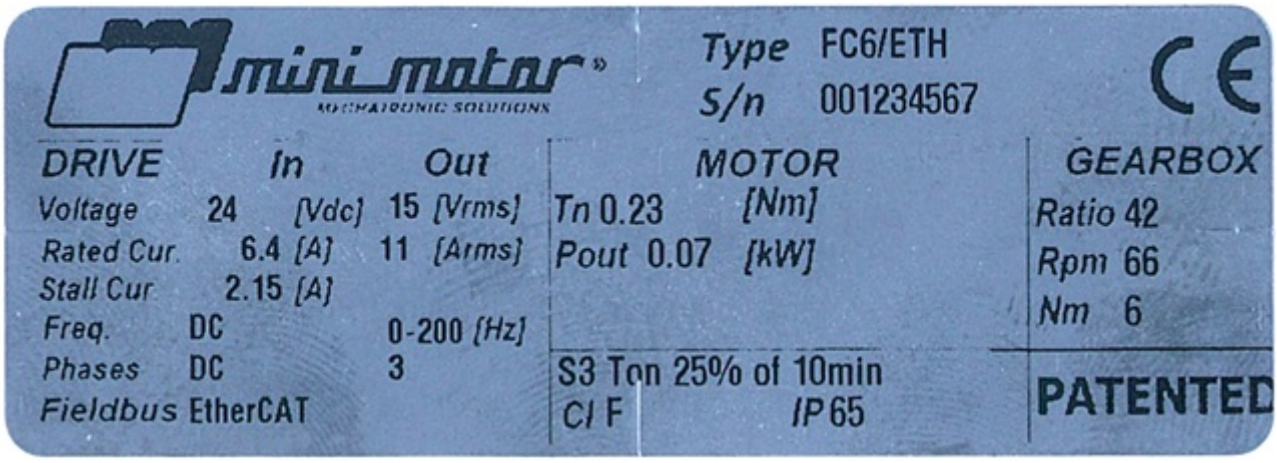
2.2. Product Checks

Upon delivery of the product, we recommend checking that it corresponds to that ordered and is complete. Keep in mind that only the connectors to wire the device (power, signals and bus) are included in the package. Contact the supplier’s technical support service for any problems relating to the product. For UR products check acceptability conditions and end user advice, which can be found on UL database at the file number 503911.

2.3. Label check

2.3.1. CE label

Example of label applied to the product:



Type	Product Name	
S/n	Serial Number	
	DRIVE	
	DRIVE IN	DRIVE OUT
Voltage	Drive Input Voltage [Vdc]	Drive Output Voltage [VRMS]
Current	Driver Input Current [A]	Drive Output Current [Arms]
Stall Cur.	Stall Current	
Freq	Drive input frequency	Drive output frequency
Phases	Drive input phases	Drive output phases
Fieldbus	Fieldbus protocol and optionals CSP/TP/DP/STO/MA	
	MOTOR	
Tn	Nominal motor torque [Nm]	
Tmax	Peak torque [Nm]	
Pout	Output motor power [kW]	
Cl	Insulation class	
IP	IP protection code	
Duty Service	Duty Service according to IEC 34-1 with on and off times. If absent, assume S1 service	
	GEARBOX	
Ratio	Gearbox ratio	
Rpm	Nominal output speed [rpm]	
Nm	Nominal output torque [Nm]	

2.4. Product Technical Specification

	FC3	FC6
Nominal Supply Voltage	24Vdc - 48Vdc	
Nominal Supply Current	5.8A @ 24Vdc - 2.8A@ 48Vdc	
Logic power draw	2,4W	
Nominal Speed	125rpm	72rpm
Nominal Torque	3Nm	6Nm
Service	S3 (25% of 10 minutes) Ton 150s (2.5 min) Toff 450s (7.5 min)	
Motor power output	63W	
PWM Frequency	4/8/12 kHz	
Motor Control Algorithm	Sinusoidal Brushless Magnetic multi turn absolute encoder (12bit single turn 20bit multi turn) 4096 pulse/rev.	
Operative mode	Velocity, Torque, Position, Homing control	
Fieldbus operating mode	Velocity, Torque, Position control, CSP (ECAT & Powerlink)	
Service communication	µUSB	
Communication fieldbus	Ethercat, EthernetIP, Powerlink, ProfiNET, Modbus RTU, canOpen	
(Main) protection features	Polarity inversion, overcurrent, overvoltage, under voltage, overheating, overload, memory error, communications error	
Cooling	Natural	
Standard ingress protection	IP65	
Ambient operating temperature	[0 + 40]°C	
Installation environment	Pollution class 3 Free from direct sunlight, vibrations, dust, corrosive or inflammable gases, oil vapour, fogs, water and humidity, with or without a high salinity level	
Humidity	[20 - 90]% RH (not condensing - not corrosive)	
Installation Position	Maximum height 1000m asl	
Storage temperature	[-20 +60]°C (transient temperatures during transport only)	
Weight	1.58kg	
Conformity	CE, complies with EMC directive 2014/30/UE, C3 class industrial environment	

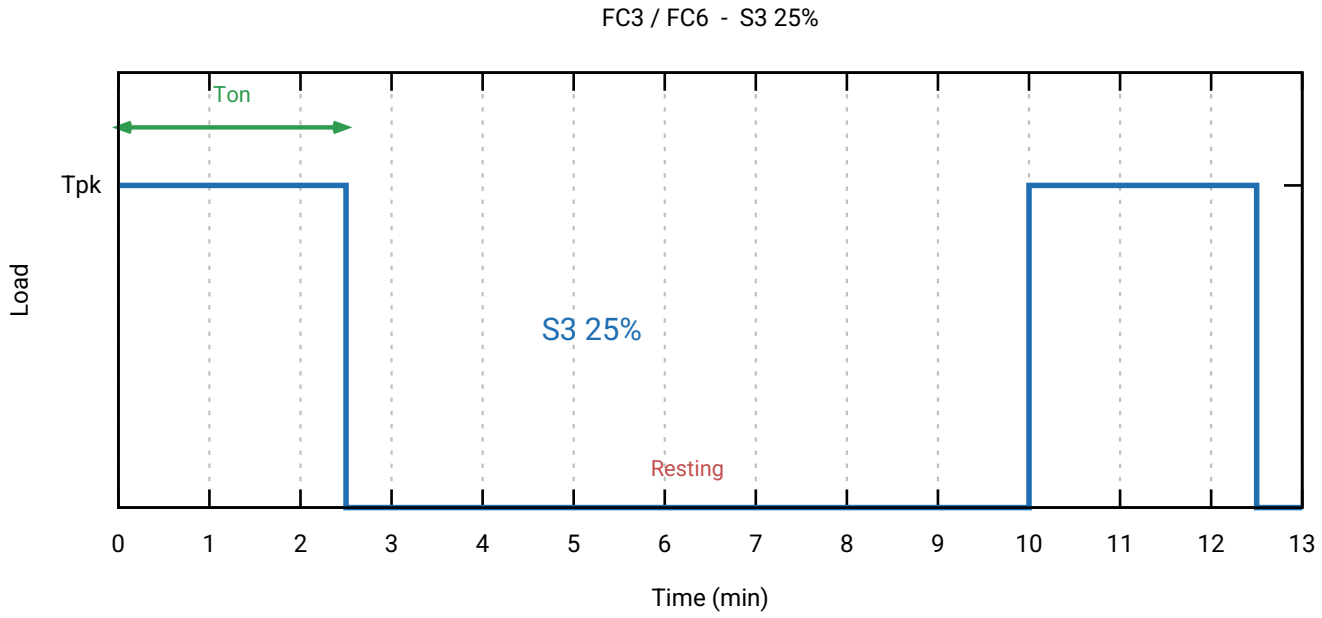
	FC10	FC20
Nominal Supply Voltage	24Vdc - 48Vdc	
Nominal Supply Current	13A @ 24Vdc - 6.5A@ 48Vdc	
Logic power draw	2,4W	
Nominal Speed	150rpm	75rpm
Nominal Torque	10Nm	20Nm
Service	S3 (15% of 10 minutes) Ton 90s (1.5 min) Toff 510s (8.5 min)	
Motor power output	180W	
PWM Frequency	4/8/12 kHz	
Motor Control Algorithm	Sinusoidal Brushless Magnetic multi turn absolute encoder (12bit single turn 20bit multi turn) 4096 pulse/rev.	
Operative mode	Velocity, Torque, Position, Homing control	
Fieldbus operating mode	Velocity, Torque, Position control, CSP (ECAT & Powerlink)	
Service communication	µUSB	
Communication fieldbus	Ethercat, EthernetIP, Powerlink, ProfiNET, Modbus RTU, canOpen	
(Main) protection features	Polarity inversion, overcurrent, overvoltage, under voltage, overheating, overload, memory error, communications error	
Cooling	Natural	
Standard ingress protection	IP65	
Ambient operating temperature	[0 + 40]°C	
Installation environment	Pollution class 3 Free from direct sunlight, vibrations, dust, corrosive or inflammable gases, oil vapour, fogs, water and humidity, with or without a high salinity level	
Humidity	[20 - 90]% RH (not condensing - not corrosive)	
Installation Position	Maximum height 1000m asl	
Storage temperature	[-20 +60]°C (transient temperatures during transport only)	
Weight	1.58kg	
Conformity	CE, complies with EMC directive 2014/30/UE, C3 class industrial environment	

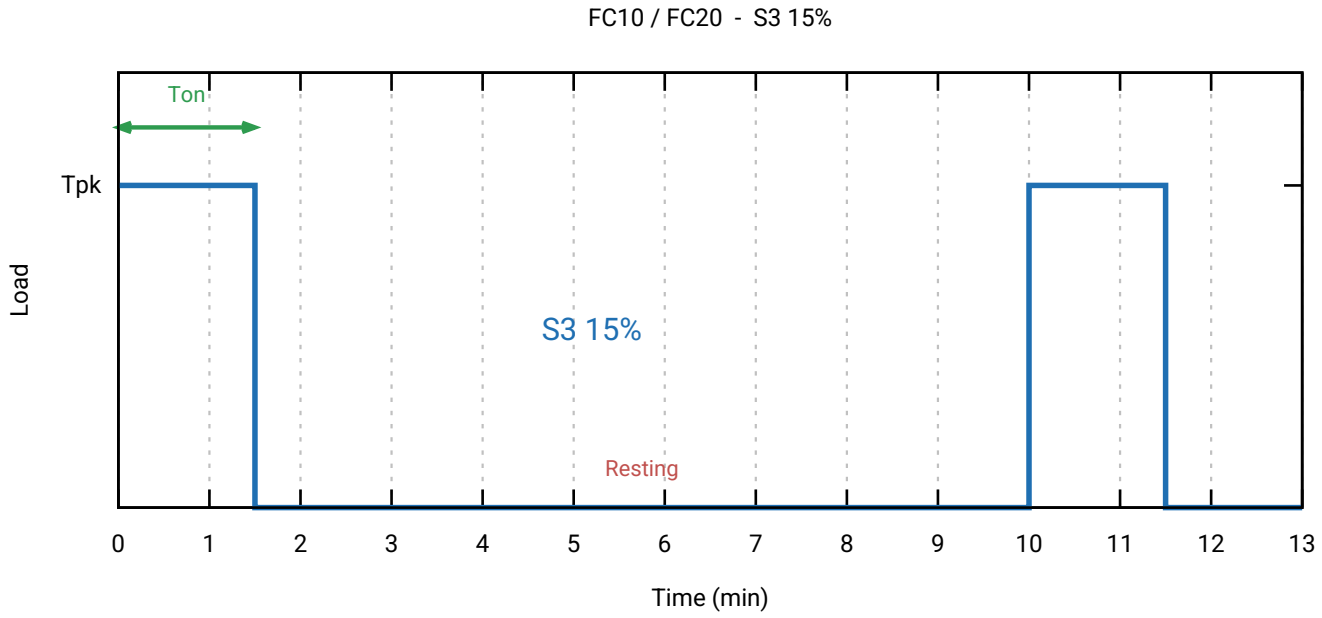
2.5. S3 Service

FC gearmotors operate in **periodic intermittent duty S3**: a sequence of identical cycles, each composed of a period of operation at constant load and a resting period without load and without electrical power. The starting current does not affect the temperature rise.

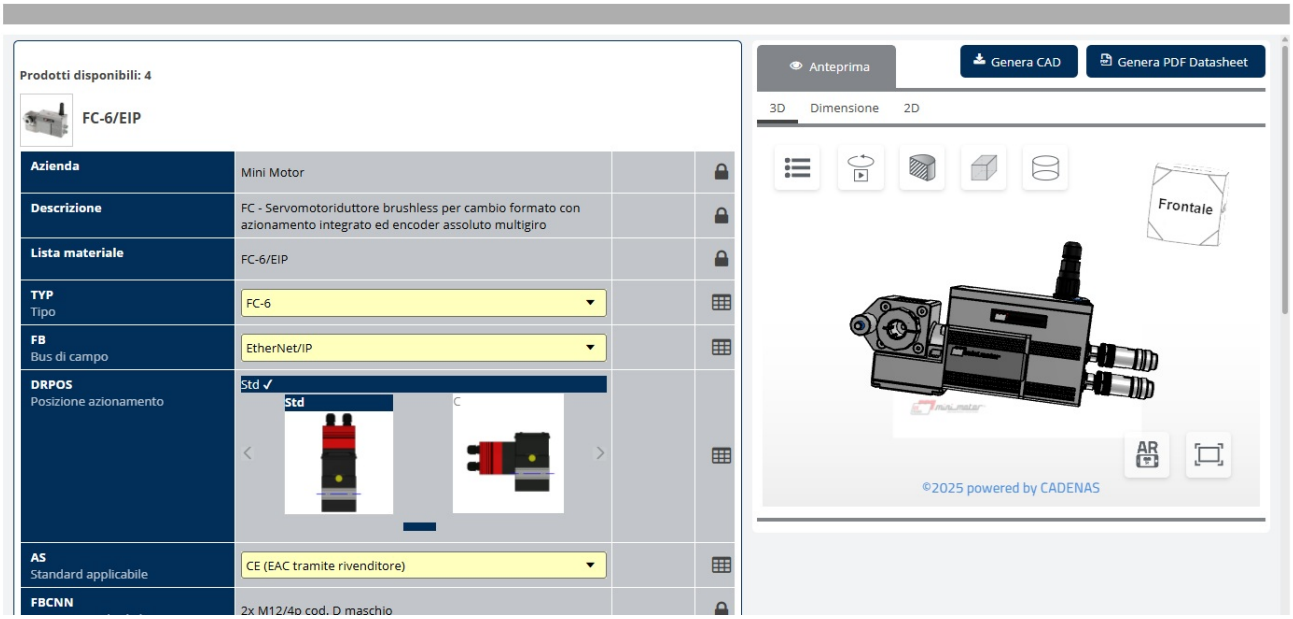
For a cycle time of 10 minutes:

Model	Duty	Time working	Resting time
FC3 / FC6	S3 25%	150 s (2.5 min)	450 s (7.5 min)
FC10 / FC20	S3 15%	90 s (1.5 min)	510 s (8.5 min)





3. Installation



For detailed information on the overall dimensions of all gearmotors in the DBS family, we recommend going to the dedicated product page from which you can find both specific Datasheet and the possibility of downloading 3D models in a variety of formats through the configurator.

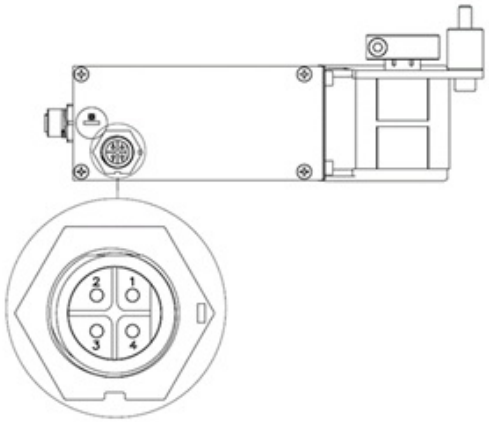
https://www.minimotor.com/en/products/fc3/

3.1. Electric Connections

Please note that the device power supply does not have a direct connection with the system earth.

WARNING: The Power must be supplied by a stabilized PSU. Protection fuse advised.

3.2. Power Supply Connector CN1 - FC



CN1		
Pin	Description	FC3/6/10/20
1	Logic GND	0V
2	Power GND	0V
3	+ Power	From 24 to 48V
4	+ Logic	From 24 to 48V

WARNING: the picture represents the top view of the connector fixed on the motor case.

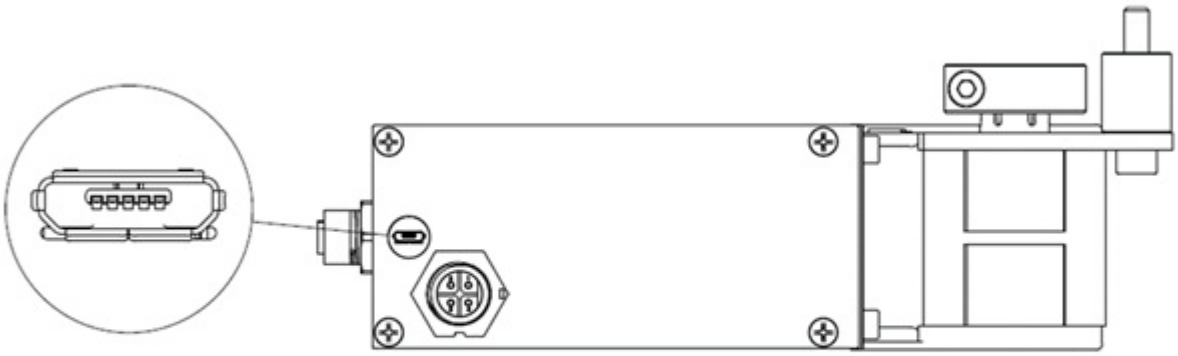
WARNING: Logic and Power GND must have common ground.

Awg 18 cable is recommended for power connection.

3.3. USB service connector CN4

Unscrew the metal cap to access the connector as shown in the picture

By using this micro USB connection, it is possible to set the device through the specific BSI software. You can download it here: https://www.minimotor.com/en/products/fc3/



⚠ The USB cable must be connected to the DBS system only with the drive powered up

3.4. Fieldbus connection

The optional available fieldbuses are:

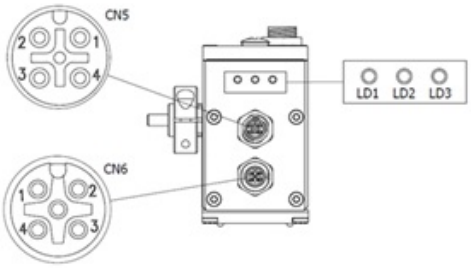
- CanOpen/Modbus RTU defined by “C” in the final part of the product name;
- Ethercat defined by “ETH” in the final part of the product name;
- Ethernet IP defined by “EIP” in the final part of the product name;
- Powerlink defined by “EPL” in the final part of the product name;
- Profinet defined by “EPN” in the final part of the product name.

Maximum tightening torques for connections: 0.5 Nm.

For the CanOpen/Modbus model the connectors on the DBS are M12 5 poles A coded, one male and one female.

For the EtherCAT/Profinet/Powerlink/EthernetIP models the connectors on the DBS are M12 4 poles D coded female

3.4.1. Canopen/Modbus RTU

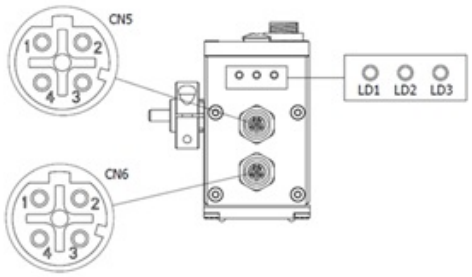


The pin function is the same for CN5 and CN6.

if more devices are connected together, pin 3 (GND) need to be connected ONLY if the logic power supply is not common (different GND).

Pin	Canopen	Modbus RTU
1	Housing shield	
2	Not connected	
3	CAN GND	GND
4	CAN H	485 B
5	CAN L	485 A

3.4.2. Ethernet IP

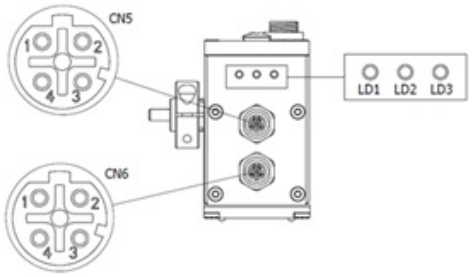


The pin function is the same for CN5 and CN6.

Connector CN5 = Connector CN6 = SWITCH

Pin	Ethernet IP
1	TX+
2	RX+
3	TX-
4	RX-

3.4.3. Profinet



The pin function is the same for CN5 and CN6.

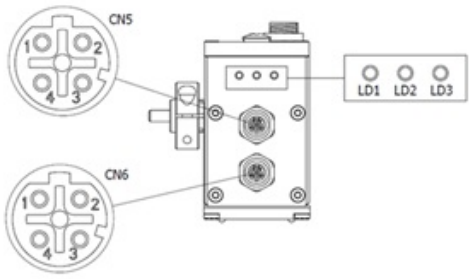
Connector CN5 = Connector CN6 = SWITCH

CN5 connector --- Port A

CN6 connector --- Port B

Pin	Profinet
1	TX+
2	RX+
3	TX-
4	RX-

3.4.4. Powerlink



The pin function is the same for CN5 and CN6.

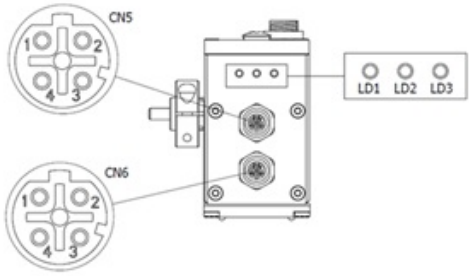
Connector CN5 = Connector CN6 = SWITCH

CN5 connector --- Port A

CN6 connector --- Port B

Pin	Powerlink
1	TX+
2	RX+
3	TX-
4	RX-

3.4.5. EtherCAT



The pin function is the same for CN5 and CN6.

CN5 connector --- Input

CN6 connector --- Output

Pin	CanOpen over EtherCAT
1	TX+
2	RX+

Pin	CanOpen over EtherCAT
3	TX-
4	RX-

3.5. Protective Device Installation

In order to prevent short circuits, the product must be connected to a power-supply line protected by devices to protect against overcurrent and short circuits, in compliance with electrical safety legislation, using standard fused, disconnect switches, etc.

For the installation of a switch up line of the power supply connection, we suggest to use isolating switch with slow blow fuse 20A

3.6. Mechanical Installation

Preliminary operations

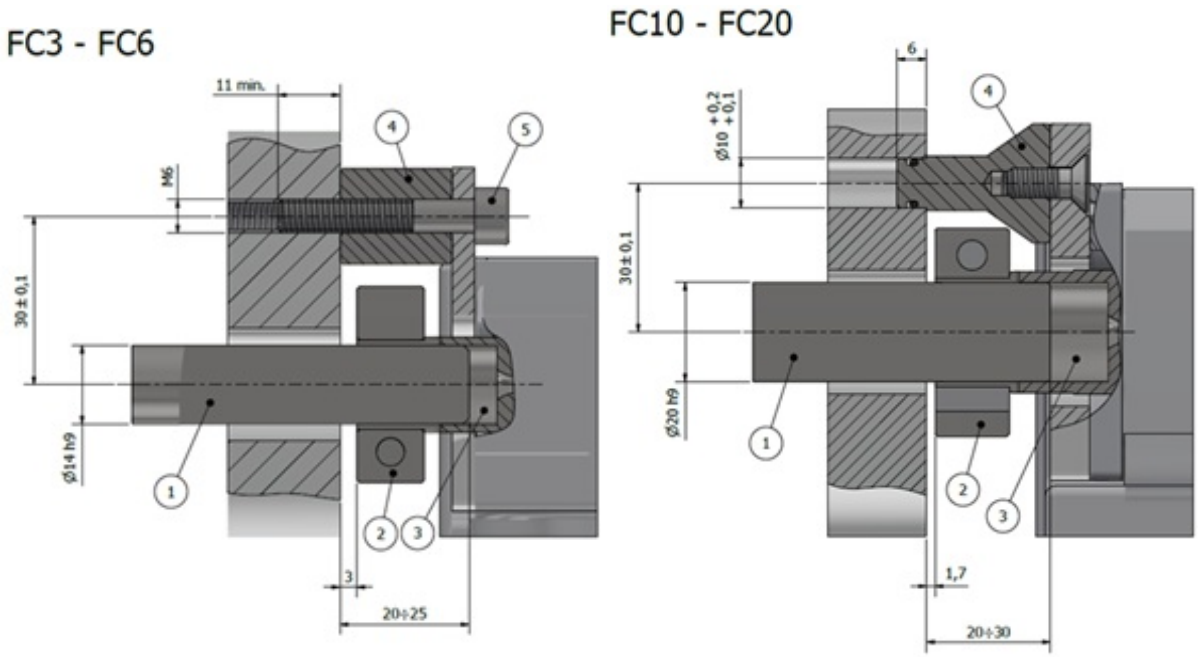
- Prepare the self-supporting drive shaft with diameter and protrusion according to fig. 1.
- Make a fixing hole with the dimensions and center distance prescribed by the drive shaft according to fig. 1.

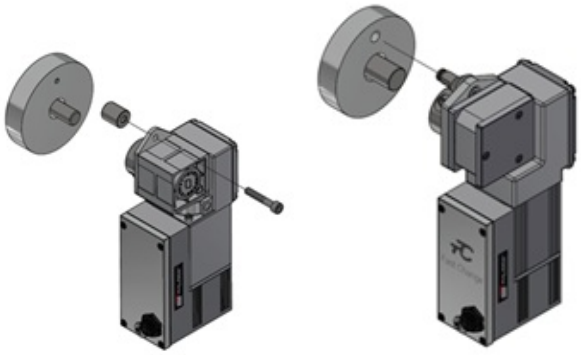
Assembly

- Carefully degrease the drive shaft and the gearmotor shaft from any oily residues at the coupling surface.
- Loosen the screws of the clamping ring .
- Insert the gearmotor shaft into the drive shaft until it reaches the stop of the anti-rotation pin .
- For models FC3/FC6 only, tighten screw with torque 7÷9 Nm, applying thread-locking adhesive.
- Tighten the clamping ring with thread-locking adhesive, tightening torque: Models FC3-FC6: 5.5÷6Nm Models FC10-FC20: 7.5÷8Nm

Note:

- Make sure to use the correct tools and follow the instructions carefully to avoid damaging the components.
- Apply thread-locking adhesive to all threaded connections to prevent them from loosening due to vibration.
- If you are not sure about how to proceed, please consult a qualified technician.





4. Parametrization interface

4.1. Software Installation

1. Download the “BSI software” and “BSI Usb Driver“ at the following link:
2. https://www.minimotor.com/prodotti/dbs/
3. Extract and run the downloaded files. Follow the installation guide procedure until it is finished.

   After the download extract the .zip file and run BSI.exe inside the folder (BSI icon).

   WARNING: this is not an installation file. It directly runs the interface software, so you do not need to install it. Because of this, every time the user wants to use the software only need to double click on this icon.

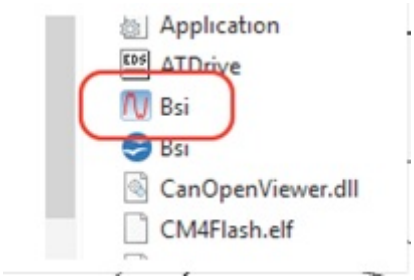


Figure 1. BSI icon

4. Connect the device through the micro USB to the pc and check that the logic is connected to the power.
5. Once the program is run, select the connection type. If choosing USB, wait for the device to be found and connect. If the DBS is connected via Fieldbus to a net that can exchange data over Ethernet (Ethernet IP, Profinet, Powerlink and EtherCAT with EoE functionality) you can connect typing the DBS IP address. (BSI connection interface). UDP port used: 25113.

Figure 2. BSI connection interface

6. If the firmware stored in the device is not up to date the screen BSI update interface

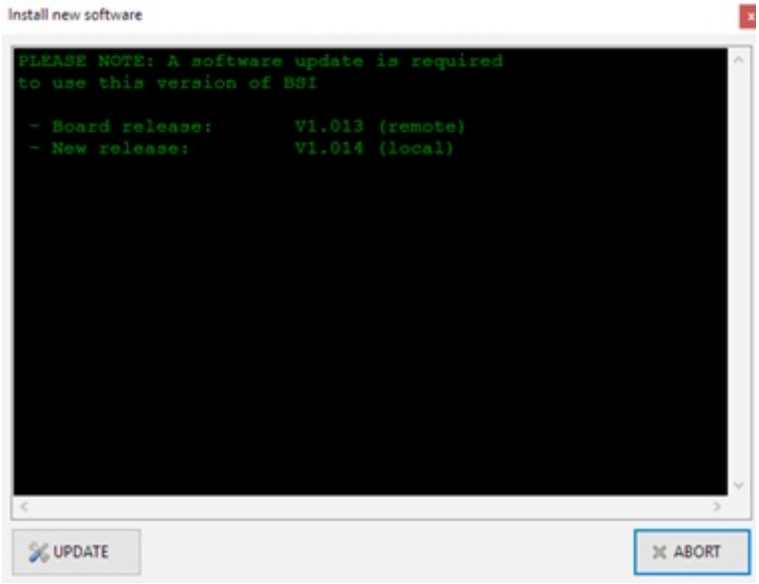


Figure 3. BSI update interface

4.2. Software

BSI update interface represents the main screen of BSI software. Following the menu it is possible to act on the motor parameter and to monitor the functional settings.

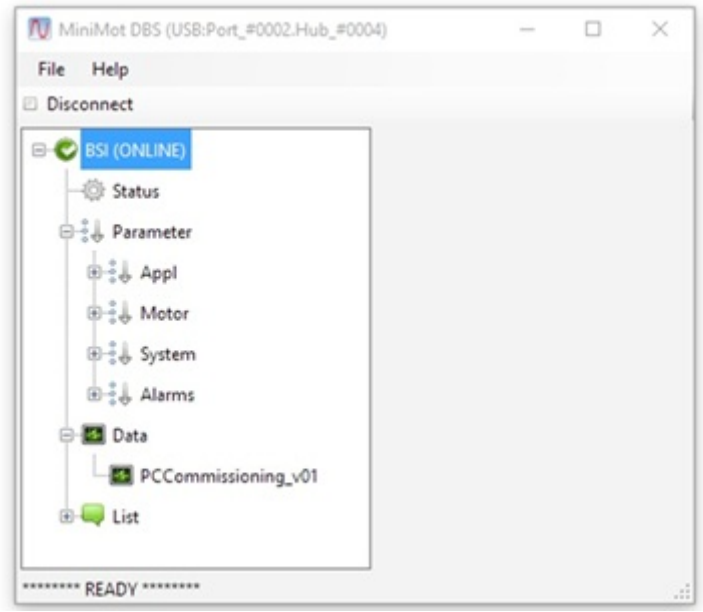


Figure 4. BSI update interface

4.2.1. Status

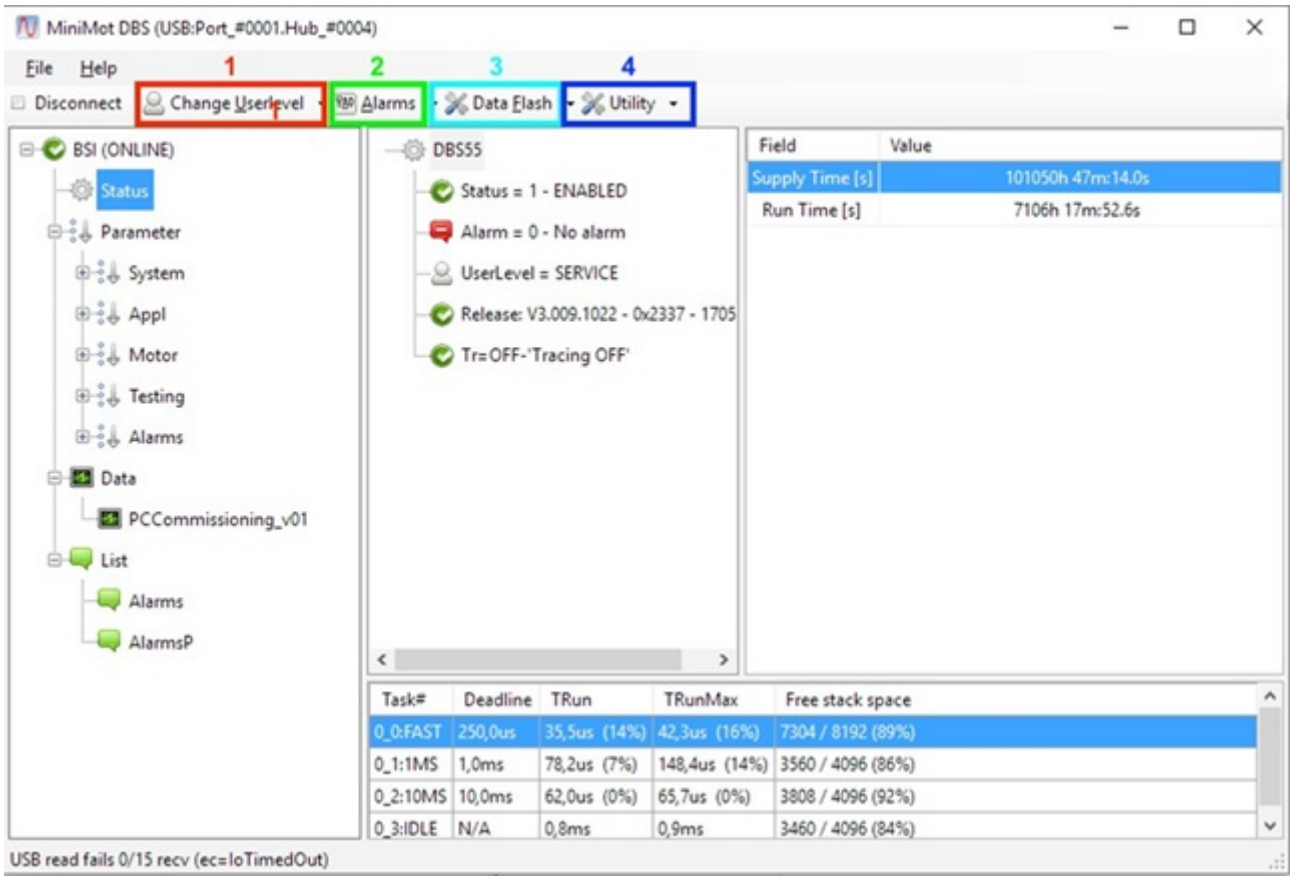


Figure 5. BSI main menu

Click on “Status” in order to know the motor status, manage the presence of fault and change the software user level.

1. “Change User Level”: use this option in order to change the software user level and have access to more setting parameters.

User	Free Access
Advance	Free Access
Eol	PW required
Service	PW required

2. “Alarms”: this button gives access to the alarm menu, in which you can reset the current alarm
3. “Data Flash”: diagnostic tool to save the Flash memory image.;
4. “Utility”: Diagnostic tool.

4.2.2. Alarm List

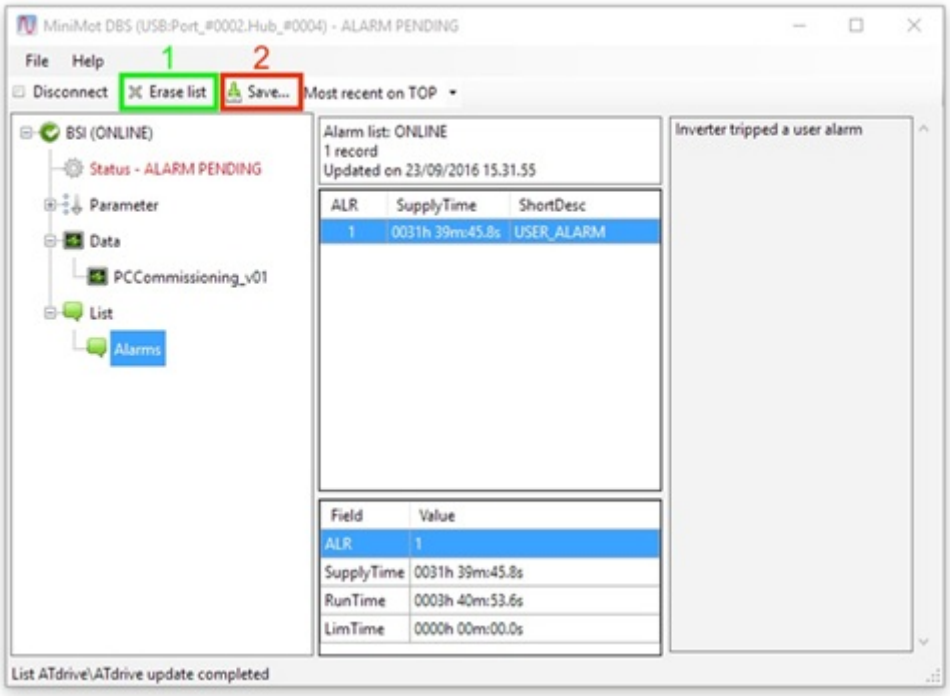


Figure 6. BSI Alarm list

By selecting “list” under the “alarm” voice, it is possible to find the alarm history list.

1. “Erase List”: use this command to erase the whole list.
2. “Save”: use this button to save the list into an external file.

4.2.3. Event List

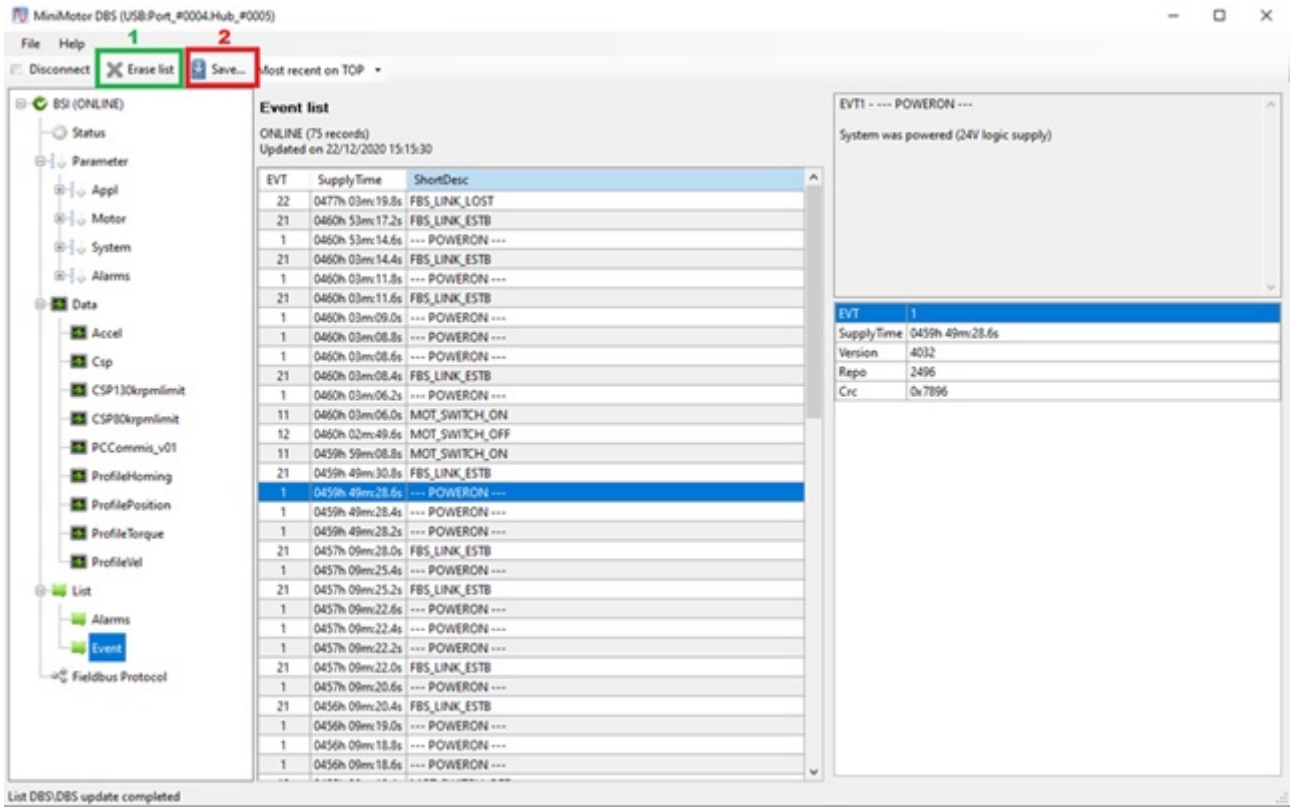


Figure 7. BSI Event List

By selecting “list” under the “event” voice, it is possible to find the event history list.

1. “Erase List”: use this command to erase the whole list.
2. “Save”: use this button to save the list into an external file.

4.2.4. Parameter

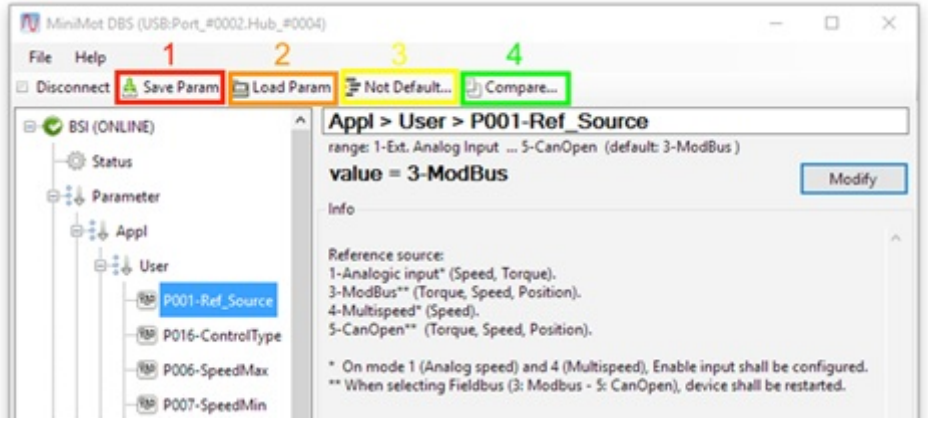


Figure 8. BSI Parameters

Under “Parameter” it is possible to find and set all the motor parameters. See the section of this guide in order to understand the meaning of all points and set the desired configuration.

In order to change the value or to select a different voice in one of the parameters, click on modify, change the value and then select Update to load the changes.

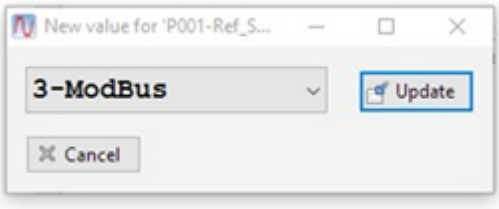


Figure 9. BSI P001

When the voice “parameter” is selected in the top of the window there are several useful options: . “Save Param”: allows to save the parameter setting into an external file.

1. “Load Param”: allows to upload the parameter stored into an external file, directly in the driver storage.
2. “Not default”: shows which parameter have been changed in respect of the default value.
3. “Compare”: allows to compare the parameter stored in the device with the parameters contained into an external file.

4.2.5. Monitoring

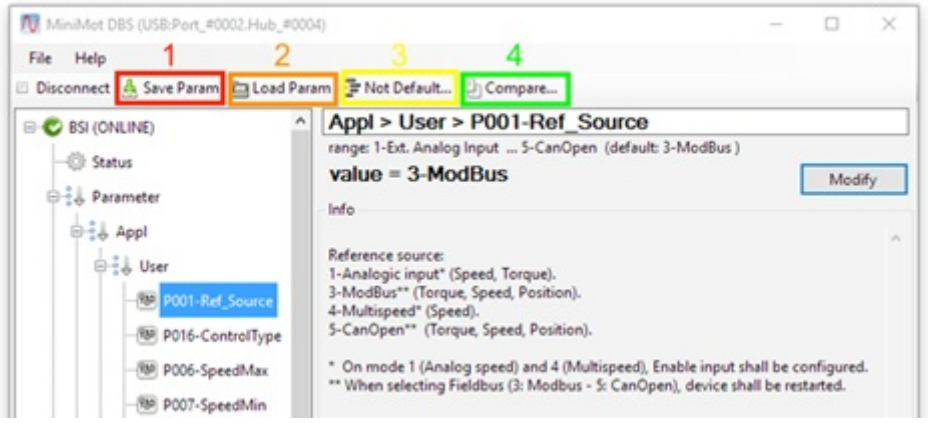


Figure 10. BSI Parameter monitoring

The program BSI is equipped with a tool useful in order to monitor the variables during the motor employment. Click on data and select PCCommissioning to see window shown in Figure 19 To activate the control of one or more variables toggle the white square under column “Upd”. The value is represented under the column “Value”. To use the oscilloscope, open the window settings and set the desired measurement, then click on single in order to obtain the measurement.

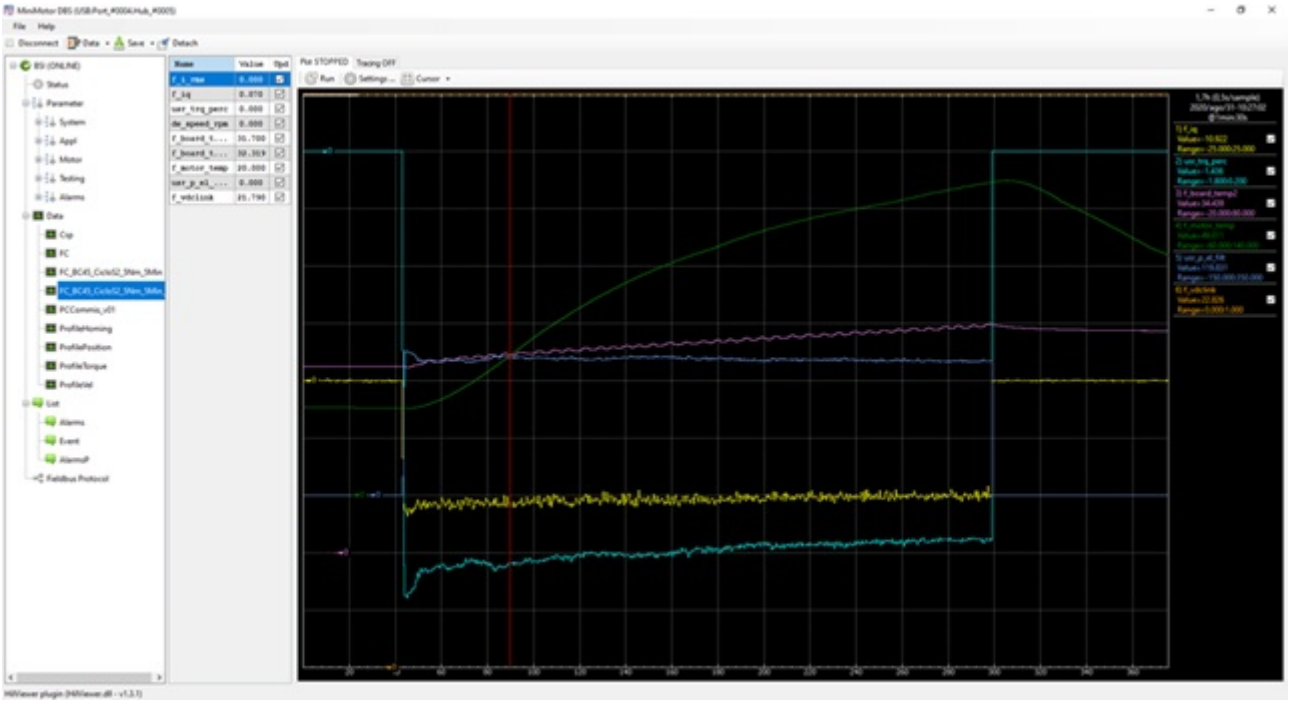


Figure 11. BSI Plot

The PLOT function, shown in BSI Plot, , enables to track values over longer period of time.

Here a table for common variables useful to monitor

NAME	DESCRIPTION
co_op_mode	DS402 6061h Mode of Operation Display
co_control_word	DS402 6040h Control Word
co_status_word	DS402 6041h Status Word
co_pp_target_pu	DS402 607Ah Target Position
co_pp_max_pu	DS402 607Dh.2 Positive position limit
co_pp_min_pu	DS402 607Dh.1 Negative position limit
co_pp_tlimit	DS402 6072h Max Torque
co_pp_wm	DS402 6082h Profile Velocity
co_pv_accel	DS402 6083h Profile Acceleration
co_pv_decel	DS402 6084h Profile Deceleration
de_speed_rpm	DS402 606Ch Velocity Actual Value
usr_trq_perc	DS402 6077h Torque Actual Value
usr_pos_pu_1ms	DS402 6064h Position Actual Value
co_pp_cmdnewset	DS402 6060h Bit 4 “new setpoint”
f_iq	Apk of current output of the drive
usr_poserror	Position error in encoder pulses from the position loop
diag_alr_code	3008h Actual Alarm
co_dc_ferror_win	DS402 6065h Following Error Window
co_dc_pos_window	DS402 6067h Position Window

5. Parameters

5.1. System

ID	NAME	DESCRIPTION	NOTE
000	MiniMotType	Motor Type	Read only
082	UndervoltMode	Undervolt error reporting logic	When Run → When the motor is running Always → Every time VDC is lost
201	Year	Production year	Read only
203	SerialNo	Motor serial number	Read only
202	StartUL	User level at start of motor	

5.2. Appl

5.2.1. User

ID	NAME	DESCRIPTION	NOTE
P001	MODE	Reference source selection	1 → Speed Analog Input1 Speed 2 → ModBus + Dig.Input1 Speed, Torque, Position, Homing 3 → ModBus Speed, Torque, Position, Homing 4 → Multispeed1 Speed 5 → CanOpen² Speed, Torque, Position, Homing 6 → CanOverECAT² Speed, Torque, Position, Homing 7 → Ethernet PowerLink² Speed, Torque, Position, Homing 8 → Ethernet/IP² Speed, Torque, Position, Homing 9 → Profinet IO² Speed, Torque, Position, Homing 10 → CanOverAir² Speed, Torque, Position, Homing 11 → TrqAnalogIn¹ Torque 12 → MultiTrq¹ Torque 13 → MultiPos¹ Position 14 → Ain Speed + Modbus Speed, Torque, Position, Homing 15 → MultiSpeedCycle¹ Speed 16 → MultiposCycle¹ Position ¹ In MODE 1, 2, 4, 11, 12, 13, 15 and 16 there must be at least one input as Enable ² Reboot is necessary when switching between modes
P006	Speed Max	Max speed reference	0 ÷ 6000 [rpm]
P007	Speed Min	Minimum speed reference in Analog mode	0 ÷ 6000 [rpm]
P008	Ramp Acc	Acceleration for speed/position control	1 ÷ 120000 [rpm/s]
P009	Ramp Dec	Deceleration for speed/position control	1 ÷ 120000 [rpm/s]

ID	NAME	DESCRIPTION	NOTE
P017	IMax	Maximum operative current	0 ÷ 22.1 / 18.7 [A] (55-50 / 55-100)
P021	MotorChangeDir1	Parameter to change motor rotation direction	1 → None (Disable) 2 → Inverted (Invert rotation) ¹ Only if P001=1,4,11,12 or 13
P164	PosMin	Minimum position limit	0 ÷ −2147483648 [position units]
P166	PosMax	Maximum position limit	0 ÷ +2147483647 [position units]

5.2.2. Input Output

ID	NAME	DESCRIPTION	NOTE
Analog			
P004	MinRefAnalog	Min analog reference	0.0% ÷ 100.0%
P005	MaxRefAnalog	Max analog reference	0.0% ÷ 100.0%
P043	AinMode	Set the analog input function	0 → Voltage ref input 0÷10 [V] 1 → Voltage ref input −10÷10 [V] 2 → Current ref input 0÷20 [mA] 3 → Current ref input 4÷20 [mA] WARNING: mode 2 and 3 are available only with MA option
P045	AinOffs	Offset correction for analog input	−10÷10 [V]
P046	AinGain	Gain correction for analog input	0.000÷2.000
P047	AinThre	Analog input threshold, as percentage	0.0÷100.0 [%]
Digital			

ID	NAME	DESCRIPTION	NOTE
P010	InputFunc1	Digital input function 1: enables control via digital pnp input 1 Defines function mapped to input 1	0 → Not Used 1 → Enable/Stand-By² 2 → Run/Stop² 3 → Run Edge Triggered² 4 → Stop Edge Triggered² 5 → Forward/Reverse² 6 → Run/Stop Forward² 7 → Run/Stop Reverse² 8 → Error Reset 9 → User Error 10 → Multispeed Binary Selection Bit 0² 11 → Multispeed Binary Selection Bit 1² 12 → Multispeed Binary Selection Bit 2² 14 → Brake Unlock 15 → CollisionSignal² 28 → Homing Start (edge triggered) 29 → Homing Start (level triggered) 30 → Home Switch¹ 31 → Fwd Limit Switch¹ 32 → Rev Limit Switch¹ ¹ Available in homing mode and local modes ² Only if P001=1,4,11,12,13
P011	InputFunc2	Digital input function 2	See P010
P012	InputFunc3	Digital input function 3	See P010 / STO 1 (optional)
P013	InputFunc4	Digital input function 4	See P010 / STO 2 (optional)
P014	InputFunc5	Digital input function 5	See P010
P032	AinFunc	Digital input associated to Analog Input	See P010
P020	DigInInvMask	Active low/high status of digital inputs	0b00001 → Input 1 Active Low 0b00101 → Inputs 1 & 3 Active Low
P015	OutFunc	Digital output — defines trigger condition for output signal	0 → Always Off 1 → Run 2 → Ready 7 → Homing Done 8 → Error 9 → Target Reached 10 → FieldBus DSP402
P064	TargReachedHyst	Window for “target reach” event	Speed [rpm] 0÷9999 Torque [mA] Position [step]
P065	TargReachHyst	Time window for “target reach” event completion	0÷9999 [ms]
P033	ExpOutEnable	Port expander configuration mask (only on WBS/WDBS)	0b00000000000 [bit]

5.2.3. Accelerometer

ID	NAME	DESCRIPTION	NOTE
P035	Endscale	Defines the Accelerometer End-Scale	2 – 4 – 8 [G]
P036	DataRate	Defines Accelerometer Output Rate	10–50–100–200–400–800 [Hz]
P037	RmsTau	Time filter constant for RMS computation of accelerations	1 ÷ 1000 [ms]
P038	EventLevel	Acceleration threshold for logging over-acceleration event	0.1 ÷ 10 [G]
P039	EventTime	Time filter constant for logging and over-acceleration event	1 ÷ 1000 [ms]
P040	EventInhibit	Inhibition time before logging a second over-acceleration event	1 ÷ 2400 [minutes]

5.2.4. Collision

ID	NAME	DESCRIPTION	NOTE
050	Deceleration	Deceleration after input trigger	Krpm/s
051	Bwspeed	Maximum speed for backward movement	rpm
052	BwAcc	Acceleration limit for backward movement	Krpm/s
053	BwDelta	Position of backward movement	Pulses

5.2.5. Multitrq

ID	NAME	DESCRIPTION	NOTE
P102	Multitorque 1	Defines the torque presets for the binary combinations of MultiTorque mode. In Multispeed/MultiPosition it works as torque limit for the corresponding profile Only with P001=4,12,13	-200 ÷ +200 [% nominal]
P103	Multitorque 2	See P102	-200 ÷ +200 [% nominal]
P104	Multitorque 3	See P102	-200 ÷ +200 [% nominal]
P105	Multitorque 4	See P102	-200 ÷ +200 [% nominal]
P106	Multitorque 5	See P102	-200 ÷ +200 [% nominal]
P107	Multitorque 6	See P102	-200 ÷ +200 [% nominal]
P108	Multitorque 7	See P102	-200 ÷ +200 [% nominal]
P109	Multitorque 8	See P102	-200 ÷ +200 [% nominal]

5.2.6. Multispeed

ID	NAME	DESCRIPTION	NOTE
P110	Multispeed 1	Defines the speed presets for the binary combinations of MultiSpeed mode. In MultiTorque/MultiPosition it works as speed limit for the corresponding profile Only with P001=4,12,13	-6000 ÷ 6000 [rpm]
P111	Multispeed 2	See P110	-6000 ÷ 6000 [rpm]
P112	Multispeed 3	See P110	-6000 ÷ 6000 [rpm]
P113	Multispeed 4	See P110	-6000 ÷ 6000 [rpm]
P114	Multispeed 5	See P110	-6000 ÷ 6000 [rpm]
P115	Multispeed 6	See P110	-6000 ÷ 6000 [rpm]
P116	Multispeed 7	See P110	-6000 ÷ 6000 [rpm]
P117	Multispeed 8	See P110	-6000 ÷ 6000 [rpm]

5.2.7. Multipos_ref

ID	NAME	DESCRIPTION	NOTE
P120	Multiposition1	Defines the position presets for the binary combinations of MultiPosition mode Only with P001=13	-2147483648 ÷ +2147483647 [user units]
P122	Multiposition2	See P120	-2147483648 ÷ +2147483647 [user units]
P124	Multiposition3	See P120	-2147483648 ÷ +2147483647 [user units]
P126	Multiposition4	See P120	-2147483648 ÷ +2147483647 [user units]
P128	Multiposition5	See P120	-2147483648 ÷ +2147483647 [user units]
P132	Multiposition6	See P120	-2147483648 ÷ +2147483647 [user units]
P134	Multiposition7	See P120	-2147483648 ÷ +2147483647 [user units]
P135	Multiposition8	See P120	-2147483648 ÷ +2147483647 [user units]

5.2.8. Multipos_mMode

ID	NAME	DESCRIPTION	NOTE
P221	Multiposref 1_mode	Defines the logic behind the positioning (absolute, relative or additive) Solo con P001=13	0 → Absolute 1 → Relative 2 → Additive
P222	Multiposref 2_mode	See P221	0 → Absolute 1 → Relative 2 → Additive
P223	Multiposref 3_mode	See P221	0 → Absolute 1 → Relative 2 → Additive
P224	Multiposref 4_mode	See P221	0 → Absolute 1 → Relative 2 → Additive
P225	Multiposref 5_mode	See P221	0 → Absolute 1 → Relative 2 → Additive
P226	Multiposref 6_mode	See P221	0 → Absolute 1 → Relative 2 → Additive
P228	Multiposref 8_mode	See P221	0 → Absolute 1 → Relative 2 → Additive

5.2.9. Multidelay

ID	NAME	DESCRIPTION	NOTE
P142	MultiDelay1	Time spent in the various positions/velocities in the automatic cyclic modes Only with P001=15,16	0.001 ÷ 32 [sec]
P143	MultiDelay2	See P142	0.001 ÷ 32 [sec]
P144	MultiDelay3	See P142	0.001 ÷ 32 [sec]
P145	MultiDelay4	See P142	0.001 ÷ 32 [sec]
P146	MultiDelay5	See P142	0.001 ÷ 32 [sec]
P147	MultiDelay6	See P142	0.001 ÷ 32 [sec]
P148	MultiDelay7	See P142	0.001 ÷ 32 [sec]
P149	MultiDelay8	See P142	0.001 ÷ 32 [sec]

5.2.10. Homing

ID	NAME	DESCRIPTION	NOTE
P152	HomingType	Defines the homing type according to the CanOpen CIA402 standard. See section 11	Available Homing: 1 ÷ 14 33, 34, 37

ID	NAME	DESCRIPTION	NOTE
P153	HomingSwitchSpeed	Speed used to search the home or limit switch	0 ÷ 4000 [rpm]
P154	HomingIndexSpeed	Positioning speed at the index pulse	0 ÷ 4000 [rpm]
P155	HomingAccel	Motor acceleration to reach homing speed from stop	0 ÷ 10000 [rpm/s]
P156	HomingDecel	Deceleration used to stop motor running at homing speed	0 ÷ 10000 [rpm/s]
P157	HomingTimeOut	Time limit for data exchange during homing process	1 ÷ 200 [s]
P158	HomingMaxCurr	Limit of motor current available during homing	0.001 ÷ 10 [A]
P159	HomingEnable	Enable or disable homing	0 → DISABLE 1 → ENABLE
P168	HomingOffsetLOW	Defines the offset value used in homing, composed of Low Word + High Word	Examples: Offset = 68000 (dec) → 0x109A0 (hex) P169 = 0x0001 P168 = 0x09A0 Offset = −1000 (dec) → FFFFFC18 (hex) P169 = 0xFFFF P168 = 0xFC18

5.2.11. Posscaling

ID	NOME	DESCRIZIONE	NOTE
P160	PosFactorNum	Defines the numerator used to calculate the scaling factor (position factor) used to represent the position in user defined unit. See 8.1.3 section	0 ÷ 32767
P161	PosFactorDen	Defines the denominator used to calculate the scaling factor (position factor) used to represent the position in user defined unit. See 8.1.3 section	0 ÷ 32767
P162	PosFactorDecimal	Defines the number of decimal places used to represent the profile unit.	0 ÷ 32767
P163	UserPosInversion	Allows the user to invert the count of the encoder based on motor rotational direction	None → Positive counting with clockwise rotation Inverted → Negative counting with clockwise rotation

5.2.12. Fieldbus

ID	NAME	DESCRIPTION	NOTE
EthernetIP			
P170	IpAddr	Allows to set the slave address of the motor using Ethernet/IP	192.168.250.100
P172	IpMask	Defines the Ethernet/IP subnet mask	255.255.0.0
P174	IpGateway	Defines the Ethernet/IP gateway address	0.0.0.0

ID	NAME	DESCRIPTION	NOTE
P215	IpMode	Ethernet/IP address assignment mode	0-Static 1-DHCP
P176	ProcDataLayout	Set default or extended process data for EIP and Profinet	0 Default 1 Extended_A (D I/O) 2 Extended_B (Touch probe 402) 3 Extended_C (Touch probe MM) 4 Extended_D (Accelerometer)
P177	SpeedUnitMode	Speed and acceleration object scaling	0 RPM – RPM/s 1 Increments/s – Increments/s²
CSP (Cyclic Synchronous Positioning)			
P178	WrongRefDelay	Delay of CSP position reference consistency	0 ÷ 1000 [ms]
P179	WrongRefBeh	Behaviour of CSP position target protection	0 NO ACTION 1 CORRECT 2 ALARM
P180	AccelMax	Acceleration limit for CSP pos. Ref.	1 ÷ 3000 krpm/s
P190	CSP_FF_Mode	CSP feedforward speed mode	1 Offset 60B1.0 2 Computed
Modbus			
P182	ModbusAddress	Set the slave address via RS485 using Modbus	1 ÷ 500
P183	ModbusMode	Defines Modbus RTU configuration: Word, Parity, Stop bits	8-N-2 → 8bit, none, 2bits 8-E-2 → 8bit, even, 1bits 8-O-1 → 8bit, odd, 1bits
P184	ModbusBaudrate	Baudrate for Modbus communication	0 → 9.6 [Kbaud] 1 → 19.2 2 → 38.4 3 → 57.6 4 → 115.2
P185	WIPProcDataLayout	Layout selection for CoA	0 WL_Default 1 CIP_Layout0 2 WL_XS_IO1 3 WL_XS_IO2
CanOpen			
P191	CanOPEN_Nodeid	Bus node address of the device	1 ÷ 127
P192	CanOPEN_Baudrate	Baudrate for CanOPEN communication	1 → 20 [Kbaud] 2 → 50 3 → 125 4 → 250 5 → 500 6 → 1 [Mbaud]
P193	CanOPEN_GuardTime	Defines the NMT Guard Time	0–1000 [ms] (0 disables)

ID	NAME	DESCRIPTION	NOTE
P194	CanOPEN_LifeFactor	Defines the NMT Life Factor	0–50 (0 disables)
Ecat (EtherCAT)			
P195	StationAddress	Default station address	0 ÷ 32767
P198	ProductCsp	EtherCAT product-id selection	0 PROFILE 1 CSP
PowerLink			
P197	NodeId	Default Node-ID for PowerLink node	0 ÷ 32767
ProfinetIO			
P199	ByteOrder	Set byte order style	Little / Big endian
Modulo			
P205	PosOptCode	Set value for 60F2h	As for CanOpen DS402
P206	PosRangeLimitMin	Modulo minimum range	posunits
P208	PosRangeLimitMax	Modulo maximum range	posunits

5.3. Motor

5.3.1. Brake

ID	NAME	DESCRIPTION	NOTE
P022	BrakeDeactTime	Deactivation time of brake AFTER motor control start	[ms]
P023	BrakeActTime	Activation time of brake BEFORE motor control stop	[ms]
P024	BrkUnlockTime	Elapsed time between Unlock V and Hold V states	[ms]
P025	BrkUnlockV	Voltage to Unlock the brake	0–48 Volt
P026	BrkHoldV	Voltage to Hold the brake unlocked	0–48 Volt
P034	BrakeFreq	Brake PWM frequency	400–10000 Hz
P057	BrkRand	PWM frequency randomizer	OFF–200%

5.3.2. Encoder

ID	NAME	DESCRIPTION	NOTE
P094	EncStRes	Single turn resolution selector. Values higher than 12bit are only valid on motors with HRE option	12–24 [bit]

ID	NAME	DESCRIPTION	NOTE
P252	EncLPFTau	Defines the time constant considered in the low-pass filter used to filter the speed measures from the encoder	0.001 ÷ 1.000 [s]

5.3.3. Control

ID	NAME	DESCRIPTION	NOTE
P237	KpSpeed	Defines the proportional gain in PI speed control	0 ÷ 3.2000
P238	kiSpeed	Defines the integral gain in speed PI control	0 ÷ 3.2000
P253	kpPos	Defines the proportional gain in P position control	0 ÷ 3.2000
P254	WstarFiltEnable	Enable the filter on the Speed reference	Enable - Disable
P256	Jloadff	Load Inertia	0 ÷ 327.67 Kgcm²

5.4. Alarms

ID	NAME	DESCRIPTION	NOTE
P210	Nreset	Defines the number of autoresettable alarm event exceeded which the fault is generated	0 ÷ 100
P211	TminReset	Defines the time after which the alarm can be autoresetted	0.5 ÷ 30 [s]
P212	Trestore	Defines the time after which the autoresettable alarm count is resetted	0.5 ÷ 1800 [s]

6. LED status

LED color and operating modes (flashing / powering mode) depend on the fieldbus expansion Board mounted (i.e. CanOpen, Ethercat, etc).

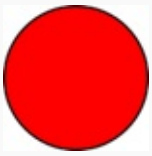
6.1. Fieldbus Board CanOpen\Modbus

The indication LEDs are three and they are mapped on the drive's LEDs in the following way:

LED	Indicazione	Stato
LD1	FLASHING (Red)	**Inverter alarm** Tacc = 0.5s Tperiod = 1s
	FLASHING (Red)	**PWM off** The number of flashes represent the type of control set • 0 flash: unchecked control • 1 flash: torque control • 2 flash: speed control • 3 flash: position control
	ON (Red)	**PWM on**
LD2	OFF	**Yellow LED always off**

LED	Indicazione	Stato
LD3	OFF	**Inverter alarm**
	FLASHING (Green)	**PWM off** The number of flashes represent the type of control set • 1 flash: Analog reference • 3 flashes: Modbus reference • 4 flashes: Multispeed reference • 5 flashes: CanOpen reference
	ON (Green)	**PWM on** (Intermittent speed proportional to rotation speed P006)

6.2. Fieldbus Board EtherCAT

LD1	LD2	LD3
RUN/ERR	ERR	L/A IN
		

LED	Indicazione	Stato
LD1	OFF	**INIT**
	ON (green)	**OPERATIONAL**
	BLINKING (2.5hz Green)	**PRE-OPERATIONAL**
	SINGLE FLASH (Green)	**SAFE-OPERATIONAL**
LD2	OFF	**No Error** The communication is working properly
	BLINKING (2.5hz Red)	**Invalid Configuration** General Configuration Error Possible reason: State change commanded by master is impossible due to register or object settings.
	SINGLE FLASH (Red)	**Local error**: Slave device application has changed the EtherCAT state autonomously. Possible reason 1: A host watchdog timeout has occurred. Possible reason 2: Synchronization Error, device enters Safe- Operational automatically.
	DOUBLE FLASH (Red)	**Local error**: Slave device application has changed the EtherCAT state autonomously. Possible reason 1: A host watchdog timeout has occurred. Possible reason 2: Synchronization Error, device enters Safe- Operational automatically.

LED	Indicazione	Stato
LD3	OFF	**No Connection** The device has no link to the Ethernet
	ON (Green)	**Connected** The device is connected but does not send/receives frames.
	FLICKERING (Green)	**Activity** Frames are being send/received.

Indicator State	Definition
ON	Constantly on
OFF	Constantly off
SINGLE FLASH	One short flash (200 ms) followed by a long off phase (1,000 ms)
DOUBLE FLASH	Two short flashes (200 ms each) separated by a short off phase (200 ms), followed by a long off phase (1,000 ms)
BLINKING (2.5 Hz)	Turns on and off at 2.5 Hz: 200 ms on, 200 ms off
FLICKERING (Load Dependent)	Approx. 10 Hz frequency to indicate high Ethernet activity: ~50 ms on, ~50 ms off. The indicator turns on and off in irregular intervals to indicate low Ethernet activity

6.3. Fieldbus Board EthernetIP

LD1	LD2	LD3
MS	NS	L/A IN

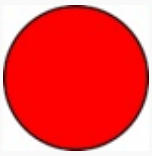
LED	Indicazione	Stato
LD1	OFF	**No Power**: the device is powered off.
	ON (green)	**Device operational** the device is operating correctly.
	FLASHING (1hz Green)	**Standby** the device has not been configured
	ON (Red)	**Major Unrecoverable fault** the device has detected a non recoverable major fault
	BLINKING (1 Hz Red)	**Major Unrecoverable fault** the device has detected a non recoverable major fault. An incorrect or inconsistent configuration would be considered a major recovereable fault
	FLASHING (Green/Red/Green)	**Major Unrecoverable fault** the device has detected a non recoverable major fault. An incorrect or inconsistent configuration would be considered a major recovereable fault

LED	Indicazione	Stato
LD2	OFF	**Not powered, no IP address** The device doesn't have an IP address (or is powered off).
	ON (Green)	**Connected** An IP address is configured, at least one CIP connection (any transport class) is established device has at least one established, and an Exclusive Owner connection has not timed out..
	BLINKING (1Hz Green)	**No Connections**: The device has obtained an IP address, the network; and no CIP connection are established and an exclusive Owner connection has not timed out.
	ON (Red)	**Duplicate IP**: The device has detected that its IP address is already in use.
	BLINKING (1 Hz Red)	**Connection Timeout**: An IP address is configured, and an Exclusive Owner connection for which the device is the target has timed out. The network status indicator returns to steady green only when all timed out Exclusive Owner connections are restablished.
	FLASHING (Green/Red/off)	**Serftest**: Thedevice is performing its power-up testing. Refer to description for module status LED self-test.
LD3	OFF	**No Connection** The device has no link to the Ethernet
	ON (Green)	**Connected** The device is connected but does not send/receives frames.
	FLASHING (Yellow)	**Activity** Frames are being send/received.
	OFF (Yellow)	**No Activity** The device does not send/receive Ethernet frames.

Indicator State	Definition
ON	Constantly on
OFF	Constantly off
BLINKING (1 Hz)	Turns on and off iso-phase at a frequency of 2.5 Hz: on for 200 ms, off for 200 ms

Indicator State	Definition
FLICKERING (Load Dependent)	Approx. 10 Hz to indicate high Ethernet activity: ~50 ms on, ~50 ms off. Irregular intervals to indicate low Ethernet activity

6.4. Fieldbus Board Ethernet Powerlink

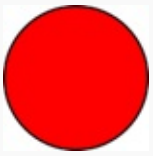
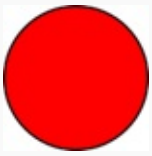
LD1	LD2	LD3
Status	Error	L/A IN
		

LED	Indicazione	Stato
LD1	OFF	**No Power**: the device is powered off.
	ON (green)	**Operational** the device in the "operational" state.
	TRIPLE FLASH (Green)	**ReadytoOperate** the device in the "Readytooperate" state.
	DOUBLE FLASH (Green)	**Pre Operational 2** the device in the "Pre Operational 2" state.
	SINGLE FLASH (Green	**Pre Operational 1** the device in the "Pre Operational 1" state.
	FLICKERING (10hz)	**Basic Ethernet** the device in the "Basic Ethernet" state.
	BLINKING (Green)	**Stopped** the device in the "Stopped" state.
LD2	OFF	**No Error** The device has no error.
	ON (Red)	**Error** The device has detected an error.

LED	Indicazione	Stato
LD3	OFF	**No Connection** The device has no link to the Ethernet
	ON (Green)	**Connected** The device is connected but does not send/receives frames.
	FLASHING (Green)	**Activity** Frames are being send/received.

Indicator State	Definition
ON	Constantly on
OFF	Constantly off
BLINKING (2.5 Hz)	Turns on and off iso-phase at 2.5 Hz: 200 ms on, 200 ms off
TRIPLE FLASH	Three short flashes (200 ms each) separated by 200 ms off phases; followed by a long off phase (1,000 ms)
DOUBLE FLASH	Two short flashes (200 ms each) separated by a 200 ms off phase; followed by a long off phase (1,000 ms)
SINGLE FLASH	One short flash (200 ms) followed by a long off phase (1,000 ms)
FLICKERING (10 Hz)	Alternates red and green LEDs at 10 Hz: 50 ms on, 50 ms off
FLICKERING (Load Dependent)	Approx. 10 Hz for high Ethernet activity: ~50 ms on, ~50 ms off. Irregular intervals indicate low Ethernet activity

6.5. Fieldbus Board Ethernet ProfiNet IO

LD1	LD2	LD3
System failure	Bus Failure	L/RX/TX
		

LED	Indicazione	Stato
LD1	OFF	**No Error**
	ON (Red)	**Error** Watchdog timeout; channel, generic or extended diagnosis present; system error.
	FLASHING (Red, 1Hz, 3s)	**DCP Signal Service** Device discovery initiated via BUS
LD2	OFF	**No Error** The device has no error.
	ON (Red)	**Error** No configuration or low speed physical link; or no physical link.
	FLASHING (2 Hz Red)	No Data exchange
LD3	OFF	**No Connection** The device has no link to the Ethernet
	ON (Green)	**Connected** The device is connected but does not send/receives frames.
	FLASHING (Yellow)	**Activity** Frames are being send/received.
	OFF (Yellow)	**No Activity** The device does not send/receive Ethernet frames.

Indicator State	Definition
ON	Constantly on
OFF	Constantly off
FLASHING (1 Hz, 3 s)	Turns on and off for 3 seconds at 1 Hz: 500 ms on, 500 ms off
FLASHING (2 Hz)	Turns on and off at 2 Hz: 250 ms on, 250 ms off
FLICKERING (Load Dependent)	Approx. 10 Hz for high Ethernet activity: ~50 ms on, ~50 ms off. Irregular intervals for low Ethernet activity

7. Alarms

ID	NAME	DESCRIPTION
0	NONE	No alarm
1	USER	User alarm
2	DATAFLASH	Data flash error
3	OVERVOLT	Overvoltage (autoresettable)
4	OVERCURR_HW1	Overcurrent hardware (autoresettable)
5	OVERCURR_SW1	Overcurrent software (autoresettable)
6	OVERTEMP_BOARD1	Board overtemperature (autoresettable)
7	OVERTEMP_MOTOR1	Motor overtemperature (autoresettable)
8	ENC_SPI_COMM	Encoder SPI communication error (autoresettable)
9	UNDERVOLT	Undervoltage (autoresettable)
10	ENC_ERR	Maximum encoder speed exceeded
11	MDB_WATCHDOG	Modbus Watchdog alarm
20	PARAM_ERROR	Parameter storage not available or damaged
21	PARAM_VAL_ERROR	Wrong value parameter
22	PARAM_RESTORE_DEFAULT	Restore parameter default error
23	MOTOR_PARAM_RESTORE_DEFAULT	Restore motor parameter default error
24	DATAFLASH_READ	Error in Flash Read
25	COLLISION_DETECTED	A collision was detected and motor stopped
30	FUNC_NOT_ENABLED	The function commanded is not enabled
90	EXT_DIG	User alarm generated by digital input
91	APPL_SWITCH_CONSISTENCY	Homing switch error
92	APPL_MOT_UNDEFINED	Motor type not selected
99	SILCHK_ALARM	Error in STO HW consistency
101	CSP_SYNC	Error in the CSP Synchronization
102	CSP_WRONGREF	Error in the CSP reference position
105	1MS_OVERRUN	Task 1ms exceeded the deadline (real-time)
106	10MS_OVERRUN	Task 10ms exceeded the deadline (real-time)
107	FB_BOARD_MISMATCH	Fieldbus board mismatched
108	FB_BOARD_FAILURE	Fieldbus board not correctly initialized
109	FB_SEC_MEMORY	Fieldbus board’s secure memory not working

ID	NAME	DESCRIPTION
110	FB_APP_FAILURE	Fieldbus mode not compatible with hardware
120	WL_BOARD_FAILURE	Wireless Board not working
150	WRONG_HOMING_PROCEDURE	Homing error
151	TIMEOUT_HOMING	Timeout Homing Error
200	PMSM_ENCNOTALIGNED	Encoder not aligned
201	PMSM_ENCALRMOTION	Encoder cannot detect the movement
202	PMSM_ENCALRVERSO	Wrong encoder direction
203[1]	PMSM_ENCSPI_MT	Multi-turn encoder error. The multiturn position is not reliable. (autoresettable)
204	PMSM_ENCSPI_ST	Single-turn encoder error. The single-turn position is not reliable. (autoresettable)
205	PMSM_ENCALRTIMEOUT	Alignment was not completed within the prescribed time
206	PMSM_24BIT_ENCSPI	Error in 24bit encoder
210	ELPARAUTOTUNE_TIMEOUT	Time-out auto tuning error

7.1. Events

ID	NAME	DESCRIPTION
1	STARTUP	System restarted
2	USER	User event
3	ACCMAX_REACHED	Acceleration Limit reached
11	MOT_SWITCH_ON	Motor was switched on by user
12	MOT_SWITCH_OFF	Motor was switched off by user
13	MOT_SWITCH_OFF_ALR	Motor was switched off by an alarm
21	FBS_LINK_ESTB	Fieldbus Link Established
22	FBS_LINK_LOST	Fieldbus Link Lost
23	FBS_UPTIME_DSC_VHIGH	Netx time discontinuity
24	FBS_UPTIME_DSC_VLOW	Netx time discontinuity with low voltage
25	FBS_UPTIME_DSC_KO	Netx time discontinuity (not responding)

7.2. Troubleshooting

7.2.1. ALR3 Overvoltage

- *Event*: The drive exceeds the voltage limit of 55V.
- *Most common cause*: Overvoltage induced by motor regeneration.

- *Solution*: Reduce deceleration/motor gains. Add a braking resistor to dissipate energy. Read the **Regenerative Braking** appendix for more information.

7.2.2. ALR5 Overcurrent SW

- *Event*: Current to be supplied calculated by the drive higher than the maximum limits.
- *Most common cause*: Dynamics required by the motor too high
- *Solution*: Check and possibly decrease the calibration of gains, accelerations, and decelerations. Check for possible mechanical jams that increase the load unexpectedly. Evaluate a larger motor.

7.2.3. ALR9 Undervoltage

- *Event*: Power voltage drops below 9V
- *Most common cause*: Electrical connections and/or power supply.
- *Solution*: Check that the power supply size is adequate to support the motor load. Check the electrical connections and the physical condition of connectors on the cable and motor.

Note: An emergency power shutdown with the motor activated generates an alarm 9. If the motor has parameter P082 set to 1-Always, this alarm is also signaled when power is lost with the motor not activated.

7.2.4. ALR8, 102, 108

- *Event*: Self-diagnosis of the drive's electronic components identifies an error.
- *Most common cause*: The specific component is not working correctly.
- *Solution*: Contact the brand's technical support.

7.2.5. ALR 6,7

- *Event*: Exceeding the temperature limits. Motor >=140°C, Electronics >=90°C.
- *Most common cause*: Heavy workload, operation outside specifications.
- *Solution*: Check ambient temperature such that ⇐40°C. Check for possible obstructions, dirt, or poor ventilation in the motor environment. Check the motor duty cycle.

7.2.6. ALR 203, 204

- *Event*: The encoder no longer has confidence in the value of the multi-turn encoder position.
- *Most common cause*: Encoder battery discharged.
- *Solution*: Leave the motor connected to the logic power supply for at least 24 hours before disconnecting it. Read the **Encoder** appendix for more information.

7.2.7. Frequent Disconnect from fieldbus Network - Event 22

- *Event*: Physical connection with the fieldbus cable lost. Signaled by events 22 on BSI.
- *Most common cause*: Inconsistent electrical connection of the cable.
- *Solution*: Check cable continuity. Check connector tightness. Check for physical damage to motor

connectors.

[1] The multiturn position is not reliable, so is mandatory to make a homing procedure.

8. Modbus Registers

Note: Available only in CanOpen/Modbus RTU driver version (DBS55/---/C)(DBS80/--/C)

8.1. Read and Write Registers

ID	NAME	DESCRIPTION	NOTE
3	Control type	Available motor control type	RW 16-bit unsigned integer + Equivalent to P016 parameter 0 None 1 Torque 2 Speed 3 Position
20	Watchdog time	Watchdog time (ms)	RW 16-bit unsigned integer Default value: 0 - If value is set to zero, watchdog is disabled - If set between 100ms–65000ms, register 840 is cleared and alarm MODBUS_WATCHDOG is triggered [1]
840	Run and direction	Enable the drive and set the rotating direction	RW 16-bit unsigned integer Equivalent to P021 parameter Bit 0: 0 Not enable / 1 Enable Bit 1: 0 Forward / 1 Reverse
829	Last alarm / clear alarm list	Shows last alarm stored and allows clearing the list	RW 16-bit unsigned integer Read = Last alarm Write = 0 → Clear alarm list
843	Alarm reset	Allows alarm reset	RW 16-bit unsigned integer 0 No reset 0 → 1 Clear on rising edge
4144	Actual alarm	Shows actual error code	See error description in Alarm Section
4177	Enable write params	Stores settled parameters in flash	0: storing disabled 0 → 1: changes are stored
11	Max current	Maximum motor current in torque control	RW 16-bit unsigned integer Equivalent to P017 parameter Range: 0–20000 [mA]
16	Max acceleration	Maximum acceleration (speed/position)	RW 16-bit unsigned integer Equivalent to P008 parameter Range: 1–15000 [rpm/s]
17	Max deceleration	Maximum deceleration (speed/position)	RW 16-bit unsigned integer Equivalent to P009 parameter Range: 1–15000 [rpm/s]
841	Speed target	Define motor target speed	RW 16-bit signed integer [rpm]
844 – 845	Position target	Define target position	RW 32-bit signed integer [User Unit]

8.2. Read only registers

ID	NOME	DESCRIZIONE	NOTE
801	Motor Speed	Shows the actual motor speed	R 16-bit signed integer [rpm]
802	Motor RMS Current	Shows the RMS current absorbed by the motor	R 16-bit signed integer [Arms*0.1]
804	Power supply voltage	Shows the DC motor bus voltage	R 16-bit unsigned integer [V*0.1]
805	Motor voltage supply	Shows the RMS voltage on the motor	R 16-bit signed integer [Vrms*0.1]
806	Motor temperature	Shows the motor temperature	R 16-bit signed integer [0.1°C]
808-809	Motor position	Shows the actual motor position	R 32-bit signed integer [User Unit]
810	Motor direction	Shows the rotating direction of the motor • 0 – Forward • 1 – Reverse	R 16-bit signed integer
812	Analog input value		R 16-bit signed integer
815	Board temperature Sensor 1	Shows the board temperature	R 16-bit signed integer [0.1°C]
816	Board temperature Sensor 2	Shows the board temperature	R 16-bit signed integer [0.1°C]
818	Digital Input	Shows the digital inputs status	R 16-bit signed integer
820	Digital output	Shows the digital outputs status	R 16-bit signed integer
822	Brake output	Shows the electromagnetic brake status	R 16-bit signed integer
825	Register Overload %	Describe the voltage overload percentage referred to the motor	R 16-bit signed integer [0–100%]
832	Supply time 16LSB	Supply time (seconds) – 16LSB	R 16-bit signed integer (latched on reading)
833	Supply time 16MSB	Supply time (seconds) – 16MSB	R 16-bit signed integer (latched on reading)
834	Inverter on 16LSB	Inverter on (seconds) – 16LSB	R 16-bit signed integer (latched on reading)
835	Inverter on 16MSB	Inverter on (seconds) – 16MSB	R 16-bit signed integer (latched on reading)
837	SW version micro	Indicates the software version stored in the drive	R 16-bit signed integer
838	SW version encoder	Indicates the software version stored in the encoder	R 16-bit signed integer

ID	NOME	DESCRIZIONE	NOTE
4132	Reference generator state	Indicates the drive status	* 0 – OFF: No enable * 1 – ENABLED: Enabled without run * 2 – HALTED: PWM on with zero speed control * 3 – ACCEL: PWM on with acceleration control * 4 – DECEL: PWM on with deceleration control * 5 – STEADY: PWM on with non-zero speed control * 6 – STOPPING: PWM with deceleration until zero speed

8.3. Homing Registers

ID	NOME	DESCRIZIONE	NOTE
257	Homing type	Defines the homing type according to the CanOpen CIA402 standard. Equivalent to P152 parameter. • Homing available: • 1 – 15 • 33 – 35	RW 16-bit signed integer
258–259	Home Offset	Defines the offset value taken into account during the homing. Equivalent to P168–169 parameters	RW 32-bit unsigned integer [User unit]
260	Homing switch search speed	Defines the motor speed during the research of the limit switch. Equivalent to P153 parameter	RW 16-bit unsigned integer [rpm]
261	Homing index search speed	Defines the motor speed during the achievement of the index pulse. Equivalent to P154 parameter	RW 16-bit unsigned integer [rpm]
262	Homing acceleration	Motor acceleration to reach the settled home velocity from zero. Equivalent to P155 parameter	RW 16-bit unsigned integer [rpm/s]
263	Homing deceleration	Motor deceleration to reach zero velocity from the target. Equivalent to P156 parameter	RW 16-bit unsigned integer [rpm/s]
264	Homing max time out	Time limit for data exchange during homing. If exceeded, an error is triggered. Equivalent to P157 parameter	RW 16-bit unsigned integer [s]
265	Homing max current	Current limit during homing. If exceeded, an error is triggered. Equivalent to P158 parameter	RW 16-bit unsigned integer [mA]
266	Homing Cmd	Enables or disables the homing procedure	RW 16-bit unsigned integer * 0 – None * 1 – Enable

[1] Note: values from 1ms–99ms are internally set as 100ms (timeout <100ms not allowed)

9. CiA402 - Servodrive application profile

9.1. DS402 overview

The **CiA 402** profile (also known as **DS402** or **DSP402**, standardised as *IEC 61800-7-201/301*) is the device profile dedicated to *drives and motion control*. Independently from the fieldbus, it defines the *object dictionary*, the *state machine*, the *modes of operation* and the encoding of the *Control Word* and *Status Word*. Its purpose is to make the drive interoperable: a master can command servo drives from different manufacturers using the same standard objects.

9.1.1. How the servo drive is commanded

The interaction between the master (the controller) and the slave (the servo drive) always follows the same pattern, regardless of the selected operating mode:

- The servo drive exposes an **object dictionary**: every quantity (commands, references, state, feedback) is an object identified by a 16-bit hexadecimal index (e.g. `6040h`).
- The master enables power by walking the **state machine** (see State Machine): by writing the **Control Word** (`6040h`) it transitions from *Switch On Disabled* up to *Operation Enabled*, the only state in which the motor produces torque.
- The actual drive state is read from the **Status Word** (`6041h`).
- The active **operating mode** is selected with the *Modes of Operation* object (`6060h`) and confirmed by the drive in *Modes of Operation Display* (`6061h`).
- References and feedback are exchanged through the objects specific to the selected mode (for example *target position* `607Ah`, *target velocity* `60FFh`, *target torque* `6071h`, *position actual value* `6064h`).

In short: power is enabled through the state machine, the operating mode is chosen with `6060h`, references are written and motion is commanded by acting on the Control Word bits, reading the outcome from the Status Word.

9.1.2. Supported modes of operation

The implemented modes of operation are the following:

Mode	Code	Description
Profile Position	1	Point-to-point positioning: the motor reaches a target position following a trapezoidal profile with settable velocity and acceleration/deceleration. Suited to classic absolute or relative positioning moves.
Profile Velocity	3	Velocity control: the motor follows a target speed reached through acceleration and deceleration ramps.
Profile Torque	4	Torque control: the motor delivers the requested torque (current), reached along a settable ramp.
Homing	6	Zero search: the motor runs a homing procedure to establish the absolute position reference according to the selected method.
Interpolated Position	7	Interpolated position (optional): the master sends a position reference every synchronisation cycle and the drive interpolates the trajectory. Typical of numeric control.
Cyclic Synchronous Position	8	Cyclic synchronous position (optional): position reference updated every cycle with optional feedforward offsets, for the tightest master control. Available only on EtherCAT/Powerlink.

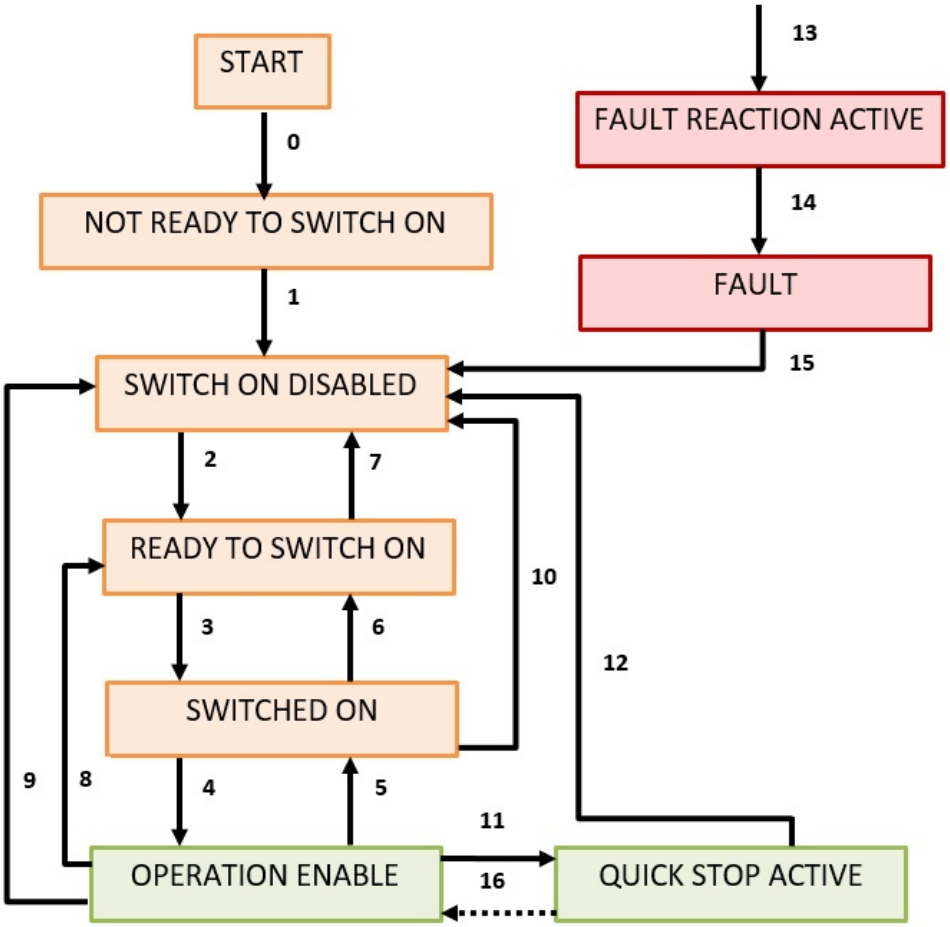
Each mode is detailed in its own section later in this chapter.

9.1.3. Fieldbuses and Modbus

The motor is commanded in DS402 over all the supported fieldbuses — **CANopen**, **EtherCAT**, **EtherNet/IP**, **PROFINET** and **Powerlink** — keeping *the same object model and the same application code*: only the transport layer changes, while the objects (`6040h`, `6041h`, `6060h`, …) and the logic described in this chapter stay identical.

The only exception is **Modbus**, which does not adopt the DS402 profile but works through *dedicated registers*: for its use refer to the Modbus Registers chapter.

9.2. State Machine



States in orange have power disabled. States in green have power enabled. Red states are alarm states. State machine description

State	Descriptions
Not Ready to Switch On	Axis not ready to start, initialization not completed.
Switch On Disable	Axis ready to be started, parameters can be transferred, power can be enabled.
Ready To Switch On	Parameters can be transferred, power can be enabled, motor cannot be actuated
Switched On	Parameters can be transferred, power can be enabled, motor cannot be actuated
Operation Enable	No fault Present, Motor functions and power are enabled.
Quick Stop Active	Drive stopped with an emergency ramp, power enabled, functions disabled.
Fault Reaction Active	An Error occurred, drive is stopping.
Fault	A fault is active, drive is stopped and disabled.

To change between states, following the numbered transitions, you need to send the correct control word, object 6040h, with specific bits enabled as shown in the following table.

Command	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	Transitions
Shutdown	0	X	1	1	0	2,6,8

Command	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	Transitions
Switch On	0	0	1	1	1	3
Disable Voltage	0	X	X	0	X	7,9,10,12
Quick Stop	0	X	0	1	X	7,10,11
Disable Operation	0	0	1	1	1	5
Enable Operation	0	1	1	1	1	4,16
Fault Reset		X	X	X	X	15

Control Word encoding is as follows:

Bit	abbreviation	Name	Specific Functionality
0	so	Switch On	
1	ev	Enable Voltage	
2	qs	Quick Stop	
3	eo	Enable Operation	
4	oms	Operating Mode Specific	New Setpoint - Homing Start - Enable Interpolation
5	oms	Operating Mode Specific	Change Setpoint Immediately
6	oms	Operating Mode Specific	Absolute/Relative
7	fr	Fault Reset	
8	h	Halt	
9	oms	Operating Mode Specific	Change on Setpoint
10	r	Reserved	
11	ms	Manufacturer Specific	Touch Probe Offset Mode (optional)
12	ms	Manufacturer Specific	Touch Probe Module mode (optional)
13	ms	Manufacturer Specific	
14	ms	Manufacturer Specific	
15	ms	Manufacturer Specific	

The drive state is communicated with object 6041h, the Status Word. The DS402 State Machine States are communicated with the following bits from the Status Word

StatusWord bit values	State Machine State
xxxx xxxx x0xx 0000	Not Ready To Switch On
xxxx xxxx x1xx 0000	Switch On Disabled
xxxx xxxx x01x 0001	Ready To Switch On

StatusWord bit values	State Machine State
xxxx xxxx x01x 0011	Switched On
xxxx xxxx x01x 0111	Operation Enabled
xxxx xxxx x00x 0111	Quick Stop Active
xxxx xxxx x0xx 1111	Fault Reaction Active
xxxx xxxx x0xx 1000	Fault

Status Word encoding is the following:

Bit	abbreviation	Nome
0	rtso	Ready to Switch On
1	so	Switched On
2	oe	Operation Enabled
3	f	Fault
4	ve	Voltage Enabled
5	qs	Quick Stop
6	sod	Switch On Disabled
7	w	Warning
8	ms	Manufacturer Specific
9	rm	Remote
10	tr	Target Reached
11	ila	Internal Limit Active
12	oms	Operating Mode Specific
13	oms	Operating Mode Specific
14	ms	Manufacturer Specific
15	ms	Manufacturer Specific

The Operating Mode Specific bits of the Status Word have the following meaning depending on the Operating Mode.

Bit	Profile Position	Profile Velocity	Profile Torque	Homing	Interpolation	CSP
10	Target Reached			Homing Done	Target Reached	Toggle
12	Set Point Ack	Zero Speed	–	–	Interpolation Active	Following Setpoint
13	Following Error	–	–	Homing Error	–	Following Error

9.3. position/speed/torque scaling

Position scaling is managed using a manufacturer specific mechanism. By changing servodrive parameters is possible to set the relation from position pulses and “position units” by means of:

- PosFactorNum (NUM)
- PosFactorDen (DEN)
- PosFactorDecimal (DEC)
- UserPosInversion (INV)

$PPulses = PPosUnits * NUM/DEN * 10^{DEC} * INV$
$PPosUnits = PPulses * DEN/NUM * 10^{-DEC} * INV$

Numeric Example:

- Motor pulses: 4096/rev.
- Gearbox Ratio: 7
- Screw pitch: 3

I need the position to be in millimeter and I want 0.1 precision.

$$(4096 * 7) / 3 = > 28672 / 3$$

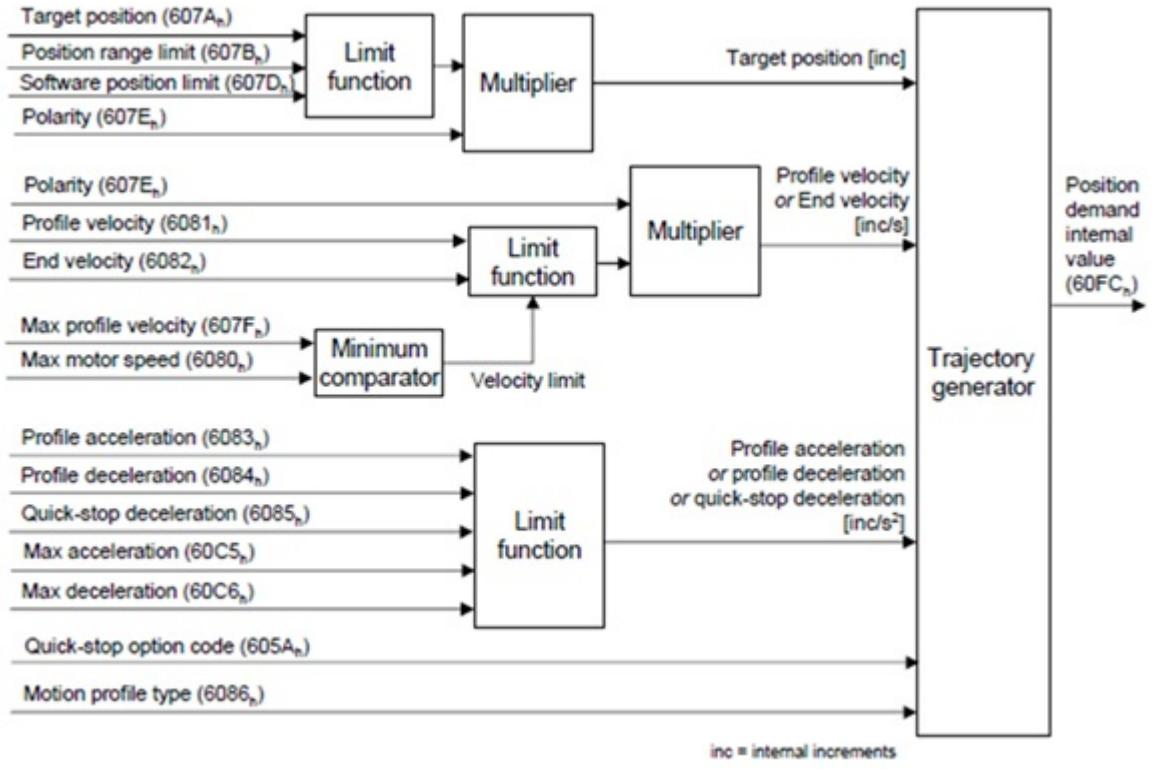
So the final numbers to put in the parameters for the scaling are:
PosFactorNum = 28672
PosFactorDen = 3
PosFactorDecimal = 1

Furthermore, is possible to set the polarity of the position and the speed information; the position scaling is not accessible via CanOpen but the position scaling defined via parameter is applied to all position objects that are specified as “Position units”.

The only exception is the polarity object (607Eh) that allows from the bus to change the direction of the position and/or velocity as defined by CiA402:

- Position Polarity Flag affects Profile position
- Velocity Polarity Flag affects Profile velocity

Homing mode and Torque mode are not affected by polarity. Velocity units are RPM (round per minute) and acceleration units are RPM/S (round per minute over a second). <<< === Profile Position



The profile position implementation includes the following features:

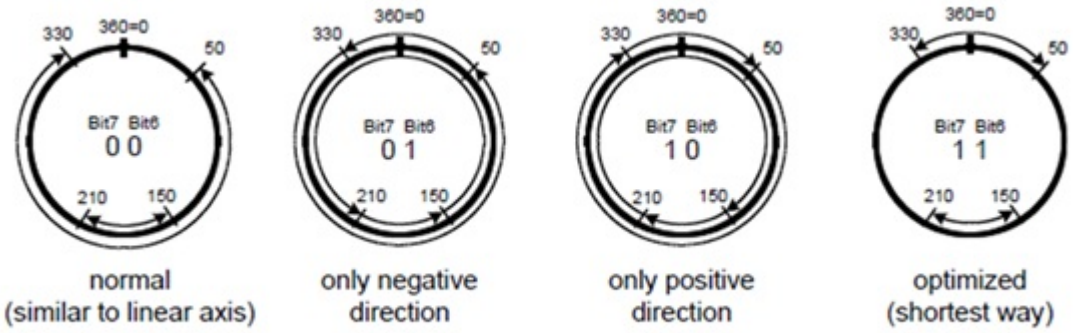
- Standard (Set of set-point) mode: up to 5 position targets can be stored using NEW_SET/SET_ACK mechanism
- Change Setpoint Immediately mode: after NEW_SET/SET_ACK the target is applied immediately
- Relative/Absolute
- Position limits (607Dh)
- Online limit change for Profile Velocity/Acceleration/Deceleration

Optional features **NOT** IMPLEMENTED:

- Change on setpoint mode: this mode is ignored and set of setpoint will be used
- Halt option code (605Dh): halt will always happen using profile deceleration
- End velocity is always zero rpm (6082h)
- Motion profile type (6086h) is not supported, profile is always speed-trapezoidal.
- The following Positioning option code (60F2h):
- Relative positioning is always referred to previous target
- Change setpoint immediately will always take place immediately
- Newset/setack will always take place as specified in section 10.2 of CiA402
- Position wrap of the range is not supported and will saturate to position limits

Modulo Implementation

- Modulo range is defined with the Position range objects (607Bh).
- Bit 6 and 7 of the Position Option Code (60F2h) regulate the behaviour of the modulo positioning.



Position Option Code 60F2h	Bit7	Bit6
Normal	0	0
Only Negative	0	1
Only Positive	1	0
Optimized (Shortest Way)	1	1

Manufacturer specific function implemented:

- using manufacturer specific bits in the control word (Bit11…15) is possible to enable manufacturer specific functions
- implemented function are triggered by values in the next table

Function	Bit15	Bit14	Bit13	Bit12	Bit11
Standard behaviour	0	0	0	0	0
Touchprobe positioning (DIN1 rising edge)	0	0	0	0	1
Touchprobe positioning (DIN1 falling edge)	0	0	0	1	0
Touchprobe modulo positioning (DIN1 rising edge)	0	0	0	1	1
Touchprobe modulo positioning (DIN1 falling edge)	0	0	1	0	0
Wait for global time	0	1	0	0	0
Wait for sync	0	1	0	0	1

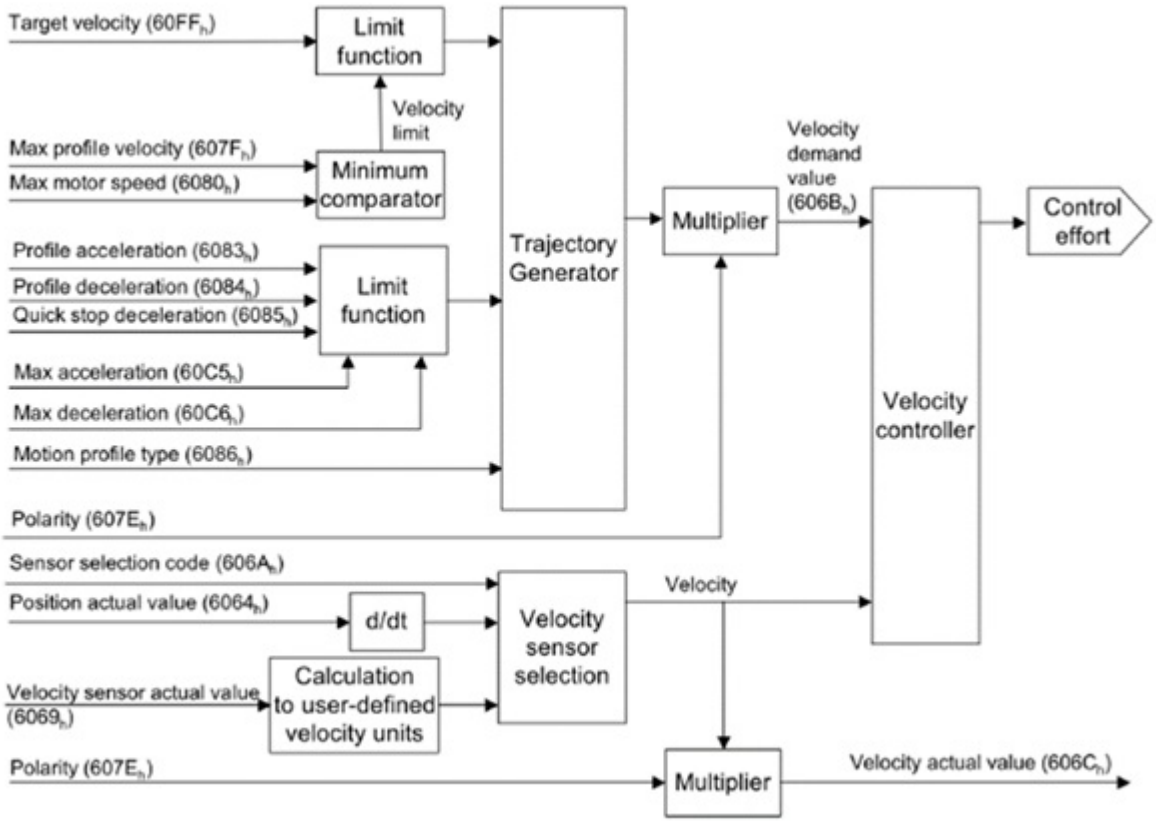
- During Touch-probe positioning the status_word is extended using manufacturer specific Bit14/Bit15 with the following meaning:

Function	Bit15	Bit14
Standard behaviour	0	0
Touchprobe positioning: armed (wait for touch_probe)	0	1
Touchprobe positioning: position sensed, wait to complete	1	1
Not allowed	*1*	*0*

See section DBS touch-probe extensions.

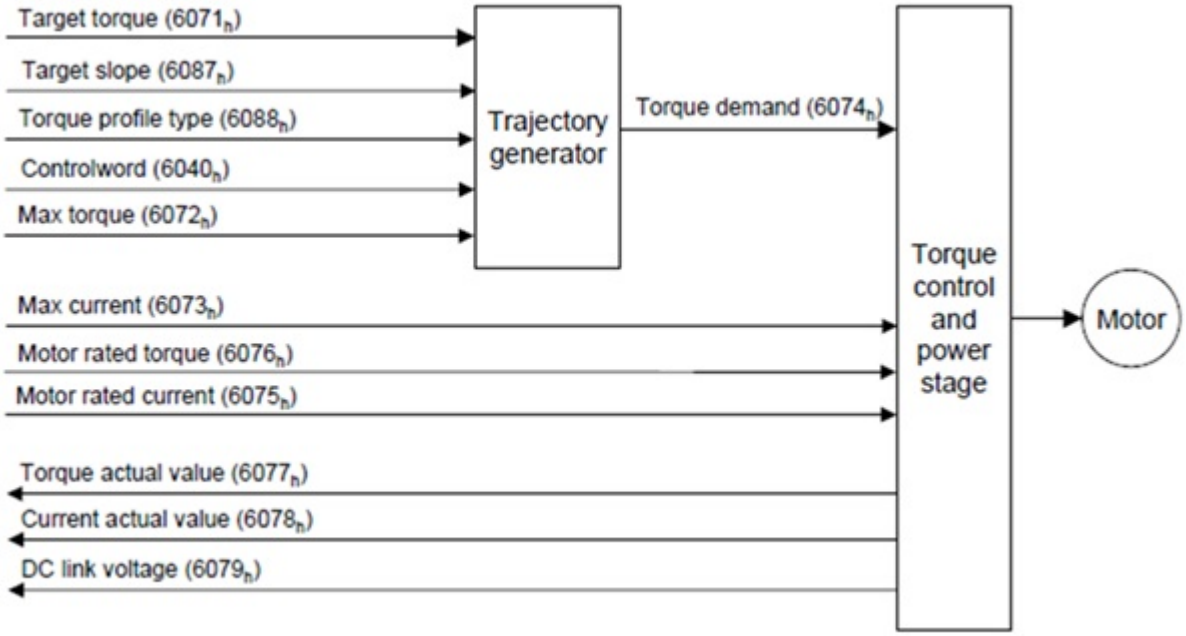
The “wait for global time” and “wait for sync” modes allow to defer the start of the profile generation on a particular event:

- Wait for global time: in fieldbus where the global time exists (e.g. Ethercat, CanopenOverAir) using object 3018h (profile position sync time) is possible to defer the profile generation until the sync time reaches a given value
- Wait for sync: in CanOpen over can-bus, is possible to defer the start of the profile until the drive detects a SYNC message on the bus <<< === Profile Velocity



Optional feature **NOT** IMPLEMENTED:

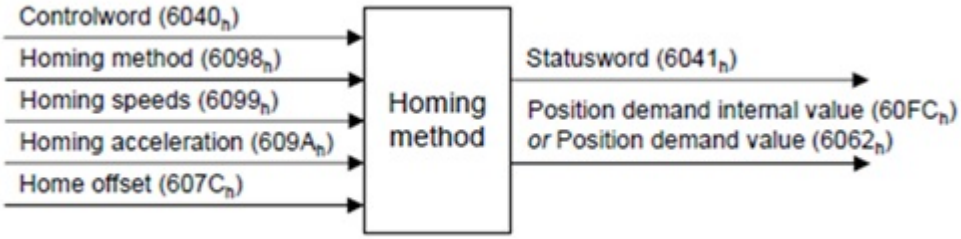
- Sensor code (6069h); the velocity is derived from the position information
- Sensor selector code (606Ah); position sensor cannot be selected
- Max slippage (60F8h) is not implemented since the max slippage control does not make sense for servodrive
- Motion profile type (6086h) is not supported, profile is always speed-trapezoidal. <<< === Profile Torque



Optional feature **NOT** IMPLEMENTED:

- Torque profile type (6088h) is not supported, profile is always torque-trapezoidal

9.4. Homing



Implemented homing types:

- Negative/positive switches (1, 2)
- Home + index (3,4,5,6)
- Home on the right + index (7,8,9,10)
- Home on the left + index (11,12,13,14)
- Index (33, 34)
- Set quota (37)

Homings are described in paragraph 11.

Optional feature **NOT** IMPLEMENTED:

- Supported homing methods object (60E3h): the supported homing methods are described in the previous section and in the servodrive manual.

9.5. Touch Probe (optional)

This functionality is implemented and uses DIN1. Touchprobe allows precise capture of the position while motor is running.

Touch-probe function 60B8h

Bit	Value	Definition
0	0	Switch off touch probe 1
	1	Enable touch probe 1
1	0	Trigger first event
	1	continous
4	0	Switch off sampling at positive edge of touch probe 1
	1	Enable sampling at positive edge of touch probe 1
5	0	Switch off sampling at negative edge of touch probe 1
	1	Enable sampling at negative edge of touch probe 1

bit 2..3 and 6..15 are reserved and should be left to zero.

Touch-probe status 60B9h

Bit	Value	Definition
0	0	Touch probe 1 is switched off
	1	Touch probe 1 is enabled
1	0	Touch probe 1 no positive edge value stored
	1	Touch probe 1 positive edge position stored
2	0	Touch probe 1 no negative edge value stored
	1	Touch probe 1 negative edge position stored

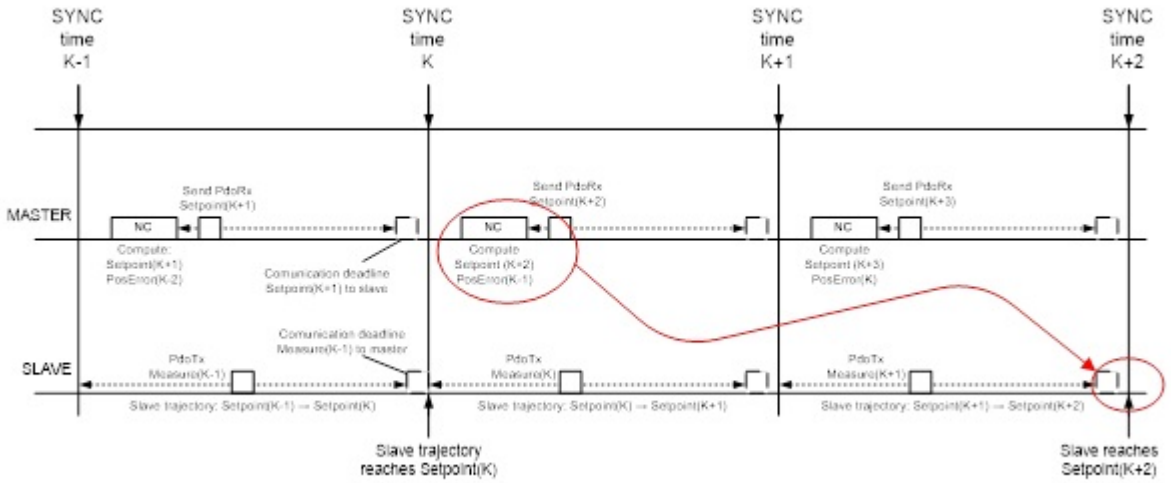
bit 3..15 are not used and will be read as zero.

Operation:

1. Touch-probe is armed via Bit0 of Function; capture can happen on first event or continuously (Bit1 of Function) and on rising or falling edge of the DIN1 input. At least one of Bit4 or Bit5 must be set in order to arm touch-probe sensing
2. Function is detected as armed via Status object Bit0
3. Whenever the first capture happens (either on rising edge or falling edge or both) Bit1 and Bit2 of Status will provide feedback to master. Objects 60BAh and 60BBh will allow master to read the probed motor position
4. If continuous operation is selected, subsequent events will update position read by master but Bit1/Bit2 of Status will retain set state

9.6. Interpolated mode (optional)

This profile allows operation with a numerical control; target buffering on the device side is NOT implemented: one value needs to be sent by the master to the device for each sync period and the device will generate a ramp that will drive the motor to that position on the end of the NEXT sync period; the next figure illustrates how the data exchange works.



Is worth noting that interpolated position mode inserts a delay of two sync periods from the master point of view: for example at time K, the numeric control will compute the position error of the previous sync period (K-1) and will compute the target that the slave will reach at time (K+2). Implemented features details:

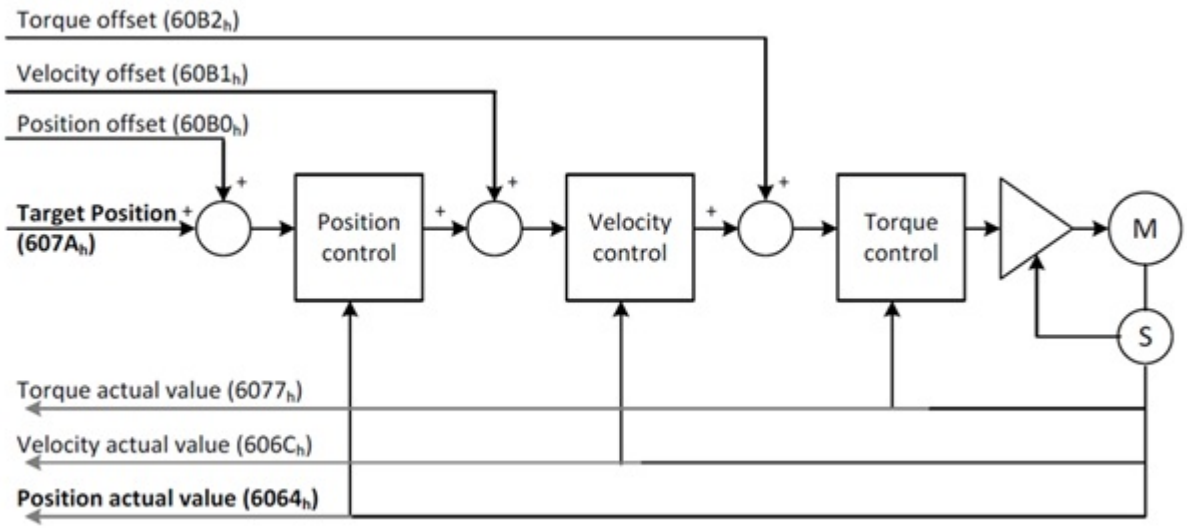
- interpolation submode selection (60C0h, type 0), only default linear interpolation can be selected
- allowed interpolation period range is 1.0ms … 100.0ms

Optional features **NOT** implemented:

- position range objects not implemented (607Bh)
- interpolation data configuration (60C4h) is not implemented since there is NO interpolation data buffer on the device; one target must be transferred from master on each synchronization period
- profile parameters don’t influence trajectory generation; the master is responsible of limiting speed/acceleration/deceleration in this mode

9.7. Cyclic Synchronous Position mode (CSP) (optional) - Only EtherCAT/Powerlink

The cyclic synchronous position mode allows interpolation in a similar way of the interpolated mode; this mode allows the master to manage more precisely the reference generation inside the servodrive. The following scheme is employed:



Using the offset objects, master can give feedforward actions in the inner loops of the servodrive allowing, for example, load torque compensation. The following optional functionalities are not supported in the CSP implementation:

- Output Cycle Counter (bit5/6 of control word) is not supported; objects 60D9h, 60DAh are not implemented
- Bit13 of the status word is the following error and not the second bit of Input Cycle Counter.

9.8. Default PDO Mapping

PDO	COB-ID	Mapping	Trasm Type
PDORX1	0200h + NODEID	6040h: control word 607Ah: target position 3012h: Max current [mA]	Enabled by default
PDORX2	0300h + NODEID	60FFh: target velocity 6071h: target torque 6060h: mode of operation	Enabled by default
PDORX3	0400h + NODEID	–	Disabled by default
PDORX4	0500h + NODEID	–	Disabled by default
PDOTX1	0180h + NODEID	6041h: status word 6064h: position actual value 3011h: current value [mA]	255 + event time 100ms Enabled by default
PDOTX2	0280h + NODEID	606Ch: velocity actual value 3010h: tension value 6061h: mode of operation display	255 + event time 100ms Enabled by default
PDOTX3	0380h + NODEID	–	Disabled by default
PDOTX4	0480h + NODEID	–	Disabled by default

9.9. EMCY codes

The following table summarizes the emergency codes emitted by the servodrive whenever an alarm is

detected.

Code	Meaning
2220h	Continuous overcurrent
2230h	Instantaneous overcurrent
3110h	Overvoltage in power supply
4210h	Overtemperature in the power stage
4310h	Overtemperature in the motor
5530h	Hardware data storage error
5531h	Software data storage error
5532h	Parameter value error
6200h	User generated alarm
9000h	Digital input generated alarm
FF04h	Motor parameters copy error
FF05h	Motor parameter copy error (speed loop)
FF15h	1ms task overtime
FF16h	10ms task overtime
FF19h	Parameter restore error
FF20h	Motor not calibrated
FF21h	No movement during encoder calibration procedure
FF22h	Wrong direction during encoder calibration procedure
FF23h	Error during autoconfiguration procedure
FF24h	Error reading encoder multiturn information
FF25h	Error reading encoder singleturn information

9.10. DBS touch-probe extension (optional)

The DBS family provide some extension to the standard touch probe mechanism.

9.10.1. Touch-probe positioning

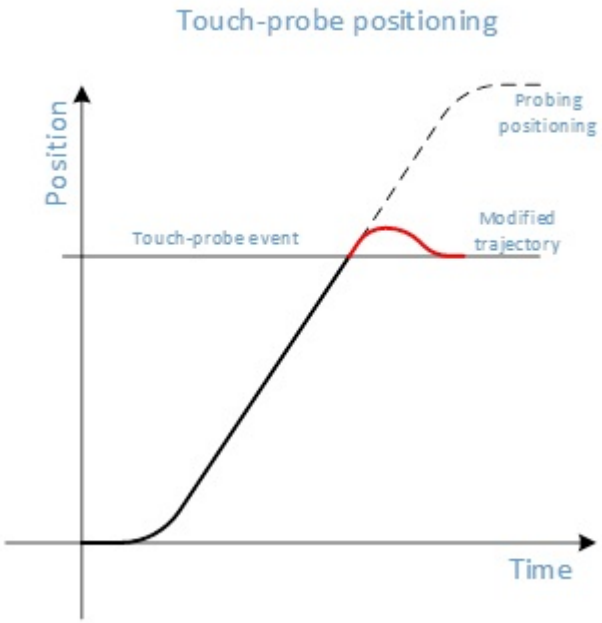
This function allows to modify on the fly the trajectory of the on-going positioning in order to reach the position related to the touch-probe event. This function is available when profile position is active.

The touch-probe positioning allows the following operation:

- motor is standstill
- master start a movement via standard “new_set” bit (control_word=001Fh)

- motor starts with defined profile position kinematic constraints (speed/acceleration)
- Master enables touch-probe positioning via Bit11..15 of the control word; for example touch-probe positioning on the rising edge of DIN1 is triggered by updating control_word=081Fh; status word Bit14 is set by motor reflecting positioning “armed”
- when the touch-probe is sensed the related position is captured (TP_Pos) and the status word Bit15 is set by motor signaling “position captured”
- motor modifies the trajectory in order to reach the target TP_Pos+TP_Offset (object 3020h)
- Motor reaches the position and stops (status word is updated as usual with Bit10 target reached).

The following diagram shows the behavior of the motor in “touch-probe positioning”:



9.10.2. Touch-probe modulo positioning

This function is similar to the previous but allows to manage positioning by means of “modulo” position. This allows for example to deal with circular axis where the same point can be reached in a given position P1, in the next modulo P1+MOD, after two modulo P1+2*MOD etc.

The typical example is where the modulo MOD is 1 motor turn; in the DBS the resolution of one turn is 4096 pulses/rev so in this example a given application point can be reached when the motor position is 0, 4096, 8192 etc; default value for object TP_Modulo 3021h is 4096.

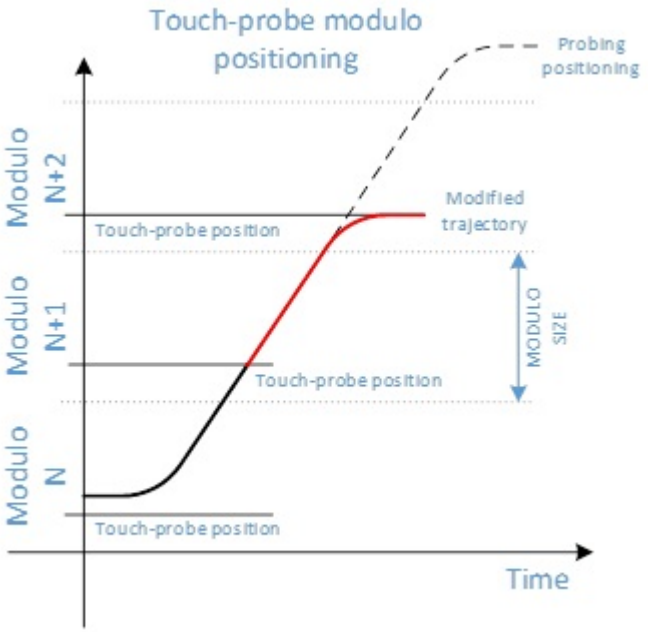
The purpose of the Touch-probe modulo positioning is reach the probed position without inverting the motion direction as soon as possible.

A typical operation is described by these steps:

- motor is standstill
- master start a movement via standard “new_set” bit (control_word=001Fh)
- motor starts with defined profile position kinematic constraints (speed/acceleration)
- master enables touch-probe positioning via Bit11..15 of the control word; for example touch-probe modulo positioning on the rising edge of DIN1 is triggered by updating control_word=181Fh; status word Bit14 is set by motor reflecting positioning “armed”

- when the touch-probe is sensed the related position is captured (TP_Pos) and the status word Bit15 is set by motor signaling “position captured”
- motor modifies the trajectory in order to reach the target TP_Pos+TP_Modulo (object 3021h) without changing direction
- motor reaches the position and stops (status word is updated as usual with Bit10 target reached).

The following diagram shows the behavior of the touch-probe modulo positioning:



Is worth noting that depending on kinematic constraints (profile velocity, acceleration, deceleration), the position that can be reached without inverting the direction could be farther than the position in the next modulo.

This example clarifies the scenario:

- modulo size: 4096
- start position: 2048
- probe position: 0 (and 4096, 8192,12288 etc.)
- the probing positioning in started in positive direction with high acceleration/speed and with low deceleration
- then motor reaches the first time the probe (position=4096);
- with given deceleration, the halt space is greater than 4096 pulses (e.g. 6000pulses); the motor modifies the target position to 12288 (2 turns after the probe)
- the motor reaches 12288 and stops.

In order to deal with this problem, when the touch probe event trigger the destination change, the DBS computes the halting space and modifies the destination in order to prevent direction change.

10. CanOpen/EtherCAT/Powerlink

10.1. Overview

Note: available only for drivers equipped with CanOpen/Modbus RTU, CanOpen over EtherCAT or Ethernet PowerLink. (---/---/C; ---/---/ETH; ---/---/EPL).

The present document describes the adherence of the Minimotor stack to the following specification:

CanOPEN

- CiA DS301 v4.02 (CanOPEN DLL)
- CiA DS402 v3.0.1.15 (servodrive profile)

CanOPEN over EtherCAT:

- ETG1000-2 v1.0.1 (physical layer)
- ETG1000-3 v1.0.1 (DLL services)
- ETG1000-4 v1.0.1 (DLL protocols)
- ETG1000-5 v1.0.1 (AL services)
- ETG1000-6 v1.0.1 (AL protocols)
- CiA DS402 v3.0.1.15 (servodrive)
- ETG6010 v1.1.0 (CiA 402 impl. directives)

Ethernet PowerLink (EPL)

- EPSG DS 301 V1.2.0 (Powerlink DLL)

The purpose of the present document is to describe what optional features are implemented by the servo-motor CanOPEN, CoE or EPL.

Applicable ---/---/--- software release: ≥v4.036

10.2. DS301 Communication Layer

This paragraph describes the details of the DS301 layer of the CanOpen interface. It applies only to the CanOPEN over can-bus implementation.

10.2.1. Physical Layer

Supported baudrate:

- 1Mbps tq=71ns SamplingPoint=11tq
- 500Kbps tq=142ns SamplingPoint=12tq

- 250Kbps tq=284ns SamplingPoint=12tq
- 125Kbps tq=571ns SamplingPoint=12tq
- 50Kbps tq=1.43us SamplingPoint=12tq
- 20Kbps tq=3.57us SamplingPoint=12tq

10.2.2. SDO protocol

- Maximum supported object size: 32bit
- Expedited transfer supported SUPPORTED
- Segmented transfer: NOT SUPPORTED
- Block transfer: NOT SUPPORTED

10.2.3. TIME protocol

The time protocol is NOT supported.

10.2.4. PDO protocol

The implementation supports up to 4 Pdo TX and 4 Pdo RX; each object can map up to 8 objects; mapping can be done only using the whole size of the object (i.e. is not possible to map 8 bit of a 32 bit object).

The following transmit type are implemented for PDO TX:

Transmission Type	Description
0	PDO is emitted when changes occur after SYNC is received
1..240	PDO emitted after SYNC, depending on SYNC message occurrence (e.g. 1 = every SYNC, 2 = every 2 SYNC, etc.)
241..251	Not supported
252	PDO emitted after RTR reception * Data sampled on SYNC * Emitted after RTR
253	PDO emitted after RTR reception * Data sampled and emitted after RTR
254	Not supported
255	PDO emitted asynchronously Two features supported for emission control: * Inhibit time (emission only when PDO content changes) * Event timer (periodical asynchronous emission)

Defaults:

- inhibit time: 100ms
- event time: 0 (periodic emission on TT=255 disabled by default)

See application profile for default PDO configuration. ==== EMCY protocol

Emergency protocol is implemented for reporting alarms from the servodrive; to control the emission of

EMCY messages, the inhibit time (object 1015h) can be modified. See the application profile for detailed list of alarms. (See paragraph 8.3)

10.2.5. NMT protocol

The NMT implementation includes standard commands for NMT state machine management. For node alive checking, the NODE GUARDING protocol is implemented; the behaviour of node guarding is set by objects 100Ch (guard time) and 100Dh (life factor).

10.2.6. HEARTBEATING

Heartbeating Protocol is supported. Register 1016h defines the time in multiples of 1ms for the Heartbeat protocol consumer. Register 1017h determines the time in multiples of 1ms for the producer of the protocol. Object 1029h determines the behaviour when the error is triggered.

10.3. Object Dictionary

Object dictionary is divided in the following sections:

Range Objects	Functions
1000h .. 1FFFh	Communication profile area
2000h .. 21FFh	Manufacturer parameter area
3000h .. 3FFFh	Manufacturer specific object area
6000h .. 67FFh	Profiled objects

Other areas are unused; the object list is common between CanOPEN, CoE and EPL implementation.

10.3.1. Communication area

The following table shows the list of communication area objects; since some objects are relevant only for CanOPEN or CoE implementation, the table shows in which fieldbus the object is relevant.

Index	Description	CANopen	CoE	EPL
1000	Device type	X	X	X
1001	Error register	X	X	X
1003	Predefined error	X	X	
1006	Cycle length			X
100C	Guard time	X		
100D	Life factor	X		
1010	Store parameters, see Saving parameters to non-volatile memory	X	X	X
1014	EMCY cob-id	X		

Index	Description	CANopen	CoE	EPL
1015	EMCY inhibit time	X	X	
1016	Consumer Heartbeat Time	X		
1017	Producer Heartbeat Time	X		
1018	Identity object	X	X	X
1020	CFM Verify Configuration			X
1029	Error Behaviour	X		
1030	NMT Interface Group 0h			X
1300	SDO Sequ Layer Timeout			X
1400	PDO RX 1 – config [1]	X	X	X
1401	PDO RX 2 – config	X		
1402	PDO RX 3 – config	X		
1403	PDO RX 4 – config	X		
1600	PDO RX 1 – mapping [1]	X	X	X
1601	PDO RX 2 – mapping	X		
1602	PDO RX 3 – mapping	X		
1603	PDO RX 4 – mapping	X		
1800	PDO TX 1 – config [1]	X	X	X
1801	PDO TX 2 – config	X		
1802	PDO TX 3 – config	X		
1803	PDO TX 4 – config	X		
1A00	PDO TX 1 – mapping [1]	X	X	X
1A01	PDO TX 2 – mapping	X		
1A02	PDO TX 3 – mapping	X		
1A03	PDO TX 4 – mapping	X		
1C00	Sync manager types		X	
1C0A	DLL_CnCollision REC			X
1C0B	DLL_CnLoss REC			X
1C0C	DLL_CnLossSoC_REC			X
1C0D	DLL_CnLossSoA REC			X
1C0E	DLL_CnSpCJitter REC			X
1C0F	DLL_CnCRCError REC			X
1C10	Sync manager 0 config		X	
1C11	Sync manager 1 config		X	

Index	Description	CANopen	CoE	EPL
1C12	Sync manager 2 config		X	
1C13	Sync manager 3 config [1]		X	
	DLL_CnSocJitterRange [1]			X
1C14	DLL_CnLossOfSocTolerance			X
1C32	Output sync parameters		X	
1C33	Input sync parameters		X	
1E40	NWL_IpAddrTable_0h_REC			X
1E4A	NWL_IpGroup_REC			X
1F82	NMT_FeatureFlags_U32			X
1F83	NMT_EPLVersion_U8			X
1F8C	NMT_CurrNMTState_U8			X
1F93	NMT_EPLNodeID_REC			X
1F98	NMT_CycleTiming_REC			X
1F99	NMT_CNBasicEthernetTimeout_U32			X
1F9A	NMT_HostName_VSTR			X
1F9E	NMT_ResetCmd_U8			X

Saving parameters to non-volatile memory

Writing an object in the 2000h … 21FFh range only alters the RAM image of the parameter: at the next power-up the drive reloads the values stored previously.

To make a change permanent, perform a 32 bit write of the value `65766173h` (the ASCII string "SAVE") to the *Store parameters* object `1010h.1`.

10.3.2. Manufacturer parameter area

The object range 2000h … 21FFh allows to access servodrive parameters; the following table summarize the details of a parameter object:

Object details

Campo	Valore
Subidx	0
Object type	16bit, signed integer
Access	Read/Write
PDO mappable	NO

Object dictionary for manufacturer parameters

Index	Description
2000h	Parameter 0
2001h	Parameter 1
…	…
21FFh	Parameter 511

For details of the parameter meaning and encoding refer to the servodrive manual. IMPORTANT: Writing objects in the 2000h … 21FFh range will alter only the RAM image of the parameter; to actually save a parameter to non-volatile storage a 32 bit write of the value `65766173h` ("SAVE") to object `1010h.1` needs to be performed, as described in Saving parameters to non-volatile memory.

10.3.3. Manufacturer Specific Area

This area contains several object that allows to read/write manufacturer specific data.

Index	Access	Data type	PDO mappable	Description
3000h	RO	INT16	YES	Ain0 value (-32768 to 32768)
3001h	RO	UINT16	YES	Digital input bits 0–4
3002h	RW	UINT16	YES	Digital output bit0
3003h	RO	INT16	YES	Heatsink temperature (x0.1 °C)
3005h	RO	INT16	YES	Motor temperature (x0.1 °C)
3008h	RO	UINT16	YES	Actual alarm code (not EMCY)
3009h	RO	UINT	NO	Events (0–16, see section 6.1)
3010h	RO	UINT16	YES	Power stage tension (x0.1 V)
3011h	RO	INT16	YES	Motor IQ current (mA)
3012h	RW	UINT16	YES	Motor IQ limit (mA)
3014h	RO	INT16	YES	Output power (1W/lsb)
3020h	RW	INT32	YES	Touchprobe positioning offset
3021h	RW	UINT32	YES	Touchprobe positioning modulo
3030h	RO	UINT16	YES	Acceleration RMS modulus (milli-G)
3031h	RO	UINT16	YES	Acceleration RMS x (milli-G)
3032h	RO	UINT16	YES	Acceleration RMS y (milli-G)
3033h	RO	UINT16	YES	Acceleration RMS z (milli-G)
3034h	RO	UINT16	YES	Acceleration DC x (milli-G)
3035h	RO	UINT16	YES	Acceleration DC y (milli-G)
3036h	RO	UINT16	YES	Acceleration DC z (milli-G)

Index	Access	Data type	PDO mappable	Description
3037h	RO	INT16	YES	Instantaneous Acceleration x (milli-G)
3038h	RO	INT16	YES	Instantaneous Acceleration y (milli-G)
3039h	RO	INT16	YES	Instantaneous Acceleration z (milli-G)

10.3.4. Profiled Objects

The range 6000h … 67FFh of objects are defined as specified by CiA402; refer to the application area for further information.

Index	Access	Type	PDO mappable	Description
6040h	RW	U16	YES	Control word
6041h	RO	U16	YES	Status word
605Ah	RW	U16		Quickstop option code
605Bh	RW	U16		Shutdown option code
605Ch	RW	U16		Disable operation option code
605Dh	RW	U16		Halt option code
605Eh	RW	U16		Fault reaction option code
6060h	RW	S8	YES	Mode of operation
6061h	RO	S8	YES	Mode of operation display
6062h	RO	S32	YES	Position demand value
6064h	RO	S32	YES	Position actual value
6065h	RW	U32		Following error window
6066h	RW	U16		Following error time
6067h	RW	U32		Position window
6068h	RW	U16		Position window time
606Bh	RO	S32	YES	Velocity demand
606Ch	RO	S32	YES	Velocity actual value
606Dh	RW	U16		Velocity window
606Eh	RW	U16		Velocity window time
606Fh	RW	U16		Velocity threshold
6070h	RW	U16		Velocity threshold time
6071h	RW	S16	YES	Target torque
6072h	RW	U16	YES	Max torque
6073h	RW	U16	YES	Max current

Index	Access	Type	PDO mappable	Description
6074h	RO	S16	YES	Torque demand value
6075h	RO	U32	YES	Motor rated current
6076h	RO	U32	YES	Motor rated torque
6077h	RO	S16	YES	Torque actual value
6078h	RO	S16	YES	Current actual value
6079h	RO	U32	YES	DC link voltage
607Ah	RW	S32	YES	Target position
607Bh.0	RO	U8	YES	Position Range Limit – number of subindex
607Bh.1	RW	S32	YES	Negative limit
607Bh.2	RW	S32	YES	Positive limit
607Ch	RW	S32	YES	Home offset
607Dh.0	RO	U8	YES	Software position limit – number of subindex
607Dh.1	RW	S32	YES	Negative limit
607Dh.2	RW	S32	YES	Positive limit
607Eh	RW	U8		Polarity
607Fh	RO	U32	YES	Max speed
6080h	RO	U32	YES	Max speed (duplicate index)
6081h	RW	U32	YES	Profile velocity
6083h	RW	U32	YES	Profile acceleration
6084h	RW	U32	YES	Profile deceleration
6085h	RW	U32	YES	Quick stop deceleration
6087h	RW	U32	YES	Torque slope
6098h	RW	S8	YES	Homing method
6099h.0	RO	U8	YES	Homing speed – number of subindex
6099h.1	RW	U32	YES	Switch speed
6099h.2	RW	U32	YES	Index speed
609Ah	RW	U32	YES	Homing acceleration
60B0h	RW	S32	YES	Position offset
60B1h	RW	S32	YES	Velocity offset
60B2h	RW	S16	YES	Torque offset
60B8h	RW	U16	YES	Touch-probe function
60B9h	RW	U16	YES	Touch-probe status

Index	Access	Type	PDO mappable	Description
60BAh	RW	S32	YES	Touch-probe positive edge latched position
60BBh	RW	S32	YES	Touch-probe negative edge latched position
60C0h	RO	S16	NO	IpData submode selection
60C1h.0	RO	U8	YES	IpDataRecord – number of subindex
60C1h.1	RW	S32	YES	IpData record
60C2h.0	RO	U8		Interpolation time period – number of subindex
60C2h.1	RW	S8		Ip time units
60C2h.2	RW	S8		Ip time index
60F2h	RW	U16	YES	Position Option Code
60F4h	RO	U32	YES	Following Error
60FDh	RO	U32	YES	Digital inputs
60FEh.0	RO	U8	YES	Digital outputs – number of subindex
60FEh.1	RW	U32	YES	Physical outputs
60FEh.2	RW	U32	YES	Output mask
60FFh	RW	S32	YES	Target velocity
6502h	RO	U32	NO	Supported mode of operation

[1] PDO config and mapping have different encoding for CoE/CanOPEN and EPL. Please check DLL manual for details.

11. Ethernet IP

Note: available only for drivers equipped with Ethernet IP optional board (---/---/EIP).

This document outlines the conformance of the Ethernet/IP and CIP stack implemented in the servodrive; the following documents apply unless otherwise stated:

- CIP Network library – Volume 1 (edition 3.3, november 2007)
- CIP Network library – Volume 2; Ethernet/IP adaptation of CIP (edition 1.4, november 2007)
- CiA DS402 v3.0.1.15

Furthermore the CiA DS402 adaptation to CIP are described in the relevant section.

11.1. CIP Objects

11.1.1. Overview

The servodrive implements the following objects of the CIP object model:

- CIP common objects
  - Identity object – 01h
  - Assembly object – 04h
  - TCP object – F5h
  - Ethernet object – F6h;
- Manufacturer specific objects
  - COM object – 0x64
  - PAR object – 0x65
  - MISC object – 0x66
  - CIA402 objects – 0x67

Furthermore the following ASSEMBLIES contain process data:

- O>>T: Assembly 0x64

- T>>O: Assembly 0x65
- CFG: Assembly 0x66

11.1.2. COM object - 0x64

Instance	Attribute ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT		RO
0x1	0x01	Software edition	UINT		RO
	0x02	Software CRC	UDINT		RO
	0x10	Save parameters	UDINT		RW

Saving parameters to non-volatile memory

Writing an attribute of the PAR object only alters the RAM image of the parameter: at the next power-up the drive reloads the values stored previously.

To make a change permanent, perform a 32 bit write of the value `65766173h` (the ASCII string "SAVE") to the *Save parameters* attribute of the COM object: class `0x64`, instance `0x1`, attribute `0x10`.

11.1.3. COM object - 0x65

Instance	Attribute ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT		RO
0x1	0x00–0xFF	P000–P255	UINT		RW
0x2	0x00–0xFF	P256–P511	UINT		RW

Special parameters All 32 Bit parameter are divided in two 16 bits parameters of continguous numbering for the most significant and least significant bits.

These parameters are:

- P170 - Ip Addr
- P172 - Ip Mask
- P174 - Ip Gateway
- P120-135 - Multipos Ref 1 - 8

11.1.4. Misc objects - 0x66

Instance	Attribute ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT		RO

Instance	Attribute ID	Name	Data Type	Unit meas.	Access
0x30 (48)	0x00	Analog input 0	UINT		RO
	0x01	Digital inputs	UINT		RO
	0x02	Digital outputs	UINT		RW
	0x03	Heatsink temperature	UINT	0.1 °C	RO
	0x05	Board temperature	UINT	0.1 °C	RO
	0x08	Actual Alarm	UINT	N/A	RO
	0x10	Dclink voltage	UINT	0.1 V	RO
	0x11	Current actual value	UINT	mA	RO
	0x12	Current limit for PP/PV/PT	UINT	mA	RO

11.1.5. CiA 402 Objects - 0x67

Instance	Attr. ID	Name	Data Type	Unit meas.	Access
0x0	0x01	Revision	UINT		RO
0x60 (96)	0x40	Control word	UINT		RW
	0x41	Status word	UINT		RO
	0x5A	Quick-stop option code	INT		RW
	0x5C	Disable operation option code	INT		RW
	0x5E	Fault reaction code	INT		RW
	0x60	Mode of operation	SINT		RW
	0x64	Position actual value	DINT	pulses	RO
	0x65	Following error window	DINT	pulses	RW
	0x66	Following error time	INT	ms	RW
	0x67	Position window	DINT	pulses	RW
	0x68	Position window time	INT	ms	RW
	0x6C	Velocity actual value	DINT	rpm	RO
	0x6D	Velocity window	DINT	rpm	RW
	0x6E	Velocity window time	INT	ms	RW
	0x6F	Velocity threshold	DINT	rpm	RW
	0x70	Velocity threshold time	INT	ms	RW
	0x71	Target torque	INT	thousand-th of rated	RW
	0x72	Max torque	INT	thousand-th of rated	RW

Instance	Attr. ID	Name	Data Type	Unit meas.	Access
	0x73	Max current	INT	thousand-th of rated	RW
	0x75	Motor rated current	DINT	milliA	RO
	0x76	Motor rated torque	DINT	milliNm	RO
	0x77	Torque actual value	INT	thousand-th of rated	RO
	0x78	Current actual value	INT	thousand-th of rated	RO
	0x7A	Target position	DINT	pulses	RW
	0x7B	Position Range index1: min limit index2: max limit	ARRAY[1..2] OF DINT	pulses	RW
	0x7C	Home offset	DINT	pulses	RW
	0x7D	Position limits index1: min limit index2: max limit	ARRAY[1..2] OF DINT	pulses	RW
	0x7E	Polarity	USINT		RW
	0x80	Max motor speed	DINT	rpm	RO
	0x81	Profile velocity	DINT	rpm	RW
	0x83	Profile acceleration	DINT	rpm/s	RW
	0x84	Profile deceleration	DINT	rpm/s	RW
	0x85	Quick stop deceleration	DINT	rpm/s	RW
	0x87	Torque slope	INT	thousandth of rated / s	RW
	0x98	Homing method	SINT		RW
	0x99	Homing speed index1: switch speed index2: index speed	ARRAY[1..2] OF DINT	rpm	RW
	0x9A	Homing acceleration	DINT	rpm/s	RW
	0xF2	Profile Option Code	INT		RW
	0xFF	Target velocity	DINT	rpm	RW

11.2. Process Data

_Valid from Firmware ≥ v4.036 _ There are four possible configurations of the Ethernet IP process data.

- **Default (DFL)**: the standard process data implementing the profile velocity/torque/position/homing
- **A**: Standard process data with the possibility to read Digital and Analog inputs and command via fieldbus the Digital Out. The latter needs the parameter P015 on 10-Fieldbus to receive commands.
- **B**: A process data with the CiA 402 Touch probe functionality
- **C**: A process data with our manufacturer custom Touch Probe implementation

- **D**: A process data with acceleration three axis rms values

You must select the corresponding device description and set the parameter P176 to the desired value.

11.2.1. Layout Default (DFL)

The configuration for process data is the following:

Assembly 0x64 (Originator to Target, DBS/FC inputs - PLC outputs)
Assembly Length : 50 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see DSP402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota [1]

Assembly 0x65 (Target to Originator, DBS/FC outputs - PLC inputs)
Assembly Length : 20 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual

11.2.2. Layout A

The configuration for process data is the following:

Assembly 0x64 (Originator to Target, DBS/FC inputs - PLC outputs)
Assembly Length : 50 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see DSP402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota [1]

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used [2]

Assembly 0x65 (Target to Originator, DBS/FC outputs - PLC inputs)
Assembly Length : 30 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	0	-32768 … +32767
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

11.2.3. Layout B

The configuration for process data is the following:

Assembly 0x64 (Originator to Target, DBS/FC inputs - PLC outputs)
Assembly Length : 50 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see DSP402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	× 0.1% Trq

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota [1]
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used [2]
0x2C	U16	Touch Probe Function	0x67	0x60	0xB8	0	[1]

Assembly 0x65 (Target to Originator, DBS/FC outputs - PLC inputs)
Assembly Length : 40 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	0	-32768 … +32767

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	Touch Probe Status	0x67	0x60	0xB9	0	[1]
0x18	S32	Touch Probe Rising Edge position	0x67	0x60	0xBA	0	position units
0x1C	S32	Touch Probe Falling Edge position	0x67	0x60	0xBB	0	position units

11.2.4. Layout C

The configuration for process data is the following:

Assembly 0x64 (Originator to Target, DBS/FC inputs - PLC outputs)
Assembly Length : 60 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see DSP402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota [1]
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used [2]
0x2C	S32	Touch Probe positioning offset	0x67	0x30	0x20	0	Position Units
0x2C	S32	Touch Probe positioning modulo	0x67	0x30	0x21	0	Position Units

Assembly 0x65 (Target to Originator, DBS/FC outputs - PLC inputs)
Assembly Length : 30 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	0	-32768 … +32767
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

11.2.5. Layout D

The configuration for process data is the following:

Assembly 0x64 (Originator to Target, DBS/FC inputs - PLC outputs)
Assembly Length : 50 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	ControlWord	0x67	0x60	0x40	0000h	see DSP402
0x02	S32	TargetPosition	0x67	0x60	0x7A	+0	position units
0x06	U16	TorqueLimit	0x67	0x60	0x72	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x67	0x60	0xFF	+0	rpm
0x0C	S16	TargetTorque	0x67	0x60	0x71	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]
0x0F	U8	Padding Byte					
0x10	U32	ProfileVelocity	0x67	0x60	0x81	+3000	rpm
0x14	U32	ProfileAccel	0x67	0x60	0x83	+3000	rpm/s
0x18	U32	ProfileDecel	0x67	0x60	0x84	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x67	0x60	0x7D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x67	0x60	0x7D.2	+2147483647	position units
0x24	S32	HomingOffset	0x67	0x60	0x7C	+0	position units
0x28	S8	HomingMethod	0x67	0x60	0x98	37	SetQuota [1]
0x29	U8	Padding Byte					
0x2A	U16	Digital output	0x67	0x30	0x02	0	bit0: DOUT1 bit1..15: not used [2]

Assembly 0x65 (Target to Originator, DBS/FC outputs - PLC inputs)
Assembly Length : 30 Bytes

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x00	U16	StatusWord	0x67	0x60	0x41	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x67	0x60	0x64	+0	position units
0x06	S16	TorqueActualValue	0x67	0x60	0x77	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x67	0x60	0x6C	+0	rpm
0x0C	U16	VDclink	0x67	0x30	0x10	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x67	0x60	0x60	1	Profile [1]

Offset	Type	Attribute	Class	Instance	Attr. ID	Default	Meaning
0x0F	U8	Padding Byte					
0x10	U16	ActualAlarm	0x67	0x30	0x08	0	see DBS Manual
0x12	S16	Analog Input	0x67	0x30	0x00	0	-32768 … +32767
0x14	U16	Digital Input	0x67	0x30	0x01	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	Acceleration rms	0x67	0x30	0x31		
0x19	U16	Acceleration rms	0x67	0x30	0x32		
0x1A	U16	Acceleration rms	0x67	0x30	0x33		

[1] see DSP402

[2] P015=10 required for fieldbus operation

12. ProfiNET

Note: available only for drivers equipped with Profinet optional board (---/---/EPN).

This document outlines the details of the Profinet implementation for Minimotor DBS55 integrated servomotor. By enabling Profinet/DSP402 “reference mode”, the servodrive is commanded by means of CanOpen DSP402 state machine and profiles.

For further information, please refer to:

- DBS55 user manual: information about parametrization through USB and PC interface
- Dsp402 v3: information about device state machine, control word/status word encoding and object dictionary.

12.1. Device identity

- Vendor_ID: 0x0419
- Device_ID: 0x0001

12.2. Process Data

There are four possible configurations of the Profinet process data.

- **Default(DFL)**: the standard process data implementing the profile velocity/torque/position/homing
- **A**: Standard process data with the possibility to read Digital and Analog inputs and command via fieldbus the Digital Out. Parameter P015 must be on 10-Fieldbus.
- **B**: A process data with the CiA 402 Touch probe functionality
- **C**: A process data with our manufacturer custom Touch Probe implementation
- **D**: A process data with rms acceleration values of the three axis

You must select the corresponding device description and set the P176 parameter to the desired value.

12.2.1. Layout Default (DFL)

The configuration for process data is the following:

PLC Outputs (42 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota [1]
0x29	U8	Padding Byte			

PLC Inputs (18 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual

12.2.2. Layout A

The configuration for process data is the following:

PLC outputs (44 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota [1]
0x29	U8	Padding Byte			
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used [2]

PLC inputs (22 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 … +32767
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

12.2.3. Layout B

The configuration for process data is the following:

PLC outputs (46 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota [1]
0x29	U8	Padding Byte			
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used [2]
0x2C	U16	Touch Probe Function	0x60B8	0	[1]

PLC inputs (32 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 … +32767
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	Touch Probe Status	0x60B9	0	[1]
0x18	S32	Touch Probe Rising Edge position	0x60BA	0	position units
0x1C	S32	Touch Probe Falling Edge position	0x60BB	0	position units

12.2.4. Layout C

The configuration for process data is the following:

PLC outputs (52 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota [1]
0x29	U8	Padding Byte			
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used [2]
0x2C	S32	Touch Probe positioning offset	0x3020	0	Position Units
0x2C	S32	Touch Probe positioning modulo	0x3021	0	Position Units

PLC inputs (22 bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 … +32767
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used

12.2.5. Layout D

The configuration for process data is the following:

PLC outputs (44 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	ControlWord	0x6040	0000h	see DSP402
0x02	S32	TargetPosition	0x607A	+0	position units
0x06	U16	TorqueLimit	0x6072	1000	× 0.1% Trq
0x08	S32	TargetVelocity	0x60FF	+0	rpm
0x0C	S16	TargetTorque	0x6071	+0	× 0.1% Trq
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U32	ProfileVelocity	0x6081	+3000	rpm
0x14	U32	ProfileAccel	0x6083	+3000	rpm/s
0x18	U32	ProfileDecel	0x6084	+3000	rpm/s
0x1C	S32	PositionLimit.Neg	0x607D.1	-2147483648	position units
0x20	S32	PositionLimit.Pos	0x607D.2	+2147483647	position units
0x24	S32	HomingOffset	0x607C	+0	position units
0x28	S8	HomingMethod	0x6098	37	SetQuota [1]
0x29	U8	Padding Byte			
0x2A	U16	Digital output	0x3002	0	bit0: DOUT1 bit1..15: not used [2]

PLC inputs (28 Bytes)

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x00	U16	StatusWord	0x6041	0250h	Motor Status [1]
0x02	S32	PositionActualValue	0x6064	+0	position units
0x06	S16	TorqueActualValue	0x6077	+0	× 0.1% Trq
0x08	S32	VelocityActualValue	0x606C	+0	rpm
0x0C	U16	VDclink	0x3010	+0	× 0.1 V
0x0E	U8	ModeOfOperation	0x6060	1	Profile [1]
0x0F	U8	Padding Byte			
0x10	U16	ActualAlarm	0x3008	0	see DBS Manual
0x12	S16	Analog Input	0x3000	0	-32768 … +32767

Offset	Type	Attribute	CIA402 Object	Default	Meaning
0x14	U16	Digital Input	0x3001	0	Bit0: DIN1 Bit1: DIN2 Bit2: DIN3 Bit3: DIN4 Bit4: DIN5 Bit5..15: not used
0x16	U16	Acceleration rms x	0x3031		
0x19	U16	Acceleration rms y	0x3032		
0x1A	U16	Acceleration rms z	0x3033		

12.3. Saving parameters

Saving parameters to non-volatile memory

An acyclic write to a parameter only alters its RAM image: at the next power-up the drive reloads the values stored previously.

To make a change permanent, perform a 32 bit write of the value `65766173h` (the ASCII string "SAVE") to the *Store parameters* index `1010h`, whose subindex is not addressable.

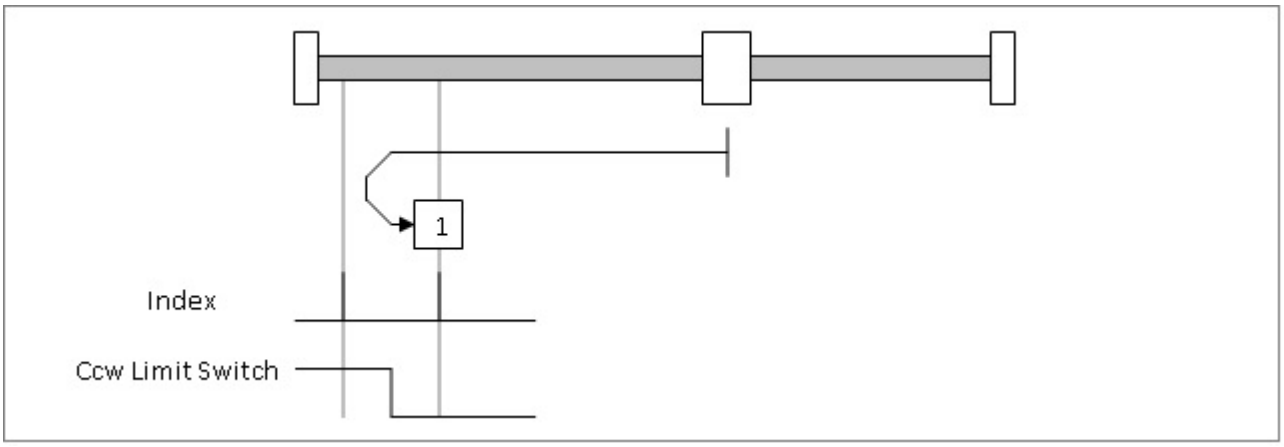
[1] see DSP402

[2] P015=10 required for fieldbus operation

13. Homing Types

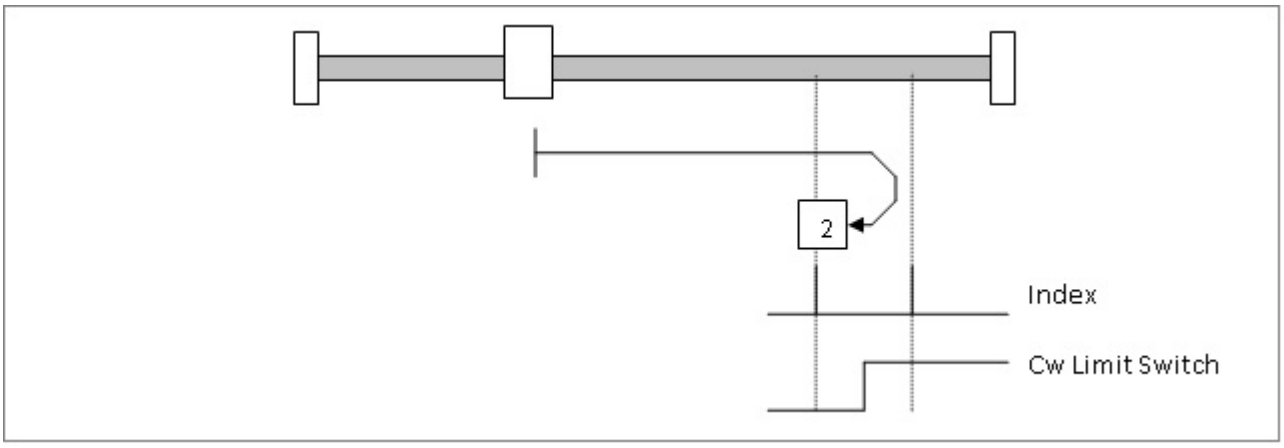
13.1. Type 1: Homing on anticlockwise limit switch and index pulse (resolver zero).

The initial direction of movement is anticlockwise towards the (anticlockwise) limit switch if this is inactive. The reference (home) position is on the first index pulse (resolver zero) to the right of the anticlockwise limit switch, when the value on the latter switches to low.



13.2. Type 2: Homing on clockwise limit switch and index pulse (resolver zero)

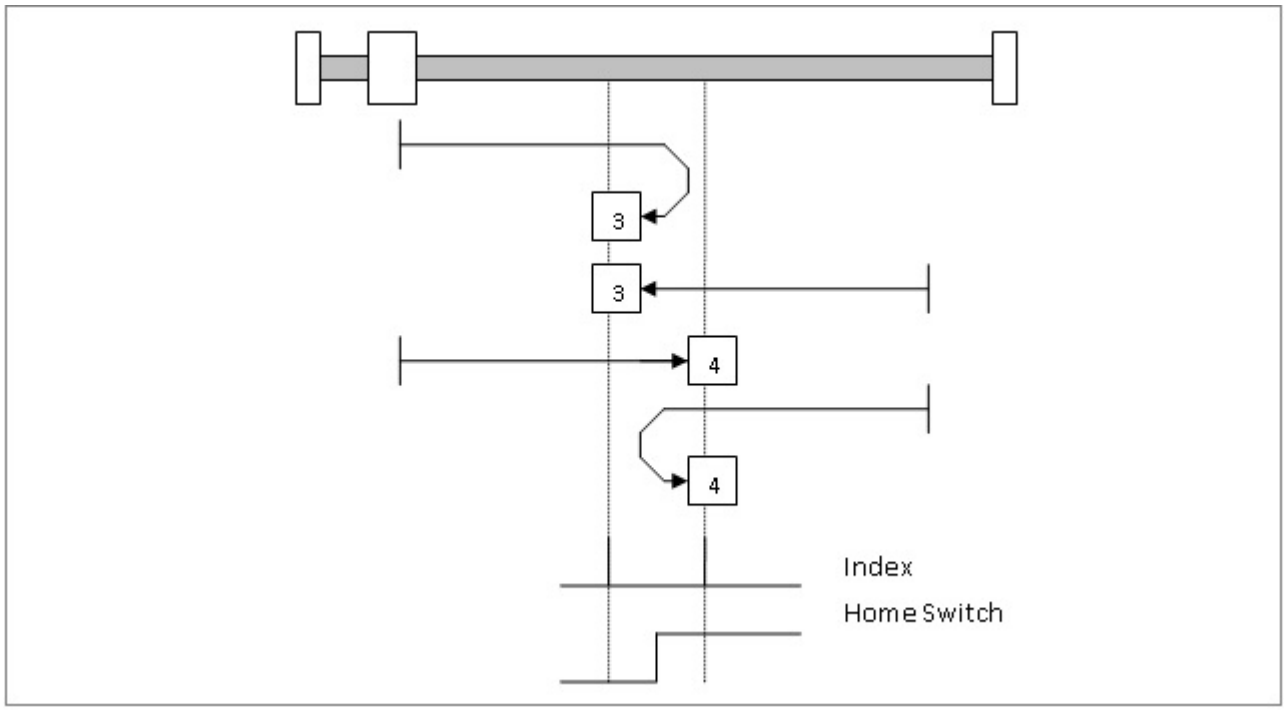
The initial direction of movement is clockwise towards the (clockwise) limit switch if this is inactive. The reference (home) position is on the first index pulse (resolver zero) to the left of the clockwise limit switch, when the value on the latter switches to low



13.3. Type 3 and 4: homing on the positive Home Switch and Index Pulse (resolver zero)

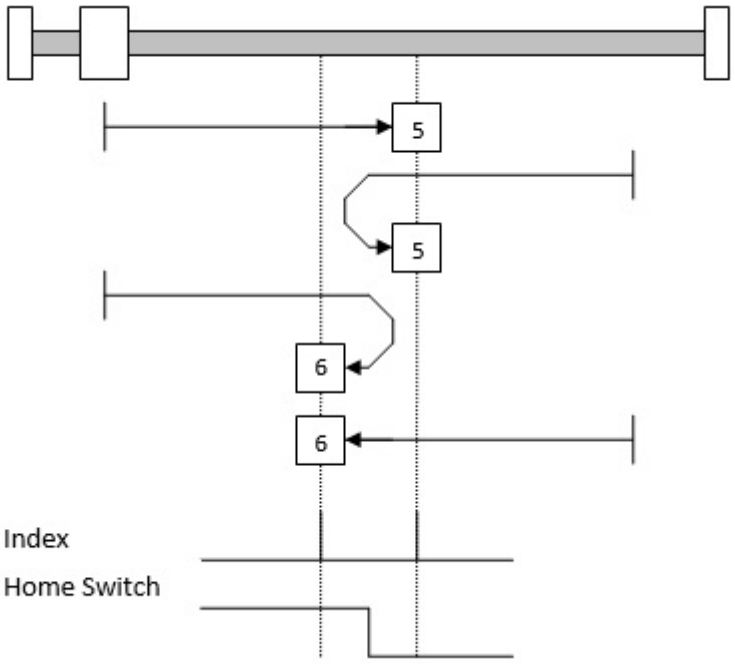
The initial direction of movement depends on the state of the home switch. The reference (home) position is on the index pulse (resolver zero) to the left (type 4) or right (type 3) of the switching point of the home switch. If the starting position is such that a reversal of direction is required, the reversal takes place after the

state of the home switch changes.



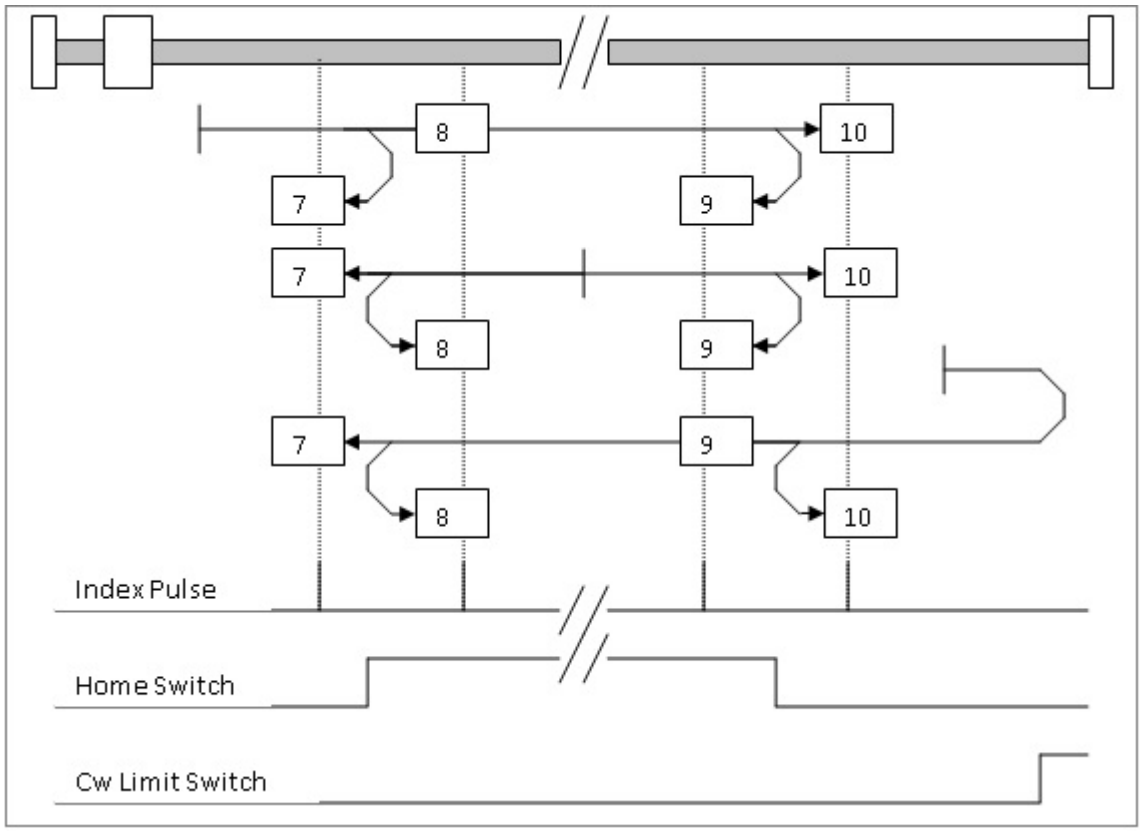
13.4. Type 5 and 6: homing on the negative Home Switch and Index Pulse (resolver zero)

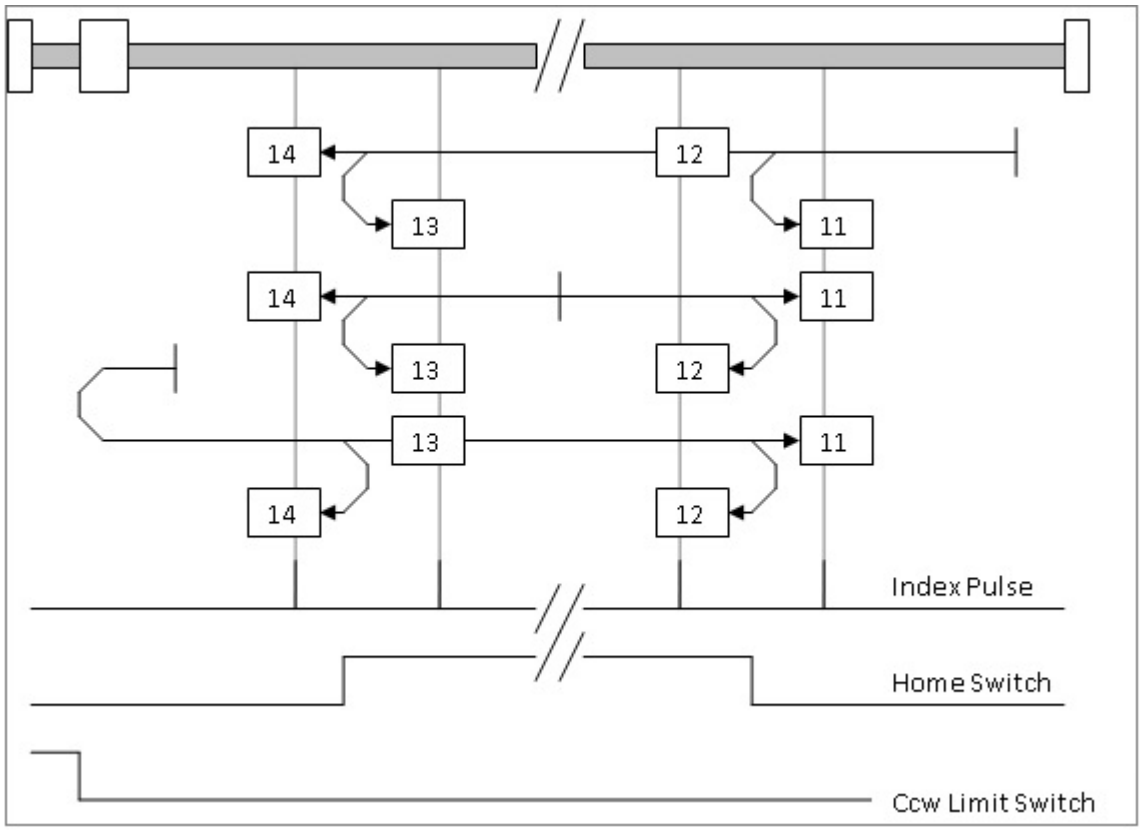
The initial direction of movement depends on the state of the home switch. The reference (home) position is on the index pulse (resolver zero) to the left (type 6) or right (type 5) of the switching point of the home switch. If the starting position is such that a reversal of direction is required, the reversal takes place after the state of the home switch changes



13.5. Type 7, 8, 9, 10, 11, 12, 13, 14: homing on the Home Switch and Index Pulse (resolver zero)

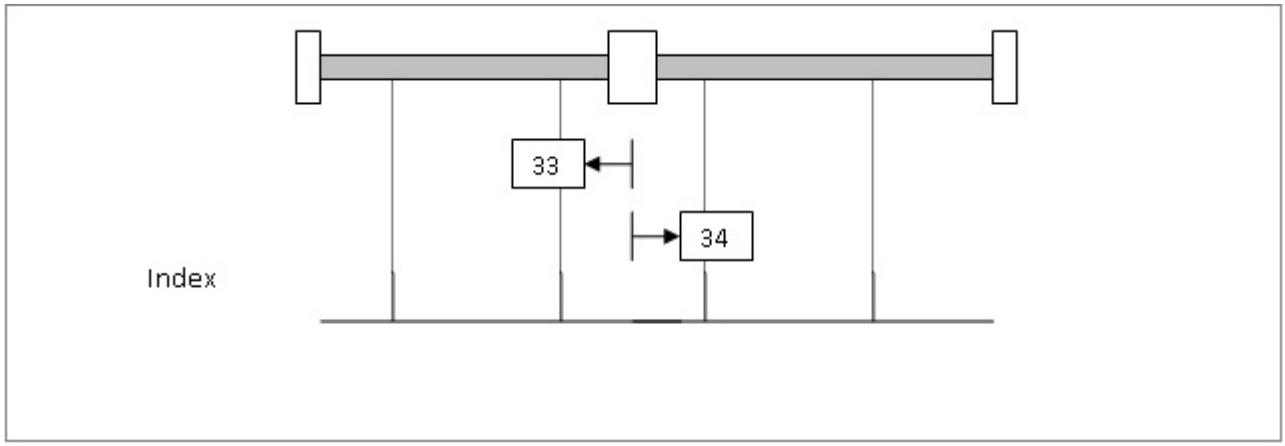
The initial direction of movement is clockwise for types 7 to 10 and anticlockwise for types 11 to 14, except when the Home Switch is high at the beginning of movement. These types of homing procedures use the home switch, which is at a high logic level only for a portion of the excursion. In this case, the initial direction of movement depends on the desired change in state of the home switch. The reference (home) position is on the index pulse (resolver zero) to the left or right of the ascending or descending change in state of the home switch. If the initial direction does not meet the home switch, the direction is reversed toward the limit switch.





13.6. Types 33 and 34: Homing on Index Pulse (resolver zero).

With Homing type 33, initial motion direction is counter clockwise; vice versa, with type 34 it is clockwise. The reference position (Home Position) is on the first Index Pulse (resolver zero) found in the direction selected.



13.7. Type 37: homing in current position.

14. Warranty Conditions

We recommend reading this document closely and ensuring you have understood it before purchasing the product. Contact the manufacturer directly with any queries.

The manufacturer guarantees its products from defects in materials and/or workmanship for a period of **twelve (12) months** (unless otherwise specified in contractual documentation) from the date of delivery. The warranty is limited to repair of the product, or replacement, at the manufacturer's discretion.

The warranty does not cover damage, malfunctions, losses or requests for compensation due to:

- Operational errors in use and/or installation
- Modifications performed by the purchaser
- Unauthorised repairs
- Dropping of the device
- Natural disasters (fire, flood, lightning strikes etc.)
- Incorrect storage and/or maintenance

The burden of proof of any defect (and for any request for assistance in the place of use) lies with the purchaser.

15. Disclaimer

The manufacturer shall not be liable:

- For determining the suitability of the product to meet the customer's needs; this shall remain the responsibility of the customer themselves
- For the use of this product as a safety device for machinery which represents a hazard to persons or property
- For conformity of the product with legislation, standards and regulations applicable to the collection of products required for the purchaser's application. The purchaser shall therefore be responsible for guaranteeing the conformity of their machinery to such standards.

Furthermore:

- The product performance specified in this document does not represent a guarantee but merely a reference for the choice of the most appropriate solution for the purchaser's requirements, as it is the result of the manufacturer's testing conditions
- The product may be subject to modifications in order to make improvements, or for other reasons. Contact the manufacturer to confirm these.
- The information contained in the following document has no contractual validity and may contain omissions, typographical and/or spelling errors, and may therefore be subject to modification and/or updating without notice.

Appendix A: Gains tuning

The drive features a PIV control with two nested loops, position and speed, whose gains are accessible to the user. The default values cover most applications: only act on them if the behaviour of the system is not satisfactory.

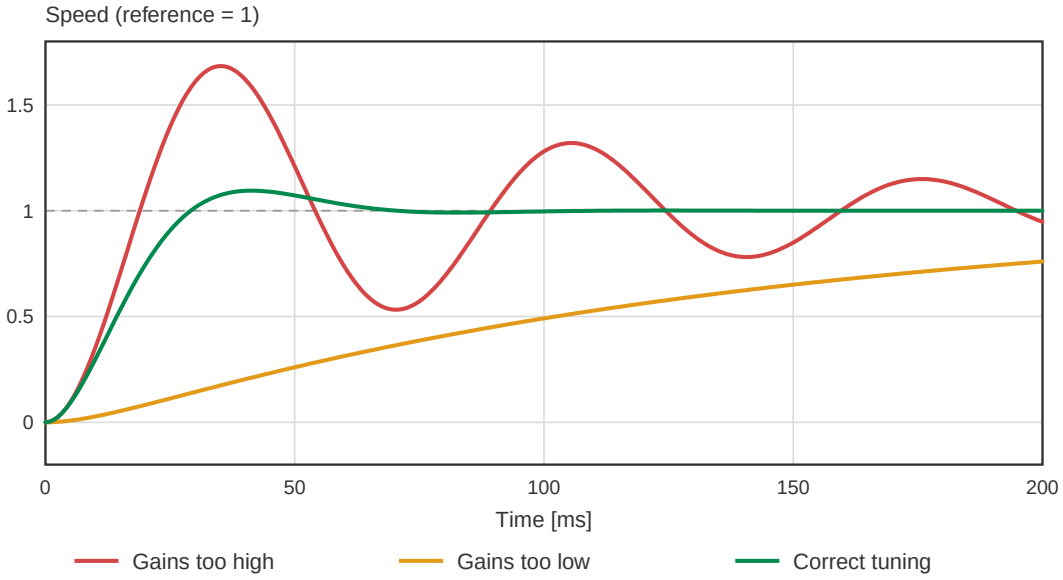


Figure 12. Response to a speed step with three different tunings

Tuning that is too aggressive leads to oscillation, with consequences ranging from unwanted behaviour to actual danger for the machine. Tuning that is too conservative gives up performance without there being any fault to look for.

A.1. Tuning parameters

ID	BSI NAME	FUNCTION	DEFAULT	RANGE	OBJECT
P237	kpSpeed	Proportional gain of the speed loop (K_{ps})	0.0030	0 ÷ 3.2000	20EDh
P238	kiSpeed	Integral gain of the speed loop (K_{is})	0.2000	0 ÷ 3.2000	20EEh
P253	kpPos	Proportional gain of the position loop (K_{pp})	0.050	0 ÷ 3.2000	20FDh
P252	EncLPFTau	Time constant of the filter on the speed estimate	0.005	0.001 ÷ 1.000 [s]	20FCh
P254	WstarFiltEnable	Enables the filter on the speed reference	Enable	Enable – Disable	20FEh
P256	Jloadff	Load inertia for the acceleration feedforward	0.00	0 ÷ 327.67 [kgcm²]	2100h

In BSI the parameters are found under **Motor > Control**, with the exception of P252 which is in **Motor > Encoder**, and they are only visible once the user level is set to **Advanced** from the **Change Userlevel** button. The OBJECT column gives the address in the fieldbus parameter area, reachable with an acyclic write: SDO on CanOpen, EtherCAT and Powerlink, Explicit Message on Ethernet/IP, Record Data on ProfiNET.

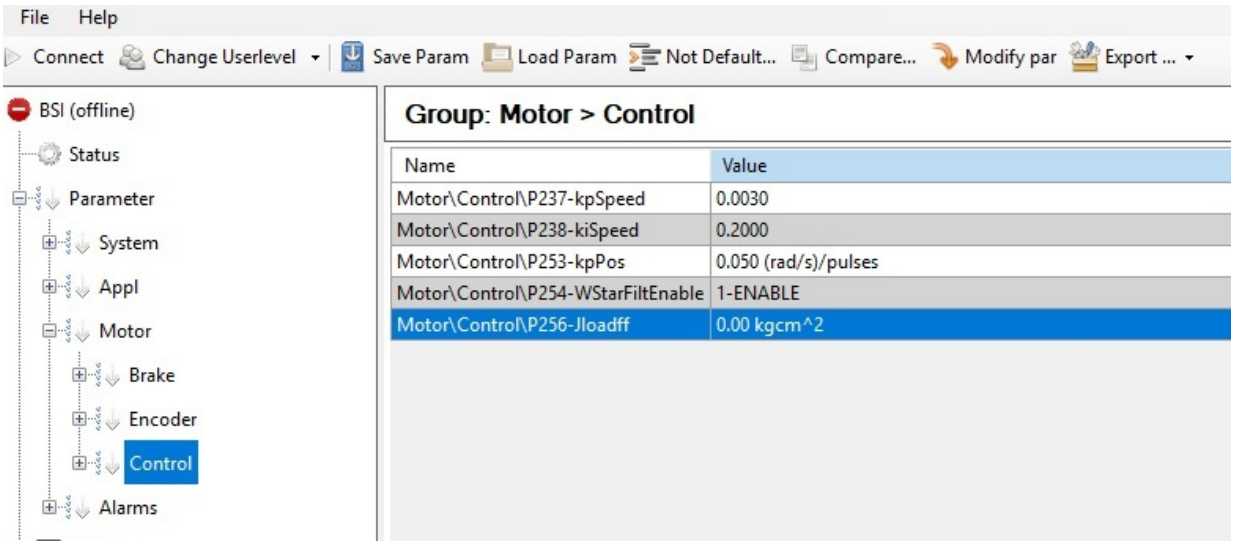


Figure 13. The Motor > Control group in the BSI parameter tree

When a parameter is changed from BSI it is saved to non-volatile memory automatically. Over the fieldbus it is not.

Writes in the 2000h … 21FFh area only affect the RAM image of the parameter. To make a tuning performed over the fieldbus permanent, a 32-bit write with the value 65766173h ("SAVE") must be performed on the *Store parameters* object: `1010h.1` on CanOpen, EtherCAT and Powerlink, `1010h` on Ethernet/IP and ProfiNET, where the subindex is not addressable. Without this step the gains revert to their previous values at the first restart.

A.2. Control scheme

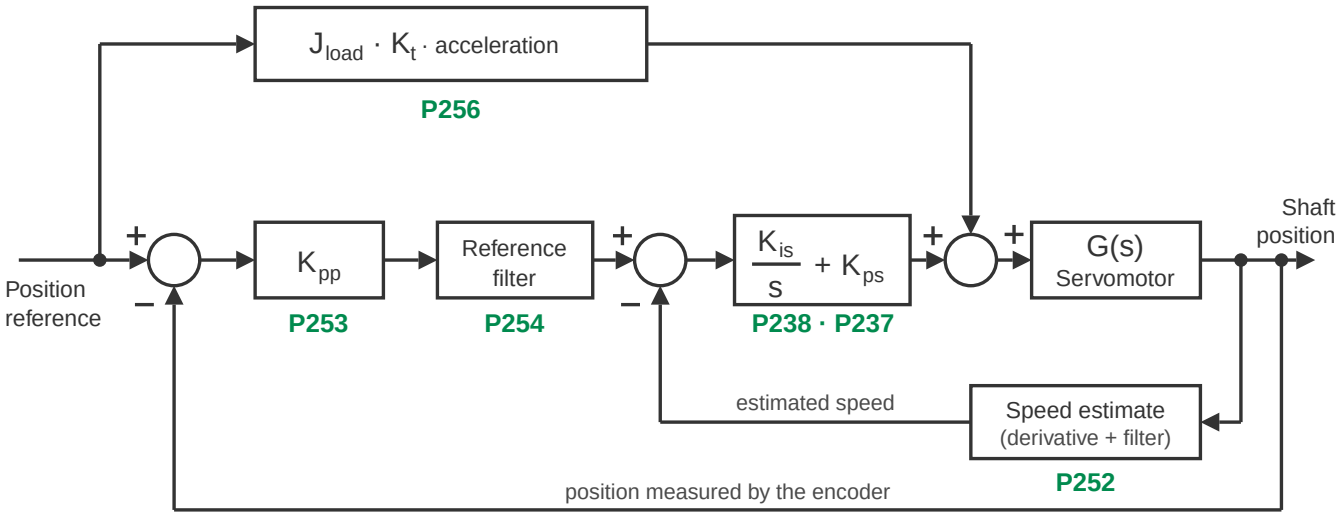


Figure 14. PIV loop with the tuning parameter associated to each block

The position reference is compared with the position read by the encoder and the resulting error is multiplied by K_{pp} (P253), which turns it into a speed reference. This reference, filtered or not depending on P254, is compared with the estimated speed and the error feeds the speed PI, made up of the proportional action K_{ps} (P237) and the integral action K_{is} (P238). The output of the PI is the torque reference.

Two details of the scheme have a bearing on tuning:

- The feedback speed is not measured: it is the derivative of the encoder position, filtered with the time constant P252. The bandwidth of the speed loop cannot exceed that of this estimate.
- The feedforward branch (P256) does not belong to any loop. It adds to the torque reference the contribution required by the acceleration of the profile directly, without waiting for an error to build up.

The proportional action reacts to the instantaneous error and is tied to the stiffness of the system: the higher it is, the more the motor opposes deviations, up to the limit beyond which the system oscillates. The integral action accumulates the error over time and cancels it at steady state, but it introduces delay and therefore overshoot.

A.3. Preparation

Tuning means deliberately driving the system towards its stability limit on a real machine. Work with the load disconnected or on free travel, keep the emergency stop within reach and, where possible, reduce P017 (IMax) to limit the torque available during the tests.

Tuning is normally performed with the motor already connected to the fieldbus and commanded in DS402: references and ramps are therefore set on the profile objects.

1. Select **Profile Velocity** mode by writing 3 to *Modes of Operation* (`6060h`) and check the confirmation on `6061h`.
2. Set the reference on *Target velocity* (`60FFh`) and the ramps on *Profile acceleration* (`6083h`) and *Profile deceleration* (`6084h`).
3. Bring acceleration and deceleration to the maximum operating values of the machine and write down the starting values: they must be restored at the end.
4. In BSI open **Data > Trace** and plot `de_speed_rpm` (actual speed), `usr_poserror` (error of the position loop) and `f_iq` (output current), triggering on the command.
5. Use the BSI “Not Default” function to take a snapshot of which parameters already differ from the factory values, before adding more.

An ideal step is not required: tune with the most demanding profile the machine actually runs, because that is the condition that has to be stable. Bear in mind, though, that if the ramps are gentle compared to the speed of the loop what you are measuring is the profile generator and not the control: the response looks clean but tells you nothing about how much margin you have. To find out the margin, steepen the ramps until overshoot appears.

In local control, without a fieldbus, the equivalents are P001 for mode selection and P008 / P009 for the acceleration and deceleration ramps.

A.4. Tuning the speed loop

The speed loop is tuned first: the position loop works on top of it and cannot be better than what it has underneath.

1. Command a speed transient of roughly half the cycle speed, with the ramps at the maximum operating

values, and watch `de_speed_rpm`.

2. Increase K_{ps} (P237), doubling it at each step, until overshoot appears. Go back to the last good value and refine it in 10 ÷ 20% increments. Stop just below the overshoot threshold.
3. Increase K_{is} (P238) until a contained overshoot appears, in the order of 5 ÷ 10% of the target speed. That is the sign that the integral action is enough to cancel the steady-state error without destabilising the response.

If a whistle or a hum appears before the overshoot does, or if ALR4/ALR5 Overcurrent trips, you are exciting a mechanical resonance: the stability margin is already used up and the value to keep is the previous one.

A.5. Tuning the position loop

1. Select **Profile Position** mode (value 1 on `6060h`), set *Profile velocity* (`6081h`) to roughly half the speed of the final cycle and leave the `6083h` / `6084h` ramps at the maximum operating values. Command the targets on *Target position* (`607Ah`).
2. Start from a low value of K_{pp} (P253) and increase it until overshoot appears, then go back to the previous step.
3. Bring `6081h` back to the speed of the real cycle, restore `6083h` and `6084h` to the operating values and check the behaviour over the whole cycle.

The acceptance criterion is `usr_poserror`: while moving it must stay within the following error window (6065h, `co_dc_ferror_win`) and at the end of the positioning it must fall inside the position window (6067h, `co_dc_pos_window`).

A.6. Fine adjustments

The three items below are only to be touched if, once tuning is complete, performance is still not enough.

Inertia feedforward (P256). This is not a gain to be raised by trial and error: it is the physical value of the load inertia referred to the motor shaft, in kgcm². Entered correctly, it anticipates the torque required by the acceleration phases and reduces the following error without touching the gains. An overestimated value produces overshoot during acceleration; an underestimated one leaves part of the error you had before.

Bandwidth of the speed estimate (P252). Lowering the time constant of the filter on the encoder increases the useful bandwidth of the control and allows higher gains. In exchange, the noise on the speed estimate rises and is passed on to the torque reference: come down in small steps and keep an eye on `f_iq`. If the current becomes visibly noisier or the motor whistles at standstill, you have gone too far down.

Filter on the speed reference (P254). When the profile is generated inside the drive the reference is already continuous and the filter is superfluous: disabling it recovers bandwidth. If instead the reference comes from the master at every bus cycle, as in CSP mode (optional, available on EtherCAT and Powerlink), leave the filter enabled: without it the discontinuities of the reference are passed straight on to the torque.

A.7. Diagnostics

SYMPTOM	LIKELY CAUSE	ACTION
Marked overshoot on the speed transient	K_{ps} or K_{is} beyond the useful value	Reduce P237; if it persists, reduce P238
Oscillation that does not settle	Gains beyond the mechanical stability limit	Reduce P237, then P253
Slow response, the motor lags the reference	Gains too low	Increase P237, then P253
Constant steady-state error in speed	Insufficient integral action	Increase P238
High following error only during acceleration	Feedforward contribution missing	Enter the real load inertia in P256
Whistle or hum at standstill	Noise of the speed estimate amplified by the gain	Increase P252, or reduce P237
ALR4 / ALR5 Overcurrent during transients	Required torque beyond the limit	Reduce P237 and the profile accelerations; check P017
Tuning lost after a restart	Fieldbus write not committed to non-volatile memory	Write 1010h.1 as indicated above

Appendix B: Accelerometer

The drive features a three-axis MEMS accelerometer that measures the mechanical vibration on the motor. It serves to keep mechanical stress under control during operation and to log excessive acceleration events even when nobody is watching.

- Selectable full scale ±2 / ±4 / ±8 g (P035)
- 16-bit resolution, data expressed in milli-g
- Sampling rate from 10 to 800 Hz (P036)

B.1. Axes and available data

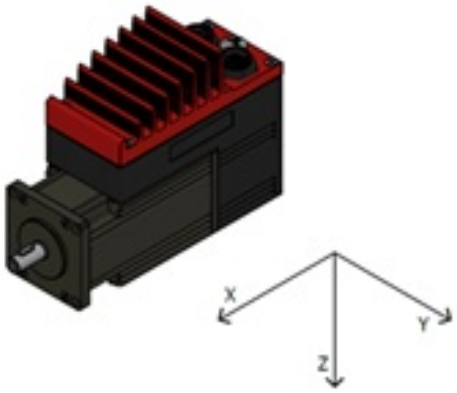


Figure 15. Reference frame of the accelerometer, fixed to the motor body

The frame is fixed to the motor: **X** is parallel to the shaft axis and positive towards the output shaft, **Y** is perpendicular to the shaft, **Z** is perpendicular to the face the drive is mounted on and positive downwards.

For each axis the drive makes three quantities available, all in milli-g:

QUANTITY	BSI VARIABLES	DESCRIPTION
Instantaneous acceleration	`usr_acc_x` `usr_acc_y` `usr_acc_z`	Sampled value, net of the DC component
DC component	`usr_acc_x_ofs` `usr_acc_y_ofs` `usr_acc_z_ofs`	Constant part of the measurement: it is the gravity vector projected on the three axes and is used to work out the mounting orientation
RMS value	`usr_acc_x_rms` `usr_acc_y_rms` `usr_acc_z_rms`	Moving root mean square, filtered with the time constant P037

The DC component is recomputed at every power-up and subtracted from the instantaneous accelerations, so that these are not affected by constant forces. The motor must therefore be at standstill and in its final orientation when it receives the logic supply: if it is powered up while moving, or tilted differently from how it will work, the offset stays wrong for the whole session.

The RMS value is the figure best suited to continuous monitoring: it changes slowly, it is not affected by the single sample and it is the only one that makes sense to exchange at every fieldbus cycle.

B.2. Parameters

ID	BSI NAME	FUNCTION	RANGE	OBJECT
P035	Endscale	Full scale of the accelerometer	2 – 4 – 8 [g]	2023h
P036	DataRate	Sampling rate	10 ÷ 800 [Hz]	2024h
P037	RmsTau	Time constant of the RMS filter	1 ÷ 1000 [ms]	2025h
P038	EventLevel	Acceleration threshold that generates the event	0.1 ÷ 10 [g]	2026h
P039	EventTime	Time constant of the filter on the event	1 ÷ 1000 [ms]	2027h
P040	EventInhibit	Inhibition time between two consecutive events	1 ÷ 2400 [minutes]	2028h

In BSI the parameters are found under **Appl > Accelerometer**.

The full scale P035 is a trade-off: a low value gives more resolution on small vibration but saturates on impacts, a high value does not saturate but loses sensitivity. Start from ±2 g and only go up if you see the measurement clipped at full scale.

B.3. Sampling

Parameter P036 sets the rate at which the microcontroller acquires data from the sensor.

SAMPLING RATE	PERIOD	OBSERVABLE BANDWIDTH
10 Hz	100 ms	5 Hz
50 Hz	20 ms	25 Hz
100 Hz	10 ms	50 Hz
200 Hz	5 ms	100 Hz
400 Hz	2.5 ms	200 Hz
800 Hz	1.25 ms	400 Hz

The observable bandwidth is half the sampling rate: sampling at 200 Hz, components up to 100 Hz can be distinguished, beyond which two different vibrations become indistinguishable once sampled.

At the two highest rates, 400 and 800 Hz, the microcontroller cannot read the sensor at every conversion. The resulting undersampling produces aliasing: high-frequency components appear in the measurement as low-frequency ones, indistinguishable from the real ones.

The same reasoning applies downstream, on the data exchange cycle with the PLC. Sampling at 400 Hz, that is every 2.5 ms, and then exchanging data every 4 ms means undersampling a second time: the PLC receives samples taken at arbitrary instants of the vibration. This is why, over the fieldbus, the figure to acquire is the

RMS value and not the instantaneous acceleration.

B.4. Tuning the RMS filter

The time constant of the RMS filter must span several cycles of the vibration to be measured: if it is too short the RMS value oscillates instead of giving a stable reading. It therefore has to be tuned on the specific application.

Before starting, check that P036 is set so that the vibration of interest falls inside the observable bandwidth.

1. In BSI open **Data > Trace** and plot the instantaneous accelerations `usr_acc_x`, `usr_acc_y` and `usr_acc_z`.
2. Enable the cursors and place them on two consecutive peaks of the vibration.
3. Read Δ, the interval between the two cursors, which BSI also reports as an equivalent frequency.
4. Multiply Δ by a factor between 3 and 5 and write the result into P037.

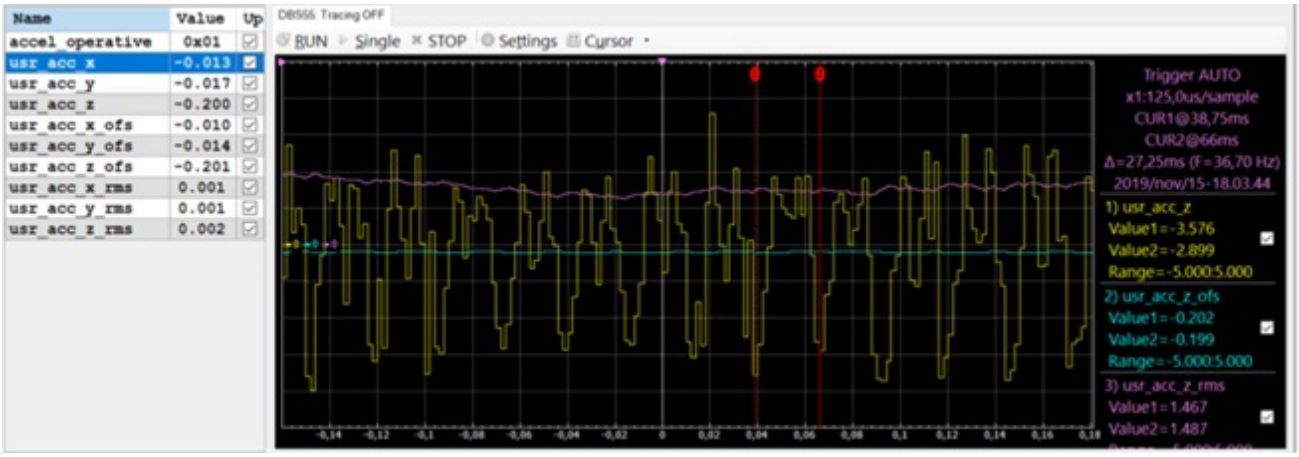


Figure 16. Measuring the vibration period with the Trace cursors

In the example in the figure Δ is 27.25 ms, so P037 is to be set between 82 and 136 ms. Longer time constants give a more stable reading but react more slowly; shorter ones react sooner but make the value jump around.

B.5. Over-acceleration events

The drive logs by itself the episodes in which the acceleration exceeds a set threshold, without the PLC having to monitor it. The function is governed by three parameters:

- **P038 EventLevel** is the acceleration threshold beyond which the event is triggered.
- **P039 EventTime** is the time constant of the filter applied before the comparison with the threshold, and prevents a single isolated sample from generating a record.
- **P040 EventInhibit** is the inhibition time between one event and the next. It avoids a persistent condition filling the log with identical records.

The event logged is number 3, **ACCMAX_REACHED**, according to the coding of the Alarms chapter.

The log is read in BSI from **List > Event**. Each row reports the event code and the SupplyTime, that is the cumulative logic supply time of the drive at the moment of the record: events are therefore correlated with one another by supply hours, not by date and time. From there the list is exported to a file with “Save” and cleared with “Erase list”.

On CanOpen, EtherCAT and Powerlink the events are also accessible from object 3009h with an acyclic read; the object is not PDO mappable. On Ethernet/IP and ProfiNET it is not exposed.

B.6. Reading over the fieldbus

OBJECT	DATA	TYPE	PDO
3030h	Acceleration RMS modulus	UINT16	Yes
3031h ÷ 3033h	Acceleration RMS x, y, z	UINT16	Yes
3034h ÷ 3036h	DC component x, y, z	UINT16	Yes
3037h ÷ 3039h	Instantaneous acceleration x, y, z	INT16	Yes

All values are expressed in milli-g.

On CanOpen, EtherCAT and Powerlink these objects can be freely mapped into the PDOs. On Ethernet/IP and ProfiNET the RMS values of the three axes are not part of the default process data: profile **D** must be selected, which adds them to the assemblies.

B.7. Vibration analysis

The accelerometer returns data in the time domain. From these, with an offline analysis, the quantities that describe the vibration are obtained.

- **Peak-to-peak**: maximum excursion of the wave, an indicator of the severity of the mechanical stress or of backlash.
- **Peak**: maximum positive or negative excursion, an indicator of the intensity of short impacts.
- **RMS**: root mean square value, tied to the energy content of the vibration.

$$RMS = \sqrt{\frac{1}{T}\int_0^T x^2(t)\,dt}$$

- **Crest factor**: ratio between peak value and RMS value, that is between the two arrows on the right of the figure below. High values identify phenomena of an impulsive nature, low values a distributed vibration. A pure sine wave gives 1.41; the vibration in the figure gives 1.7.

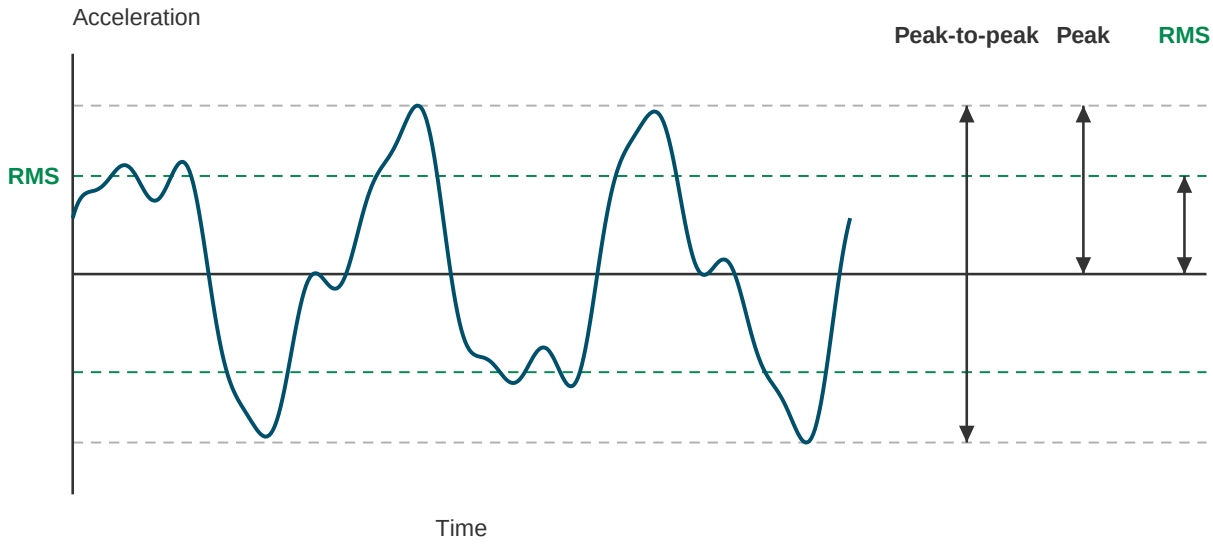


Figure 17. The characteristic quantities measured on a vibration in the time domain

The next step is the analysis in the frequency domain, obtained by applying a Fourier transform to the time data with any computation tool. The spectrum shows which components make up the vibration and with what amplitude.

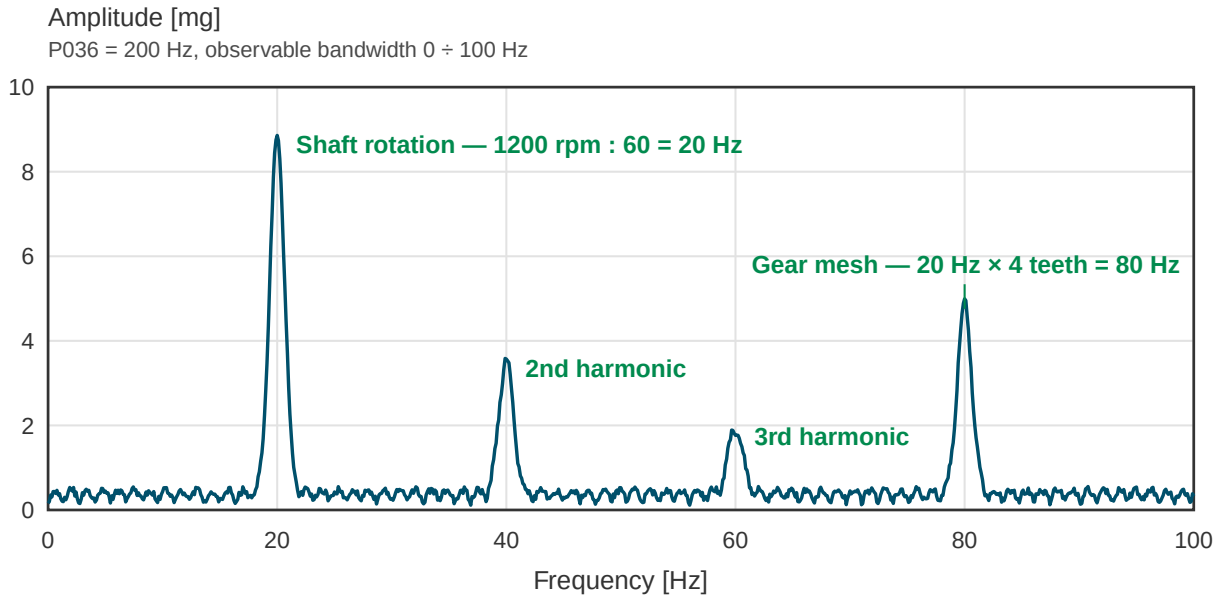


Figure 18. Amplitude spectrum with the lines traced back to the characteristic frequencies of the machine

The lines of the spectrum are interpreted by comparing them with the characteristic frequencies of the machine, which you can work out in advance: the shaft rotation frequency (revolutions per minute divided by 60), the gear mesh frequency of the gearbox (shaft frequency multiplied by the number of teeth) and their respective harmonics.

In the example in the figure the motor runs at 1200 rpm, so the dominant line at 20 Hz is the shaft rotation. The lines at 40 and 60 Hz are the second and third harmonic: they are not independent sources, but the sign that the phenomenon at 20 Hz is not sinusoidal. The line at 80 Hz, on the other hand, is not a dominant harmonic but coincides with the gear mesh frequency of the gearbox, 20 Hz times the 4 teeth of the pinion: it is a separate source, and is to be followed on its own.

There is no vibration threshold that is valid in absolute terms: it depends on the machine and on the

application, and providing an all-encompassing guide is beyond the scope of this manual. The method that works is to measure the vibration signature on the machine under correct operating conditions and to monitor its drift over time: it is the departure from the baseline that flags the problem, not the absolute value. As a general frame of reference for vibration levels, the ISO 20816 series of standards can be used.

Appendix C: Encoder

The servo gearmotor features a 32-bit absolute multiturn magnetic encoder. Absolute means that the shaft position is known from power-up, with no need for a homing run, and that the motor keeps it even when switched off, even if the shaft is turned by hand in the meantime.

Figure 19. The encoder board with the integrated backup battery

C.1. Resolution

The 32 bits are split into two parts:

PART	BITS	MEANING
Single turn (ST)	12	Angular position within the turn: 4096 pulses per motor revolution
Multiturn (MT)	20	Count of the revolutions performed by the shaft: 1,048,576 turns

The split can be changed with parameter **P094 EncStRes**, which sets the bits assigned to the single turn. The total always stays at 32 bits, so **every bit gained on the single turn is a bit lost on the revolution count**: taking P094 from 12 to 24 bits raises the angular resolution to 16,777,216 pulses per revolution, but the multiturn range drops from 1,048,576 to 256 turns.

Values above 12 bits are only allowed on motors with the **HRE** option.

Before raising P094, work out how many revolutions the axis has to travel between one homing and the next. On a continuously rotating axis, or on a long screw, 256 turns are used up quickly.

C.2. The backup battery

The multiturn count is the only part that needs to be powered while the motor is off: the position within the turn is absolute magnetic and is read back by itself at every power-up. To retain the revolution count, the encoder electronics are supplied by a rechargeable battery integrated in the gearmotor.

- **Autonomy**: 6 months with the motor disconnected from the supply.

- **Charging**: automatic whenever the logic supply is present. A full charge takes at least 24 consecutive hours.
- **Replacement**: the battery is not replaceable. It is rechargeable, and has to be kept charged with the cycles described below.

Under normal operating conditions the battery is invisible to the system: with the logic supply present it stays fully charged and the motor runs on mains power.

There is no reading of the battery charge state. The system gives no advance notice: the first sign is the alarm at the next power-up, when the multiturn position has already been lost. This is why the preventive charging cycles must be observed.

The product receives a full charge before shipping. At the first power-up, or after a period of inactivity, leave the gearmotor with the logic supply alone switched on for at least 24 hours before putting it into service.

C.3. Storage management

Every 6 months of non-use the battery must be recharged with a full cycle of at least 24 hours of logic supply, as also stated among the maintenance warnings of the Safety chapter.

The typical case in which the battery runs flat is not storage but transport: a machine that has been tested and packed and then spends months on a ship arrives at its destination with the multiturn count lost, even though it was working at final testing.

C.4. Loss of the multiturn position

If the battery runs flat, at the next power-up the motor comes up in fault: bit 3 of the Status Word goes to 1 and register `3008h` *Actual Alarm* reports code **203 PMSM_ENCSPI_MT**, according to the coding of the Alarms chapter.

Under a very deep discharge, code **204** may appear as well. In that case a double reset is required, first for the 204 and then for the 203, before being able to proceed.

The error is reset by taking bit 7 of the Control Word to a logic high level.

After a 203 error the multiturn position is no longer reliable. The axis **must be homed again** before resuming normal operation: resetting the alarm puts the motor back in a condition to work, but it does not restore the position reference that was lost.

Register `3008h` is exchanged in the process data of Ethernet/IP and ProfiNET, and is freely accessible and PDO mappable on CanOpen, EtherCAT and Powerlink.

Our advice is to have a routine in the PLC that, faced with a motor in fault, reads the alarm code and handles it accordingly. If it detects a 203, it must tell the operator that this servomotor needs realigning before putting the machine back in cycle, rather than simply resetting the fault.

Appendix D: Prewired Cables

One of the DBS accessories are pre-wired cables. For availability, minimum quantity, delivery times and lengths available, check the DBS configurator or contact Mini Motor.

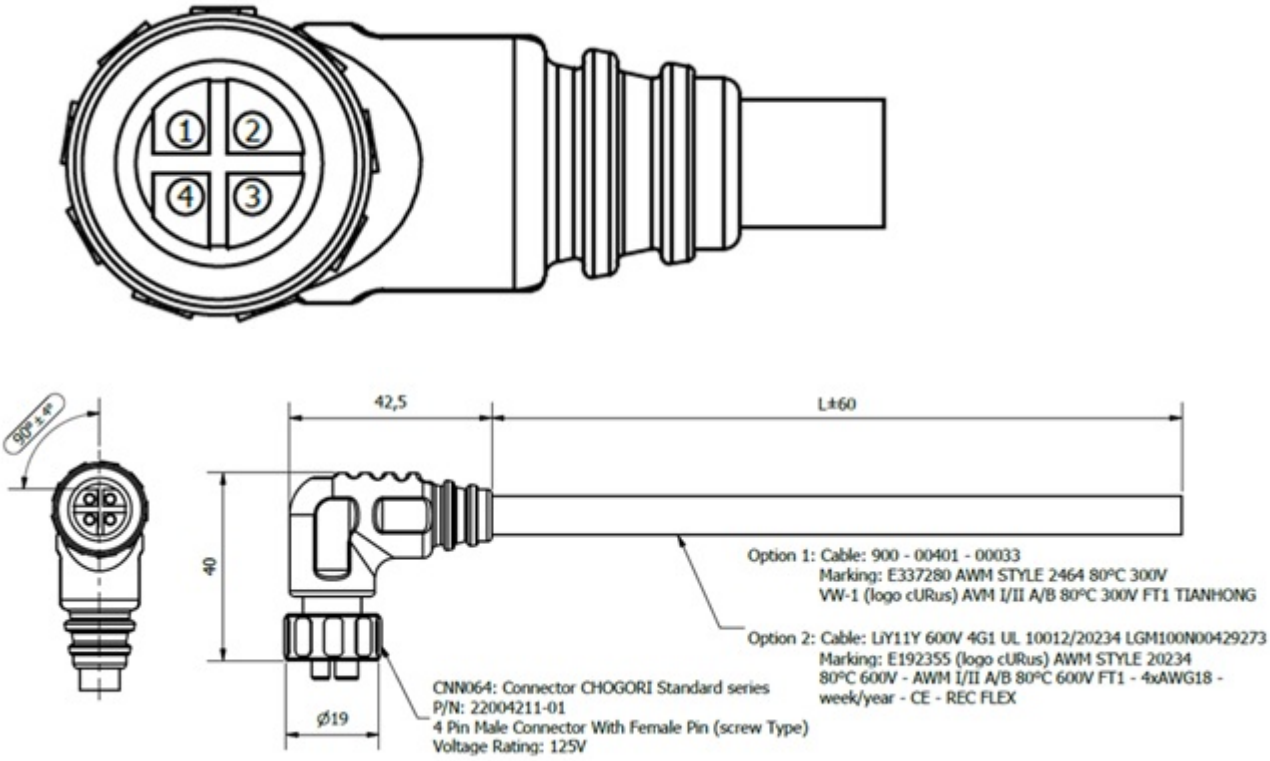
The available codes are the following:

Product Code	Installation	Angle	Function
CBL023	Fixed	90°	CN1 DBS55
CBL024	Fixed	90°	CN2
CBL031	Mobile	90°	CN1 DBS55
CBL032	Mobile	90°	CN2
CBL033	Fixed	Straight	CN1 DBS55
CBL034	Fixed	Straight	CN2
CBL035	Mobile	Straight	CN1 DBS55
CBL036	Mobile	Straight	CN2
CBL037	Fixed	Straight	CN1 DBS 80
CBL038	Mobile	Straight	CN1 DBS 80
CBL039	Fixed	90°	CN1 DBS 80
CBL040	Mobile	90°	CN1 DBS 80

The connection schemes are the following: (Note, Logica is Logic, Potenza is Power)

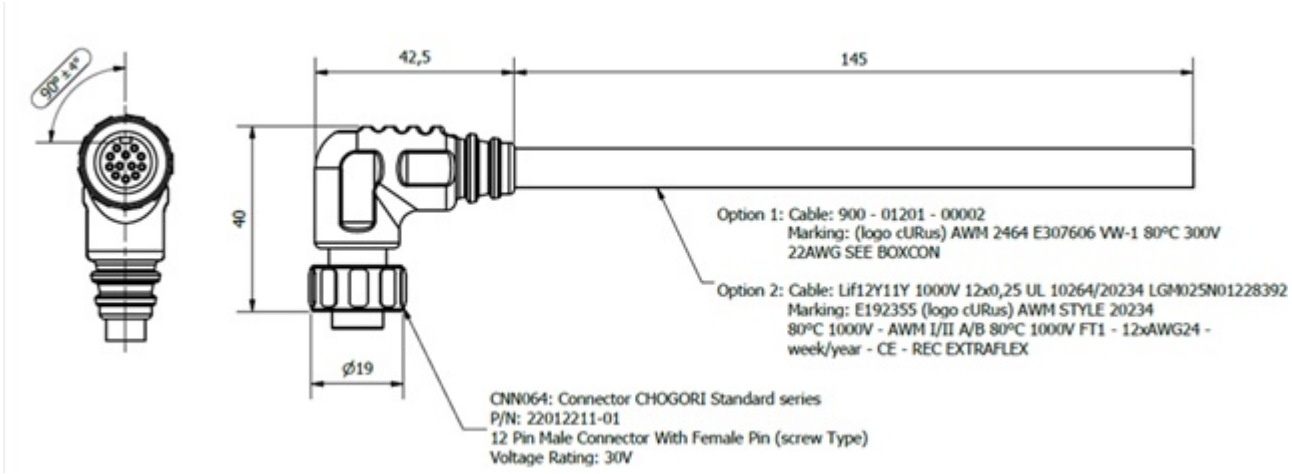
D.1. CBL023

Pin Assignments Front View			
Conn	Mode	Wire Color Option 1	Wire Color Option 2
4	+ Logica	Brown	Brown
3	+ Potenza	Red	Gray
2	- Potenza	Black	Black
1	- Logica	Blue	Blue



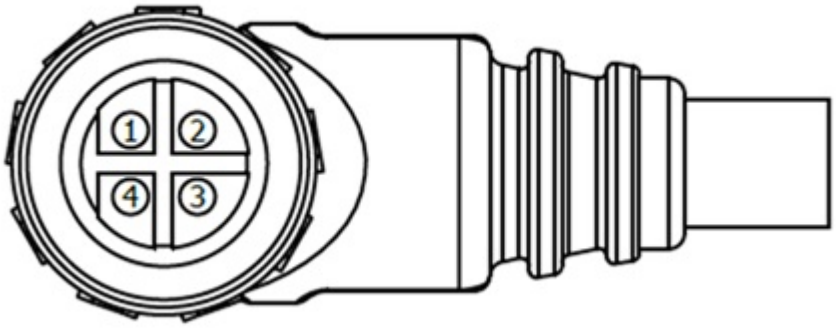
D.2. CBL024

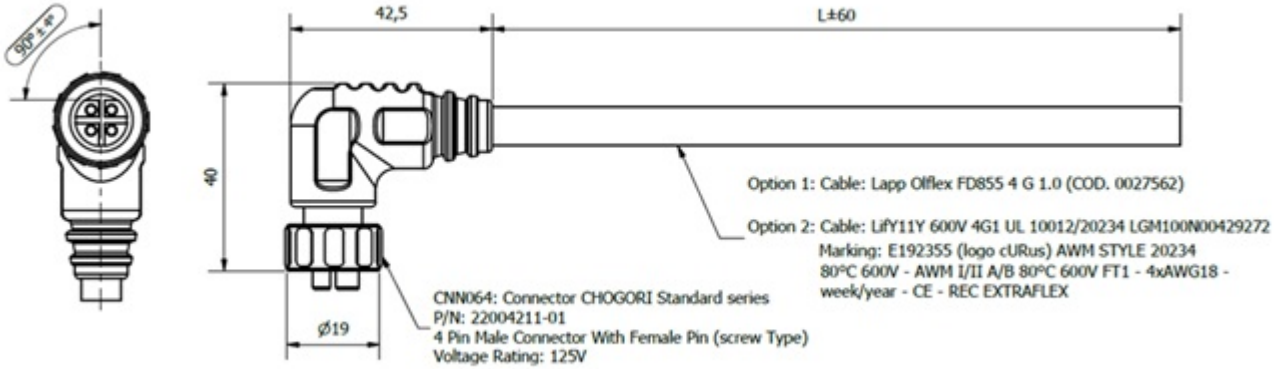
Pin Assignments Front View		
PIN	Wire Color Option 1	Wire Color Option 2
1	Black	Black
2	Brown	Brown
3	Red	Red
4	Orange	Gray/Pink
5	Yellow	Yellow
6	Green	Green
7	Blue	Blue
8	Purple	Purple
9	Grey	Grey
10	White	White
11	Pink	Pink
12	Light green	Red/Blue



D.3. CBL031

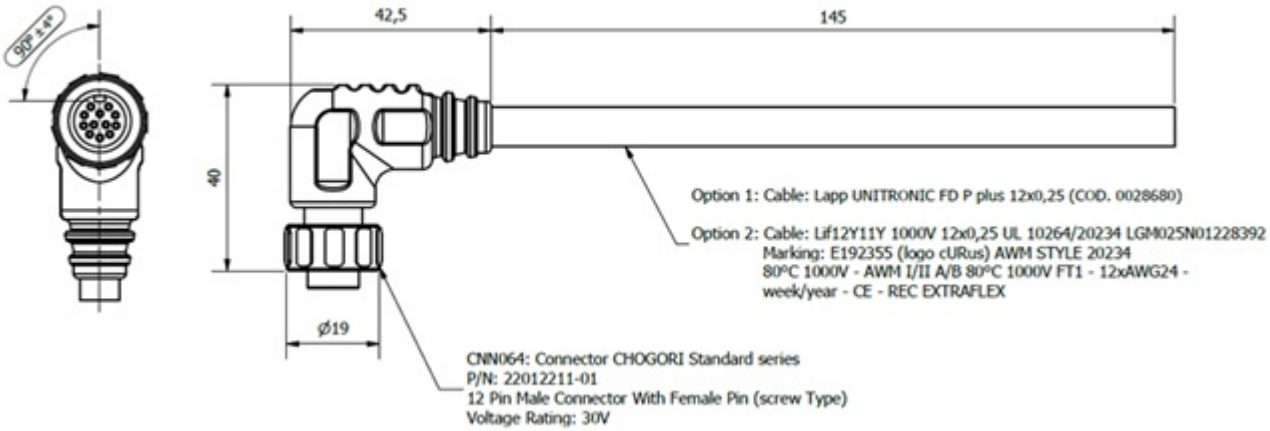
Pin Assignments Front View			
Conn	Mode	Wire Color Option 1	Wire Color Option 2
4	+ Logica	3	Brown
3	+ Potenza	2	Gray
2	- Potenza	1	Black
1	- Logica	Yellow/Green	Blue





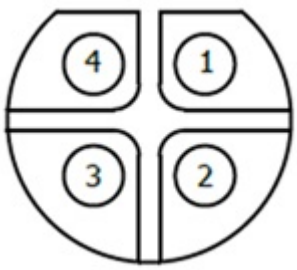
D.4. CBL032

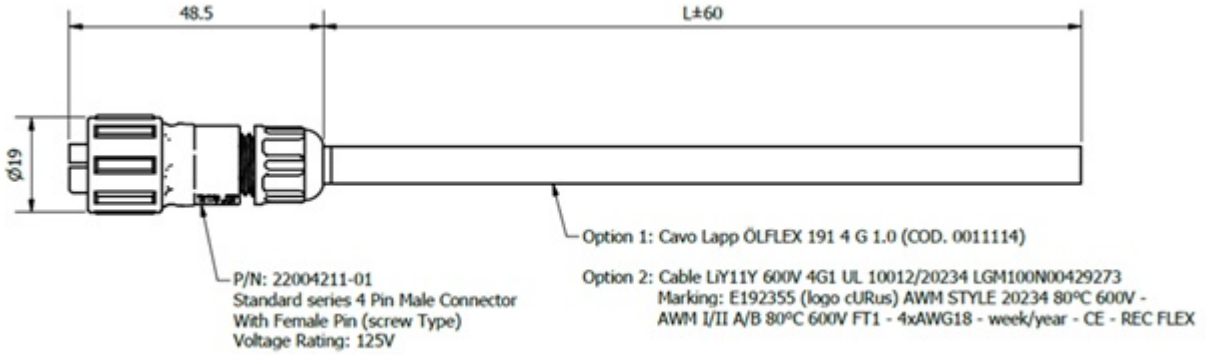
Pin Assignments Front View	
PIN	COLOR
1	Black
2	Brown
3	Red
4	Gray/Pink
5	Yellow
6	Green
7	Blue
8	Violet
9	Grey
10	White
11	Pink
12	Red/Blue



D.5. CBL033

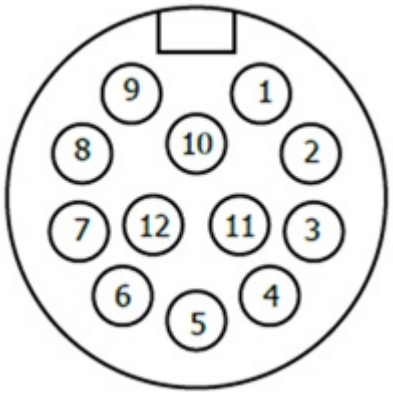
Pin Assignments Front View			
Conn	Mode	Wire Color Option 1	Wire Color Option 2
4	+ Logica	3	Brown
3	+ Potenza	2	Gray
2	- Potenza	1	Black
1	- Logica	Yellow/Green	Blue

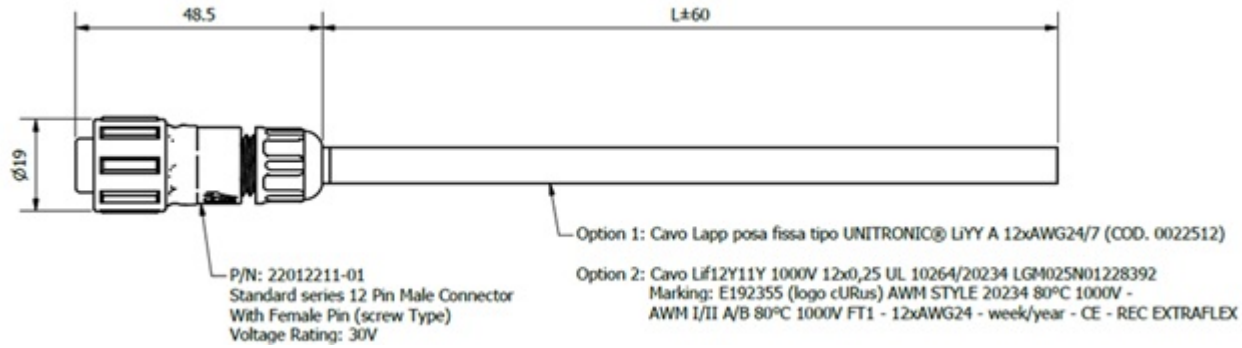




D.6. CBL034

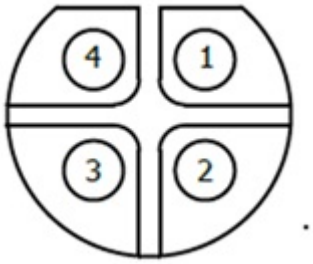
Pin Assignments Front View	
PIN	COLOR
1	Black
2	Brown
3	Red
4	Gray/Pink
5	Yellow
6	Green
7	Blue
8	Violet
9	Grey
10	White
11	Pink
12	Red/Blue

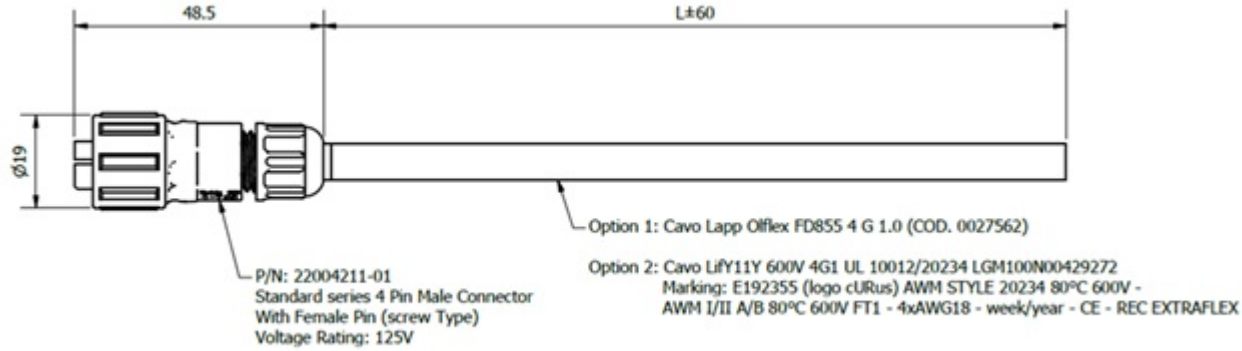




D.7. CBL035

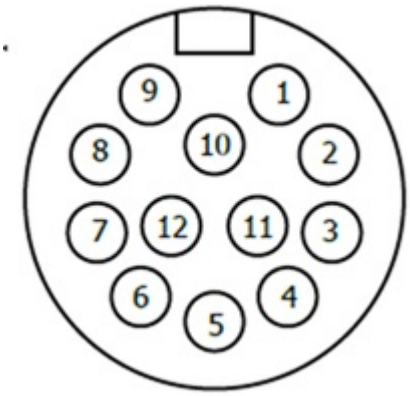
Pin Assignments Front View		
Conn	Wire Color	Mode
4	3	+ Logica
3	2	+ Potenza
2	1	- Potenza
1	Yellow/Green	- Logica

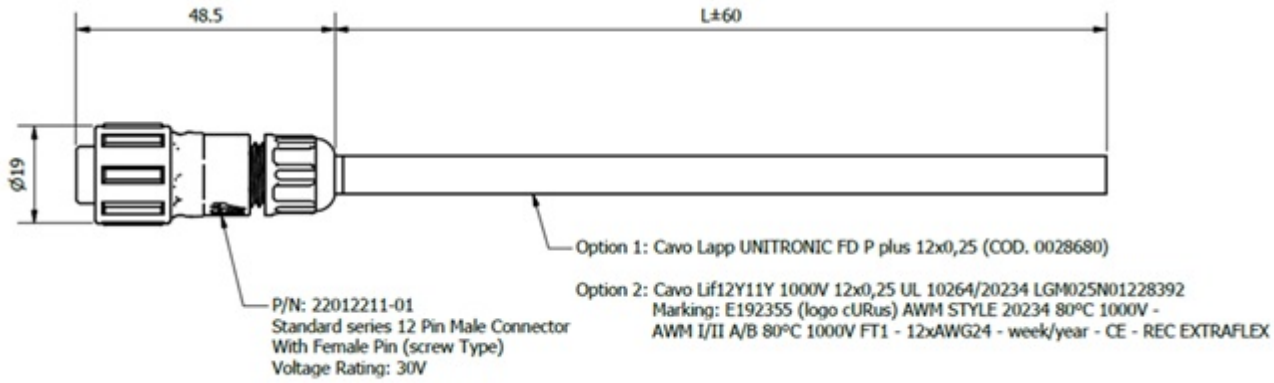




D.8. CBL036

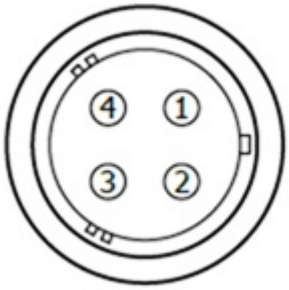
Pin Assignments Front View	
PIN	COLOR
1	Black
2	Brown
3	Red
4	Grey/Pink
5	Yellow
6	Green
7	Blue
8	Violet
9	Grey
10	White
11	Pink
12	Red/Blue

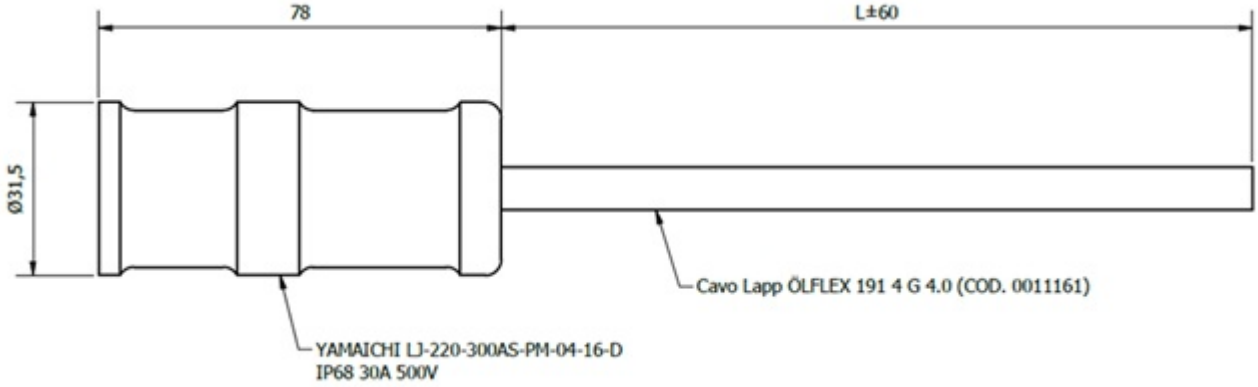




D.9. CBL037

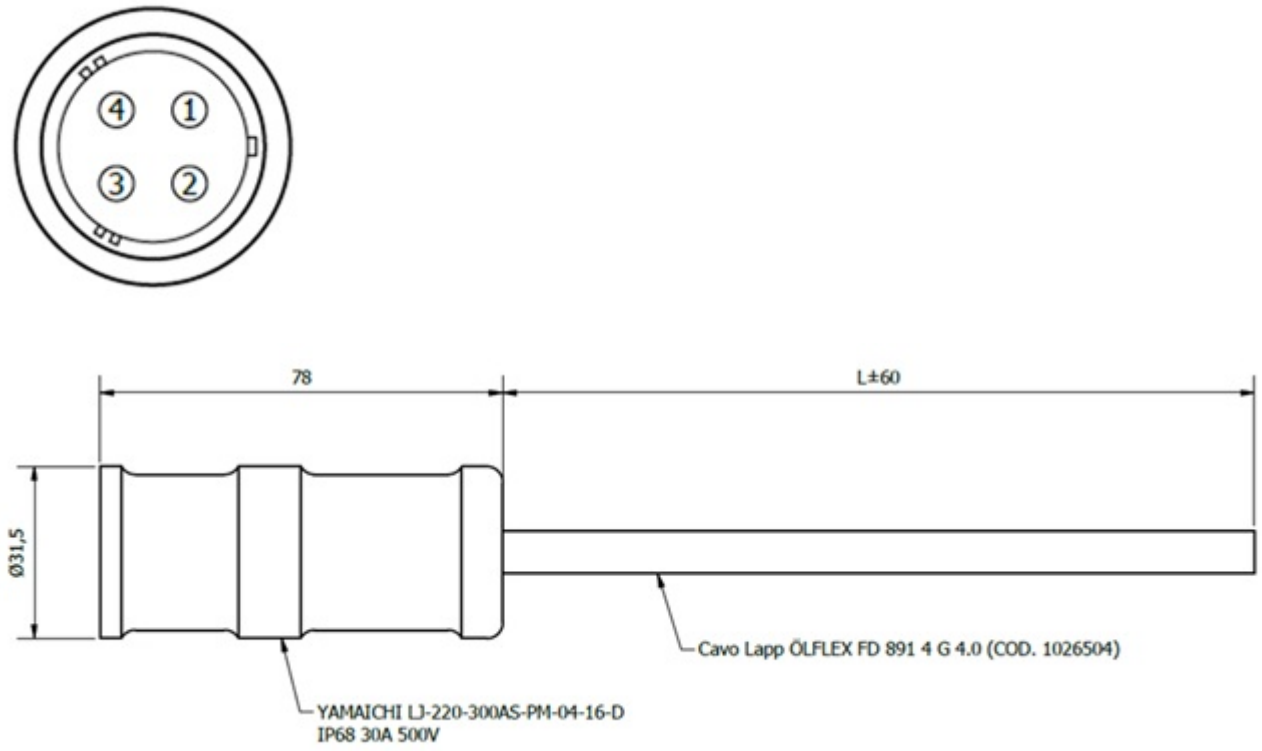
Pin Assignments Front View		
Conn	Wire Color	Mode
4	3	+ Logica
3	2	- Potenza
2	1	+ Potenza
1	Yellow/Green	- Logica





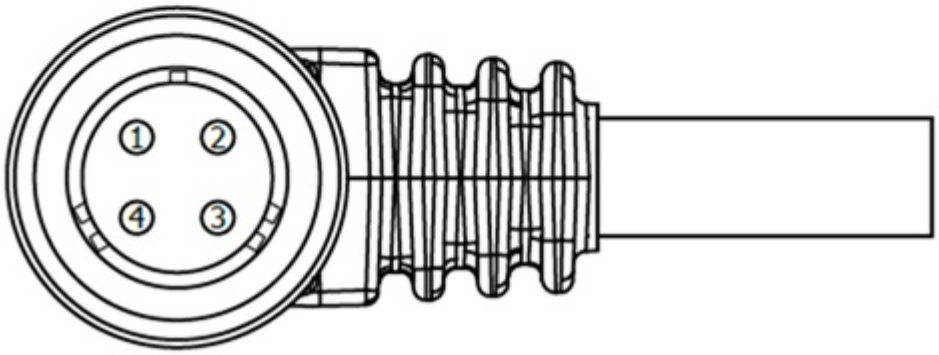
D.10. CBL038

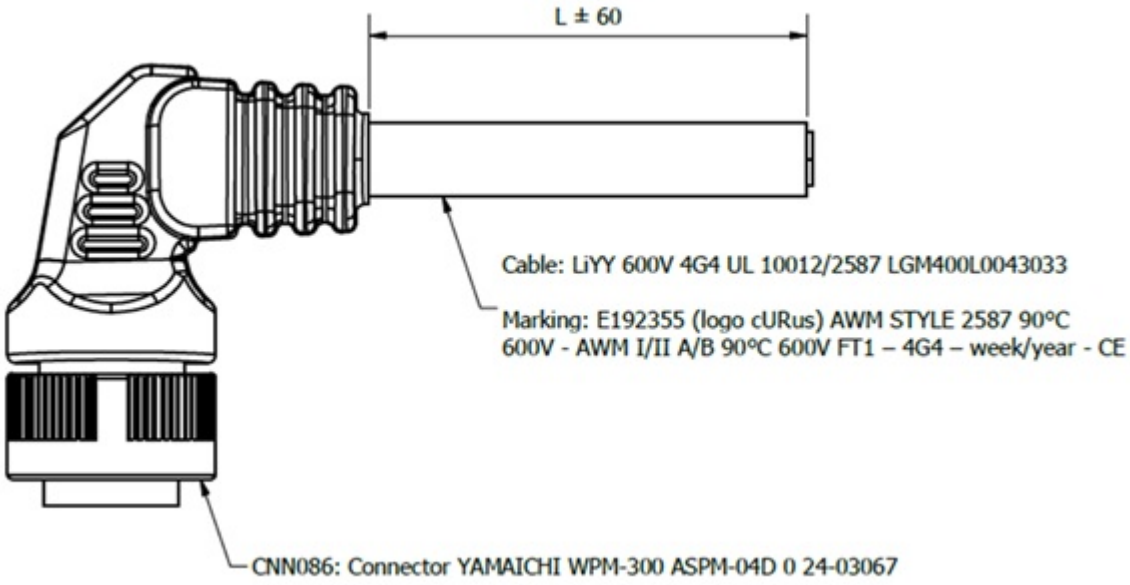
Pin Assignments Front View			
Conn	Mode	Wire Color Option 1	Wire Color Option 2
4	+ Logica	3	Brown
3	- Potenza	2	Black
2	+ Potenza	1	Gray
1	- Logica	Yellow/Green	Blue



D.11. CBL039

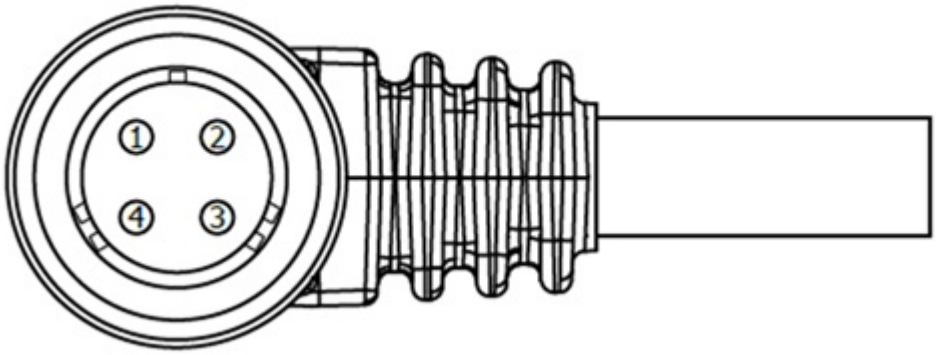
Pin Assignments Front View		
Conn	Wire Color	Mode
4	Brown	+ Logica
3	Black	- Potenza
2	Gray	+ Potenza
1	Blue	- Logica

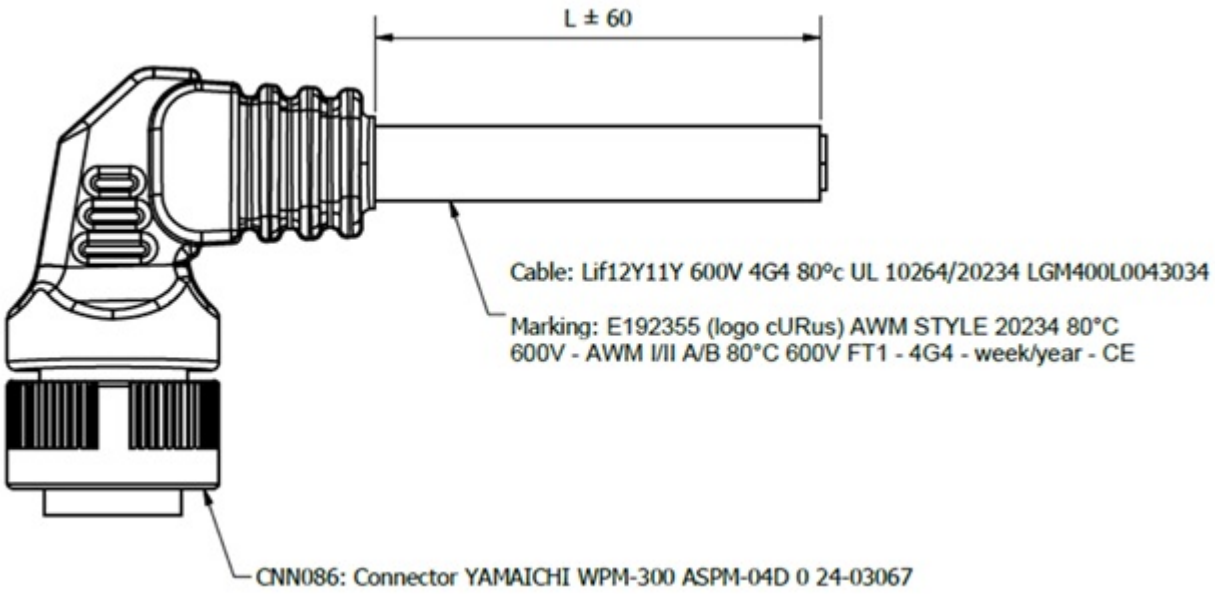




D.12. CBL040

Pin Assignments Front View		
Conn	Wire Color	Mode
4	Brown	+ Logica
3	Black	- Potenza
2	Gray	+ Potenza
1	Blue	- Logica




L ± 60
Cable: Lif12Y11Y 600V 4G4 80°c UL 10264/20234 LGM400L0043034
Marking: E192355 (logo cURus) AWM STYLE 20234 80°C
600V - AWM I/II A/B 80°C 600V FT1 - 4G4 - week/year - CE
CNN086: Connector YAMAICHI WPM-300 ASPM-04D 0 24-03067

Appendix E: Regenerative Braking

An electric motor is a bidirectional machine: it converts electrical energy into kinetic energy and, just as well, kinetic energy into electrical energy. The conversion runs the second way every time the system applies a torque opposing the direction of rotation of the shaft, that is, on every deceleration.

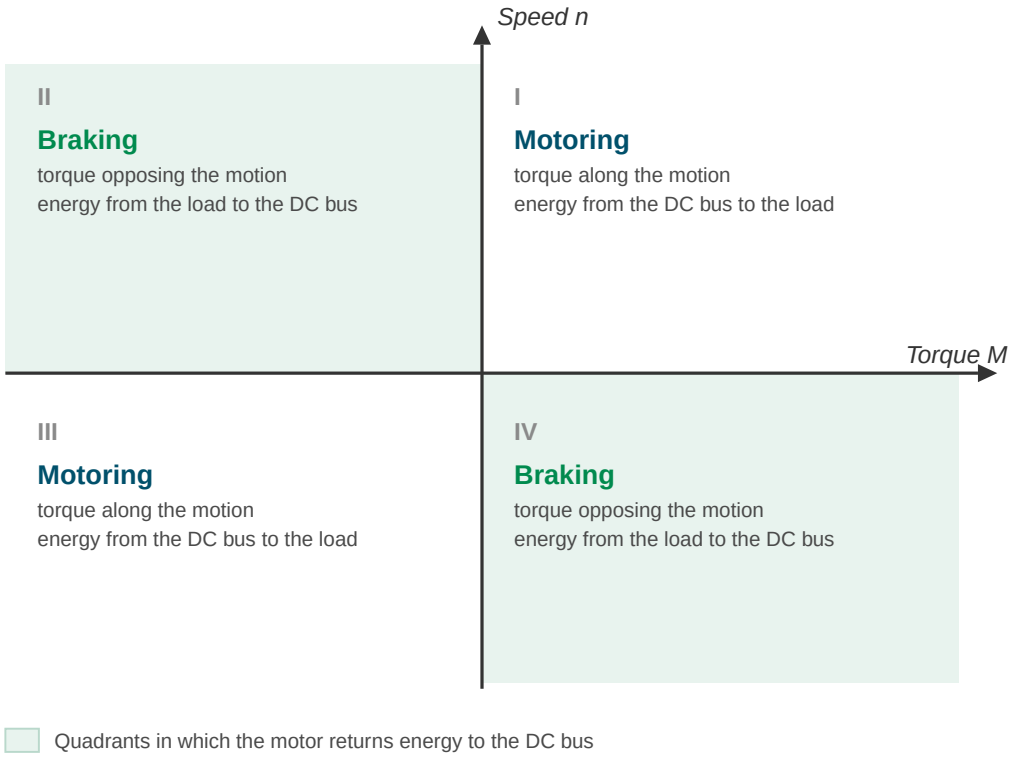


Figure 20. The four operating quadrants: in II and IV the motor returns energy to the DC bus

The energy produced this way has to go somewhere. In low-voltage systems, fed by a single power supply, it ends up stored on the DC bus, on the capacitances of the network.

E.1. When braking energy becomes a problem

If the energy to be absorbed exceeds the capacity of the DC bus, the bus voltage rises. The drive has an overvoltage threshold at **55 V**: past that, bit 3 of the Status Word goes to 1 and register `3008h` *Actual Alarm* reports code **3 OVERVOLT**, according to the coding of the Alarms chapter. The alarm is reset by taking bit 7 of the Control Word to a logic high level.

A single sporadic event is handled with a reset. If instead the alarm comes back regularly during the normal machine cycle, the system is producing more regenerated energy than it can absorb and the problem has to be tackled at its root.

E.2. First check: lengthen the deceleration

Before adding components, check whether the cycle allows it: the power to be disposed of is inversely proportional to the deceleration time, so doubling T_{dec} halves the regenerated power. It is often enough, and it costs nothing.

Gain tuning that is too aggressive also contributes, because it produces sharper torque transients than

necessary: see the appendix on gains tuning.

If the machine cycle does not allow gentler decelerations, move on to sizing a dedicated component.

E.3. Sizing

Consider the classic cycle made up of acceleration, a constant-speed stretch and deceleration, and focus on the deceleration phase. The kinetic energy to be disposed of is:

$$E_k = \frac{1}{2}(J_m + J_l) \cdot \left(\frac{2\pi N}{60}\right)^2$$

with J_m the motor inertia and J_l the load inertia referred to the motor shaft, both in kg·m², and N the speed in revolutions per minute.

Once the deceleration time T_{dec} of your cycle is known, the average power to be dissipated during braking is:

$$P_{dec} = \frac{E_k}{T_{dec}}$$

The figure obtained is conservative: part of the kinetic energy is lost in friction and in the copper and iron losses of the motor, and never reaches the DC bus.

Example

Motor at 48 V, cycle speed N = 3000 rpm, deceleration in T_{dec} = 0.2 s. As example values, motor inertia $0.5 \cdot 10^{-4}$ kg·m² and load inertia referred to the shaft $4.5 \cdot 10^{-4}$ kg·m²: the actual figures depend on the motor model and on the gearbox in between.

$J_m + J_l = 5.0 \cdot 10^{-4}$ kg·m²

$\omega = 2\pi \cdot 3000 / 60 = 314$ rad/s

$E_k = ½ \cdot 5.0 \cdot 10^{-4} \cdot 314^2 \approx$ **24.7 J**

$P_{dec} = 24.7 / 0.2 \approx$ **123 W**

If the cycle repeats every 2 seconds, the average power over the whole cycle is instead 24.7 / 2 ≈ **12.4 W**: these are the two figures to compare with the limits of the chosen component, the first for the peak and the second for thermal endurance.

E.4. Capacitors or braking resistor

The two possible technical solutions are opposite in philosophy:

- **Capacitors**: they store the braking energy and make it available for the next start, behaving as a reservoir. They recover energy, but they take up space and are only worthwhile if the cycle reuses it straight away.
- **Braking resistor**: it dissipates the excess power as heat. It recovers nothing, but it is compact, inexpensive and works with any cycle.

To size the capacitors, start from the energy they have to absorb:

$$E_{DC} = \frac{1}{2}C\left(V_h^2 - V_l^2\right)$$

from which the capacitance to be added:

$$C = \frac{2\,E_{DC}}{V_h^2 - V_l^2}$$

with V_h the maximum voltage allowed before the overvoltage alarm and V_l the nominal bus voltage. The margin between the two is what determines the capacitance needed: on a 48 V bus with a threshold at 55 V the window is narrow, and the required capacitance grows quickly. This is why, in most applications, the braking resistor is the practical choice.

E.5. The braking boards

Our range of accessories includes three braking boards, distinguished by motor supply voltage and by power:

PARAMETER	SCH108	SCH109	SCH110
Motor supply	24 V	48 V, low power	48 V, high power
Operating voltage	12 ÷ 40 V	33 ÷ 65 V	12 ÷ 60 V
Switch-on threshold	18 ÷ 39 V adjustable	33 ÷ 65 V adjustable	20 ÷ 60 V adjustable
Braking load	5 Ω	15 Ω	1.4 Ω or 5.6 Ω selectable
Switch-on current	5 A at 25 V	3 A at 50 V	10 ÷ 20 A typical
Idle consumption	10 mA	2 mA	5 mA
Hysteresis	10% typical	10% typical	10% typical
Overtemperature limit	190 °C	130 °C	160 °C

All three work the same way: a comparator watches the supply voltage and, when this exceeds the set threshold, switches the resistors in parallel with the supply, dissipating the excess energy. An LED on the board indicates that the resistor is conducting. All of them include built-in overtemperature protection.

The peak power dissipated at the switch-on threshold is V^2/R: it is the braking load values in the table that determine it. The **average** power over the cycle has to be compared instead with the continuous rating of the board: if the overtemperature protection trips regularly, the board is undersized for that cycle.

SCH109

SCH110

E.6. Choosing the switch-on threshold

The threshold has to be set above the no-load voltage of the power supply, otherwise the board would conduct all the time, and below the alarm threshold of the drive, otherwise it would step in too late.

The rule of thumb is to place it **10 ÷ 15% above the no-load voltage of the power supply**. With a 26 V supply, set the threshold to 30 V.

> On a 48 V system the usable window is narrow: only a few volts are left between the supply voltage and the 55 V alarm threshold. Set the threshold in the lower part of the range, roughly between 50 and 53 V, so that the board steps in before the drive goes into OVERVOLT.

E.7. Connection

The board is connected in parallel with the power supply, on the same conductors that feed the CN1 of the servo gearmotor.

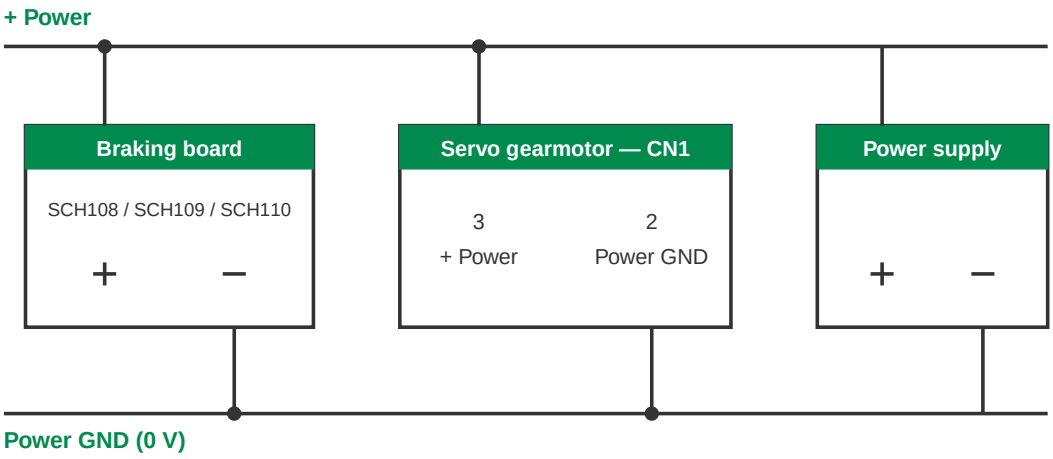


Figure 21. Connecting the braking board in parallel with the power supply

Terminal numbering is not the same on all boards: on SCH108 and SCH109 terminal 1 is the negative and 2 the positive, on SCH110 it is the other way round. Refer to the silk screen of the board you have in your hands, not to the terminal number.

Use the same conductor cross-section specified for the power wiring and keep the cable as short as possible: it carries impulsive currents of several amperes.

Mini Motor S.p.a.

Via E.Fermi, 5

42011 Bagnolo in Piano (RE), Italy

Tel. +39 0522 951889

Fax. +39 0522 952610

info@minimotor.com

www.minimotor.com