LAM Technologies
electronic equipment

DMS7xE

Integrated Stepper Motors
with EtherCAT fieldbus

EtherCAT®

User Manual

(Hardware rev. 1.00 Firmware rev. 0.14)

1 Introduction 7
1.1 Series 7
1.2 Terms, symbols and abbreviations 8
1.3 Documents 9
1.4 Contents of the pack 10
1.5 Intended use 10
1.6 Safety and use conditions 10
1.7 Warranty 12
2 Torque curves 13
2.1 NEMA23 13
2.2 NEMA34 14
3 Installation 16
3.1 Physical Characteristics 17
3.1.1 DMS7xEx2 (NEMA23 flange) 17
3.1.2 DMS7xEx3 (NEMA34 flange) 19
3.2 Connection 21
3.2.1 NEMA23, DMS7xEx2 series 21
3.2.2 NEMA34, DMS7xEx3 series 22
3.2.3 CN1 – Power Supply: AC models 23
3.2.4 CN1 – Power Supply, DC models 28
3.2.5 CN2 – Digital and analog I/O 34
3.2.6 CN3 CN4 – EtherCAT Bus 40
4 EtherCAT 41
4.1 ESI File 41
4.2 CoE Service 41
4.3 Synchronization 42
4.4 PDO 42
4.5 LED 43
5 Software Omni Automation IDE 45
5.1 Firmware update 46
5.2 Status and Diagnostics 47
5.3 Visualization and modification of the dictionary objects 49
6 Operation 53
6.1 Minimum settings 53
6.1.1 Running and idle current configuration 53
6.2 Saving and restoring of default values 55

6.3 Motor Holding Brakes 57
6.4 Closed-loop 58
6.5 Operation Modes 59
6.5.1 Profile position (pp) 60
6.5.2 Profile velocity mode (pv) 63
6.5.3 Profile torque mode (tq) 66
6.5.4 Homing mode (hm) 69
6.5.5 Cyclic synchronous position (csp) 76
6.5.6 Cyclic synchronous velocity (csv) 80
7 Errors and diagnostics 83
7.1 LED *On* 83
7.2 LED *En* 83
7.3 LED *Fault* 83
7.4 Errors 84
8 Objects Dictionary 88
8.1 Object with index between 1000_h and $1FFF_h$ 89
8.1.1 1000_h Device type 89
8.1.2 1001_h Error register 89
8.1.3 1008_h Manufacturer device name 90
8.1.4 1009_h Manufacturer hardware version 90
8.1.5 $100A_h$ Manufacturer software version 90
8.1.6 1010_h Store Parameters 91
8.1.7 1011_h Restore Default Parameters 92
8.1.8 1018_h Identity object 94
8.1.9 $10F1_h$ Error Settings 96
8.1.10 1600_h, 1601_h, 1602_h, 1603_h RPDO mapping parameter 97
8.1.11 $1A00_h$, $1A01_h$, $1A02_h$, $1A03_h$ TPDO mapping parameter 99
8.1.12 $1C00_h$ Sync Manager Communication Type 100
8.1.13 $1C12_h$ Sync Manger 2 PDO assign 102
8.1.14 $1C13_h$ Sync Manger 3 PDO assign 102
8.1.15 $1C32_h$ Sync Manager Output Parameter 103
8.1.16 $1C33_h$ Sync Manager Input Parameter 106
8.2 Objects with index between 2000_h and $5FFF_h$ 109
8.2.1 2010_h Fieldbus Setup 109
8.2.2 2120_h Error and Fault Setup 110
8.2.3 2210_h Digital I/O Setup 112

8.2.4 2310_h Motor Data 113
8.2.5 2330_h Motor Incremental Encoder Data 115
8.2.6 2338_h SSI Encoder Data 117
8.2.7 2360_h Holding Brake Setup 121
8.2.8 $23A0_h$ Current Regulation 123
8.2.9 2410_h Motion Setup 124
8.2.10 2440_h Position Loop 126
8.2.11 2480_h Errors Class Setup 127
8.2.12 2810_h Digital Inputs Assignment 128
8.2.13 2830_h Digital Outputs 0 Assignment 134
8.2.14 2831_h Digital Output 1 Assignment 135
8.2.15 2832_h Digital Output 2 Assignment 136
8.2.16 3080_h Sync monitoring 137
8.2.17 3120_h Fault and Error Status 141
8.2.18 3140_h Device Status 145
8.2.19 3210_h Digital Inputs Value 146
8.2.20 3230_h Digital Outputs Value 147
8.2.21 3250_h Analog Input 0 Value 148
8.2.22 3251_h Analog Input 1 Value 149
8.2.23 3310_h Motor Configuration 149
8.2.24 3312_h Motor Value 151
8.2.25 3330_h Motor Incremental Encoder Value 151
8.2.26 3338_h SSI Encoder Value 153
8.2.27 3360_h Holding Brake Control 155
8.2.28 3450_h Profile Velocity Configuration 157
8.2.29 3470_h Profile Torque Configuration 157
8.2.30 3490_h Interpolated Profile Configuration 158
8.2.31 $34B0_h$ Cyclic Synchronous Velocity Configuration 160
8.3 Object with index between 6000_h and $9FFF_h$ 161
8.3.1 $603F_h$ Error code 161
8.3.2 6040_h Controlword 161
8.3.3 6041_h Statusword 163
8.3.4 6060_h Modes of operation 164
8.3.5 6061_h Modes of operation display 165
8.3.6 6062_h Position demand value 165
8.3.7 6064_h Position actual value 165

8.3.8 6065_h Following error window 166
8.3.9 6066_h Following error time out 166
8.3.10 6067_h Position window 166
8.3.11 6068_h Position window time 167
8.3.12 $606B_h$ Velocity demand value 167
8.3.13 $606C_h$ Velocity actual value 167
8.3.14 $606D_h$ Velocity window 167
8.3.15 $606E_h$ Velocity window time 168
8.3.16 $606F_h$ Velocity threshold 168
8.3.17 6070_h Velocity threshold time 168
8.3.18 6071_h Target Torque 168
8.3.19 6072_h Max torque 169
8.3.20 6075_h Motor rated current 169
8.3.21 6076_h Motor rated torque 169
8.3.22 6077_h Torque actual value 170
8.3.23 6078_h Current actual value 170
8.3.24 $607A_h$ Target position 170
8.3.25 $607C_h$ Home offset 170
8.3.26 $607D_h$ Software position limit 171
8.3.27 $607F_h$ Max profile velocity 172
8.3.28 6081_h Profile velocity 172
8.3.29 6083_h Profile acceleration 172
8.3.30 6084_h Profile deceleration 172
8.3.31 6085_h Quick Stop deceleration 173
8.3.32 6087_h Torque slope 173
8.3.33 6098_h Homing method 174
8.3.34 6099_h Homing speeds 179
8.3.35 $60B8_h$ Touch probe function 180
8.3.36 $60B9_h$ Touch probe status 181
8.3.37 $60BA_h$ Touch probe 1 positive edge 182
8.3.38 $60BB_h$ Touch probe 1 negative edge 182
8.3.39 $60BC_h$ Touch probe 2 positive edge 182
8.3.40 $60BD_h$ Touch probe 2 negative edge 182
8.3.41 $60C2_h$ *Interpolation Time Period* 183
8.3.42 $60F4_h$ Following error actual value 184
8.3.43 $60F8_h$ Max slippage 184

8.3.44 $60FD_h$ Digital inputs 184
8.3.45 $60FF_h$ Target velocity 185
8.3.46 6502_h Supported drive modes 185
8.3.47 $67FE_h$ Version number 186

1 Introduction

The DMS7xE motors with integrated drives series is equipped with EtherCAT fieldbus and use vector bipolar driving technique.

The I/O equipment is complete and includes digital and analog inputs and outputs.

Some models have encoder, which allows the closed-loop moto control thus eliminating stall problems and improving efficiency.

The free software *Omni Automation IDE*, running under Windows (Windows 7, Windows 8.1 and Windows 10 32bit or 64bit), assists the commissioning of the product. For the connection to the PC the UDP30 interface is required.

1.1 Series

The series comprises 32 different models, different in functionalities, power supply and motor torque.

Power Supply Min…Max / Motor Torque 24Vdc Auxiliary Power Supply	3 Digital Inputs 3 Digital Inputs/Outputs 2 Analog Inputs	3 Digital Inputs 3 Digital Inputs/Outputs 2 Analog Inputs 1 Encoder A, B, I	Notes
20..50Vdc / 1.1Nm	**DMS71E4241**	**DMS72E4241**	NEMA 23 DC Power Supply
20..50Vdc / 1.8Nm	**DMS71E4264**	**DMS72E4264**	
20..50Vdc / 3.0Nm	**DMS71E4271**	**DMS72E4271**	
24..90Vdc / 1.1Nm	**DMS71E7241**	**DMS72E7241**	
24..90Vdc / 1.8Nm	**DMS71E7264**	**DMS72E7264**	
24..90Vdc / 3.0Nm	**DMS71E7271**	**DMS72E7271**	
20..50Vdc / 4.4Nm	**DMS71E4321**	**DMS72E4321**	NEMA 34 DC Power Supply
20..50Vdc / 6.8Nm	**DMS71E4331**	**DMS72E4331**	
20..50Vdc / 9.2Nm	**DMS71E4350**	**DMS72E4350**	
20..50Vdc / 12.1Nm	**DMS71E4360**	**DMS72E4360**	
24..90Vdc / 4.4Nm	**DMS71E7321**	**DMS72E7321**	
24..90Vdc / 6.8Nm	**DMS71E7331**	**DMS72E7331**	
24..90Vdc / 9.2Nm	**DMS71E7350**	**DMS72E7350**	
24..90Vdc / 12.1Nm	**DMS71E7360**	**DMS72E7360**	
16..36Vac / 4.4Nm	**DMS71E4321A**	**DMS72E4321A**	NEMA 34 AC Power Supply
16..36Vac / 6.8Nm	**DMS71E4331A**	**DMS72E4331A**	
16..36Vac / 9.2Nm	**DMS71E4350A**	**DMS72E4350A**	
16..36Vac / 12.1Nm	**DMS71E4360A**	**DMS72E4360A**	
20..65Vac / 4.4Nm	**DMS71E7321A**	**DMS72E7321A**	
20..65Vac / 6.8Nm	**DMS71E7331A**	**DMS72E7331A**	
20..65Vac / 9.2Nm	**DMS71E7350A**	**DMS72E7350A**	
20..65Vac / 12.1Nm	**DMS71E7360A**	**DMS72E7360A**	

1.2 Terms, symbols and abbreviations

To indicate features common to a whole group of products the character "x" is used in place of any other character. For example, the term DMS7xE4350 implies the models DMS71E4350 and DMS72E4350.

The terms *manual* and document have the same meaning, moreover the words integrated drive, *integrated motor, motor, device* and *product* always refer to the DMS7xE Series.

In the manual some symbols are used to underline necessary topics of particular concern or deserving interest. The meaning of each one of them is detailed here below:

It refers to a dangerous condition that must be accurately evaluated and avoided. Failing to follow instructions marked with this symbol can be cause of serious damages to people, animals and things.

It draws the attention to important issues that, if not understood or implemented, may affect the good functioning of the product.

It highlights a valuable feature or functionality of the product that is difficult to find elsewhere or shows a shortcut to reach a target.

The characteristic names of registers, parameters, objects, functions, modes, etc. are always provided in English to avoid confusion or doubts in the interpretation.

To describe the data type of registers, parameters, objects, etc. abbreviations are used. The following table shows their meaning, as well as the range of values allowed for each type:

Abbreviation	Bits	Description	Min	Max
bool	1	Boolean	0	1
i8	8	Signed Byte	-128	127
i16	16	Signed Word	-32.768	32767
i32	32	Signed Integer	-2.147.483.648	2.147.483.647
u8	8	Unsigned Byte	0	255
u16	16	Unsigned Word	0	65.535
u32	32	Unsigned Integer	0	4.294.967.295
f32	32	Floating Point	-3.402823e38	3.402823e38
str	---	String		

The symbols ↑ and ↓ are used to indicate respectively the rising edge (0->1 transition) or the falling edge (1->0 transition) of a digital signal or bit.

Other abbreviations used:

Abbreviations	Description
AC, ac	Alternate current
AI	Analog input
AO	Analog output
DC, dc	Direct current
hm	Homing mode
DIO	Digital input / output
DI	Digital input
DO	Digital output
pp	Position mode
pv	Velocity mode
PDO	Process Data Object
RMS, rms	Root mean square
RO, ro	Read-only
WO, wo	Write-only
rPreOPw	Read Pre-Operational Write
RPD	Receive Process Data
RW, rw	Read-write
TPD	Transmit Process Data
tq	Torque mode

1.3 Documents

The present manual applies to the standard series of DMS7xE integrated motors with Hardware and Firmware revisions as shown on the cover. Customized products or with a different Hardware or Firmware revision may have features and behaviors different to what herein described. It is technician and user's responsibility to use the documents appropriate to the products used updated to the latest available version, also by sending a specific request to your supplier or by writing to support@lamtechnologies.com Hardware or Firmware revisions less than 1.00 (for example 0.24) indicate the prototype products.

LAM Technologies reserves the right to modify at any moment the present document without obligation to give prior notice. This includes, for example, but not limited to, diagrams, images, organization of chapters, technical specifications of the product, features, warranty, etc.

The information contained herein replace any previously issued revision.

This document contains reserved and proprietary information. All rights are reserved. It may not be copied, disclosed or used for any purposes not expressly authorized by LAM Technologies.

The manual has been compiled with the intention to make it clear and complete. LAM Technologies, in order to continuously improve its products and documents quality, will appreciate any suggestion, be in change, addition or else.

LAM Technologies is a registered trademark.

1.4 Contents of the pack

The device is supplied without cables that must be purchased separately.

Technical documentation and software can be downloaded from the website www.lamtechnologies.com or may be required to your supplier or by writing to support@lamtechnologies.com.

1.5 Intended use

Each product described in this manual is a component part intended to be incorporated into a product for professional use and must not be installed or used by the consumer or private user.

The products described in this manual can only be used in industrial applications. It is expressly forbidden to use the products described in this manual in military, aeronautical, aerospace, nuclear, medical applications and any other sector other than industrial.

1.6 Safety and use conditions

This manual is intended for technicians specialized in automation or similar disciplines. In case the arguments, the terms, or the concepts expressed should not be clear you can contact our technical support writing to support@lamtechnologies.com. It is prohibited to use the products herein described if you are not sure to have understood their features and how to use and install them.

ATTENTION
The following are safety warnings and practices of primary importance that need to be fully understood and applied by the user. The user who does not fully understand the content below, or was not able to apply it totally, should not use the product for any reason.

The devices described in this manual are components. The user is responsible of the installation and use of the product that must be used only if in compliance with the rules and regulations in force. Furthermore, the user must have the technical skills needed to fully understand the features, the setting parameters and the instructions given herein. The user must also apply all the laws and specific rules of the Country and/or application in which the product is used.

The user must make the housing of the motor with integrated drive inaccessible when the same is powered on. The user must also consider that, because of the capacitors inside the motor, it is necessary to wait at least 30 seconds from the power off before accessing the drive. According to the external capacitors eventually mounted on the power supply circuit, it is possible that the wait time is considerably longer.

During operation the product generates heat that can raise the temperature of certain parts (the heat sink for example, but not only) to values which can cause burns. Such condition persists for a long time even after the product has been turned off. The user must provide protections and appropriate warnings as well as instructing the user, the technical support and maintenance staff. The user must also describe this condition in the service manual of the finished product.

The high performance integrated motor is able to generate strong accelerations, with high motor torque. It is therefore essential to never touch the mechanical parts with the device powered on. The user must prepare the application so that this condition is always granted.

Because of an incorrect wiring, incorrect configuration or else, the integrated motor can command to the load unexpected movements. Before supplying the motor, assure that an unexpected movement of the motor does not represent danger for people, animals and things.

The power supply of the product must be isolated from the mains line (for example through a transformer). In series to the power supply circuit, the user must always provide a protective fuse.

In normal working conditions, many control signals are isolated from the power supply; however consider that, under fault conditions, these lines can reach the same potential of the power supply and it is therefore necessary to design the application giving attention to this eventuality.

The EMC interferences can cause unexpected behavior in the whole application, therefore it is essential to minimize the spread of the EMC interferences with the use of a shielded cable, through a correct connection of the shields and of the equipotential points, etc. Furthermore, it is important to carry out a complete commissioning test after installation is complete.

The product could be permanently damaged by corrosive substances (such as gas, salts, etc.), liquid or corruptive dusts. Even a long and strong exposure to strong vibrations can cause its damage.

In some fault conditions, the integrated motor can start sparks and fire. The housing and the components placed in the nearby must be chosen to tolerate this eventuality and to avoid the spread of fire.

The products must never be used in explosive atmospheres (Ex areas).

The products were not designed for use in vital support systems, equipment used in hazardous environments, nuclear control systems or other similar applications, or where the failure of the product, even in part, can cause death or damage to people, animals or things, or cause economic loss. The user not able to ensure this condition should not use the products described in this manual.

Do not dismantle the product, do not try and repair it and do not modify it unless expressly authorized in writing by LAM Technologies.

Failure to follow the indications included in this manual can cause permanent damage to the product. For example, to power supply the product with voltage higher than the maximum one allowed, to invert the polarity of the same, to connect or disconnect the integrated motor with the power supplied, etc. are cause of permanent damage.

Even if the product has been designed and manufactured with extreme care, there is always the possibility that in certain circumstances it may malfunction. Therefore, for no reason, the products described in this manual can be used in life support application and in all those cases in which the unexpected failure of the product could be cause of death or damage to people, things, animals or cause economic loss.

LAM Technologies reserves the right to modify the aspect, the technical and functional specifications, the construction characteristics and the prices of products at any time, with immediate effect, without obligation to give prior notice to anyone.

Any request, restriction or condition imposed by the buyer is void unless expressly approved in writing by LAM Technologies. For anything not expressly regulated by this document, the Italian Laws apply. For any dispute, as an example and not limited to the products, the documentation, the technical assistance, the supply execution, etc. the Court of Florence will have exclusive jurisdiction.

The user who disagrees with the conditions of use of the products here described, should not use them

1.7 Warranty

LAM Technologies warrants the products described in this manual against defects in materials or workmanship for a period of 12 months from the delivery of goods. This warranty does not apply to defects, damages caused by improper use, incorrect installation or inadequate maintenance. This warranty does not apply in case the products are received modified or integrated with other parts and/or products not expressly authorized or provided for by LAM Technologies. This warranty does not apply also in case the product's label has been removed or modified.
Any request for assistance must be sent to the supplier of the product. In case of direct purchase from LAM Technologies, a returned material authorization number (RMA) must be obtained, before shipping the device, from support@lamtechnologies.com clearly specifying the product's code, the serial number, the problem found and the assistance required. The RMA number must be clearly written on each shipping document otherwise the parcel could be rejected. The customer shall be responsible for the packaging and shipping of the defective product to LAM Technologies and shipment must be made charges prepaid. The product inspected, repaired or replaced will be available to be collected at LAM Technologies'. In case of repair under warranty LAM Technologies can, at its own discretion, repair or replace the product. No cost for material or service will be charged in case of repair under warranty. During inspection or repair of the product, any configuration or programming stored in the device may be removed. Therefore, before sending a product for service, always ensure that you have all the information (configuration, program source, etc.) necessary to reconfigure or reprogram the device once it is returned.

The above warranty does not apply to the software. LAM Technologies shall not be liable for any direct or indirect damages such as, but not limited to, costs of removal and installation, lost profits, deriving from the use or the impossibility to use the software. The user who disagree with or cannot accept what stated herein, should not use or install the software.

2 Torque curves

The torque curves characteristic of each model are shown below.

The power supply voltage affects the motor torque, in particular to a high power supply voltage corresponds a more constant torque of the motor as the speed increases. On the contrary, a low power supply voltage produces a torque curve that rapidly decreases as the speed increases.

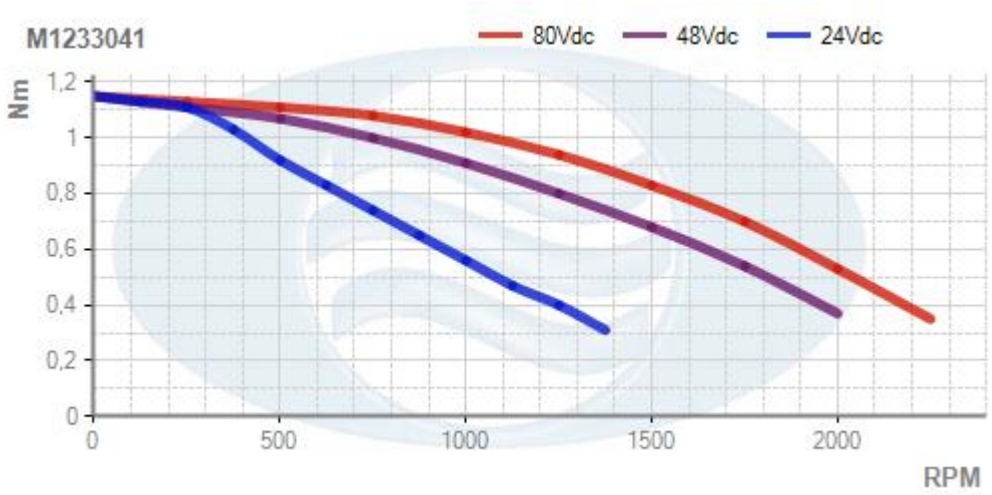
For each motor with integrated electronics, more different torque curves are shown, each one corresponding to a different power supply voltage. The voltage values are expressed in continuous (Vdc), for the devices with AC power supply (Vac) the following formula must be applied to identify the corresponding curve:

Vdc = (Vac * 1,41) – 2.5

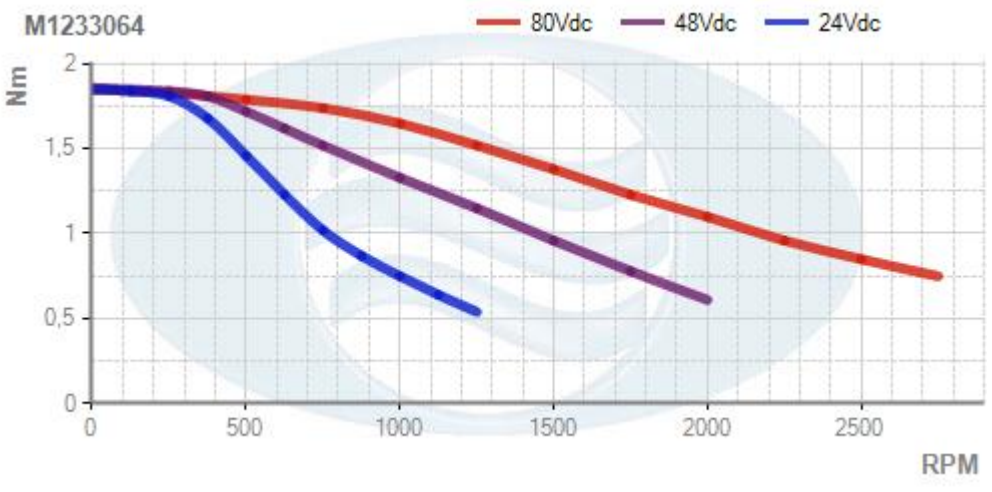
For example. if the AC power supply is 36Vac, we will use the torque curve that is closest to (36 * 1.41) -2.5 = 48.3, which is the curve corresponding to a power supply of 48Vdc.

2.1 NEMA23

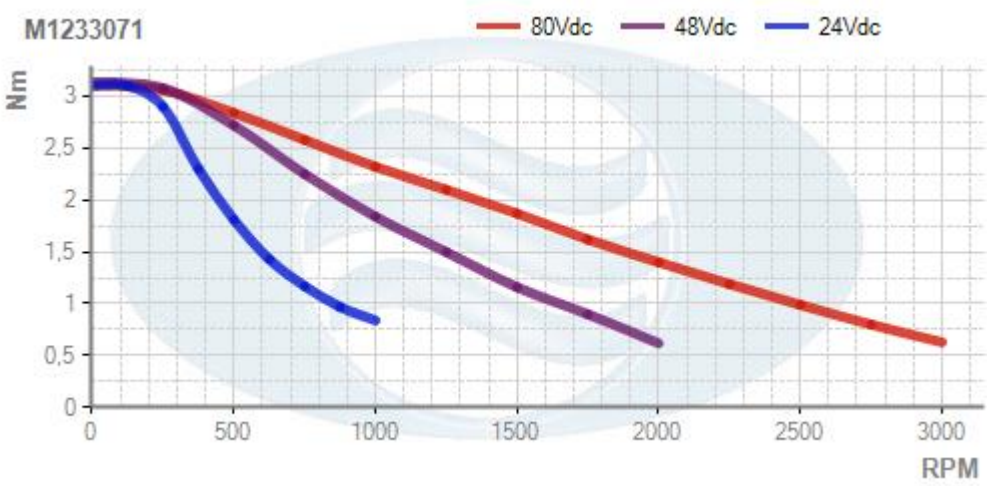
DMS7xEx241



DMS7xEx264

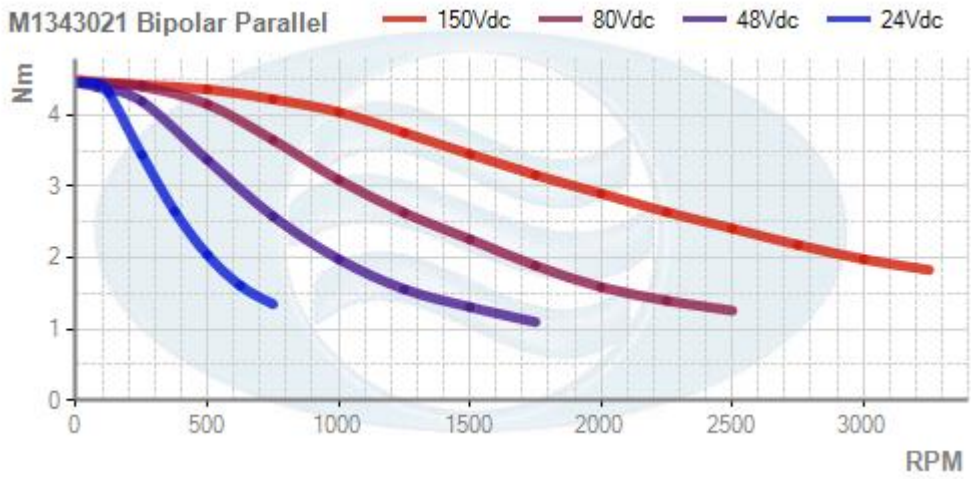


DMS7xEx271

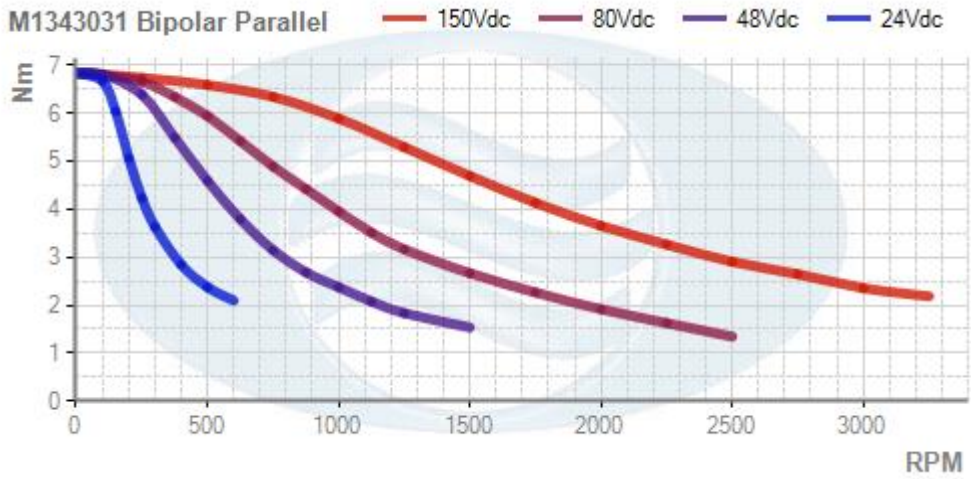


2.2 NEMA34

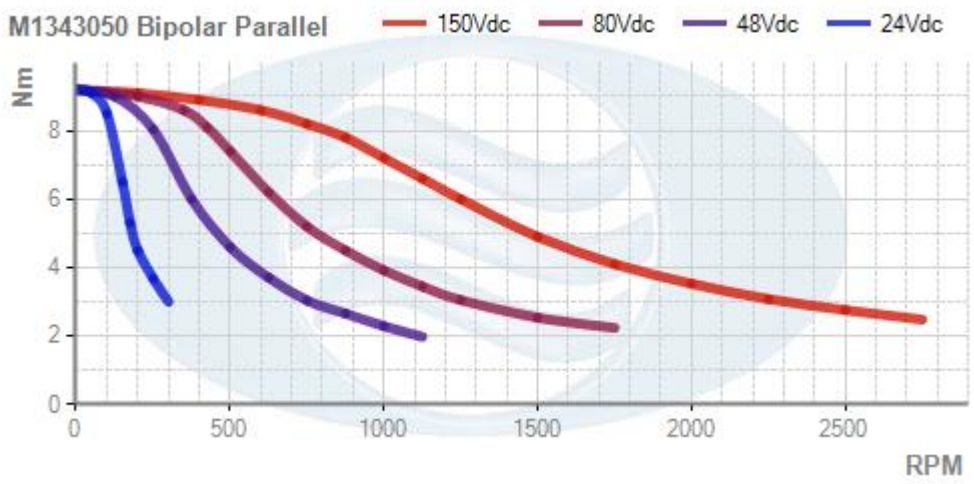
DMS7xEx321x



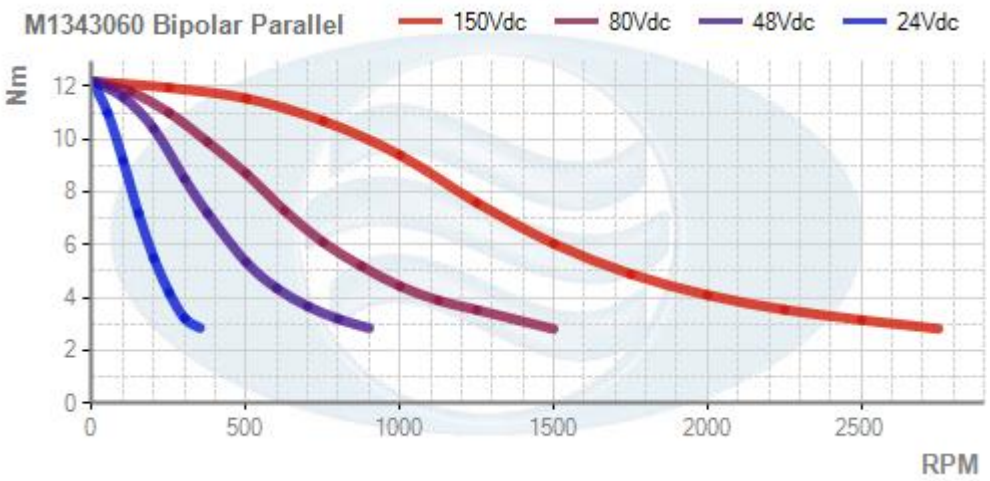
DMS7xEx331x



DMS7xEx350x



DMS7xEx360x



3 Installation

ATTENTION, the DMS7xE series integrated drive motors are component parts. The user is responsible for the choice, the installation and use of the product within his own equipment.

ATTENTION, the products described in this document can only be used if compatible with the regulations and standards in force applicable to the device and the country of use. Furthermore, the use must also have the technical skills necessary to fully understand the features, the setting parameters and the indication given herein.

ATTENTION, the user must apply all the rules and regulations specific to the country and/or application in which the product is used.

ATTENTION, the use of the products described in this document is permitted only in an industrial contest and the installation must be executed by expert professional personnel and after having read and understood the instructions contained in this manual.

ATTENTION, the products described in this documentation are for fixed installation and must be mounted on static supports. Occasional or periodic accelerations may cause product malfunction and/or failure.

ATTENTION, the motor temperature measured at the hottest point must not exceed 90°C.

ATTENTION, the connection cables must be blocked in order their weight does not weigh on the integrated motor and no pressure, tractions or twisting is exerted on it or on the connectors.

3.1 Physical Characteristics

The products must always be stored and used in compliance with the following environmental limits:

Environmental storage conditions					
Symbol	Description	Unit	Value		
			Min	Nominal	Max
SCa	Altitude	m			4000
SCt	Temperature	°C	-20		70
SCh	Non-condensing humidity	%	5		90

Environmental operating conditions					
Symbol	Description	Unit	Value		
			Min	Nominal	Max
FCa	Altitude	m			2000
Idr	Phase current downgrading every 1000m above the FCa value	%		10	
FCt	Temperature	°C	0		40
FCh	Non-condensing humidity	%	5		90

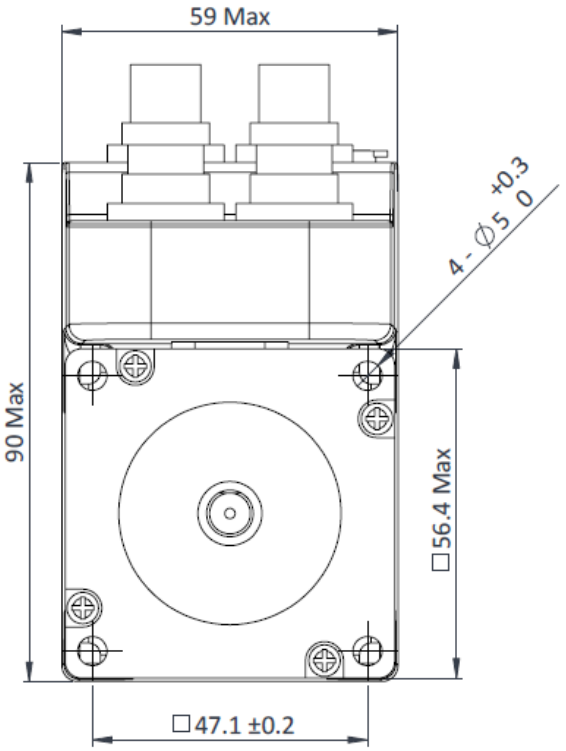
The protection rating is IP60 except for the motor shaft, which has a protection rating of IP40.

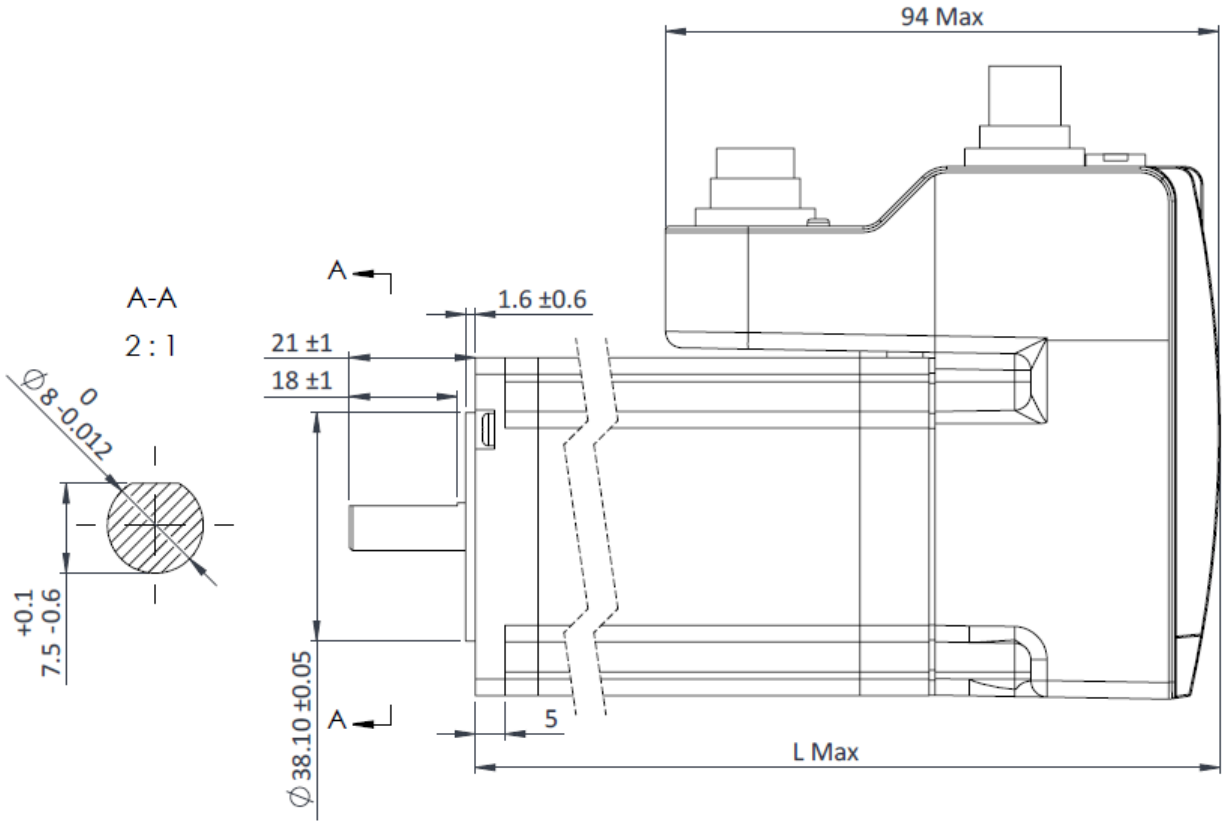
3.1.1 DMS7xEx2 (NEMA23 flange)

All the models of motors with integrated drives with NEMA23 flange share the same dimensions with the sole exception of the length *L Max* which changes according to the maximum torque that can be delivered by the motor.

The following table shows the *L Max* value and the weight of each device:

Model	L Max	Weight
DMS7xEx241	106mm	1000g
DMS7xEx264	126mm	1400g
DMS7xEx271	150mm	1850g


59 Max
4 - ∅5 +0.3 0
90 Max
□56.4 Max
□47.1 ±0.2

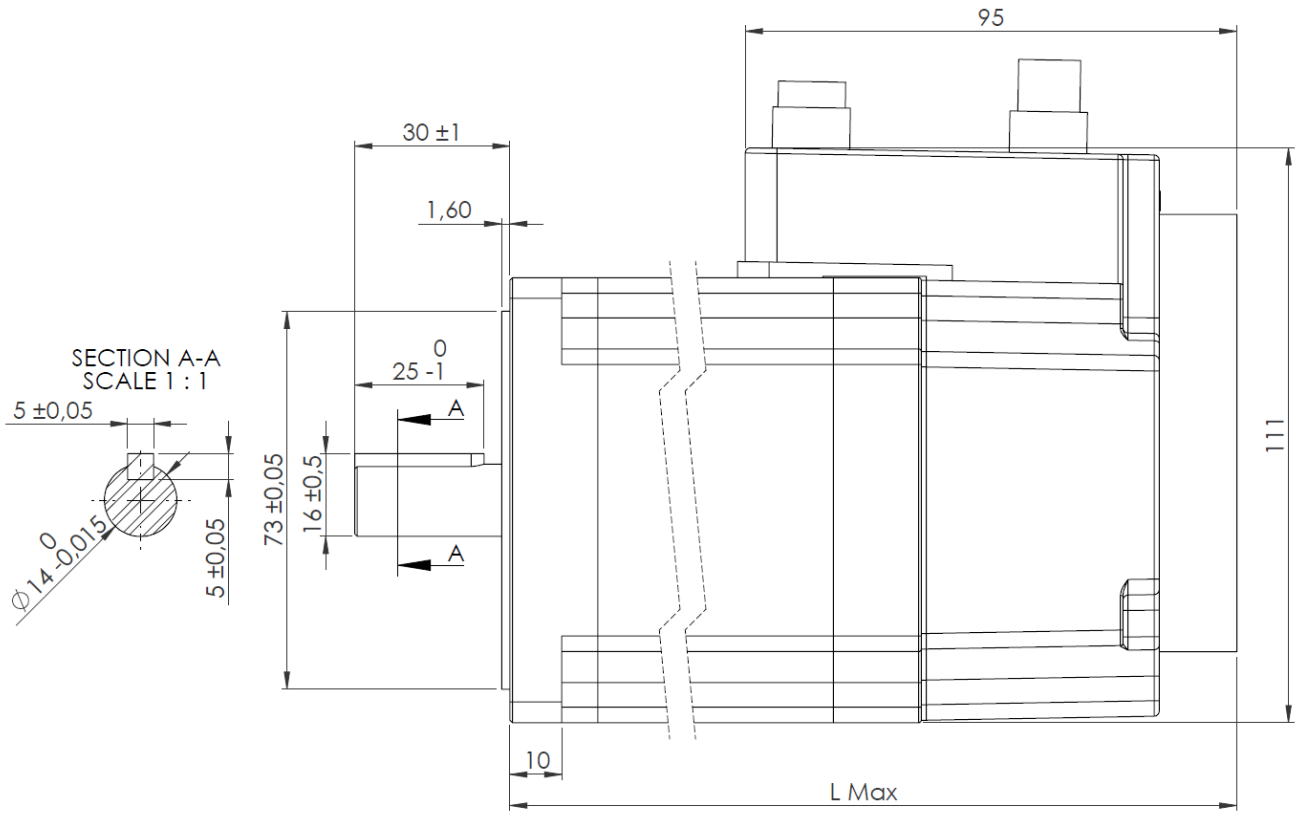

94 Max
A
1.6 ±0.6
21 ±1
18 ±1
A-A
2 : 1
∅8 0 -0.012
+0.1
7.5 -0.6
∅38.10 ±0.05
A
5
L Max

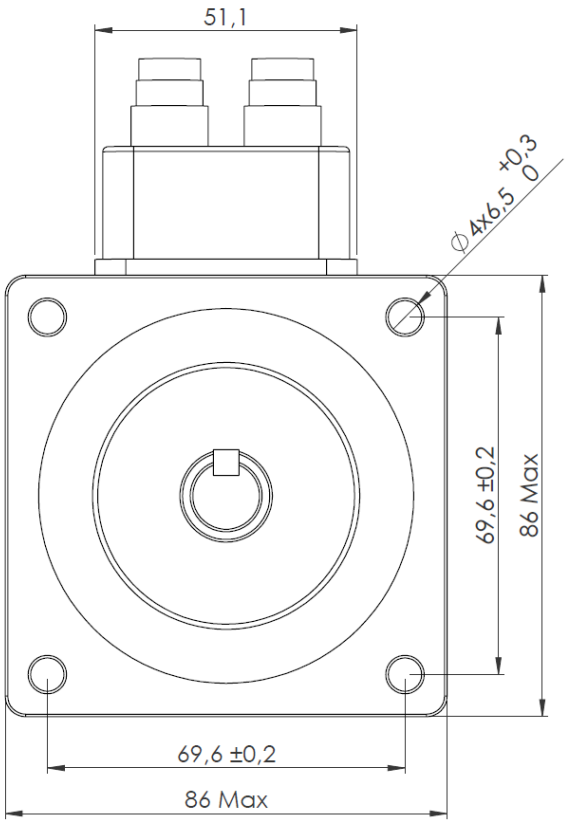
3.1.2 DMS7xEx3 (NEMA34 flange)

All the models of motors with integrated drive with NEMA34 flange share the same dimensions with the sole exception of the length *L Max* that changes according to the maximum torque that can be delivered by the motor.

The following table shows the *L Max* value and the weight of each device:

Model	L Max	Weight
DMS7xEx321x	141mm	2700g
DMS7xEx331x	161mm	3400g
DMS7xEx350x	189mm	4500g
DMS7xEx360x	213mm	5400g



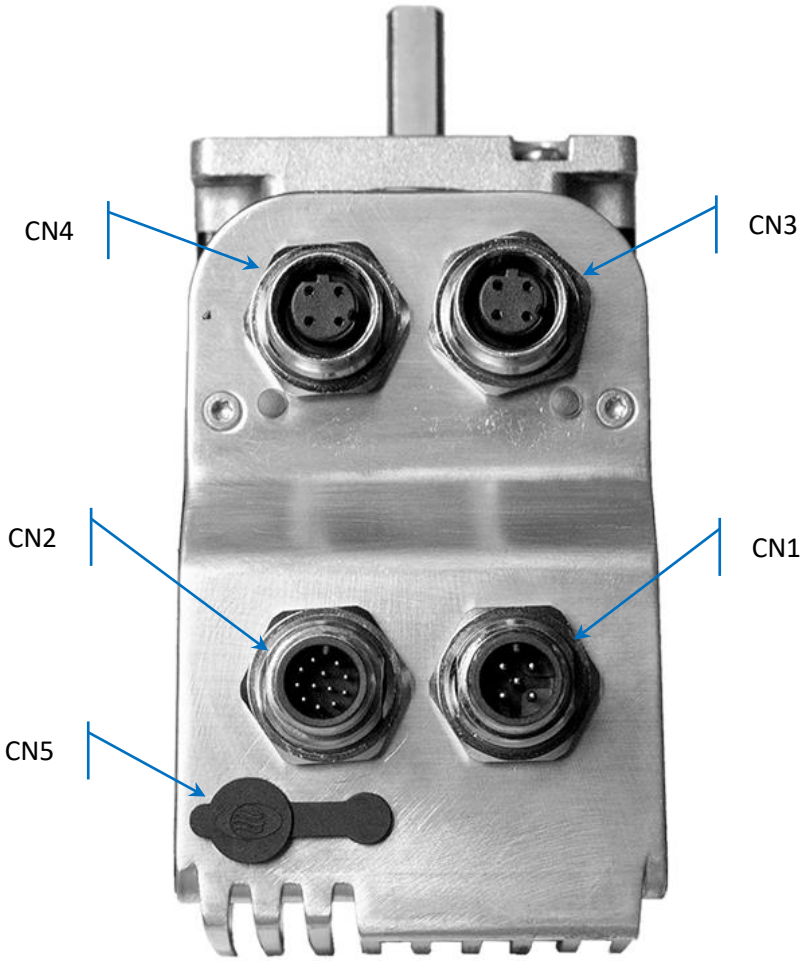

51,1
⌀4x6,5 +0,3 0
69,6 ±0,2
86 Max
69,6 ±0,2
86 Max

3.2 Connection

The DMS7xE Series of motors with integrated drive is provided with 4 M12 connectors reserved to the power supply, the connection of the digital and analog I/Os and of the EtherCAT bus.

Connector	Type		Function
	DMS7xEx2 (NEMA23 flange)	DMS7xEx3 (NEMA34 flange)	
CN1	M12 Coding A Male 5ct	M12 Coding L Male 5ct	Power supply
CN2	M12 Coding A Male 12ct	M12 Coding A Male 12ct	Digital and analog I/O
CN3	M12 Coding D Female 4ct	M12 Coding D Female 4ct	EtherCAT In
CN4	M12 Coding D Female 4ct	M12 Coding D Female 4ct	EtherCAT Out
CN5			DUP port for the connection to a UDP30

3.2.1 NEMA23, DMS7xEx2 series



The CN5 connector is equipped with a protective rubber cap.

3.2.2 NEMA34, DMS7xEx3 series

The CN5 connector is equipped with a protective rubber cap.

3.2.3 CN1 – Power Supply: AC models

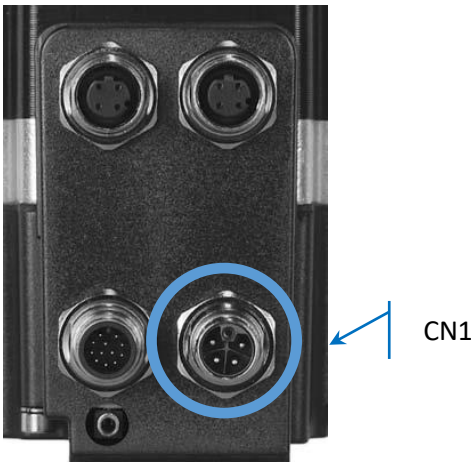
The DMS7xEx3 series integrated motors with NEMA 34 flange are also available in the AC supply version and are identified by the letter **A** placed at the end of the code (ex. DMS72E7331**A**). They integrate a rectifier bridge and the capacitors necessary to rectify and filter the AC power supply voltage.

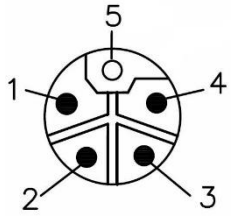
Therefore, this series of products does not need an external power supply and can be directly connected to the output of a transformer with adequate voltage.

The overcoming of the *Vacbrk* voltage limit can permanently damage the device.

Do not supply the motor with integrated drive before the wiring is complete.

Do not connect the device with the power supply turned on.



CN1 – Power supply		
Pin	Description	M12 Coding L Male 5ct
1	Vac, AC power supply input	
2	Vac, AC power supply input	
3	+24V Auxiliary power supply	
4	0V (GND) Auxiliary power supply	
5	PE, earth (internally connected to the motor housing)	View from top

3.2.3.1 Auxiliary Power Supply

The auxiliary power supply is optional and, if provided, it allows to maintain the control section of the integrated motor supplied, even if the power supply is removed (for example, to secure the application).

Keeping supplied the logic section of the integrated motor, the signals, the fieldbus and the encoder reading (for DMS72E models), that allows to track the position of the motor even if not supplied and moved manually, are maintained active.

The auxiliary power supply must be within the range shown in the table below:

Symbol	Description	Unit	Value		
			Min.	Nominal	Max
V24	DC auxiliary supply voltage	V	20		35
V24brk	Permanent damage voltage	V	-0.5		40

3.2.3.2 Power Supply

The Vac power supply must be with a sinusoidal waveform and voltage according to the values shown in the following table:

Model	Symbol	Description	Unit	Value		
				Min	Nominal	Max
DMS7xE4xxxA	Vac	AC supply voltage	Vac	16	32	36
	Vacbrk	Voltage causing permanent damage	Vac			42
DMS7xE7xxxA	Vac	AC supply voltage	Vac	20	48	65
	Vacbrk	Voltage causing permanent damage	Vac			75

The motor with integrated drive has protections that intervene when the power supply voltage has a value that no longer guarantees correct operation.

Model	Symbol	Description	Unit	Value		
				Min	Nominal	Max
DMS7xE4xxxA	Vacl	Under voltage protection intervention threshold	Vac		15	
	Vach	Over voltage protection intervention thresholdOver voltage protection intervention	Vac		38	
DMS7xE7xxxA	Vacl	Under voltage protection intervention threshold	Vac		18	
	Vach	Over voltage protection intervention threshold	Vac		68	

For the connection with the transformer it is necessary to use a conductor with section adequate to the chosen motor's size. The following table resumes the cable section suggested for each motor with integrated electronics:

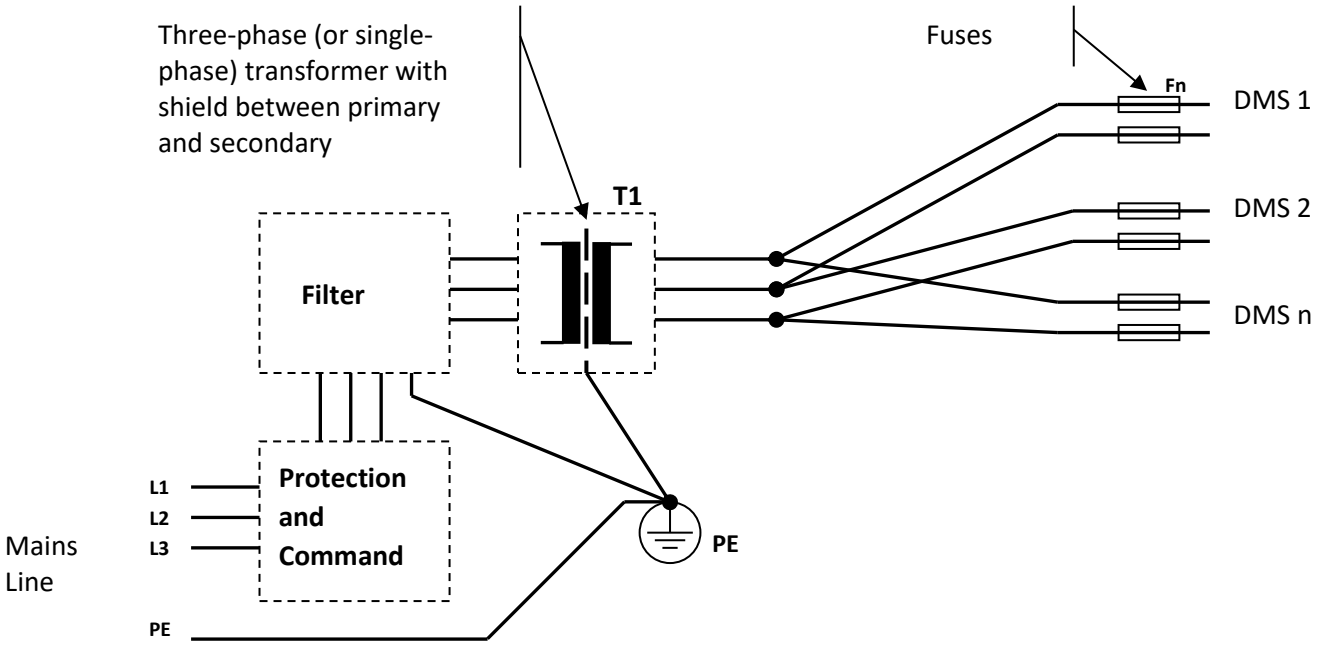
Model	Section (mm^2)
DMS7xE4321A	1
DMS7xE4331A	1.5
DMS7xE4350A	1.5
DMS7xE4360A	1.5
DMS7xE7321A	1
DMS7xE7331A	1.5
DMS7xE7350A	1.5
DMS7xE7360A	1.5

It is recommended not to place the power supply cable nearby the signal ones.

While choosing the transformer secondary voltage it is important to take into account the maximum mains line fluctuation expected in the worst operative conditions, the maximum open load voltage and the minimum full load voltage and to ensure that the maximum and minimum values, results of the combination of these components, are within the maximum and minimum voltage values specified for the chosen integrated motor model.

The power that the transformer must handle is given by the one absorbed by the load (dependent therefore on the torque required to the motor as well as the rotation speed), by the efficiency of the motor and the efficiency of the drive.

The following is an example of base connection.



The above scheme includes a three-phase transformer (note the distribution of the drives on the three phases). If necessary, it is also possible to use a single-phase transformer.

Also note that the wiring must be star-like, where the earth connections of the various components ends in one only point electrically connected to the metal chassis and the earth of the plant.

Do not connect the secondary of the transformer to earth otherwise there is the risk to permanently damage the motor with integrated drive.

As shown in the scheme, it is necessary to put in series to the transformer primary winding a filter able to stop the emissions coming from the device and/or present on the mains line. Furthermore, the filter must be able to support the maximum power required by the application plus the transformers losses.

The noise reduction level that the filter must guarantee can vary according to the laws applied to the field to which the application and/or installation belong.

The producers of filters SHAFFNER and CORCOM can represent a good reference to find the right filter.

It is mandatory to provide on each phase of the transformer primary winding a fuse able to intervene in case of short circuit or malfunctioning. It is also mandatory to use a fuse on each device's power supply conductor.

The following table shows the suggested value for some components according to the number of motors with integrated drive present in the application.

The calculation considers also an oscillation of the mains supply voltage included in the +10/-20% range.

Model	Fuses Fn (A T)	Number of motors	Secondary T1 (Vac)	Power T1 (VA)	Current D1 (Arms)
DMS7xE4321A	8	1	32	200	25A
		2		400	25A
		3		600	25A
		4..5		1000	35A
		6..8		1400	50A
DMS7xE4331A	8	1	32	300	25A
		2		600	25A
		3		900	35A
		4..5		1500	50A
		6..8		2100	80A
DMS7xE4350A	8	1	32	300	25A
		2		600	25A
		3		900	35A
		4..5		1500	50A
		6..8		2100	80A
DMS7xE4360A	12	1	32	300	25A
		2		600	25A
		3		900	35A
		4..5		1500	50A
		6..8		2100	80A

DMS7xE7321A	8	1	48	400	25A
		2		800	25A
		3		1200	35A
		4..5		2000	50A
		6..8		2800	80A
DMS7xE7331A	8	1	48	500	25A
		2		1000	35A
		3		1500	35A
		4..5		2500	50A
		6..8		3500	80A
DMS7xE7350A	8	1	48	500	25A
		2		1000	35A
		3		1500	35A
		4..5		2500	50A
		6..8		3500	80A
DMS7xE7360A	12	1	48	500	25A
		2		1000	35A
		3		1500	35A
		4..5		2500	50A
		6..8		3500	80A

The working voltage of the T1 transformer primary winding must be chosen according to the mains supply voltage available in the installation. The transformer must have a shield between primary and secondary windings which must be connected to earth with a short and not inductive connection. The secondary winding voltage is meant open load, with the primary winding supplied at the nominal voltage.

In the application with more than one motor with integrated drive, if the working cycle is not simultaneous, the power of the transformer can be considerably reduced. In some cases this can also be made when the motors' speed is limited.

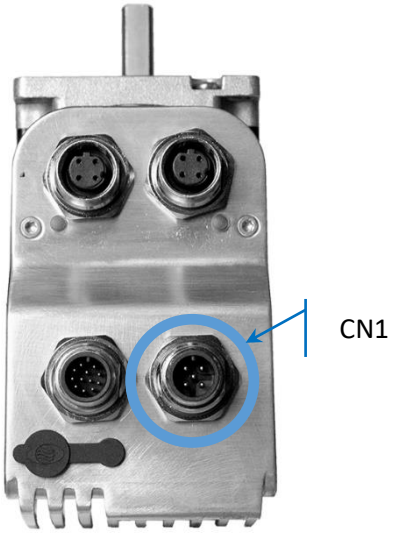
The set composed by the filter and the transformer must be used only to supply the power voltage to the drives. It is recommended not to derive other supplies from any of these parts. On the contrary, it is suggested to obtain auxiliary supplies using directly the mains line upstream of the filter.

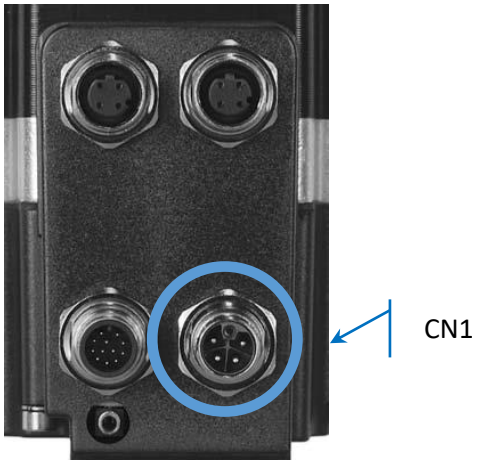
3.2.4 CN1 – Power Supply, DC models

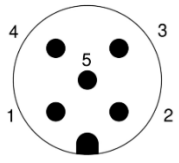
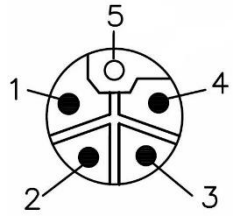
Reverse polarity connection can permanently damage the motor with integrated drive as well as the exceeding of the *Vpbrk* voltage limit.

Do not supply the motor before the wiring is complete.

Do not connect the drive with integrated drive with the power supply on.





CN1 – Power supply			
Pin	Description	DMS7xEx2	DMS7xEx3
1	+Vp, positive DC supply voltage	M12 Coding A Male 5ct	M12 Coding L Male 5ct
2	-Vp (GND), negative DC supply voltage		
3	+24V Auxiliary power supply		
4	0V (GND) Auxiliary power supply		
5	PE, earth (internally connected to the motor housing)	 View from top	 View from top

The negative of the –Vp power supply and the ground of the 0V auxiliary power supply are connected together inside the motor.

3.2.4.1 *Auxiliary Power Supply*

The auxiliary power supply is optional and, if provided, allows to maintain the control section of the integrated motor powered, even if the power supply is removed (for example, to secure the application).

By keeping supplied the logic section of the integrated motor, the signals, the fieldbus and the encoder reading (for DMS72E models), that allows to track the position of the motor even if not supplied and moved manually, are maintained active.

The auxiliary power supply must be within the range shown in the table below:

Symbol	Description	Unit	Value		
			Min.	Nominal	Max
V24	DC auxiliary supply voltage	V	20	24	35
V24brk	Permanent damage voltage	V	-0.5		40

3.2.4.2 *Power Supply*

The VP power supply must be supplied according to the values specified in the following table:

Model	Symbol	Description	Unit	Value		
				Min.	Nominal	Max
DMS7xE4xxx	Vp	DC supply voltage	V	20	48	50
	Vpbrk	Voltage causing permanent damage	V	-0.5		60
DMS7xE7xxx	Vp	DC supply voltage	V	24	70	90
	Vpbrk	Voltage causing permanent damage	V	-0.5		105

The motor with integrated drive has protections that intervene when the supply voltage has a value such as not to ensure a correct operation.

Model	Symbol	Description	Unit	Value		
				Min	Nominal	Max
DMS7xE4xxx	Vpl	Under voltage protection intervention threshold	V		18	
	Vph	Over voltage protection intervention threshold	V		52	
DMS7xE7xxx	Vpl	Under voltage protection intervention threshold	V		22	
	Vph	Over voltage protection intervention threshold	V		96	

If the distance between the motor and the power supply is more than 2m, it is necessary to place near the integrated motor (less than 10cm) an electrolytic capacitor whose minimum characteristics are specified in the following table:

Model	Voltage (V)	Capacity (µF)
DMS7xE4241, DMS7xE4264, DMS7xE4271, DMS7xE4321, DMS7xE4331, DMS7E4350	63	1000
DMS7xE4360	63	2200
DMS7xE7241, DMS7xE7264, DMS7xE7271, DMS7xE7321, DMS7xE7331, DMS7xE7350	100	1000
DMS7xE7360	100	2200

To connect the transformer, it is necessary to use a conductor with section adequate to the chosen size of the motor. The following table resumes the cable section suggested for each motor with integrated drive:

Model	Sectione (mm^2)
DMS7xEx241	0.,5
DMS7xEx264	0.5
DMS7xEx271	0.5
DMS7xEx321	1
DMS7xEx331	1.5
DMS7xEx350	1.5
DMS7xEx360	1.5

We recommend avoiding placing the power supply cable near the signal ones.

The power supply can be regulated or unregulated type.

The use of a regulated power supply ensures a constant output voltage, immune to mains line fluctuations, and this allows to supply the motor with integrated drive with voltage values near to the maximum allowed ones with an immediate benefit in terms of torque supplied by the motor at high speed. The disadvantage of the regulated power supplies is their cost.

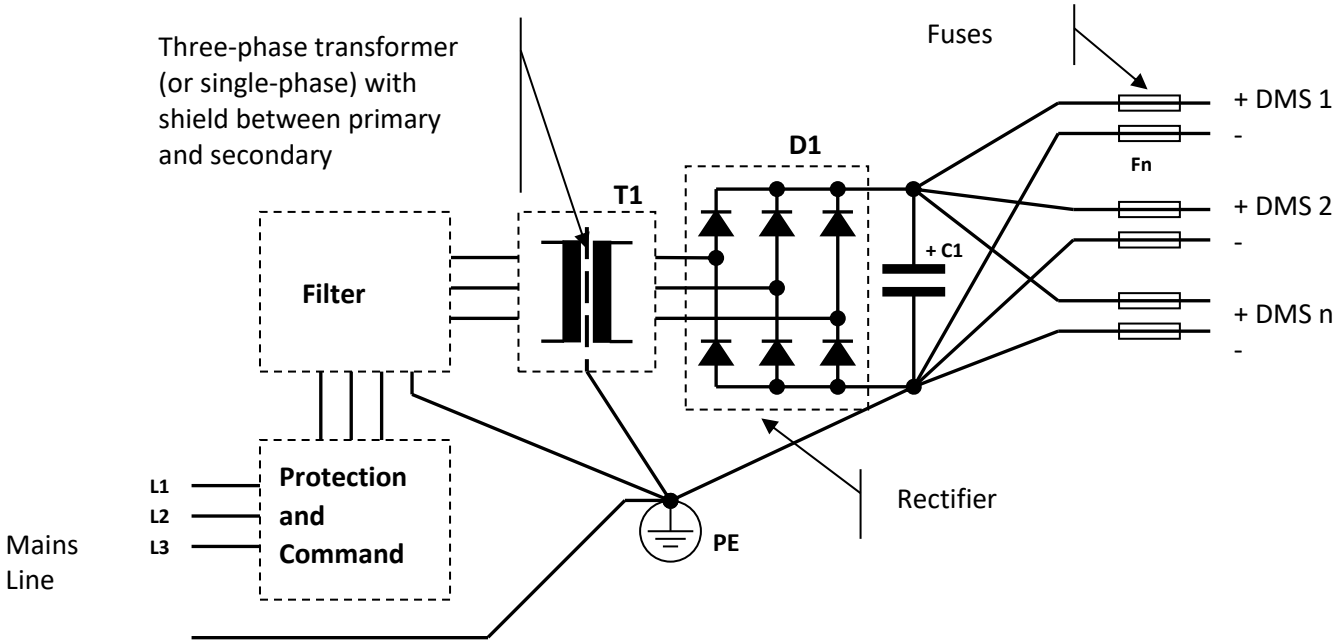
An unregulated power supply is cheaper but it forces to consider a safety's tolerance in sizing so that, in case of mains line and load fluctuations, voltage remains however within the allowed operating limits.

A detailed description of the sizing of the power supply is outside of this manual. The user who decides to assemble his own power supply must be technically qualified to size it, to ensure the correct working and to fulfill each safety requirements. To determine the power supply output voltage it must be considered the maximum mains line fluctuation expected in the worst operative

conditions, the maximum open load voltage and the minimum full load voltage, and to ensure that the maximum and minimum values resulting from the combination of these components are within the range of the maximum and minimum voltage specified for the chosen integrated motor model.

The power that the power supply must deliver is given by the one absorbed by the load (thus depending from the torque required to the motor as well as from the rotation speed) and by the motor and drive efficiency.

As an example, not to be considered exhaustive nor necessarily suitable to the application, it is the following basic electric diagram of an unregulated power supply with a brief indication of the values of components.



Note that the earth connection must be star-like, where the earth connections of the various components terminate into one single point electrically connected to the electrical cabinet and the earth of the plant.

In addition, the wiring to the drives must be star-like fixing the star center on the poles of the filter capacitor C1.

It is mandatory to provide a fuse on each primary winding of the transformer, able to intervene in case of short circuit or malfunctioning. It is also obligatory to place a fuse on each motor's power supply conductor.

As shown in the diagram, it is necessary to put in series to the transformer primary a filter able to block the emissions generated by the device and/or present on the mains line.

The reduction level that the filter must guarantee may vary a lot according to the rules applied to the field to which the application and/or installation belong. Manufacturers of SHAFFNER and CORCOM filters can be a good reference to find the suitable filters for your application.

The following table shows the characteristic values of the main components of the power supply. The calculations take into account a medium power application and an oscillation of the mains line voltage included in the +10/-20% range.

Model	Fuses Fn (A rit.)	Number of motors	Secondary T1 (Vac)	Power T1 (VA)	Current D1 (Arms)	Voltage C1 (Vdc)	Capacity C1 (µF)
DMS7xE4241 DMS7xE4264 DMS7xE4271 DMS7xE4321	8	1	32	200	25A	63	2200
		2		400	25A		4700
		3		600	25A		6800
		4..5		1000	35A		8200
		6..8		1400	50A		10000
DMS7xE4331	8	1	32	300	25A	63	2200
		2		600	25A		4700
		3		900	35A		6800
		4..5		1500	50A		8200
		6..8		2100	80A		10000
DMS7xE4350	8	1	32	300	25A	63	2200
		2		600	25A		4700
		3		900	35A		6800
		4..5		1500	50A		8200
		6..8		2100	80A		10000
DMS7xE4360	12	1	32	300	25A	63	2200
		2		600	25A		4700
		3		900	35A		6800
		4..5		1500	50A		10000
		6..8		2100	80A		12200
DMS7xE7241 DMS7xE7264 DMS7xE7271 DMS7xE7321	8	1	48	400	25A	100	2200
		2		800	25A		4700
		3		1200	35A		6800
		4..5		2000	50A		8200
		6..8		2800	80A		10000
DMS7xE7331	8	1	48	500	25A	100	2200
		2		1000	35A		4700
		3		1500	35A		6800
		4..5		2500	50A		8200
		6..8		3500	80A		10000
DMS7xE7350	8	1	48	500	25A	100	2200
		2		1000	35A		4700
		3		1500	35A		6800
		4..5		2500	50A		8200
		6..8		3500	80A		10000
DMS7xE7360	12	1	48	500	25A	100	2200
		2		1000	35A		4700
		3		1500	35A		6800
		4..5		2500	50A		10000
		6..8		3500	80A		12200

The capacity values suggested for C1 can also be obtained placing more capacitors in parallel amongst them. Eventual approximations must be made in excess. In parallel with the capacitor C1 it is recommended to place a resistor, sized appropriately, to ensure the discharge of the capacitor when the power supply is turned off.

The working voltage of the T1 transformer primary winding must be chosen according to the mains line voltage available during the installation of the application. The transformer must have a shield between primary and secondary windings which must be connected to earth by a short and not inductive connection. The secondary winding voltage is meant open load, with the primary winding supplied at the nominal voltage.

The rectifier, besides supporting the maximum current required by the integrated motor, must be able to tolerate the current generated during the charge of the C1 capacitor. Such current, as being essentially limited only by the internal resistor of the transformer secondary winding, usually very low, and by the wiring, can also be of elevated entity, even if of short length (it exhausts when the capacitor is charged).

Furthermore, the rectifier needs a heat sink able to maintain the temperature within the range defined by the manufacturer (usually 70°C). The working voltage of the D1 rectifier must then be chosen according to the T1 transformer secondary winding voltage, multiplied by 2 at least.

In the configurations with more than one motor with integrated drive, if the drives are not all set to the maximum current and/or if the working cycle is not simultaneous, the power of the transformer can be considerably reduced. In some cases this can also be done when the motors' speed is limited.

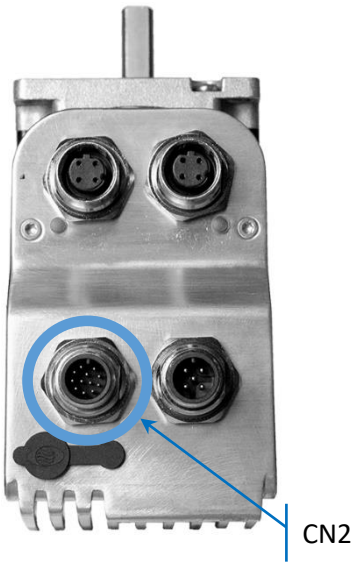
The diagram and the components' values refer to a three-phase power supply. Dimensioning the components in a different way, it is also possible to realize a single-phase power supply, which is not recommended when the required power is greater than 800W.

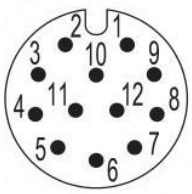
The filter, transformer and power supply unit must be used only to supply voltage to the motors. It is not recommended to derive other supplies from any of these parts. Rather, it is suggested to get auxiliary supplies using the mains line before the filter.

3.2.5 CN2 – Digital and analog I/O

The connection of the digital and analog I/O is through a 12pin M12 connector.

2



CN2 – digital and analog I/O		
Pin	Description	M12 Coding A Male 12ct  View from top
1	DI0+ (Digital Input 0)	
2	DI0- (Digital input 0)	
3	DI1+ (Digital Input 1)	
4	DI1- (Digital input 1)	
5	DI2 (Digital input 2)	
6	DI3 (Digital input 3) / DO0 (Digital Output 0)	
7	DI4 (Digital input 4) / DO1 (Digital Output 1)	
8	DI5 (Digital input 5) / DO2 (Digital Output 2)	
9	DOCOM+ (common DO power supply)	
10	AI0 (Analog Input 0)	
11	AI1 (Analog Input 1)	
12	DAGND (Digital and Analog Input GND)	

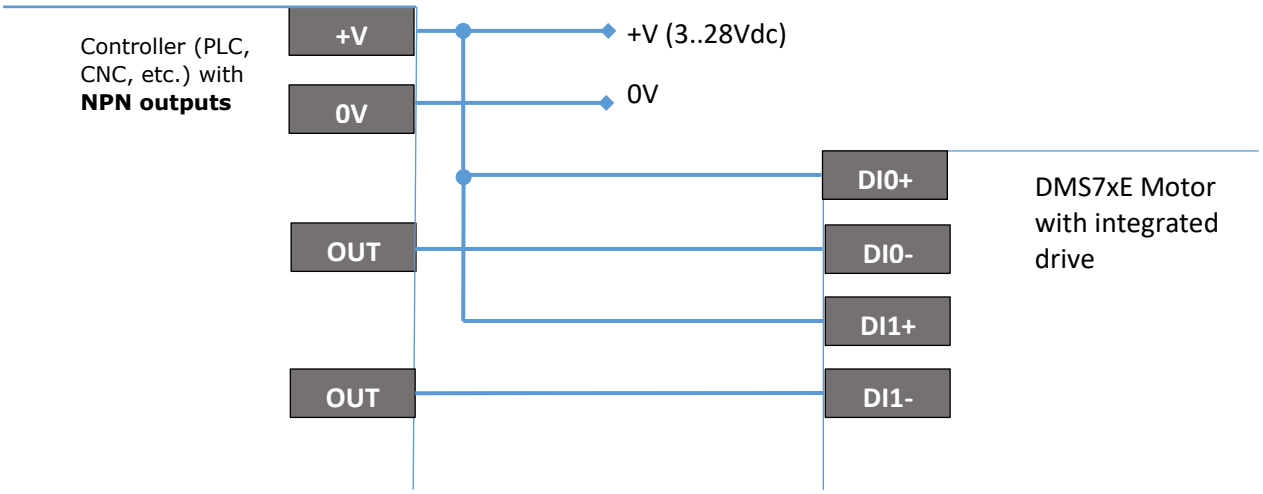
3.2.5.1 Digital Inputs

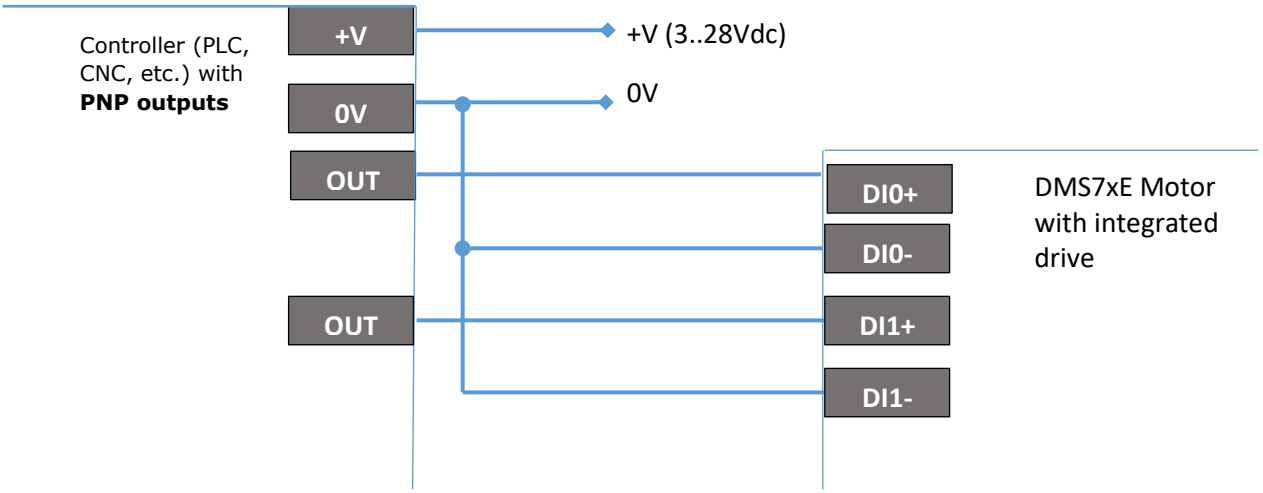
The DI0 and DI1 digital inputs are optocoupled and have a current regulation circuit which allows to apply voltages between 3Vdc and 28Vdc without the need to insert any current limit resistor. The current in fact remains constant independently from the voltage applied to the input. This simplifies the installation and the wiring.

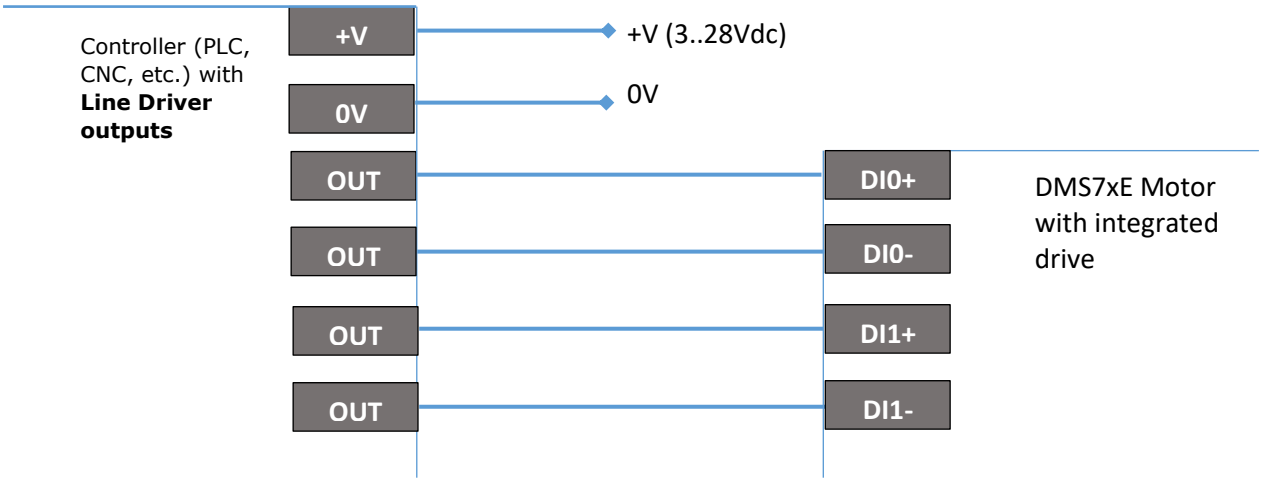
The DI0 and DI1 inputs have both connections (+ and -) and can be used in configuration NPN, PNP and Line Driver. To use the input in NPN configuration you just have to connect the + to the positive reference and the – to the output of the master controller (PLC, CNC, etc.), while in case of PNP connection just connect the – to the GND (0V) and the + to the output of the control system. Finally, in case of configuration in Line Driver mode you just have to connect the +

of the input to the + of the output (also called direct signal) and the signal – to the – of the output of the control system (also called negate signal).

The described NPN, PNP and Lien Driver connections are shown in the below images:





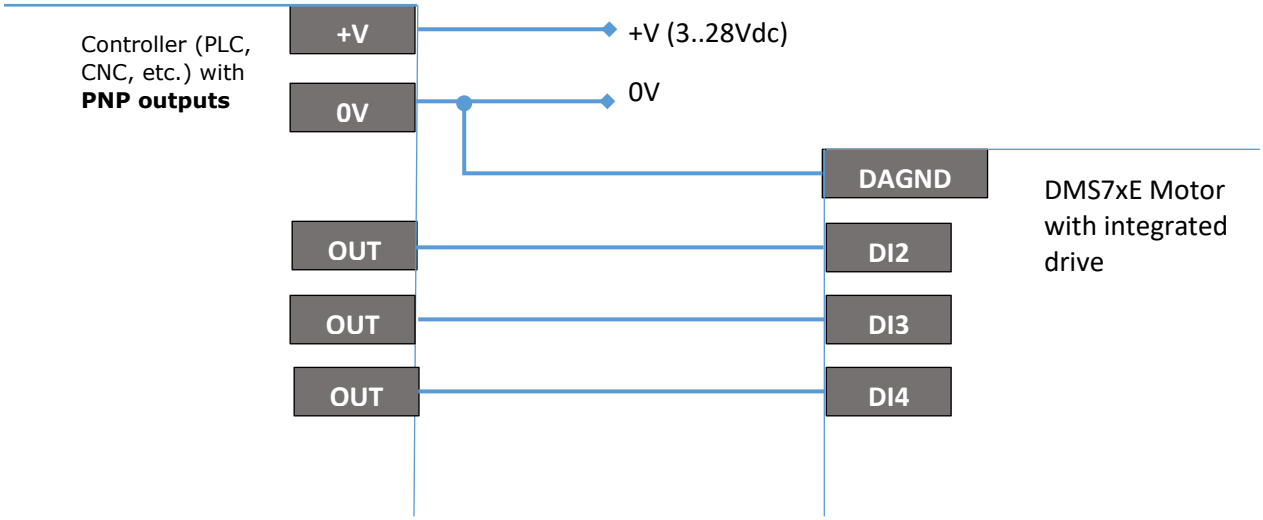


The following table shows the voltage values which correspond to the *Active* and *Inactive* status of the inputs DI0 and DI1, together with other parameters:

Symbol	Description	Unit	Value		
			Min	Typ	Max
Vdi01	Voltage for input DI0, DI1 active	Vdc	3		
Vdi01off	Voltage for input DI0, DI1 inactive	Vdc			1
Vdi01brk	Breakdown voltage of digital inputs DI0, DI1	Vdc	-30		+30
Idi01	Current absorbed by the digital inputs DI0, DI1 (24Vdc)	mA		5	

The inputs DI2, DI3, DI4 and DI5 refer to DAGND and can be used in PNP configuration. Inside the integrated motor there is a 10K pull-down resistor. Keep in mind that the DAGND signal is connected internally to the motor with the ground of the auxiliary power supply and the power ground.

An example of a PNP connection is shown below.



The following table shows the voltage values which correspond to the *Active* and *Inactive* status of the inputs DI2, DI3, DI4 and DI5, together with other parameters:

Symbol	Description	Unit	Value		
			Min.	Typ	Max
Vdi25	Voltage for input DI2..DI5 active	Vdc	4.5		
Vdi25off	Voltage for input DI2..D5 inactive	Vdc			2.5
Vdi25brk	Breakdown voltage of digital inputs DI2..D5	Vdc	-30		+30
Idi25	Current absorbed by the digital inputs DI2..D5 (24Vdc)	mA		2.4	

On the CN2 connector the pins of the inputs DI3, DI4 and DI5 are shared with the outputs DO0, DO1 and DO2. To work, the latter must first be configured through the *Direction_DIOSTP* object, as described in the following chapter.

Most of the inputs do not have a dedicated functionality and their use is defined when configuration the integrated motor, however some inputs have specific functions that cannot be assigned to other inputs.

The following table shows the inputs with dedicated functions:

Input	Function	
DI0	DI0	Generic input, 100KHz
	---	Input reserved for future implementation (ex. counter)
DI1	DI1	Generic input, 100KHz
	---	Input reserved for future implementation (ex. Counter)
DI2	DI2	Generic input
DI3	DI3	Generic input
DI4	DI4	Generic input
DI5	DI5	Generic input

3.2.5.2 Digital Outputs

The digital outputs do not have a specific functionality and their use depends on how the integrated motor is configured.

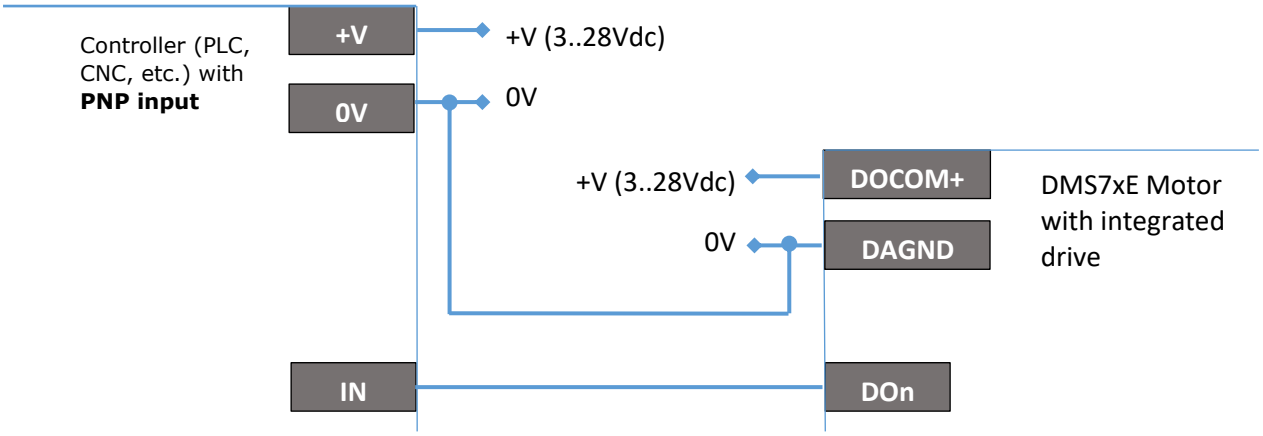
All the outputs are PNP type and are powered through the common DOCOM+ signal. Therefore, the output voltage depends on the power supply voltage applied between DOCOM+ and DAGND and this allows to freely interface systems with 5V, 12V or 24V inputs. Keep in mind that the DAGND signal is connected internally to the motor with the ground of the auxiliary power supply and the power ground.

In parallel to each output is placed a Zener diode which allows the connection of medium entity inductive loads (for example signal relays) without the need of an external recirculation diode.

The following table shows the outputs electrical characteristics:

Symbol	Description	Unit	Value		
			Min.	Typ	Max
Vdocom	Digital output supply voltage	Vdc	3.3		30
Vdo	Digital output operating voltage	Vdc	1		30
Vdobrk	Digital output breakdown voltage	Vdc	-0.5		37
Vdoz	Zener diode voltage placed in parallel to each output	Vdc	37	39	42
Ido	Digital output available current	mA			80
Idobrk	Digital output breakdown current	mA	120		
Pwdo	Digital output dissipable power	mW			300

The image below shows an example of connection of the digital outputs:



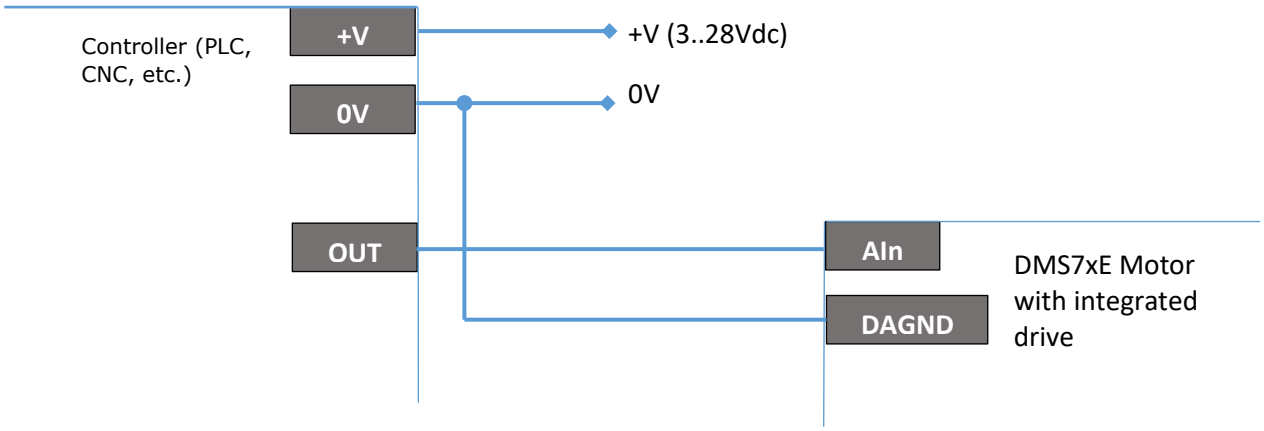
The digital outputs share the pins with the digital inputs DI3, DI4 and DI5 on the CN2 connector. The latter are always active while the outputs to be used must first be activated through the *Direction_DIOSTP* object. For more details, see the object's description in the chapter 8.2.3 2210h Digital I/O Setup.

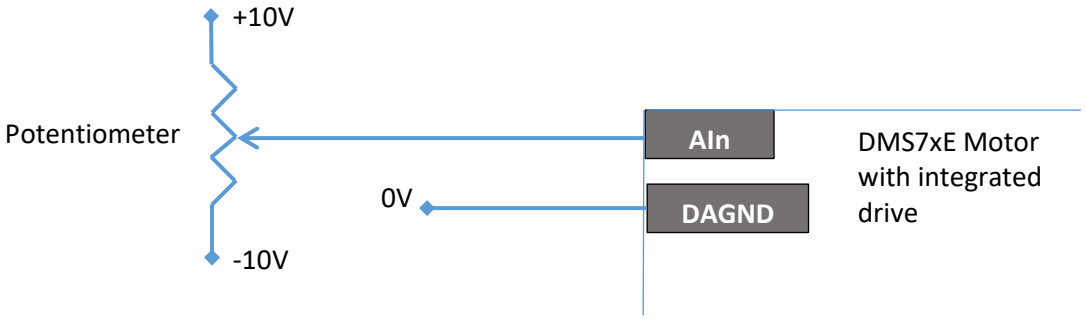
3.2.5.3 Analog Input

The analog input AI0 and AI1 are able to measure voltage between -10V and +10V.

In order to drive the analog input from a control device (PLC, CNC, etc.) simply connect respectively the AIn and DAGND signals of the integrated motor to the voltage output of the control device and to its ground reference. Please take note that the DAGND signal is connected internally to the motor with the ground of the auxiliary power supply and the power ground.

Example of connection of the analog input:





The following table shows the electrical characteristics of the analog inputs:

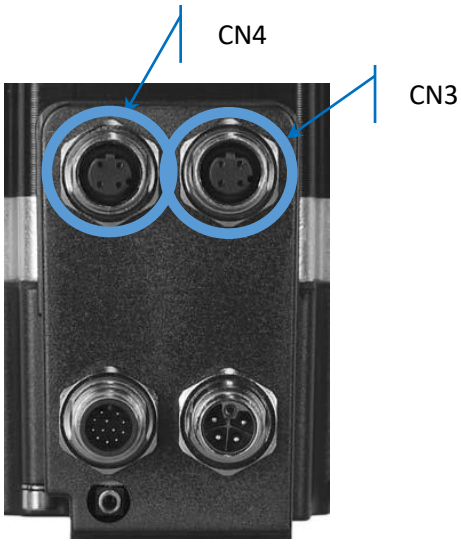
Symbol	Description	Unit	Value		
			Min	Typ	Max
Vai	Analog input operating voltage	Vdc	-10.2		+10.2
Vaibrk	Analog input breakdown voltage	Vdc	-45		+45
Rai	Analog inputs impedance	KΩ		47	
ADst	A/D converter conversion time	ms		1	
ADsoff	A/D converter start offset	%fs		1	
ADdoff	A/D converter offset drift	%fs		0.2	
ADline	A/D converter linearity error	%fs		1	

3.2.6 CN3 CN4 – EtherCAT Bus

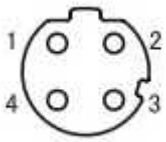
The CN3 and CN4 connectors are used for connection to the EtherCAT fieldbus. For the connection it is recommended to use a shielded cable in category CAT5 or higher. The connectors must be of metallic type and their casing must be electrically connected to the cable shield with a resistance of less than 10 Ohm.

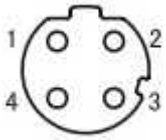
To improve the communication reliability in environments with high noises, or when the connections are particularly long, it is possible to experiment the connection to the functional ground of the cable shield using a low inductance connection.

The bus input must be connected to the CN3 connector while the CN4 connector can be used to connect the next EtherCAT device.



The following table shows the correspondence of signals to the connector's pins:

CN3 –EtherCAT Bus **Input**		
Description	Pin	M12 Coding D Female 4ct  View from top
TXD+	1	
RXD+	2	
TXD-	3	
RXD-	4	

CN4 –EtherCAT Bus **Output**		
Description	Pin	M12 Coding D Female 4ct  View from top
TXD+	1	
RXD+	2	
TXD-	3	
RXD-	4	

4 EtherCAT

The EtherCAT communication bus is a real-time industrial bus originally developed by Beckhoff and described in the document ETG1000 and ETG1020 available on the official website www.ethercat.org. For an exhaustive description of the protocol, see the official documentation, while below is an introduction to the specific functionalities and features of the DMS7xE motor with integrated drive.

Technical features of the EtherCAT communication and implemented functionalities:

Physical layer	100BASE-T
Baud rate	100Mbps Full-duplex
Sync Manager 0	Rx Mailbox
Sync Manager 1	Tx Mailbox
Sync Manager 2	RPDO
Sync Manager 3	TPDO
FMMU0	RPDO
FMMU1	TPDO
FMMU2	Mailbox status
Services	CoE (DS402)
Communication	SDO, PDO, Emergency Message
Synchronization	Free Run, SM2, DC SYNC0
Cycle time	1ms…10ms freely settable
RPDO, TPDO	Assignable and configurable (maximum 32byte)

4.1 ESI File

The master controller (PLC, CNC, etc.) acquires the necessary information for the correct communication with the integrated motor through a file in XML format file called EtherCAT Slave Information (ESI) file. The ESI file can be downloaded from the website www.lamtechnologies.com or required to the technical support writing to support@lamtechnologies.com.

Normally the ESI file must be imported into the master controller before scanning the network. To learn about the import procedure, consult the documentation of the master used.

The detailed description of ESI file format is contained in the document ETG2000 available on the official website www.ethercat.org.

4.2 CoE Service

The *CANopen over EtherCAT* (CoE) service implemented in the Mailboxes provides access to dictionary's object and regulates their methods of use and functionalities. For a detailed description, refer to the CANopen documentation on profiles /CiA301/ and /CiA402/ available on the official website www.can-cia.org.

4.3 Synchronization

The device is able to receive the process data both asynchronously (Free-Run) and synchronously (SM2 e DC SYNC0).

If the application uses the motion profiles *Profile position* (pp), *Profile velocity* (pv), *Torque Mode* (tq) and *Homing mode* (hm), it is possible to use either the asynchronous or synchronous communication mode as the trajectory is generated internally to the integrated motor and the master controller is limited to parametrize the motion profile and to command its start and stop.

On the contrary, if the motor with integrated drive is used with an interpolated profile, such as *Cyclic synchronous position* (csp), it is important to configure the master to use one of the supported synchronous modes since in the interpolated profiles the trajectory generator is in the master controller and the synchronism of the data transmitted and received amongst all the drives must be guaranteed. Furthermore, it is of fundamental importance that the data are updated isochronously.

The SM2 synchronous mode causes the updating of the process data upon receipt of the Sync Manager 2 sync signal. Due to the jitter present on the network, the receipt of the Sync Manager 2 can occur at different times therefore using this mode, the exchange of data amongst the various devices on the network is synchronized but not isochronous.

The DC SYNC0 synchronous mode includes the generation of a clock signal distributed to all the devices on the network, used to update the process data. This solution guarantees the isochronous update of process data and it is therefore the preferred synchronization mode.

For a correct operation and an optimal movement of the motor, it is of fundamental importance to set the objects $60C2_h:01_h$ *Interpolation_time_period_value* and $60C2_h:02_h$ *Interpolation_time_index* coherently with the chosen cycle time. When the synchronism mode is set to DC SYNC0, at the moment of the transition from the *PreOperation* to the *SafeOperational* state, the objects $60C2_h:01_h$ *Interpolation_time_period_value* and $60C2_h:02_h$ *Interpolation_time_index* are automatically filled in with the cycle time transmitted to the master controller, as long as it is included within the allowed range.

It is possible to disable the automatic compiling through the $2010_h:01_h$ *Options_FLDBSTP* object.

4.4 PDO

Process data are exchanged using the PDO which can be freely configured to meet the application's need.

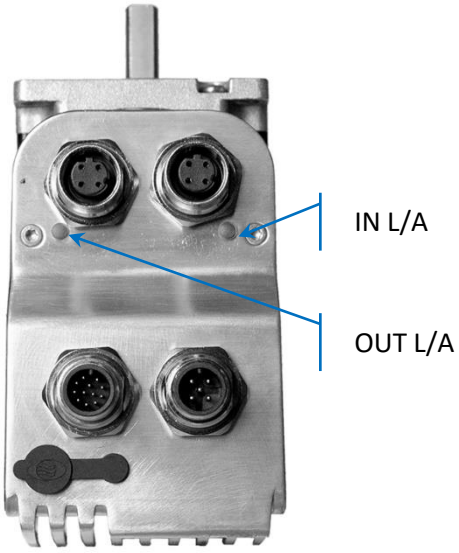
4 RPDO are available for the Outputs (data in output from the master and in input to the integrated motor) and 4 TPDO for the Inputs (data in input to the master and in output from the integrated motor).

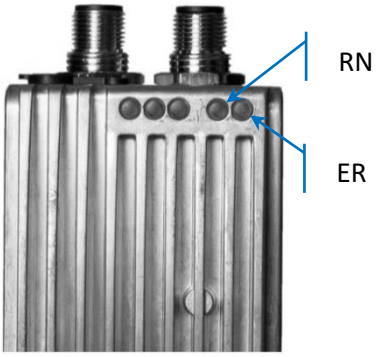
Each PDO can contain up to a maximum of 8 objects and totally 32byte of data.

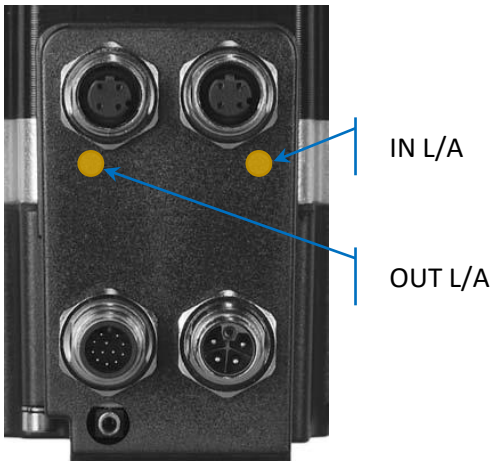
If an error occurs during PDO processing, for example due to the writing of a value outside the allowed limits or not allowed in the current status, the motor signals an error and enters the Quick Stop status executing the emergency deceleration ramp.

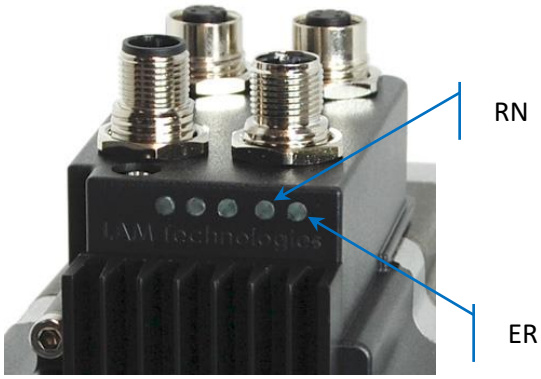
4.5 LED

In the upper and rear part of the motor there are 4 LEDs that show the status of the EtherCAT communication as shown in the table below:









<table>
<tr><th>Led</th><th>Status</th></tr>
<tr><td>IN L/A, yellow</td><td>
<table>
<tr><td>Steady on</td><td>Ethernet Link</td></tr>
<tr><td>Flashing</td><td>EtherCAT Activity</td></tr>
</table>
</td></tr>
<tr><td>OUT L/A, yellow</td><td>
<table>
<tr><td>Steady on</td><td>Ethernet Link</td></tr>
<tr><td>Flashing</td><td>EtherCAT Activity</td></tr>
</table>
</td></tr>
<tr><td>RN, green</td><td>It shows the status of the EtherCAT State Machine (ESM).
<table>
<tr><th>Led</th><th>ESM status</th></tr>
<tr><td>Off</td><td>Init</td></tr>
<tr><td>Slow flashing</td><td>Pre-Operational</td></tr>
<tr><td>Single flashing</td><td>Safe-Operational</td></tr>
<tr><td>Steady on</td><td>Operational</td></tr>
<tr><td>Fast flashing</td><td>Bootstrap</td></tr>
</table>
</td></tr>
<tr><td>ER, red</td><td>It shows errors
<table>
<tr><th>Led</th><th>Error</th></tr>
<tr><td>Off</td><td>No error</td></tr>
<tr><td>Continuous flashing</td><td>Configuration error</td></tr>
<tr><td>Single flashing</td><td>Generic error</td></tr>
<tr><td>Double flashing</td><td>Cyclic data watchdog</td></tr>
</table>
</td></tr>
</table>

5 Software Omni Automation IDE

The free Omni Automation IDE software (hereinafter OAI) running under Windows (Windows 7, Windows 8.1 and e Windows 10 32bit or 64bit) allows to perform device diagnostics and to read and write registers in real time.

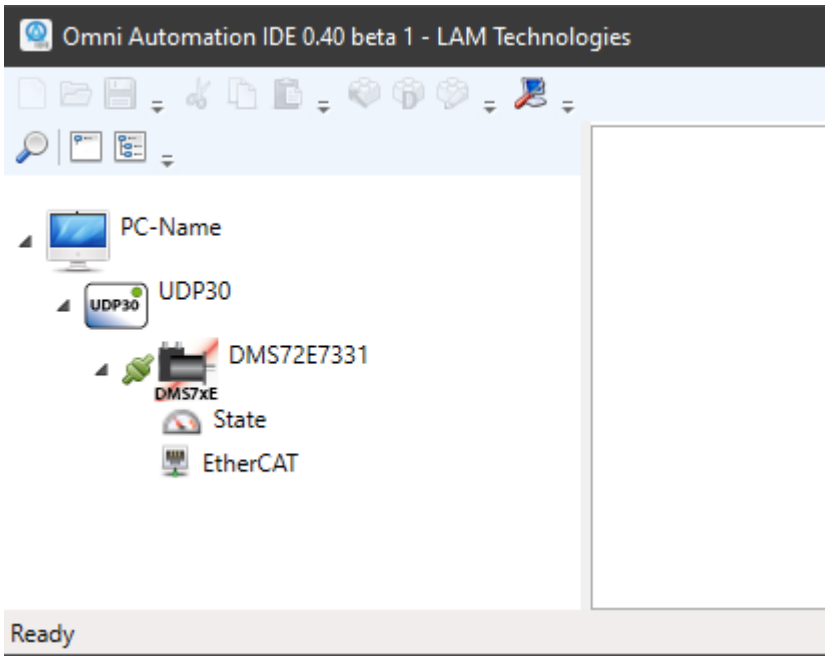
The connection between the PC and the motor with integrated drive is made through the UDP30 interface which also galvanically isolates the PC from the device. Furthermore, the UDP30 interface is able to power the logic section, allowing for example to update the firmware even without the power supply.

When the integrated motor is powered only by the UDP30 interface and receives the auxiliary power (+24V) and/or the power supply, it automatically restarts.

When the integrated motor is powered only by the UDP30 interface, the power stage, the field bus and the I/Os (digital, analog, encoder, etc.) are not operating.

After connecting the UDP30 interface to the DUP port on the upper part of the motor, it is possible to push the *Search* button so that OAI starts scanning the UDP30 interfaces and the connected devices.

When the search is complete, the tree of the devices connected to the PC appears, looking similar to the following image:



By clicking with the right button of the mouse on the symbol of the motor it is possible to save and load the settings and data of each device's tab .

5.1 Firmware update

By double-clicking on the symbol or name of the integrated motor (DMS72E7331 in the image) it appears the tab with the characteristic data of the device as the serial number, the firmware version, etc.

Through the *Update* link it is possible to update the device firmware.

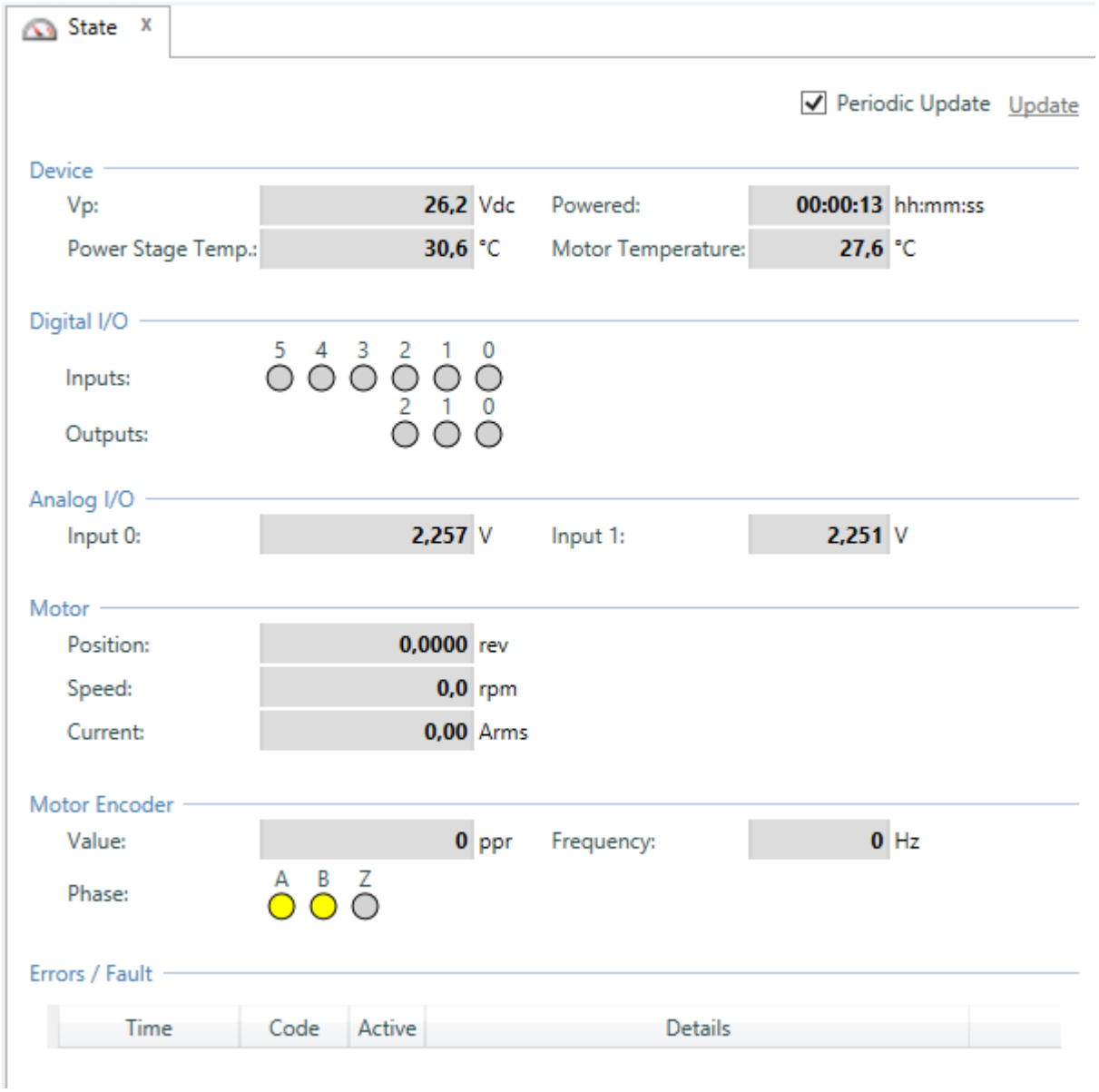
The products are typically sold with the latest released firmware which may include new functionalities, the improvement of the existing performances, bugs fixes, etc. A different firmware may produce a different behavior of the device and whether the correct functioning of the application is affected to characteristics or behaviors of the product not defined in the manual, or beyond the limits indicated, malfunctions in the application itself could occur.

The user who wishes to have an identical behavior of the product with respect to a specific revision of the firmware, must upload this firmware into the products itself before putting it into operation. The specific revision of the firmware may be required by writing to support@lamtechnologies.com. New firmware revisions may be released at any time and without prior notice.

The firmware update may overwrite the device configuration and any data stored on it with default values.

5.2 Status and Diagnostics

In Omni Automation IDE, on the left in the tree view list of the connected devices, double-clicking on the voice *Status* visible under the symbol of the integrated motor, opens the tab that shows the status of the device and the eventual errors.



The check box *Periodic Update*, when selected, maintains updated the device status view. By removing the check mark from the box, to update the status you will need to click on the near link *Update*.

The link *Update* is activated when the check box is not selected and allows you to manually update the status display of the integrated motor.

In the section *Device,* the field *Vp* shows the voltage value of the DC power bus inside the drive.

In the section *Device,* the field *Powered* shows the time passed from the last power on or reset of the device.

In the section *Device*, the field *Power Stage Temp.* shows the temperature value of the power stage of the drive while the field *Motor Temperature* shows the temperature reached by the motor.

Under the section *Digital I/O,* it is shown the status of the digital inputs and outputs. When the signal is associated with the yellow color it means that it is in the Active status while if the color is grey it is in the Inactive status.

Under the section *Analog I/O,* the field *Input 0* shows the voltage value applied to the analog input 0 while the field *Input 1* shows the voltage value applied to the analog input 1.

Following is the description of the fields inside the section *Motor*.

The field *Position* shows the actual position reached by the motor. The integer part of the value indicates the number of complete revolution, while the decimal part shows the fraction of the revolution reached by the motor with a resolution of 1/10000 rev. For example, the value 0.5000 indicates that the motor is half revolution forward with respect to 0 position, while the value -3.7500 indicates that the motor is 3 and 3 quarters of revolution backward with respect to the 0 position.

The field *Speed* shows the actual speed reached by the motor.

The field *Current* shows the actual phase current which flows in the motor phases. It is not surprising if the field value differs from the configuration because, particularly at high speed, or with low power supply voltage, due to the inductance and counter-electromotive force of the motor, the current cannot be able to reach the set rated value.

The field *Load Ratio* becomes visible only when the drive operates at closed-loop and indicates the relation between the torque supplied by the motor and the load resistant torque. The value is expressed in percentage and when it reached 100% the motor stops. This condition, in fact, indicates that the load resistant torque has exceeded the one supplied by the motor. If the value is positive it means that the load applies a resisting torque in the direction opposite to that of the motor rotation, while if the value is negative it means that the load is trying to drag the motor, in the same direction of the rotation, beyond its position.

Under the section *Motor Encoder,* the field *Value* indicates the cyclical position of the encoder over a revolution. The value, expressed in 4x encoder resolution, resets to zero at each revolution of the encoder itself.

Under the section *Motor Encoder,* the field *Frequency* indicates the frequency of the encoder A and B signals.

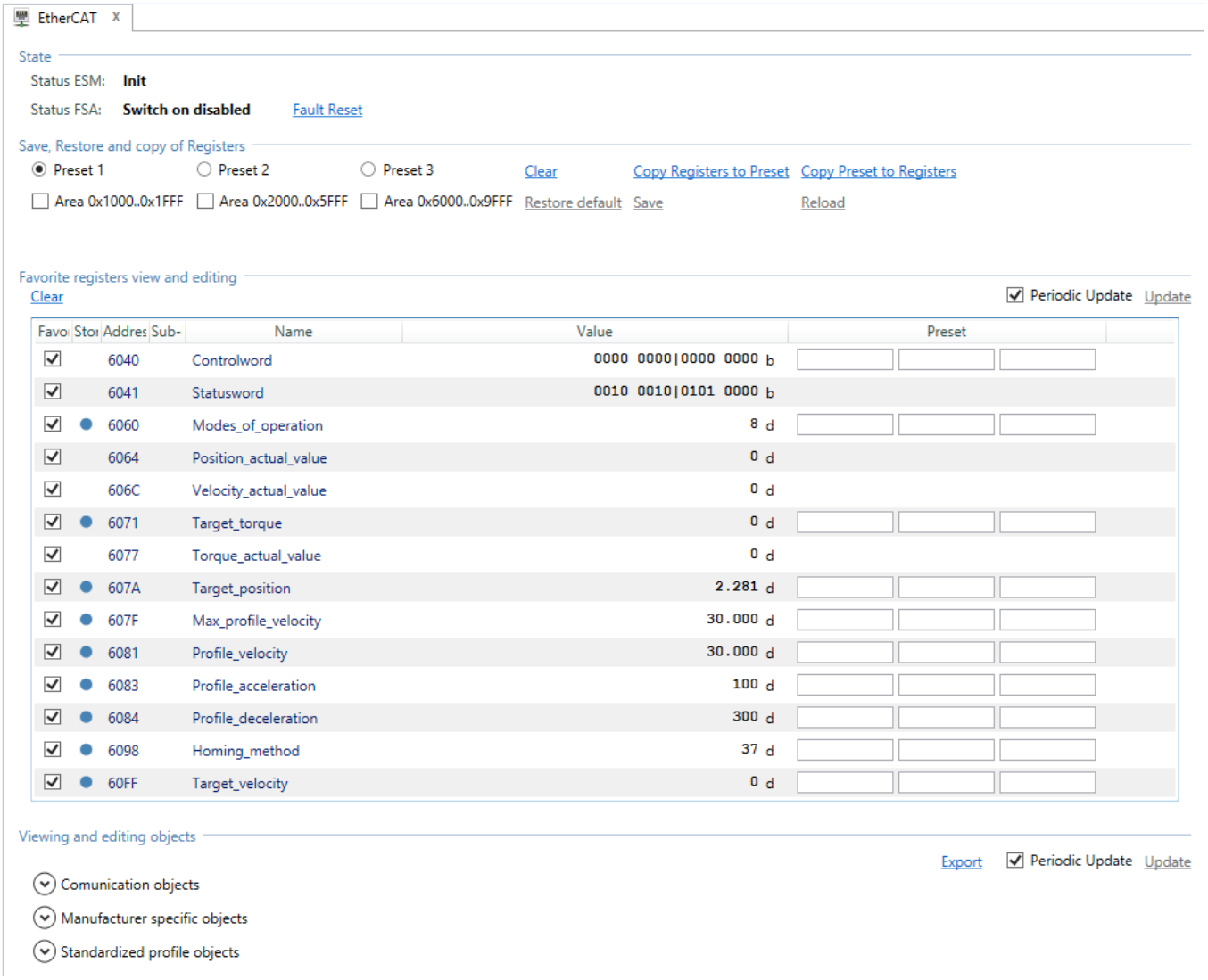
Under the section *Motor Encoder,* it is possible to know the logic level of the encoder *Phases* A, B and Z according to the associated color. When the signal is associated with the yellow color it means that it is in the Active status, while if the color is grey it means that the signal is in the Inactive status.

The section *Errors/Fault*, in table form, shows the history of the errors occurred from the last power on. When the error is active, the column *Active* of the table contains an exclamation mark, while when the error is stored but no more active

the column *Active* is empty and the line background is red. When the error reset is executed the background becomes white. In the table are stored up to 10 errors, then the latest errors replace the oldest ones. The column *Time* shows the moment when the error occurred after the latest integrated motor power on or reset. The column *Code* contains the error numeric code while the column *Details* shows a brief description of the error. Positioning the mouse pointer over the contents of each column a tooltip provides more details.

5.3 Visualization and modification of the dictionary objects

In Omni Automation IDE, on the left in the tree view list of the connected devices, by double-clicking on the entry *EtherCAT*, visible below the motor with integrated drive's symbol, you can access the tab that allows to view and edit the device dictionary objects.



Scrolling the tab from top to bottom the first section met shows the device status. The link *Fault Reset* on the right allows you to reset faults and errors that are no longer active.

Removing the fault, the integrated motor can immediately go to the operating state and start to rotate according to the contents of dictionary objects. Before removing the fault, properly configure the objects and make sure that an unexpected movement of the motor cannot cause any damage to things or people.

Next section allows you to select the Preset group in which you can transfer the dictionary objects values or the Preset group used to write the objects themselves.

The Presets are temporary locations containing a value that can then be transferred into the corresponding dictionary object.

To transfer the Preset value to the register, simply press the enter key or double-click with the left mouse button.

Totally, there are available 3 Preset for each register that allow you to quickly switch from a value to another by simply double-clicking the field containing the desired value. Subsequently there are 3 boxes that allow you to select the belonging area of the objects on which you want to operate. It is possible to check more than one box to select a wider range of objects.

The *Save* allows you to save the contents of the dictionary objects in the non-volatile memory, while *Restore default* invalidates the contents of the non-volatile memory restoring the default value of the objects. At the power on the objects are automatically initialized with the default values or with the values saved, it is possible to manually reload the objects through the link *Reload*.

The actions *Save*, *Restore default* and *Reload* are possible only with the motor disabled and the ESM status *Init* o *Pre-Operational*. For more details see chapter 6.2 Saving and restoring of default values.

The objects visualization and modification area is logically divided in two; in the upper area there are the most commonly used objects (favorite) while in the lower one there are all the remaining objects gathered by the functional area and ordered by address.

The *Periodic Update* check box, when selected, maintains updated the visualization of the device status. By removing the check from the box, to update the status it will be necessary to click on the near link *Update*.

The link *Update* is activated when the check box is not selected and allows you to manually update the visualization of the integrated motor status.

The objects that you want to monitor frequently can be added to the favorites area by checking the box in the *Favorite* column of the table.

The objects contained in the favorites area are more frequently updated than the others, allowing a more accurate analysis of the value.

The column *Storable* of the table shows the objects that can be saved in the non-volatile memory using the *Store Parameters* functions. If there is a blue dot it means that the objects will be stored in the non-volatile memory as a result of the *Store Parameters* command.

The subsequent columns show the object address in hexadecimal format and the name.

The column *Value* shows the object value in real time.

Every time the value changes, the background becomes green for about 5 seconds to highlight the objects recently changed.

The objects value can be visualized in decimal, hexadecimal or binary format. To change the visualization base, simply place the mouse over the value and click with the right button, then select the desired visualization base.

The letter after the value indicates the base in which the number is visualized according to the following correspondence:

Symbol	Base
d	Decimal
h	Hexadecimal
b	Binary

In the *Preset* column there are three fields that can be filled with the values you want to write in the corresponding object. By pressing the enter button or double-clicking with the left mouse button, the value of the field is copied into the dictionary object.

Please note that some registers prevent the writing in specific operative conditions.

Please note that when the cyclic process data exchange is running, the objects mapped in the process data area are continuously overwritten from the master controller, therefore an attempted modification by OAI may appear to have no effect.

The Preset can be compiled with a hexadecimal value, putting the prefix “0x”, or binary, putting the prefix “0b”.

For clarity of writing, spaces can be used to logically separate figures. For example, it is possible to write the value 212 in binary as “0b 1101 0100 to highlight more clearly the nibbles (groups of 4 bits) which compose the byte.

The link *Export*, in the objects area, allows you to export the contents of each dictionary object to a file, in CSV format.

By clicking with the right button of the mouse on the *EtherCAT* entry (on the left in the devices tree) it is possible to save and load the configuration of the

EtherCAT tab which includes the Preset values, the objects in the favorites area, the value display base, etc.

6 Operation

The motor with integrated drive operates mainly through EtherCAT fieldbus. With an appropriate configuration it is also possible to control some functions through the I/O signals integrated in the device.

ATTENTION, verify carefully that there are no conflicts between the control via fieldbus and the local control, in order to prevent unexpected movements or a failure in the activation of desired functions.

The integrated motor implements the CoE service and the profile / CiA402 / dedicated to drives and motion control devices. The operating modes supported by the firmware revision described in this manual are as shown in the table below:

Abbreviation	Description
pp	Profile position mode
pv	Profile velocity mode
tq	Torque profile mode
hm	Homing mode
csp	Cyclic synchronous position mode
csv	Cyclic synchronous velocity mode

Note: It is suggested to always verify if there are firmware revisions with new operating modes or new implemented functionalities.

Before operating, the drive requires some preliminary settings by writing appropriate values in the dictionary objects. Many objects are provided by the profile /CiA402/ and contained in the standardized area *(Standardized profile area*), while others are peculiarities of the DMS7xE Series and are included in the specific area (*Manufacturer-specific profile area*).

6.1 Minimum settings

The minimum settings that must be made before enabling the motor are the setting of the running and idle current (3310_h *Motor Configuration* object).

If the integrated motor operates in an interpolated mode, such as *Cyclic synchronous position mode*, and the EtherCAT communication is synchronous (SM2 or DC SYNC0), it is important to set the $60C2_h:01_h$ *Interpolation_time_period_value* and $60C2_h:02_h$ *Interpolation_time_index* objects with the same cycle time used by the master controller, otherwise communication errors or irregular movements might occur.

6.1.1 Running and idle current configuration

The integrated motor allows to freely define the running and idle current of the motor to optimally adapt it to the application.

The running current is impressed on the motor during the rotation while the idle current is applied after the stop. The time from the motor stop, after which the current is set to the idle value, is configurable.

When the integrated motor is equipped with Encoder and configured for closed-loop control, it is possible to set the current regulation so that it adapts to the load applied ($23A0_h$:01_h *Mode_CRRG* object).

When the current regulation is configured dynamically, the running and idle current correspond respectively to the current supplied to the motor in absence of load and to the current at full load (locked rotor).

The parameters related to the operating current of the motor are within the 3310_h *Motor Configuration* object. The object is Record type and provided with sub-indices.

6.1.1.1 Current_Min_MTRCNF (3310h:02h)

The *Current_Min_MTRCNF* object allows to specify the current applied to the motor in idle mode. The current is applied after the motor stop after the time defined by the *Current_Idle_Delay_MTRCNF* object described below.

The object is expressed as a percentage of the motor rated current (2310_h:08_h *Rated_Current_MTRDT* object) and each unit is equal to 0.01%. For example, if you want to set an idle current equal to the 30% of the rated current, it is necessary to write the *Current_Min_MTRCNF* object with the value 3000 (30 / 0.01). If the configured motor rated current is for example 4Arms, the idle current will be equal to 1.2Arms (30% of 4A).

6.1.1.2 Current_Max_MTRCNF (3310h:03h)

The *Current_Max_MTRCNF* object allows to specify the current applied to the motor during rotation.

The object is expressed as a percentage of the motor rated current (2310_h:08_h *Rated_Current_MTRDT* object) and each unit is equal to 0.01%. For example, if you want to set a running current equal to the 80% of the rated current, it is necessary to write the *Current_Max_MTRCNF* object with the value 8000 (80 / 0.01). If the configured motor rated current is for example 4Arms, the idle current will be equal to 3.2Arms (80% of 4A).

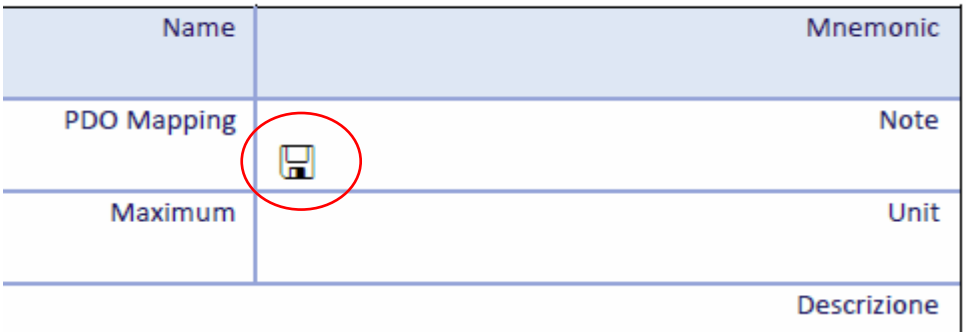
6.1.1.3 Current_Idle_Delay_MTRCNF (3310h:04h)

The object *Current_Idle_Delay_MTRCNF* allows to specify the waiting time from the motor stop before the current is set to the value defined by the *Current_Min_MTRCNF* object.

Each unit is equal to 1ms, therefore setting for example the value 500, the drive will wait for 500ms from the motor stop before changing the phase current.

6.2 Saving and restoring of default values

The device is able to save many of its objects in the non-volatile memory. The dictionary objects that can be saved are highlighted with the symbol 🖫 in the field *Note* of the table that describe the objects itself, as in the following example:



When the value of an object is saved in the non-volatile memory it is automatically restored at the power on or in case of NMT *Service Reset Node*.

By saving a value other than the default it is possible to adapt the device to the application without having to configure it every time. Apparently this seems to be a simplification but it forces to prepare the device (saving the desired values in the dictionary objects) before it can be used in the application. When there are many applications, or they are updated over time, you are forced to keep an archive with all the objects values used in each application and in each version and over time this can become complex and give rise to errors. On the contrary, by delegating the configuration of each device to the master controller, it will not be necessary to keep track of how the device should be initialized. In this case, in fact, the specific application will initialize the objects with the desired values and without errors. Furthermore, if the device should need to be replaced, the technical support can simply send a new device without worrying about the application and the version in which it will be installed.

When possible it is therefore recommended not to use the *Save* function to modify the default value of the objects. On the contrary, it is recommended to always initialize every object used in the application with the desired value, independently from the saving or default. Usually, the master controller provides the possibility to initialize the dictionary objects with the desired value during the PS transition (by *Pre-Operational* a *Safe Operational*).

According to the /CiA301/ profile, saving is done by writing an appropriate key in one of the sub-entries of the *Store Parameter* object.

The dictionary objects can be also restored to the default value writing an appropriate key in one of the sub-entries of the *Restore Default Parameters* object.

It is possible to operate on all the dictionary objects or on a subset of them choosing the appropriate sub-index. According to the /CiA301/ profile the following subsets have been created:

- *Communication Parameters*, entry included between 1000_h and $1FFF_h$
- *Manufacturer Defined Parameters*, entry included between 2000_h and $5FFF_h$
- *Application Parameters*, entry included between 6000_h and $9FFF_h$

The defaults values can be saved or restored only with the motor disabled and with the EtherCAT State Machine (ESM) in the *Init* or *Pre-Operational* status. Trying the operation in a different status an error code is received.

It is possible to save or restore the default values for 10,000 times max.

6.3 Motor Holding Brakes

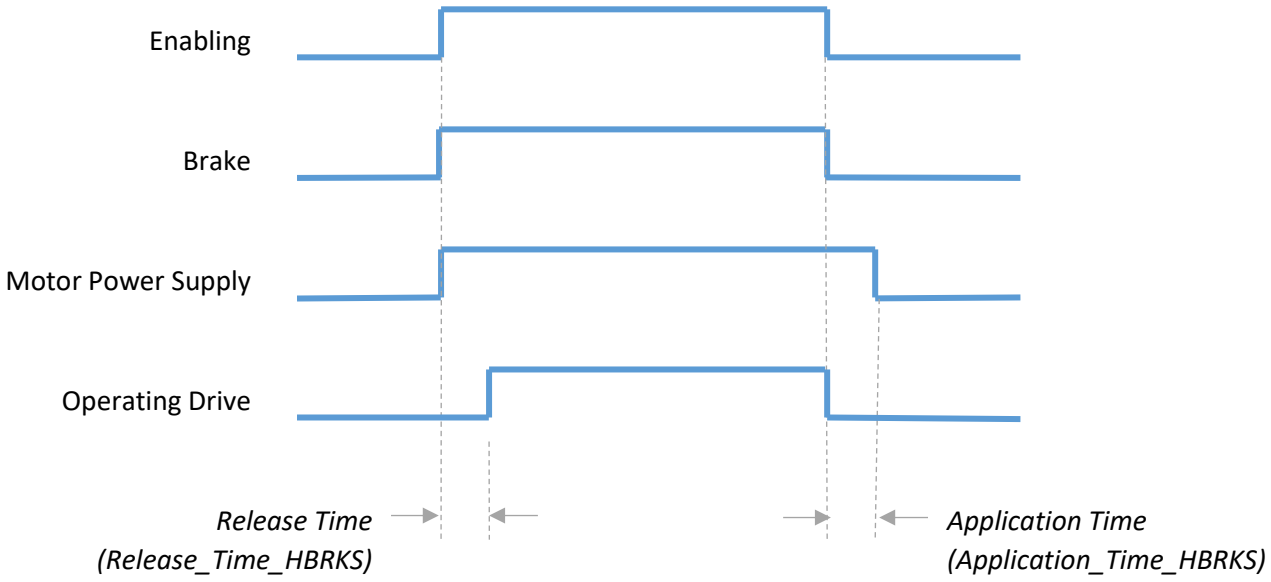
The drive is able to control the holding brake of the motor through one of the digital outputs.

Through the *Option_HBRKS* object it is possible to coordinate the handle of the brake with motor enabling, taking into account the brake's characteristic engaging and disengaging time.

With the brake control enabled, by enabling the motor, the digital output predisposed to control the break is immediately activated to release it and simultaneously the motor is supplied to keep it in position. The transition to the operating state is delayed by the time the brake requires to completely disengage. This time can be freely set through the *Release_Time_HBRKS* object.

With the brake handling enabled, by disabling the motor the drive immediately exits the operating state and simultaneously deactivates the digital output predisposed to the control of the brake to engage it. The motor remains powered for the time the brake requires to completely engage. This time can be freely set through the *Application_Time_HBRKS* object.

The following graph shows the temporal relationship between the described events:



The *Option_HBRKS* object is also useful to enable and set the manual control of the brake that can intervene in an exclusive way, with respect to the control operated by the drive, or shared.

The manual control can be performed via a digital input configurable through the *Holding_Brake_DIA* object or through the bit0 of the *Control_HBRKC* object.

Through the Status_HBRKC object it is possible to know in real time the status of the brake and the status of the associated output.

When the digital output predisposed to control the brake is active, the brake is considered released.

6.4 Closed-loop

The integrated motors equipped with Encoder belonging to the DMS72 series are able operate also in closed-loop with the following benefits:

- No loss of step
- 100% use of motor torque
- Possibility of control of the motor torque
- Dynamic current supply proportional to the load
- Less motor heating
- Position Error monitoring and Following Error alarm

In order for the drive to operate in closed-loop, you just have to enable the closed-loop control mode by setting the bits 1..0 of the *Feedback_MTNSTP* object to 1.

At the first enabling of the motor, following the start or reset of the device, and with the closed-loop operation active, phasing occurs between the Encoder and the rotor. During this operation, the rotor can make small movements within approximately +/-2° and the user must always ensure that the application allows such rotation. If, on the contrary, the rotor is blocked, even in just one direction, for example because it is in mechanical stop, the correct phasing of the Encoder may fail, with consequent malfunction of the motor with integrated drive. If this possibility occurs, it is possible to enable the motor in open-loop and move it to a position that guarantees +/-2° of freedom, then disable the motor, activate the closed-loop operation and finally enable the motor again. In this way, the rotor will have the space necessary to make the small rotations essential to correctly complete the Encoder phasing.

If the Encoder supply is removed, or in case of errors, the phasing between the Encoder and the rotor will be repeated the first time the motor is enabled.

By acting on bit 3 of the register *Feedback_MTNSTP* it is possible to obtain the phasing of the encoder at each single enabling of the motor.

6.5 Operation Modes

The motor with integrated drive implements many of the operationing modes provided by the profile /CiA402/ to meet the most different applications.

The firmware revision described in this manual supports the following operation modes:

Abbreviation	Description
pp	Profile position mode
pv	Profile velocity mode
tq	Torque profile mode
hm	Homing mode
csp	Cyclic synchronous position mode
csv	Cyclic synchronous velocity mode

We suggest to always verify if there are new firmware revision with new operation modes or implemented features.

The following paragraphs describe the different operation modes.

6.5.1 Profile position (pp)

In this mode the integrated drive executes a positioning profile by controlling the speed and position of the motor. The master controller can command absolute or relative positioning, moreover a buffer and handshake mechanism allows to chain consecutively multiple positioning, with no delays or interruptions due to the slow fieldbus communication.

The movement is performed according to the set values of maximum speed, acceleration and deceleration.

The main objects involved in the *Profile position* mode are shown in the table below:

Object associated with the operative *Profile position* mode			
OD Entry	Name	Unit Data type PDO	Description
6040$_h$	*Controlword*	--- UINT16 RPDO	Command controlling the FSA.
6041$_h$	*Statusword*	--- UINT16 RPDO	Provide the status of the FSA
6060$_h$	*Modes_of_operation*	--- INT8 RPDO	Requested operation mode
6061$_h$	*Modes_of_operation_display*	--- INT8 TPDO	Actual operation mode
6062$_h$	*Position_demand_value*	0.0001rev INT32 TPDO	Provide the demanded position value
6064$_h$	*Position_actual_value*	0.0001rev INT32 TPDO	Provide the actual value of the position measurement device
6065$_h$	*Following_error_window*	0.0001rev UINT32 RPDO	Indicate the configured range of tolerated position values symmetrically to the position demand value
6066$_h$	*Following_error_time_out*	ms UINT16 RPDO	Indicate the configured time for a following error condition, after that the bit 13 of the Statusword is set to 1
6067$_h$	*Position_window*	0.0001rev UINT32 RPDO	Indicate the configured symmetrical range of accepted positions relative to target position
6068$_h$	*Position_window_time*	ms UINT16 RPDO	Indicate the configured time, during which the actual position within the position window is measured
607A$_h$	*Target_position*	0.0001rev INT32 TPDO	Indicate the commanded position to reach
607D$_h$:01$_h$	*Min_software_position_limit*	0.0001rev INT32 TPDO	Min position range limit
607D$_h$:02$_h$	*Max_software_position_limit*	0.0001rev INT32 TPDO	Max position range limit

607F $_h$	*Max_profile_velocity*	0.1rpm UINT32 RPDO	Indicate the configured maximal allowed velocity in either direction
6081 $_h$	*Profile_velocity*	0.1rpm UINT32 RPDO	Indicate the configured velocity attained at the end of the acceleration ramp. It is valid for both directions of motion
6083 $_h$	*Profile_acceleration*	rpm/s UINT32 RPDO	Indicate the configured acceleration
6084 $_h$	*Profile_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration
6085 $_h$	*Quick_stop_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration used to stop the motor when the quick stop function is activated

To activate the *Profile position* mode you need to write the 6060$_h$ *Modes_of_operation* object with the value 01$_h$.

The *Controlword* object allows you to enable or disable the motor, to cause an absolute or relative movement and, together with the *Statusword* object, is responsible for the handshake movement useful in preparing a new movement while a positioning is in progress.

The following table shows the *Controlword* object and the meaning of its component bits.

Operative mode															
Profile position mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						cosp	h	fr	abrl	csi	nsp	eo	qs	ev	so

Controlword bits organization		
Bit		Description
15		Reserved, set to 0
14		Reserved, set to 0
13		Reserved, set to 0
12		Reserved, set to 0
11		Reserved, set to 0
10		Reserved, set to 0
9	cosp	Change on setpoint
8	h	Halt
7	fr	Fault reset
6	abrl	Absolute / Relative
5	csi	Change set immediately
4	nsp	New set-point
3	eo	Enable operation
2	qs	Quick stop
1	ev	Enable voltage
0	so	Switch on
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The following table shows the *Statusword* object and the meaning of its component bits.

Operative mode															
Profile position mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		fe	spa	ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Statusword bits organization		
Bit		Description
15		Reserved, ignore the value
14		Reserved, ignore the value
13	fe	Following error, 1=following error
12	spa	Set-point acknowledge, 1=New target accepted, 0=New target can be set
11	ila	Internal limit active, 1= Restriction of one or more parameters for internal limit
10	tr	Target reached, 1= Target position reached. In case of Halt or QuickStop, motor halted
9	rm	Remote, 1=*Controlword* executed
8	h	Halt, 1=Active request
7	w	Warning, 1=Presence of one or more warnings
6	sod	Switch on disabled
5	qs	Quick stop, 0= Quick Stop procedure in progress or concluded
4	ve	Voltage enabled, 1= Power supply applied to the device
3	f	Fault, 1= Error or Fault procedure in progress or concluded
2	oe	Operation enabled, 1=Motor enabled
1	so	Switched on, 1= Power stage of the device powered
0	rsto	Ready to switch on, 1= Device ready to supply the power stage
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The minimum steps required to execute a positioning in the *Profile position* mode are the configuration of the motion profile (acceleration, deceleration and standard velocity), the setting of the target position and the start of the movement through the bit *nsp* contained in the object *Controlword*.

The described operations can be carried out through the communication objects SDO, PDO or a combination of the two.

6.5.2 Profile velocity mode (pv)

In this mode the drive controls the motor in speed, always according to the set acceleration and deceleration ramps. The master controller can update the target velocity simply updating the *Target_velocity* object.

Two bits contained in the *Statusword* object inform the master about the status of the motor; the bit *Target-reached* becomes active when the motor actual speed approximates the target velocity of a value lower than the *Velocity_window* object, while the bit *Speed* becomes active when the motor rotates at a speed higher, in absolute value, than the *Velocity_threshold* object.

The main objects involved in the *Profile velocity* mode are shown in the table below:

Object associated with the operative *Profile velocity* mode			
OD Entry	Name	Unit Data type PDO	Description
6040h	*Controlword*	--- UINT16 RPDO	Command controlling the FSA.
6041 h	*Statusword*	--- UINT16 RPDO	Provide the status of the FSA
6060 h	*Mode_of_operation*	--- INT8 RPDO	Requested operation mode
6061 h	*Mode_of_operation_display*	--- INT8 TPDO	Actual operation mode
606D h	*Velocity_window*	0.1rpm UINT16 RPDO	Indicate the configured velocity window
606E h	*Velocity_window_time*	Ms UINT16 RPDO	Indicate the configured velocity window time
606B h	*Velocity_demand_value*	0.1rpm INT32 TPDO	Provide the output value of the trajectory generator
606C h	*Velocity_actual_value*	0.1rpm INT32 TPDO	Provide the actual velocity value
606F h	*Velocity_threshold*	0.1rpm UINT16 RPDO	Indicate the configured velocity threshold
6070 h	*Velocity_threshold_time*	Ms UINT16 RPDO	Indicate the configured velocity threshold time
607F h	*Max_profile_velocity*	0.1rpm UINT32 RPDO	Indicate the configured maximal allowed velocity in either direction
6083 h	*Profile_acceleration*	rpm/s UINT32 RPDO	Indicate the configured acceleration
6084 h	*Profile_deceleration*	rpm/s	Indicate the configured deceleration

		UINT32 RPDO	
6085 h	*Quick_stop_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration used to stop the motor when the quick stop function is activated
60F8 h	*Max_slippage*	0.1rpm INT32 TPDO	Indicate the configured maximal slippage
60FF h	*Target_velocity*	0.1rpm INT32 TPDO	Indicate the configured target velocity

To activate the *Profile velocity* mode you need to write the 6060_h *Modes_of_operation* object with the value 03_h.

The *Controlword* object allows you to enable or disable the motor while the *Statusword* object provides information on the status of the motor and the movement in progress.

The following table shows the *Controlword* object and the meaning of its component bits.

Operative mode															
Profile velocity mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							h	fr				eo	qs	ev	so

Controlword bits organization		
Bit		Description
15		Reserved, set to 0
14		Reserved, set to 0
13		Reserved, set to 0
12		Reserved, set to 0
11		Reserved, set to 0
10		Reserved, set to 0
9		Reserved, set to 0
8	h	Halt
7	fr	Fault reset
6		Reserved, set to 0
5		Reserved, set to 0
4		Reserved, set to 0
3	eo	Enable operation
2	qs	Quick stop
1	ev	Enable voltage
0	so	Switch on
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The following table shows the *Statusword* object and the meaning of its component bits.

Operative mode *Profile velocity mode*															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		mse	spd	ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Statusword bits organization		
Bit		Description
15		Reserved, ignore the value
14		Reserved, ignore the value
13	mse	Max slippage error, 1=velocity error
12	spd	Speed is equal 0, 1=motor halted
11	ila	Internal limit active, 1= Restriction of one or more parameters for internal limit.
10	tr	Target reached, 1= Target position reached. In case of Halt or QuickStop, motor halted
9	rm	Remote, 1=*Controlword* executed
8	h	Halt, 1=Active request
7	w	Warning, 1= Presence of one or more warnings
6	sod	Switch on disabled
5	qs	Quick stop, 0= Quick Stop procedure in progress or concluded
4	ve	Voltage enabled, 1= Power supply applied to the device
3	f	Fault, 1= Error or Fault procedure in progress or concluded
2	oe	Operation enabled, 1= Motor enabled
1	so	Switched on, 1= Power stage of the device powered
0	rsto	Ready to switch on, 1= Device ready to supply the power stage
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The minimum steps required to rotate the motor in the *Profile velocity* mode are the configuration of the motion profile (acceleration and deceleration) and the setting of the target velocity. This can be carried out through the communication objects SDO, PDO or a combination of the two.

6.5.3 Profile torque mode (tq)

The *Profile torque mode* can be used only when the integrated motor is equipped with encoder and allows you to control the torque available at the motor shaft. The master controller can set the torque updating the *Target_torque* object. The torque variation on the motor shaft always occurs according to the ramp set through the *Torque_slope* object.

In this mode the speed of the motor is limited only by the maximum one allowed by the motor itself and by the value of the 2310_h:$0B_h$ *Max_Speed_MTRDT* object.

The main objects involved in the *Profile torque* mode are shown in the table below:

Objects associated with the operative *Profile position* mode			
OD Entry	Name	Unit Data type PDO	Description
6040_h	*Controlword*	--- UINT16 RPDO	Command controlling the FSA.
6041_h	*Statusword*	--- UINT16 RPDO	Provide the status of the FSA
6060_h	*Modes_of_operation*	--- INT8 RPDO	Requested operation mode
6061_h	*Modes_of_operation_display*	--- INT8 TPDO	Actual operation mode
6071_h	*Target_torque*	0.1% INT16 RPDO	Indicate the configured input value for the torque controller
6077_h	*Torque_actual_value*	0.1% INT16 TPDO	Provide the actual value of the available torque on motor shaft
6087_h	*Torque_slope*	0.1%/s UINT32 TPDO	Indicate the configured rate of change of torque

To activate the *Profile torque* mode you need to write the *Modes_of_operation* (6060_h) object with the value 04_h.

The *Controlword* object allows you to enable or disable the motor while the *Statusword* object provides information on the status of the motor and the movement in progress.

The following table shows the *Controlword* object and the meaning of its component bits.

Operative mode															
Profile torque mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							h	fr				eo	qs	ev	so

Controlword bits organization		
Bit		Description
15		Reserved, set to 0
14		Reserved, set to 0
13		Reserved, set to 0
12		Reserved, set to 0
11		Reserved, set to 0
10		Reserved, set to 0
9		Reserved, set to 0
8	h	Halt
7	fr	Fault reset
6		Reserved, set to 0
5		Reserved, set to 0
4		Reserved, set to 0
3	eo	Enable operation
2	qs	Quick stop
1	ev	Enable voltage
0	so	Switch on
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The following table shows the *Statusword* object and the meaning of its component bits.

Operative mode															
Profile torque mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Statusword bits organization		
Bit		Description
15		Reserved, ignore the value
14		Reserved, ignore the value
13		Reserved, ignore the value
12		Reserved, ignore the value
11	ila	Internal limit active, 1= Restriction of one or more parameters for internal limit
10	tr	Target reached, 1= Target position reached. In case of Halt, null torque. In case of QuickStop, motor stopped
9	rm	Remote, 1=*Controlword* executed
8	h	Halt, 1=Active request
7	w	Warning, 1= Presence of one or more warnings
6	sod	Switch on disabled
5	qs	Quick stop, 0= Quick Stop procedure in progress or concluded
4	ve	Voltage enabled, 1= Power supply applied to the device

3	f	Fault, 1= Error or Fault procedure in progress or concluded
2	oe	Operation enabled, 1=Motor enabled
1	so	Switched on, 1= Power stage of the device powered
0	rsto	Ready to switch on, 1= Device ready to supply the power stage
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The minimum steps required to regulate the torque to the motor shaft in the *Profile torque* mode are the configuration of the ramp and the setting of the target torque. This can be done through the communication objects SDO, PDO or a combination of the two.

6.5.4 Homing mode (hm)

Through the Homing mode the integrated motor is able to find the zero position (also called reference). It is possible to choose among various homing methods which make use of limit switches (right and left), home switches, encoder index pulse or a combination of them.

To perform the homing procedure, the master controller must first configure the homing method through the *Homing_method* object and then start the homing by setting to 1 the *Homing_operation_start* bit in the *Controlword* object.

Two bits contained in the *Statusword* object inform the master controller on the status of the motor; the bit *Target_reached* becomes active when the homing procedure is concluded, the bit *Homing_attained* is active when the zero position is found and is valid, while the bit *Homing_error* becomes active if an error occurred during the procedure.

The main objects involved in the *Homing mode* are shown in the table below:

Object associated with the operative *Homing* mode			
OD Entry	Name	Unit Data type PDO	Description
6040_h	*Controlword*	--- UINT16 RPDO	Command controlling the FSA.
6041_h	*Statusword*	--- UINT16 RPDO	Provide the status of the FSA
6060_h	*Modes_of_operation*	--- INT8 RPDO	Requested operation mode
6061_h	*Modes_of_operation_display*	--- INT8 TPDO	Actual operation mode
6064_h	*Position_actual_value*	0.0001rev INT32 TPDO	Provide the actual value of the position measurement device
$607C_h$	*Home_offset*	0.0001rev INT32 RPDO	Indicate the configured difference between the zero position for the application and the machine home position
6098_h	*Homing_method*	--- INT8 RPDO	Indicate the configured homing method
$6099:01_h$	*Speed_during_search_for_switch*	0.1rpm UINT16 RPDO	Speed during search for switch
$6099:02_h$	*Speed_during_search_for_zero*	0.1rpm UINT32 RPDO	Speed during search for zero

To activate the *Homing* mode you need to write the *Modes_of_operation* (6060_h) object with the value 06_h.

The *Controlword* object allows you to enable or disable the motor and start the search for zero, while the *Statusword* object provides information on the status of the motor and the movement in progress.

The following table shows the *Controlword* object and the meaning of its component bits.

Operative mode															
Homing mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							h	fr			hos	eo	qs	ev	so

Controlword bits organization		
Bit		Description
15		Reserved, set to 0
14		Reserved, set to 0
13		Reserved, set to 0
12		Reserved, set to 0
11		Reserved, set to 0
10		Reserved, set to 0
9		Reserved, set to 0
8	h	Halt
7	fr	Fault reset
6		Reserved, set to 0
5		Reserved, set to 0
4	hos	Homing operation start
3	eo	Enable operation
2	qs	Quick stop
1	ev	Enable voltage
0	so	Switch on
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The following table shows the *Statusword* object and the meaning of its component bits.

Operative mode															
Homing mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		he	ha	ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Statusword bits organization		
Bit		Description
15		Reserved, ignore the value
14		Reserved, ignore the value
13	he	Homing error, 1= Error in the search for zero
12	ha	Homing attained, 1= Zero found and valid
11	ila	Internal limit active, 1= Restriction of one or more parameters for internal limit
10	tr	Target reached, 1= Target position reached. In case of Halt or QuickStop, motor halted
9	rm	Remote, 1=*Controlword* executed
8	h	Halt, 1=Active request
7	w	Warning, 1= Presence of one or more warnings
6	sod	Switch on disabled

5	qs	Quick stop, 0= Quick Stop procedure in progress or concluded
4	ve	Voltage enabled, 1= Power supply applied to the device
3	f	Fault, 1= Error or Fault procedure in progress or concluded
2	oe	Operation enabled, 1=Motor enabled
1	so	Switched on, 1= Power stage on the device powered
0	rsto	Ready to switch on, 1= Device ready to supply the power stage
For a complete description on the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/		

The motor with integrated drive can perform the homing procedure in many different modes which make use of limit switches (positive and negative), home switches, encoder index pulse or a combination of them.

The sensors connected with the motor must be associated with the corresponding digital input so that they are correctly read and the homing procedure successfully completed. To associate the sensors with the inputs, use the 2810_h *Digital_Input_Action* object.

The mode used by the motor to perform the homing is selectable through the *Homing_method* object. The following table describes the homing methods available and the sensors used by each one of them.

The abbreviations used have the following meaning: **PLS**=positive limit switch, **NLS**=negative limit switch, **HS**=homing switch, **IDX**=Index.

Homing methods selectable through the *Homing_method* object					
Code.	Description	Sensors used			
		PLS	NLS	HS	IDX
-80	Homing at the position read by the SSI encoder				
1	At the start, if negative limit switch inactive counterclockwise direction up to the limit switch, then reverse and homing at the first index outside the negative limit switch. At the start, if negative limit switch active clockwise direction up to leave the limit switch, then homing at the first index outside the negative limit switch.		●		●
2	At the start, if positive limit switch inactive clockwise direction up to the limit switch, then reverse and homing at the first index outside the positive limit switch. At the start, if positive limit switch active counterclockwise direction up to leave the limit switch, then homing at the first index outside the positive limit switch.	●			●
3	At the start, if home switch inactive initial direction clockwise up to home switch, then reverse and homing at the first index outside the home switch. At the start, if home switch active initial direction counterclockwise up to leave the switch, then homing at the first index outside the home switch.			●	●
4	At the start, if home switch inactive initial direction clockwise up to home switch, then homing at the first index inside the home switch. At the start, if home switch active initial direction counterclockwise up to leave the switch, then reverse and homing at the first index inside the home switch.			●	●

5	At the start, if home switch active initial direction clockwise up to leave the switch, then homing at the first index outside the home switch. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then reverse and homing at the first index outside the home switch.			●	●
6	At the start, if home switch active initial direction clockwise up to leave the switch, then reverse and homing at the first index inside the home switch. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then reverse and homing at the first index inside the home switch.			●	●
7	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, reverse and homing at the first index outside the switch. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then homing at the first index outside the switch. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then homing at the first index outside the switch.	●		●	●
8	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, homing at the first index inside the home switch. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then reverse up to find again the home switch and finally homing at the first index inside the switch. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then reverse and homing at the first index inside the switch.	●		●	●
9	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it then reverse and homing at the first index inside the switch. In case of positive limit switch, reverse up to find the home switch, then homing at the first index inside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the first index inside the home switch.	●		●	●
10	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it then homing at the first index outside the home switch. In case of positive limit switch, reverse up to find the home switch, then reverse up to leave the switch and homing at the first index outside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the first index outside the home switch.	●		●	●
11	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, reverse and homing at the first index outside the home switch. In case of negative limit switch, reverse up to find the home switch, continue up to leave the home switch and then homing at the first index outside the switch.		●	●	●

	At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the first index outside the home switch.				
12	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, homing at the first index inside the home switch. In case of negative limit switch, reverse up to find the home switch, continue up to leave the home switch and then reverse up to find again the home switch and finally homing at the first index inside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the first index inside the home switch.		●	●	●
13	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and homing at the first index inside the switch. In case of negative limit switch, reverse up to find the home switch, then homing at the first index inside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the first index inside the home switch.		●	●	●
14	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then homing at the first index outside the home switch. In case of negative limit switch, reverse up to find the home switch, reverse up to leave the switch and then homing at the first index outside the switch. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then homing at the first index outside the switch.		●	●	●
17	At the start, if negative limit switch inactive initial direction counterclockwise up to find the limit switch, then reverse and homing at the active/inactive switch transition. At the start, if negative limit switch active initial direction clockwise with homing at the active/inactive switch transition.		●		
18	At the start, if positive limit switch inactive initial direction clockwise up to the limit switch, then reverse and homing at the active/inactive switch transition. At the start, if negative limit switch active initial direction counterclockwise with homing at the active/inactive switch transition.	●			
19	At the start, if home switch inactive initial direction clockwise up to the home switch, then reverse and homing at the active/inactive switch transition. At the start, if home switch active initial direction counterclockwise with homing at the active/inactive switch transition.			●	
20	At the start, if home switch inactive initial direction clockwise up to the home switch, then homing at the inactive/active switch transition. At the start, if home switch active initial direction counterclockwise up to leave the switch, then reverse and homing at the inactive/active switch transition.			●	
21	At the start, if home switch active initial direction clockwise up to leave the switch, then homing at the active/inactive switch transition. At the start, if home switch inactive initial direction counterclockwise up			●	

	to find the switch, then reverse and homing at the active/inactive switch transition.				
22	At the start, if home switch active initial direction clockwise up to leave the switch, then reverse and homing at the inactive/active switch transition. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then homing at the inactive/active switch transition.			●	
23	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, reverse and homing at the active/inactive switch transition. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then homing at the active/inactive switch transition. At the start, if home switch active initial direction counterclockwise up to leave the switch, then homing at the active/inactive switch transition.	●		●	
24	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, homing at the inactive/active switch transition. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then reverse up to find again the home switch and finally homing at the inactive/active switch transition. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.	●		●	
25	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then reverse and homing at the inactive/active switch transition. In case of positive limit switch, reverse up to find the home switch, homing at the inactive/active switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.	●		●	
26	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then homing at the active/inactive switch transition. In case of positive limit switch, reverse up to find the home switch, then reverse up to leave the switch and finally homing at the active/inactive switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the active/inactive switch transition.	●		●	
27	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, reverse and homing at the active/inactive switch transition. In case of negative limit switch, reverse up to find the home switch, continue up to leave the home switch, then homing at the active/inactive switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the active/inactive switch transition.		●	●	
28	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, homing at the inactive/active switch transition. In case of		●	●	

	negative limit switch, reverse up to find the home switch, continue up to leave the home switch, then reverse up to find again the home switch and finally homing at the inactive/active switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.				
29	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and homing at the inactive/active switch transition. In case of negative limit switch, reverse up to find the home switch, then homing at the inactive/active switch transition. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.		●	●	
30	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then homing at the active/inactive switch transition. In case of negative limit switch, reverse up to find the home switch, then reverse up to leave the switch, then homing at the active/inactive switch transition. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then homing at the active/inactive switch transition.		●	●	
33	Initial direction counterclockwise with homing at the first index found.				●
34	Initial direction clockwise with homing at the first index found.				●
37	Homing at the actual position.				

The minimum steps required to execute a homing are the configuration of the operative *Homing* mode, the setting of acceleration and deceleration, of the speed during search for switch and for homing, the selection of the homing method and the homing start through the bit *Homing_operation_start*. This can be carried out through SDO, PDO or a combination of the two.

If the homing is successfully completed the *Position_actual_value* object is set to the value of the *Home_offset* object. Subsequent modifications to the *Home_offset* object do not modify the 0 reference.

6.5.5 Cyclic synchronous position (csp)

Selecting the *Cyclic synchronous position* (*csp*) mode, it is possible to command the motor with integrated drive to execute a positioning path by sending consecutive set points. For an optimal operation the set points must be sent with a known and regular frequency (period, cycle time), moreover if the application involves multiple axes interpolated together the set points must be applied synchronously to all the axes involved in the interpolation. For this reason it is recommended the use of the synchronization in DC SYNC0 mode, that allows processing the PDO synchronized to the Sync signal.

The motor with integrated drive has an internal micro-interpolator able to interpolate the path between two consecutive set points with micro position intermediate positions in order to improve the motor movement and make it continuous, even with slow updating frequencies. Furthermore, through the *Dampening_A_Value_IPCNF* object it is possible to filter possible trajectory discontinuities, due for example to an irregular update of the set points, and thus improve the movement.

The motor speed is limited by the *Max_Speed_MTRDT* Object. Optionally, it is also possible to enable the limitation of acceleration and deceleration (which can be configured through the *Profile_acceleration* and *Profile_deceleration* objects respectively).

Finally, a complete monitoring system has been set up to verify the correct receipt of the Sync signal able to detect the reception in advance or in delay with respect to the nominal period. Modifying the alarm threshold it is possible to trigger an emergency at the first violation of the timing or to tolerate sporadic violations by reserving the alarm condition only for repeated violations in the reception of the synchronism. All the objects for configuring and monitoring the Sync are included in the record *Sync Guard*.

The main standard objects involved in the *Cyclic synchronous position* mode are shown in the table below:

Object associated to the operative *Cyclic synchronous position* mode			
OD Entry	Name	Unit Data type PDO	Description
6040_h	*Controlword*	--- UINT16 RPDO	Command controlling the FSA
6041_h	*Statusword*	--- UINT16 RPDO	Provide the status of the FSA
6060_h	*Modes_of_operation*	--- INT8 RPDO	Requested operation mode
6061_h	*Modes_of_operation_display*	--- INT8 TPDO	Actual operation mode
6062_h	*Position_demand_value*	0.0001rev INT32 TPDO	Provide the demanded position value
6064_h	*Position_actual_value*	0.0001rev INT32 TPDO	Provide the actual value of the position measurement device
6065_h	*Following_error_window*	0.0001rev UINT32 RPDO	Indicate the configured range of tolerated position values symmetrically to the position demand value

6066 h	*Following_error_time_out*	Ms UINT16 RPDO	Indicate the configured time for a following error condition, after that the bit 13 of the *Statusword* is set to 1
6067 h	*Position_window*	0.0001rev UINT32 RPDO	Indicate the configured symmetrical range of accepted positions relative to target position
6068 h	*Position_window_time*	Ms UINT16 RPDO	Indicate the configured time, during which the actual position within the position window is measured
607A h	*Target_position*	0.0001rev INT32 TPDO	Indicate the commanded position to reach
607D h:01h	*Min_software_position_limit*	0.0001rev INT32 TPDO	Min position range limit
607D h:02h	*Max_software_position_limit*	0.0001rev INT32 TPDO	Max position range limit
6083 h	*Profile_acceleration*	rpm/s UINT32 RPDO	Indicate the configured acceleration
6084 h	*Profile_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration
60C2h :01h	*Interpolation_time_period_value*	--- UINT8 RPDO	Cycle time, value.
60C2h :02h	*Interpolation_time_index*	--- INT8 RPDO	Cycle time, exponent
60F4 h	*Following_error_actual_value*	0.0001rev INT32 TPDO	Indicate the actual value of the following error

The *Controlword* object allows to enable or disable the motor and to start the set-points process.

The *Statusword* object gives information about the status of the drive and of the interpolation.

The following table shows the *Controlword* object and the meaning of its component bits.

Operative mode

Cyclic synchronous position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	0	fr	r	r	r	eo	1	ev	so

Bit	Description		Bit	Description	
15	r	Reserved	7	fr	Fault reset
14	r	Reserved	6	r	Reserved
13	r	Reserved	5	r	Reserved
12	r	Reserved	4	r	Reserved
11	r	Reserved	3	eo	Enable operation
10	r	Reserved	2	1	Set to 1
9	r	Reserved	1	ev	Enable voltage
8	0	Seto to 0	0	so	Switch on

For a complete description of the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/

The following table shows the *Statusword* object and the meaning of its component bits.

Operative mode

Cyclic synchronous position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		fe	dfcv	ila	r	rm	r	w	sod	r	ve	f	oe	so	rsto

Bit	Description		Bit	Description	
15	r	Reserved	7	w	Warning
14	r	Reserved	6	sod	Switch on disabled
13	fe	Following error	5	r	Reserved
12	dfcv	Drive follows the command value	4	ve	Voltage enabled
11	ila	Internal limit active	3	f	Fault
10	r	Reserved	2	oe	Operation enabled
9	rm	Remote	1	so	Switched on
8	r	Reserved	0	rsto	Ready to switch on

For a complete description of the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/

The *Cyclic synchronous position* mode is selected writing the 6060_h *Modes_of_operation* object with the value 08_h.

The minimum steps required to control the motor in *Cyclic synchronous position* mode are limited to the setting of the cycle time through the *Interpolation_time_period_value* and *Interpolation_time_index* objects.

To obtain a correct movement of the motor it is fundamentally important to initialize the *Interpolation_time_period_value* object with the identical cycle time set on the master controller.

The described operations can be carried out through the communication objects SDO, PDO or a combination of the two. Even the setting of the set-point can be controlled through SDO or PDO, however we recommend the use of the PDO as they have a higher transmission priority and allow the synchronization of more devices through the DC SYNC0 synchronism.

6.5.6 Cyclic synchronous velocity (csv)

Selecting the *Cyclic synchronous velocity* (*csv*) mode, it is possible to command the drive to execute a velocity profile by sending consecutive set points. For an optimal operation the set points must be sent with a known and regular frequency (period, cycle time), moreover if the application involves multiple axes interpolated together the set points must be applied synchronously to all the axes involved in the interpolation. For this reason it is recommended the use of the synchronization in DC SYNC0 mode, that allows processing the PDO synchronized to the Sync signal.

The drive has an internal micro-interpolator able to interpolate the velocity between two consecutive set points in order to improve the motor movement and make it continuous, even with slow updating frequencies.

The motor speed is limited by the *Max_Speed_MTRDT* object. Optionally, through the *Mode_CSVCNF* it is also possible to enable the limitation of acceleration and deceleration (which can be configured through the *Profile_acceleration* and *Profile_deceleration* objects respectively).

Finally, a complete monitoring system has been set up to verify the correct receipt of the Sync signal able to detect the reception in advance or in delay with respect to the nominal period. Modifying the alarm threshold it is possible to trigger an emergency at the first violation of the timing or to tolerate sporadic violations by reserving the alarm condition only for repeated violations in the reception of the synchronism. All the objects for configuring and monitoring the Sync are included in the record *Sync Guard*.

The main standard objects involved in the *Cyclic synchronous velocity* mode are shown in the table below:

Object associated to the operative *Cyclic synchronous velocity* mode			
OD Entry	Name	Unit Data type PDO	Description
6040_h	*Controlword*	--- UINT16 RPDO	Command controlling the FSA
6041_h	*Statusword*	--- UINT16 RPDO	Provide the status of the FSA
6060_h	*Modes_of_operation*	--- INT8 RPDO	Requested operation mode
6061_h	*Modes_of_operation_display*	--- INT8 TPDO	Actual operation mode
6064_h	*Position_actual_value*	0.0001rev INT32 TPDO	Provide the actual value of the position measurement device
$606C_h$	*Velocity_actual_value*	0,1rpm INT32 TPDO	Actual velocity value
6077_h	*Torque_actual_value*	0,1% INT16 TPDO	Actual torque available at the shaft expressed as a percentage of the nominal torque
6083_h	*Profile_acceleration*	rpm/s UINT32	Indicate the configured acceleration

		RPDO	
6084 h	*Profile_deceleration*	rpm/s UINT32 RPDO	Indicate the configured deceleration
60C2h :01h	*Interpolation_time_period_value*	--- UINT8 RPDO	Cycle time, value
60C2h :02h	*Interpolation_time_index*	--- INT8 RPDO	Cycle time, exponent
3450h :01h	*Max_Slippage_TimeOut_PVCNF*	ms UINT16 RPDO	Time with active slippage error after which the drive signals an error
60F8 h	*Max_slippage*	0,1rpm INT32 TPDO	Maximum error allowed between the velocity of the motor and the one required by the trajectory generator
60FF h	*Target_velocity*	0,1rpm INT32 TPDO	Target velocity

The *Controlword* object allows to enable or disable the motor.

The *Statusword* object gives information about the status of the drive.

The following table shows the *Controlword* object and the meaning of its component bits.

Mode of operation *Cyclic synchronous velocity mode*															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	r	r	r	r	r	0	fr	r	r	r	eo	1	ev	so

Bit	Description		Bit	Description	
15	r	Reserved	7	fr	Fault reset
14	r	Reserved	6	r	Reserved
13	r	Reserved	5	r	Reserved
12	r	Reserved	4	r	Reserved
11	r	Reserved	3	eo	Enable operation
10	r	Reserved	2	1	Set to 1
9	r	Reserved	1	ev	Enable voltage
8	0	Set to 0	0	so	Switch on

For a complete description of the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/

The following table shows the *Statusword* object and the meaning of its component bits.

Mode of operation															
Cyclic synchronous velocity mode															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r	r	fe	dfcv	ila	r	rm	r	w	sod	r	ve	f	oe	so	rsto

Bit	Description		Bit	Description	
15	r	Reserved	7	w	Warning
14	r	Reserved	6	sod	Switch on disabled
13	fe	Following error	5	r	Reserved
12	dfcv	Drive follows the command value	4	ve	Voltage enabled
11	ila	Internal limit active	3	f	Fault
10	r	Reserved	2	oe	Operation enabled
9	rm	Remote	1	so	Switched on
8	r	Reserved	0	rsto	Ready to switch on

For a complete description of the meaning and use of the bits, refer to the official documentation /CiA301/ and /CiA402/ available on the CAN in Automation (CiA) website at the address https://www.can-cia.org/

The *Cyclic synchronous velocity* mode is selected writing the 6060_h *Modes_of_operation* object with the value 09_h.

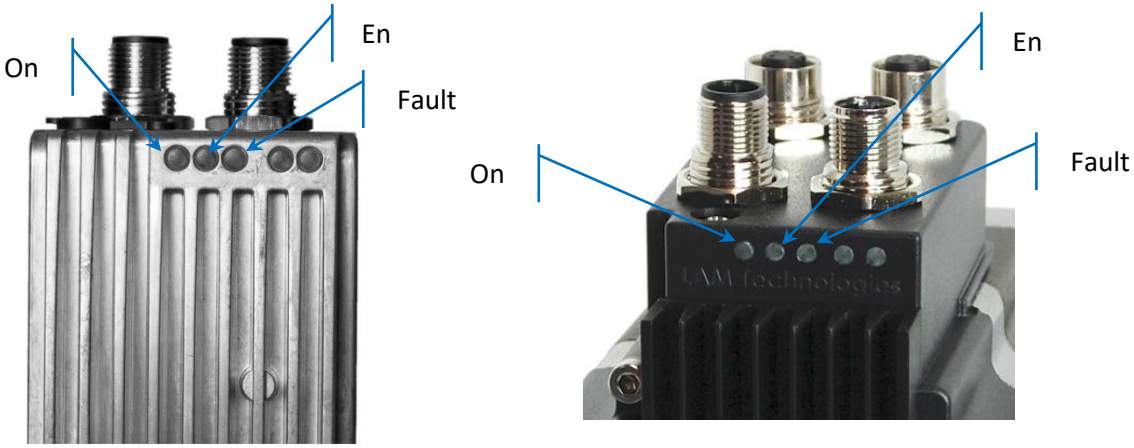
The minimum steps required to control the motor in *Cyclic synchronous velocity* mode are limited to the setting of the cycle time through the *Interpolation_time_period_value* and *Interpolation_time_index* objects.

To obtain a correct movement of the motor it is fundamentally important to initialize the *Interpolation_time_period_value* objects with the identical cycle time set on the master controller.

The described operations can be carried out through the communication objects SDO, PDO or a combination of the two. Even the setting of the set-point can be controlled through SDO or PDO, however we recommend the use of the PDO as they have a higher transmission priority and allow the synchronization of more devices through the DC SYNC0 synchronism.

7 Errors and diagnostics

The motor with integrated drive is equipped with 3 LEDs, positioned in the rear part, which provide information on the status of the power supply, the motor enabling and the presence of errors.



7.1 LED *On*

The green On LED provides information on the status of the power supply, as indicated in the table below:

Led *On*	
Off	Integrated motor not supplied
On	Integrated motor supplied

7.2 LED *En*

The yellow En LED provides information on the status of the motor, as indicated in the table below:

Led *En*	
Off	Motor not enabled
Fixed on	Motor enabled stopped
Flashing	Motor enabled in rotation

7.3 LED *Fault*

The red Fault LED is used to indicate presence of errors in the integrated motor, as shown in the table below:

Led *Fault*	
Off	No error
Fixed on	Corrupt firmware. Active Loader.
Flashing	Presence of errors. The number of flashes indicates the type of error as detailed below.

7.4 Errors

The motor with integrated drive is able to detect many error conditions that are signaled by the red LED Fault and the fieldbus and, and configuring appropriately the *Source_DOA_x* object, also through the digital outputs.

The errors are divided into classes. Each class identifies a specific reaction of the integrated motor, according to the table below:

Class	Description	Integrated motor's reaction
0	Warning, there is no impediment to continue with the operations in progress.	None.
2	Error which requires the stop of the motor but not the transition to FAULT.	Deceleration with Quick Stop.
4	Error which requires the stop of the motor and successively the transition to FAULT.	Deceleration with Quick Stop and subsequent motor disabling.
6	Error which requires the immediate disabling of the motor and the transition to FAULT.	Motor disabling.
8	As per class 6. The error can be reset only through a cycle of turning off and on.	Motor disabling.

For error class 8, note that the power off/on cycle must consider the eventual auxiliary power supply and the connection to the DUP port, which can keep the logic part supplied and thus prevent the reset of the motor with integrated drive.

Each time an error occurs, the integrated drive sends an emergency request (*Emergency Request*) containing the code of the detected error (2310_h *Error Code*), the content of the 1001_h *Error Register* object, the class of the error and in some cases a sub-code which helps to identify the origin of the problem.

The following table shows the organization of the *Emergency* message (for a detailed description see the document ETG.1000.6)

$603F_h$ *Error Code* 2 bytes	1001_h *Error Register* 1 byte	*Class* 1 byte	*Sub-code* 4 bytes
Error code	*Error Register*	Class	Sub-code

The errors are grouped by affinity and displayed to the user through a different number of flashes of the red LED *Fault*.

The following table summarizes the errors recognized by the integrated motor:

<table>
<tr><th>Error code</th><th>Class</th><th>LED Flash</th><th>bit</th><th>Description / Sub-code</th></tr>
<tr><td>2310_h</td><td>6</td><td>4</td><td>3</td><td>Detected motor over current
<table>
<tr><th>Sub-code</th><th>Description</th></tr>
<tr><td>0100_h</td><td>During PWM modulation</td></tr>
<tr><td>0101_h</td><td>Towards ground</td></tr>
<tr><td>0102_h</td><td>Towards positive power bus</td></tr>
<tr><td>0103_h</td><td>Short circuit between phase A+ and ground</td></tr>
<tr><td>0104_h</td><td>Short circuit between phase A- and ground</td></tr>
<tr><td>0105_h</td><td>Short circuit between phase B+ and ground</td></tr>
<tr><td>0106_h</td><td>Short circuit between Phase B- and ground</td></tr>
<tr><td>0109_h</td><td>Abnormal current between phase A+ and ground</td></tr>
<tr><td>$010A_h$</td><td>Abnormal current between phase A- and ground</td></tr>
<tr><td>$010B_h$</td><td>Abnormal current between phase B+ and ground</td></tr>
<tr><td>$010C_h$</td><td>Abnormal current between phase B- and ground</td></tr>
</table></td></tr>
<tr><td>$23A0_h$</td><td>6</td><td>5</td><td>5</td><td>Detected motor phase open (not connected or interrupted)
Through the 2120_h:01_h OpenPhase_EFSTP object it is possible to disable the open phase condition test.
<table>
<tr><th>Sub-code</th><th>Description</th></tr>
<tr><td>0201_h</td><td>Phase A</td></tr>
<tr><td>0202_h</td><td>Phase B</td></tr>
</table></td></tr>
<tr><td>$23B2_h$</td><td>6</td><td>8</td><td>4</td><td>Detected over current on the encoder power supply +V</td></tr>
<tr><td>3210_h</td><td>6</td><td>2</td><td>1</td><td>Power supply voltage higher than the maximum allowed</td></tr>
<tr><td>3220_h</td><td>6</td><td>1</td><td>0</td><td>Power supply voltage lower than the minimum necessary for a correct operation</td></tr>
<tr><td>4210_h</td><td>4</td><td>3</td><td>2</td><td>Power stage temperature higher than the maximum allowed value</td></tr>
<tr><td>5592_h</td><td>8</td><td>13</td><td>24</td><td>Invalid device descriptor</td></tr>
<tr><td>5594_h</td><td>8</td><td>13</td><td>24</td><td>Invalid device configuration</td></tr>
<tr><td>$55A0_h$</td><td>8</td><td>13</td><td>31</td><td>Inappropriate device installed firmware</td></tr>
<tr><td>$55A2_h$</td><td>8</td><td>13</td><td>31</td><td>Incompatible device firmware revision</td></tr>
<tr><td>6200_h</td><td>8</td><td>15</td><td>31</td><td>Firmware error</td></tr>
<tr><td>$63A0_h$</td><td>6</td><td>10</td><td>16</td><td>Conflicts in the configuration</td></tr>
</table>

				Sub-code / Description 0701h – Selected *Torque mode* without encode feedback (present or configured). 0702h – Selected current proportional to the load (*Mode_CRRG* object) without encoder feedback (present o configured). 0703h – Set interpolation time (objects …and….) out of allowed limits. 0704h – Activated encoder feedback without having activated or configured the encoder. 0705h – Activated the following error control without having activated or configured the encoder selected as feedback source. 0706h – Activated the incremental encoder Index management and contemporaneously the SSI encoder.
7200h	8	14	10	Measurement circuit fault Sottocodice / Descrizione 0411h – Fault in the power stage temperature measurement circuit 0412h – Fault in the motor temperature measurement circuit
7282h	8	14	11	Detected invalid offset values
8100h	6	7	28	Generic bus error
81E0h	6	7	28	EtherCAT bus error Sub-code / Description 0B01h – Process data watchdog expired 0B02h – Synchronization error 0B03h – Absent synchronism
82E0h	6	7	22	EtherCAT State Machine error Sub-code / Description 0C01h – Exit from Operational status with motor enabled
82E1h	6	7	22	RPDO processing error
82E2h	6	7	22	TPDO processing error
82E3h	6	7	22	SDO download error
82E4h	6	7	22	SDO upload error
8240h	6	7	22	SYNC of unexpected duration
82B2h	6	7	22	Violation of the SYNC period
8491h	6	6	18	Slippage error. It occurs when the actual motor velocity deviates from the requested one of an amount, in absolute value, greater than the value set in the *Max_slippage* object for a time longer than the value set in the *Max_Slippage_TimeOut_PVCNF* object.
8611h	6	6	18	Following error. The indicated class expresses the default value. The class can be modified through the *Following_Error_ERRCS* object.
8612h	2	6	19	Limit switch activation

<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th>Sub-code</th><th>Description</th></tr>
<tr><td>0801_h</td><td>Positive limit activation</td></tr>
<tr><td>0802_h</td><td>Negative limit activation</td></tr>
<tr><td>0803_h</td><td>Positive movement with active positive limit</td></tr>
<tr><td>0804_h</td><td>Negative movement with active negative limit</td></tr>
</table>
</td><td></td></tr>
<tr><td>8613_h</td><td>2</td><td>6</td><td>17</td><td>Homing error
<table>
<tr><th>Sub-code</th><th>Description</th></tr>
<tr><td>0601_h</td><td>Unfound index</td></tr>
<tr><td>0602_h</td><td>Unexpected clockwise limit</td></tr>
<tr><td>0603_h</td><td>Unexpected counterclockwise limit</td></tr>
<tr><td>0604_h</td><td>The selected homing mode uses the incremental encoder Index but the management of the Index itself is not active (Configuration_ENCMTR object)</td></tr>
<tr><td>0605_h</td><td>The selected homing mode uses the SSI encoder but the SSI encoder reading is not operative.</td></tr>
</table>
</td><td></td></tr>
</table>

8 Objects Dictionary

The objects dictionary accessible through the EtherCAT protocol can be logically divided into three areas. The objects with index between 1000_h and $1FFF_h$ mainly related to the communication, the objects with index between 2000_h and $5FFF_h$ specific of LAM Technologies products and described in this manual; finally the objects between 6000_h and $9FFF_h$ standardized by the profile /CiA402/.

Many dictionary objects have a default value, however two devices of the same type may have different default values, due for example to the use of the *Store Parameters* functions. It is therefore suggested to always initialize any dictionary object used in the application with the desired value, independently from the default.

Usually, the master controller provides the possibility to initialize the dictionary objects with the wanted values during the transition PS (da *Pre-Operational* a *Safe Operational*).

In the following description the numbers in hexadecimal format will be identified by the subscript h (for example, 5A is equivalent to 90 in decimal format).

8.1 Object with index between 1000_h and $1FFF_h$

The objects with index between 1000_h and $1FFF_h$ are mainly useful to configure the communication and the services provided by the CANopen standard.

Their function, the access mode, etc. are described in the manuals of the /CiA301/ standard available on the official site www.can-cia.org and in the documents ETG 1000 and ETG 1020 available on the website www.ethercat.org, therefore in the present manual there will not be a detailed description of each object but a brief summary of the implemented ones.

8.1.1 1000_h Device type

Index	Name			Mnemonic
1000_h	*Device_type*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	0x00040192			

Description

This object provides information about the device type. The object describes the type of the logical device and its functionality. It is made up of a 16-bit field that describes the device profile or the application profile that is used and a second 16-bit field, which gives additional information about optional functionality of the logical device. The additional information parameter is device profile specific and application profile specific.

Bits 31..16	bits 15..0
Additional information	Device profile number
0x0004 = Stepper Driver	0x0192 = /CiA402/

8.1.2 1001_h Error register

Index	Name			Mnemonic
1001_h	*Error_register*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

This object provides error information. The device maps internal errors into this object. It is a part of an emergency object.

It is made up of an 8-bit field that describes the error. The table below shows the bits mapping:

0	Generic error
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Device profile specific
6	Reserved
7	Manufacturer specific

8.1.3 1008_h Manufacturer device name

Index	Name			Mnemonic
1008_h	*Manufacturer_device_name*			
	Data Type	Access Type	PDO Mapping	Note
	str	const		
	Default Value	Minimum	Maximum	Unit
	Description			
	This object provides the name of the device.			

8.1.4 1009_h Manufacturer hardware version

Index	Name			Mnemonic
1009_h	*Manufacturer_hardware_version*			
	Data Type	Access Type	PDO Mapping	Note
	str	const		
	Default Value	Minimum	Maximum	Unit
	Description			
	This object provides the hardware version of the device.			

8.1.5 $100A_h$ Manufacturer software version

Index	Name			Mnemonic
$100A_h$	*Manufacturer_software_version*			
	Data Type	Access Type	PDO Mapping	Note
	str	const		
	Default Value	Minimum	Maximum	Unit
	Description			
	This object provides the software version of the device.			

8.1.6 1010_h Store Parameters

It allows to save many dictionary objects in the non-volatile memory of the device. The dictionary objects that can be saved are marked by the symbol 🖫 in the field *Note*.

When the value of an object is saved in the non-volatile memory, it is automatically restored at the power on or in case of device reset.

The defaults values can be saved or restored only with the motor disabled or with the EtherCAT State Machine (ESM) in the status *Init* or *Pre-Operational*. Trying the operation in a different status an error code answer is received.

At most it is possible to save and restore the default values for 10,000 times. The savings is completed in about 100ms.

Index	Name	Type
1010_h	*Store Parameters*	ARRAY

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		The default values can be saved or restored only with the motor disabled. Trying the operation with the motor enabled, an error code is obtained in response.
	Default Value	Minimum	Maximum	Unit
	04_h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Save all Parameters*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit
	Description			
	It allows to save the dictionary objects with index between 1000_h and $9FFF_h$, marked by the symbol 🖫, in the non-volatile memory. To start and save simply write the object with the value 0x65766173.			

Sub-Index	Name			Mnemonic
02_h	*Save Communication Parameters*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit
	Description			
	It allows to save the dictionary objects with index between 1000_h and $1FFF_h$, specified with the symbol 🖫, in the non-volatile memory.			

	To start and save simply write the object with the value 0x65766173.

Sub-Index	Name		Mnemonic
03_h	*Save Application Parameters*		
Data Type	Access Type	PDO Mapping	Note
u32	rw		
Default Value	Minimum	Maximum	Unit
Description			
It allows to save the dictionary objects with index between 6000_h and $9FFF_h$, specified with the symbol 🖫, in the non-volatile memory. To start and save simply write the object with the value 0x65766173.			

Sub-Index	Name		Mnemonic
04_h	*Save Manufacturer Defined Parameters*		
Data Type	Access Type	PDO Mapping	Note
u32	rw		
Default Value	Minimum	Maximum	Unit
Description			
It allows to save the dictionary objects with index between 2000_h and $5FFF_h$, specified with the symbol 🖫, in the non-volatile memory. To start and save simply write the object with the value 0x65766173.			

8.1.7 1011_h Restore Default Parameters

It allows to restore the default value for the objects saved using the *Store Parameters* function.

The default values can be restored only with the motor disabled or with the EtherCAT State Machine (ESM) in the status *Init* or *Pre-Operational*. Trying the operation in a different status an error code answer is received.

At most it is possible to restore the default values 10,000 times. The restoring is completed in about 100ms.

Index	Name	Type
1011_h	*Restore Default Parameters*	ARRAY

Sub-Index	Name		Mnemonic
00_h	*Number_of_Entries*		
Data Type	Access Type	PDO Mapping	Note
u8	const		
Default Value	Minimum	Maximum	Unit
04_h			
Description			
Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Restore all Default Parameters*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		The default values can be restored only with the motor disabled. Trying the operation with the motor enabled, an error code is obtained in response.
	Default Value	Minimum	Maximum	Unit
	Description			
	It allows to restore the dictionary objects with index between 1000_h and $9FFF_h$. To start and restore simply write the object with the value 0x64616F6C.			

Sub-Index	Name			Mnemonic
02_h	*Restore Communication Default Parameters*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit
	Description			
	It allows to restore the dictionary objects with index between 1000_h and $1FFF_h$. To start and restore simply write the object with the value 0x64616F6C.			

Sub-Index	Name			Mnemonic
03_h	*Restore Application Default Parameters*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit
	Description			
	It allows to restore the dictionary objects with index between 6000_h and $9FFF_h$. To start and restore simply write the object with the value 0x64616F6C.			

Sub-Index	Name			Mnemonic
04_h	*Restore Manufacturer Defined Default Parameters*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit
	Description			
	It allows to restore the dictionary objects with index between 2000_h and $5FFF_h$. To start and restore simply write the object with the value 0x64616F6C.			

8.1.8 1018_h Identity object

This object provide general identification information of the CANopen device.

Sub-index 01h contains the unique value that is allocated uniquely to each vendor of a CANopen device. The LAM Technologies vendor-ID is 0x0000030C.

Sub-index 02h Contains the unique device specific code.

Sub-index 03h contains the major revision number and the minor revision number of the revision of the device. The major revision number identify a specific CANopen behavior. That means if the CANopen functionality is different, the major revision number is different. The minor revision number identifies different versions of device with the same CANopen behavior.

Sub-index 04h contains the device serial number.

Index	Name	Type
1018_h	*Identity object*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	04_h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Vendor-ID*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	0x0000030C			
	Description			
	Contains the unique value that is allocated uniquely to each vendor of a CANopen device. The LAM Technologies vendor-ID is 0x0000030C.			

Sub-Index	Name			Mnemonic
02_h	*Product_code*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	Description			

Contains the unique device specific code.

The following table shows the correspondence between code and device:

Code	Device	Code	Device
20712410_h	DMS71E4241	20722410_h	DMS72E4241
20712610_h	DMS71E4264	20722610_h	DMS72E4264
20712810_h	DMS71E4271	20722810_h	DMS72E4271
20715410_h	DMS71E7241	20725410_h	DMS72E7241
20715610_h	DMS71E7264	20725610_h	DMS72E7264
20715810_h	DMS71E7271	20725810_h	DMS72E7271
20714210_h	DMS71E4321	20714211_h	DMS71E4321A

20714310_h	DMS71E4331	20714311_h	DMS71E4331A
20714510_h	DMS71E4350	20714511_h	DMS71E4350A
20714710_h	DMS71E4360	20714711_h	DMS71E4360A
20717210_h	DMS71E7321	20717211_h	DMS71E7321A
20717310_h	DMS71E7331	20717311_h	DMS71E7331A
20717510_h	DMS71E7350	20717511_h	DMS71E7350A
20717710_h	DMS71E7360	20717711_h	DMS71E7360A
20724210_h	DMS72E4321	20724211_h	DMS72E4321A
20724310_h	DMS72E4331	20724311_h	DMS72E4331A
20724510_h	DMS72E4350	20724511_h	DMS72E4350A
20724710_h	DMS72E4360	20724711_h	DMS72E4360A
20727210_h	DMS72E7321	20727211_h	DMS72E7321A
20727310_h	DMS72E7331	20727311_h	DMS72E7331A
20727510_h	DMS72E7350	20727511_h	DMS72E7350A
20727710_h	DMS72E7360	20727711_h	DMS72E7360A

Sub-Index	Name			Mnemonic
03_h	*Revision_number*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit

Description

Contains the major revision number and the minor revision number of the revision of the device. The major revision number identifies a specific CANopen behavior. That means if the CANopen functionality is different, the major revision number is different. The minor revision number identifies different versions of device with the same CANopen behavior.

Bits 31..16	Bits 15..0
Major revision number	Minor revision number

Sub-Index	Name			Mnemonic
04_h	*Serial_number*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit

Description

Contains the device serial number.

8.1.9 10F1h Error Settings

Index	Name	Type
10F1h	*Error Settings*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	02h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Local Error Reaction*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit
	0x00000001			
	Description			

Describes the reaction of the device to an error occurrence.

The following table shows the allowed values:

1	The Outputs Sync Manager is disabled and the ESM status automatically changes to ErrSafeOp

Sub-Index	Name			Mnemonic
02h	*Sync Error Counter Limit*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		
	Default Value	Minimum	Maximum	Unit
	0x0004			
	Description			

It allows to define the tolerance threshold for the synchronism reception errors.

The device is provided with a counter that increases by 3 units each time a synchronism reception error occurs and decreases by 1 for each synchronism correctly received. When the value of the counter exceeds the one set through this object, the device goes into error state.

8.1.10 1600_h, 1601_h, 1602_h, 1603_h RPDO mapping parameter

Index	Name	Type
1600_h 1601_h 1602_h 1603_h	*RPDO mapping parameter*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_mapped_application_objects_in_PDO*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		💾
	Default Value	Minimum	Maximum	Unit
	01_h 02_h 02_h 02_h			
	Description			
	Number of mapped application objects in PDO			

Sub-Index	Name			Mnemonic
01_h	*Application_object_1*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		💾
	Default Value	Minimum	Maximum	Unit
	0x60400010 0x60400010 0x60400010 0x607A0020			
	Description			
	Contains the information of the mapped application object 1. The object describes the content of the PDO by their index, sub-index and length (length of the application object in bit).			

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Length

Sub-Index	Name			Mnemonic
02_h	*Application_object_2*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		💾
	Default Value	Minimum	Maximum	Unit
	--- 0x607A0020 0x60FF0020 0x60810020			
	Description			
	Contains the information of the mapped application object 2. The object describes the content of the PDO by their index, sub-index and length (length of the application object in bit).			

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Length

Sub-Index	Name			Mnemonic
03h…08h	*Application_object_3…8*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		
	Default Value	Minimum	Maximum	Unit

Description

Contains the information of the mapped application objects.

The object describes the content of the PDO by their index, sub-index and length (length of the application object in bit).

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Length

8.1.11 $1A00_h$, $1A01_h$, $1A02_h$, $1A03_h$ TPDO mapping parameter

Index	Name	Type
$1A00_h$ $1A01_h$ $1A02_h$ $1A03_h$	*TPDO mapping parameter*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_mapped_application_objects_in_PDO*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		💾
	Default Value	Minimum	Maximum	Unit
	1 2 2 0			
	Description			
	Number of mapped application objects in PDO			

Sub-Index	Name			Mnemonic
01_h	*Mapped_application_object_1*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		💾
	Default Value	Minimum	Maximum	Unit
	0x60410010 0x60410010 0x60410010 ---			
	Description			

Contains the information of the mapped application object 1.

The object describes the content of the PDO by their index, sub-index and length (length of the application object in bit).

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Length

Sub-Index	Name			Mnemonic
02_h	*Mapped_application_object_2*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		💾
	Default Value	Minimum	Maximum	Unit
	--- 0x60640020 0x606C0020 ---			
	Description			

Contains the information of the mapped application object 2.

The object describes the content of the PDO by their index, sub-index and length (length of the application object in bit).

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Length

Sub-Index	Name		Mnemonic
03_h…08_h	*Mapped_application_object_3…8*		
Data Type	Access Type	PDO Mapping	Note
u32	rw		🖫
Default Value	Minimum	Maximum	Unit

Description

Contains the information of the mapped application objects.

The object describes the content of the PDO by their index, sub-index and length (length of the application object in bit).

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Length

8.1.12 $1C00_h$ Sync Manager Communication Type

Index	Name	Type
$1C00_h$	*Sync Manager Communication Type*	ARRAY

Sub-Index	Name		Mnemonic
00_h	*Number_of_Entries*		
Data Type	Access Type	PDO Mapping	Note
u8	const		
Default Value	Minimum	Maximum	Unit
04_h			

Description

Number of record entries

Sub-Index	Name		Mnemonic
01_h	*Communication Type SM 1*		
Data Type	Access Type	PDO Mapping	Note
u8	ro		
Default Value	Minimum	Maximum	Unit
0x01			

Description

Describes the type of communication for Sync Manager 0.

The following table shows the possible values:

0	Not used
1	Receiving Mailbox
2	Transmitting Mailbox
3	Output process data
4	Input process data

Sub-Index	Name			Mnemonic
02h	*Communication Type SM 2*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0x02			
	Description			

Describes the type of communication for Sync Manager 1.

The following table shows the possible values:

0	Not used
1	Receiving Mailbox
2	Transmitting Mailbox
3	Output process data
4	Input process data

Sub-Index	Name			Mnemonic
03h	*Communication Type SM 3*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0x03			
	Description			

Describes the type of communication for Sync Manager 2.

The following table shows the possible values:

0	Not used
1	Receiving Mailbox
2	Transmitting Mailbox
3	Output process data
4	Input process data

Sub-Index	Name			Mnemonic
04h	*Communication Type SM 4*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0x04			
	Description			

Describes the type of communication for Sync Manager 3.

The following table shows the possible values:

0	Not used
1	Receiving Mailbox
2	Transmitting Mailbox
3	Output process data

4	Input process data

8.1.13 1C12h Sync Manger 2 PDO assign

Index	Name	Type
1C12h	*Sync Manger 2 PDO assign*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		💾
	Default Value	Minimum	Maximum	Unit
	01h			
				Description
	Number of record entries			

Sub-Index	Name			Mnemonic
01h	*RPDO Mapping Index*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rPreOPw		
	Default Value	Minimum	Maximum	Unit
	0x1601			
				Description
	It allows to select the RPDO to use for the process data exchange. The values allowed are: 0x1600, 0x1601, 0x1602 e 0x1603.			

8.1.14 1C13h Sync Manger 3 PDO assign

Index	Name	Type
1C13h	*Sync Manger 3 PDO assign*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		💾
	Default Value	Minimum	Maximum	Unit
	01h			
				Description
	Number of record entries			

Sub-Index	Name			Mnemonic
01h	*TPDO Mapping Index*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rPreOPw		
	Default Value	Minimum	Maximum	Unit
	0x1A01			
				Description
	It allows to select the TPDO to use for the process data exchange.			

	The values allowed are: 0x1A00, 0x1A01, 0x1A02 e 0x1A03.

8.1.15 1C32h Sync Manager Output Parameter

Index	Name	Type
1C32h	*Sync Manager Output Parameter*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	20h			
	Description			
	Highest sub-index supported.			

Sub-Index	Name			Mnemonic
01h	*Synchronization Type*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rPreOPw		
	Default Value	Minimum	Maximum	Unit
	Description			
	Allows to define the synchronization type. The following table shows the possible values:			

0	Free Run
1	Synchronous, synchronized with SM event
2	DC Sync0, synchronized with Sync0 event

Sub-Index	Name			Mnemonic
02h	*Cycle Type*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
				1ns (Ex. 2500000 = 2.5ms)
	Description			
	Shows the cycle time set by the master controller.			

Sub-Index	Name			Mnemonic
04h	*Synchronization Types Supported*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	0x0007			
	Description			
	Shows the synchronization type supported by the device. The object is made up in bits. The following table details its meaning:			

Bits	Function
0	Free-Run Mode 0 – Asynchronous mode (Free-Run) not supported 1 – Asynchronous mode (Free-Run) supported
1	Sync Manager Mode 0 – Synchronism SM not supported 1 – Synchronism SM supported
2..4	Distributed Clock (DC) Mode bit4 / bit3 / bit2: 0 0 0 – DC not supported 0 0 1 – DC Sync0 supported 0 1 0 – DC Sync1 supported 1 0 0 – Subordinated Application Sync0

Sub-Index	Name			Mnemonic
05h	*Minimum Cycle Time*			
	Data Type: u32	Access Type: ro	PDO Mapping:	Note:
	Default Value: 1000000	Minimum:	Maximum:	Unit: 1ns (Ex. 1000000 = 1ms)
	Description: Minimum cycle time supported by the device. When the master controller manages the device synchronously, the cycle time must be set to a value equal to or greater than the one indicated by the object.			

Sub-Index	Name			Mnemonic
06h	*Calc and Copy Time*			
	Data Type: u32	Access Type: ro	PDO Mapping:	Note:
	Default Value: 500000	Minimum:	Maximum:	Unit: 1ns (Ex. 500000 = 500us)
	Description: Indicates the time required for the device to elaborate the process data from the Sync Manager and make them ready to be executed.			

Sub-Index	Name			Mnemonic
09h	*Delay Time*			
	Data Type: u32	Access Type: ro	PDO Mapping:	Note:
	Default Value: 50000	Minimum:	Maximum:	Unit: 1ns (Ex. 50000 = 50us)
	Description:			

	Indicates the time the device takes to execute the process data after the synchronism event has been received.

Sub-Index	Name			Mnemonic
$0B_h$	*SM-Event Missed*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	It is a counter that increases each time the synchronism even is not received in time.			

Sub-Index	Name			Mnemonic
$0C_h$	*Cycle Time Too Small*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	It is a counter that increases each time the synchronism event is received in advance, preventing the process data to be terminated correctly.			

Sub-Index	Name			Mnemonic
20_h	*Sync Error*			
	Data Type	Access Type	PDO Mapping	Note
	bool	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	Indicates if synchronization errors occurred			

0	There are no synchronization errors
1	There are synchronization errors

8.1.16 1C33h Sync Manager Input Parameter

Index	Name	Type
1C32h	*Sync Manager Input Parameter*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		💾
	Default Value	Minimum	Maximum	Unit
	20h			
	Description			
	Highest sub-index supported.			

Sub-Index	Name			Mnemonic
01h	*Synchronization Type*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rPreOPw		
	Default Value	Minimum	Maximum	Unit
	Description			

Allows to define the synchronization type.

The following table shows the possible values::

0	Free Run
22h	Synchronous, synchronized with SM2 Event
2	DC Sync0, synchronized with Sync0 event

Sub-Index	Name			Mnemonic
02h	*Cycle Type*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
				1ns (Ex. 2500000 = 2.5ms)
	Description			
	Shows the cycle time set by the master controller.			

Sub-Index	Name			Mnemonic
04h	*Synchronization Types Supported*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	0x0007			
	Description			

Shows the synchronization type supported by the device.

The object is made up in bits. The following table details its meaning:

Bits	Function
0	Free-Run Mode

0	Asynchronous mode (Free-Run) not supported
1	Asynchronous mode (Free-Run) supported

1	Sync Manager Mode 0 Synchronism SM not supported 1 Synchronism SM supported
2..4	Distributed Clock (DC) Mode bit4 bit3 bit2 0 0 0 DC not supported 0 0 1 DC Sync0 supported 0 1 0 DC Sync1 supported 1 0 0 Subordinated Application Sync0

Sub-Index	Name			Mnemonic
05h	*Minimum Cycle Time*			
Data Type	Access Type	PDO Mapping		Note
u32	ro			
Default Value	Minimum	Maximum		Unit
1000000				1ns (Ex. 1000000 = 1ms)
Description				
Minimum cycle time supported by the device. When the master controller manages the device synchronously, the cycle time must be set to a value equal to or greater than the one indicated by the object.				

Sub-Index	Name			Mnemonic
06h	*Calc and Copy Time*			
Data Type	Access Type	PDO Mapping		Note
u32	ro			
Default Value	Minimum	Maximum		Unit
200000				1ns (Ex. 200000 = 200us)
Description				
Indicates the time required for the device to elaborate the process data and to copy them into the Sync Manager.				

Sub-Index	Name			Mnemonic
09h	*Delay Time*			
Data Type	Access Type	PDO Mapping		Note
u32	ro			
Default Value	Minimum	Maximum		Unit
50000				1ns (Ex. 50000 = 50us)
Description				
Indicates the time the device takes to prepare the process data after the synchronism event has been received.				

Sub-Index	Name			Mnemonic
0Bh	*SM-Event Missed*			
Data Type	Access Type	PDO Mapping		Note

	u16	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	It is a counter that increases each time the synchronism even is not received in time.			

Sub-Index	Name			Mnemonic
$0C_h$	*Cycle Time Too Small*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	It is a counter that increases each time the synchronism event is received in advance, preventing the process data to be terminated correctly.			

Sub-Index	Name			Mnemonic
20_h	*Sync Error*			
	Data Type	Access Type	PDO Mapping	Note
	bool	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	Indicates if synchronization errors occurred			

0	There are no synchronism errors
1	There are synchronism errors

8.2 Objects with index between 2000_h and $5FFF_h$

The objects with index between 2000_h and $5FFF_h$ are device-specific and not part of the profiles standardized by the CANopen standard.

8.2.1 2010_h Fieldbus Setup

Index	Name	Type
2010_h	*Fieldbus Setup*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01_h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Options_FLDBSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0080_h			
	Description			

It allows to set the fieldbus options.

The object is made up in bits that operates as follows:

Bits	Function
0..6	Not used. In reading ignore the value, in writing set to 0.
7	Automatic compilation of the $60C2_h$:01_h *Interpolation_time_period_value* and $60C2_h$:02_h *Interpolation_time_index* objects. When the bit is set to 1 and the synchronism mode is set to DC SYNC0, at the moment of the transition from *PreOperation* state to the *SafeOperational* state, the $60C2_h$:01_h *Interpolation_time_period_value* and $60C2_h$:02_h *Interpolation_time_index* objects are automatically compiled with the cycle time transmitted by the master controller, provided that it is within the allowed range. 0 – Automatic compilation not active 1 – Automatic compilation active
8..15	Not used. In reading ignore the value, in writing set to 0.

8.2.2 2120h Error and Fault Setup

Index	Name	Type
2120h	*Error Fault Setup*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*OpenPhase_EFSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		💾
	Default Value	Minimum	Maximum	Unit
	1			
	Description			
	It allows to enable or disable the test of the open phase condition. The following table shows the possible values: 0 – Open phase test disabled 1 – Open phase test enabled			

Sub-Index	Name			Mnemonic
02h	*RPDO_EFSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		u8
	Default Value	Minimum	Maximum	Unit
	1			1
	Description			
	It allows to enable or disable the alarm on RPDO error. If it is not possible to process a value received via PDO, for example because the value is outside the allowed ranges or cannot be written in the state of the drives, and the RPDO alarm is active, the drive signals error 0x82E1. The following table shows the possible values: 0 – RPDO alarm disabled 1 – RPDO alarm enabled 2 – RPDO alarm only with motor enabled			

Sub-Index	Name			Mnemonic
03h	*TPDO_EFSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		U8
	Default Value	Minimum	Maximum	Unit
	1			1
	Description			
	It allows to enable or disable the alarm on TPDO error.			

If it is not possible to transmit a value via PDO, for example because the value is not available, and the TPDO alarm is active, the drive signals error 0x82E2.

The following table shows the possible values:

0	TPDO alarm disabled
1	TPDO alarm enabled
2	TPDO alarm only with motor enabled

Sub-Index	Name			Mnemonic
04h	*DownloadSDO_EFSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		u8
	Default Value	Minimum	Maximum	Unit
	1			1

Description

It allows to enable or disable the alarm on SDO Download.

If it is not possible to process a value received via SDO, for example because the value is outside the allowed ranges or cannot be written in the state of the drive, and the Download alarm is active, the drive signal error 0x82E3.

The following table shows the possible values:

0	RPDO alarm disabled
1	RPDO alarm enanled
2	RPDO alarm only with motor enabled

Sub-Index	Name			Mnemonic
05h	*UploadSDO_EFSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		u8
	Default Value	Minimum	Maximum	Unit
	1			1

Description

It allows to enable or disable the alarm on Upload SDO error.

If it is not possible to transmit a value via SDO, for example because the value is not available, and the Upload SDO alarm is active, the drive signals error 0x82E4.

The following table shows the possible values::

0	TPDO alarm disabled
1	TPDO alarm enabled
2	TPDO alarm only with motor enabled

Sub-Index	Name			Mnemonic
06h	*MotorTemperature_EFSTP*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		💾
	Default Value	Minimum	Maximum	Unit

	1			
	Description			
	It allows to enable and disable the motor over-temperature alarm. The following table shows the possible values: 0 – Motor over-temperature alarm disabled 1 – Motor over-temperature alarm enabled			

8.2.3 2210h Digital I/O Setup

Index	Name	Type
2120h	*Digital IO Setup*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping:	Note:
	Default Value: 01h	Minimum:	Maximum:	Unit:
	Description: Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Direction_DIOSTP*			
	Data Type: u8	Access Type: rw	PDO Mapping:	Note: 💾
	Default Value: 1	Minimum:	Maximum:	Unit:
	Description: It allows to define the input or output IO line. The object is made up in bits and each bit is associated with an IO line. When the bit is 0 it is configured as input, when the bit is 1, the line is configured an output. The following table shows the correspondence between the bits and the IO lines and the function performed according to the status of the bit itself. The bits of the object are associated to the IO as follows:			

Bit	Description	Value	Function
0	Digital IO 0 (DI0)	0	DI3
		1	DO0
1	Digital IO 1 (DI1)	0	DI4
		1	DO1
2	Digital IO 2 (DI2)	0	DI5
		1	DO2
3..15	Reserved, ignore the value for reading and set to 0 for writing		

8.2.4 2310h Motor Data

Index	Name	Type
2310h	*Motor Data*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	0Bh			
	Description			
	Number of record entries			

Sub-Index	Name			Mnemonic
01h	*CMC_MTRDT*			
	Data Type	Access Type	PDO Mapping	
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	It is a compact code that identifies the LAM Technologies motors and that allows the drive to configure the best control parameter for the motor. The *CMC_MTRDT* code is compiled automatically and does not need to be initialized. When the object is initialized with a value different from 0, the objects with Sub-Index from 02h to 0Ah are ignored.			

Sub-Index	Name			Mnemonic
02h	*Type_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	12			
	Description			
	Not used, ignore the value.			

Sub-Index	Name			Mnemonic
03h	*Pole_Pairs_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	50			
	Description			
	Not used, ignore the value.			

Sub-Index	Name			Mnemonic
04h	*Wiring_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	Not used, ignore the value.			

Sub-Index	Name	Mnemonic
05h	*Resistance_MTRDT*	

	Data Type	Access Type	PDO Mapping	Note
	u16	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	100		60000	10mOhm (Ex. 240 = 2.4 Ohm)
	Description			
	Not used, ignore the value.			

Sub-Index			Name	Mnemonic
06h	*Inductance_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	300		60000	10µH (Ex. 320 = 3.2 mH)
	Description			
	Not used, ignore the value.			

Sub-Index			Name	Mnemonic
07h	*Back_EMF_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	2500		60000	10mV/1000rpm (Ex. 4500 = 45V/1000rpm)
	Description			
	Not used, ignore the value.			

Sub-Index			Name	Mnemonic
08h	*Rated_Current_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	100	10	60000	10mArms (Ex. 420 = 4.2 Arms)
	Description			
	Not used, ignore the value.			

Sub-Index			Name	Mnemonic
09h	*Max_Current_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	130	10	60000	10mArms (Ex. 550 = 5.5 Arms)
	Description			
	Not used, ignore the value.			

Sub-Index			Name	Mnemonic
0Ah	*Rated_Torque_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	50	1	60000	10mNm (Ex. 180 = 1.8 Nm)
	Description			
	Not used, ignore the value.			

Sub-Index			Name	Mnemonic
0Bh	*Max_Speed_MTRDT*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit

	30000	1	30000	0.1rpm (Ex. 668 = 66x8rpm)
	Description			
	Not used, ignore the value.			

8.2.5 2330_h Motor Incremental Encoder Data

Index	Name	Type
2330_h	*Encoder Motor*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Minimum
	02_h			
	Description			
	Highest sub-index supported.			

Sub-Index	Name			Mnemonic
01_h	*Configuration_ENCMTR*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0			

Description

It allows you to set the encoder management mode. The object is made up of groups of bits that define the following functions:

Bits	Function
0	Index 0 – Index signal not managed 1 – Index signal managed
1	Pulse count inversion 0 – Normal pulses count 1 – Backwards pulses count (equivalent to change phase A and B between them)
2	Not used
3	Power Supply 0 – Encoder power supply off 1 – Encoder power supply on
7..4	Filter If the encoder signals are affected by electric noise, it is possible to use the digital filter present in the drive to eliminate the pulses shorter than a pre-defined value.

3	250ns
5	500ns
7	1us
9	2us
12	4us
15	8us
0, 1, 2, 4, 6, 8, 10, 11, 13, 14	Reserved values, do not use

The maximum equivalent frequency must be calculated considering twice the set time, as this time must elapse both for the active signal condition and for the inactive signal. For example, setting a Filter value of 1us you can apply a maximum frequency of 500KHz (1+1=2us i.e. 500KHz).

If the object is set to 0, the encoder input is disables.

Sub-Index	Name			Mnemonic
02_h	*CPR_ENCMTR*			
Data Type	Access Type	PDO Mapping		Note
u16	ro			
Default Value	Minimum	Maximum		Unit
0				Encoder pulses per revolution

Description

It allows to set the number of pulses/rev of the encoder connected to the motor.

By setting the value to 0, the encoder input is disabled.

The device is able to count each pulse edge in order to obtain a resolution 4 times higher than the native resolution of the encoder. For example, by using a 400 pulses/rev encoder, the drive will be able to recognize 1600 different positions/rev.

The encoder must be able to generate at least 360 pulses/rev.

Sub-Index	Name			Mnemonic
03_h	*PhasingTime_ENCMTR*			
Data Type	Access Type	PDO Mapping		Note
u16	ro			
Default Value	Minimum	Maximum		Unit
500				1 ms (1500 = 1.5ms)

Description

It allows you to set the waiting time to phase the encoder.

The first time that a motor equipped with an encoder is enabled, the encoder must be phased.

After been enabled, the motor could move and it is necessary to wait for the load to stabilize before proceeding to phase the encoder. The PhasingTime_ENCMTR object allows you to set this time.

The *Configuration_ENCMTR* and *CPR_ENCMTR* objects described above are automatically updated each time the *CMC_MTRDT* object is written with a valid LAM Technologies motor code.

8.2.6 2338_h SSI Encoder Data

Index	Name	Type
2338_h	*Encoder SSI*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Minimum
	$0B_h$			
	Description			
	Highest sub-index supported.			

Sub-Index	Name			Mnemonic
01_h	*Configuration_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0			
	Description			

It allows to enable the SSI encoder and set its management mode. The object is made up of groups of bits that define the following functions:

Bits	Function
0	Enabling 0 – SSI encoder reading not enabled 1 – SSI encoder reading enabled
1	Error control 0 – Error control not enabled 1 – Error control enabled When the error control is enabled, the drive signals an error condition if the SSI encoder reading returns all the bits to 1, all the bits to 0 or if the sampled bit on the first falling edge of the clock differs from 1.
2	Position copy 0 – Copy not enabled 1 – Copy enabled When the position copy is enabled, the SSI encoder quota is copied in the actual motor position (*Position_actual_value* object) the first time the quota becomes valid and at each transition to the *Operational* status of the drive. The actual motor position can be initialized at the quota read by the SSI encoder also through the *Homing* operation mode by selecting the mode -

	80.
3	Power Supply 0 SSI encoder power supply off 1 SSI encoder power supply on
7..4	Not used. Set to 0.

If the object is set to 0, the SSI encoder input is disabled.

Sub-Index	Name			Mnemonic
02h	*FrameLength_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	25	8	31	bit
	Description			
	It allows to set the number of bits that make up the data frame of the SSI encoder. The *start* bit must not be counted.			

Sub-Index	Name			Mnemonic
03h	*TurnsBits_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	12		24	bit
	Description			
	It allows to set the number of bits reserved for the full revolutions counting. In case of linear SSI encoder, set the object to 0.			

Sub-Index	Name			Mnemonic
04h	*SingleTurnBits_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	13	8	31	bit
	Description			
	It allows to set the number of bits reserved for the position on the revolution. In case of linear encoder set the object with the number of bit that make up the quota of the encoder itself.			

Sub-Index	Name			Mnemonic
05h	*MSB_Offset_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0		20	bit
	Description			

	It allows to set the position of the most significant bit (MSB) inside the frame. The position is counted from left to right with the first bit of the frame that has position 0. For example, if the most significant bit of the quota falls on the third bit of the frame (starting from left and not counting the *start* bit) the *MSB_Offset_ENCSSI* object must be set to the value 3.

Sub-Index	Name			Mnemonic
06h	*FormingTime_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	150			1 us (150 = 150us)
	Description			
	It allows to set the time required by the SSI encoder to prepare a new quota. The drive waits at least this time between one reading and the next.			

Sub-Index	Name			Mnemonic
07h	*SetupTime_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	1200			1 ms (1200 = 1.2ms)
	Description			
	It allows to set the time required by the SSI encoder to complete the initializations and provide the first valid quota. After the SSI encoder is enabled and after power has been supplied (if the SSI encoder is configured with bit3 of the *Configuration_ENCSSI* object to 1), the drive waits at least this time before executing the first reading of the quota.			

Sub-Index	Name			Mnemonic
08h	*ScalingFactor_N_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro		
	Default Value	Minimum	Maximum	Unit
	10000			
	Description			
	It allows to set the numerator of the formula used to calculate the position expressed in 0.0001rev, starting from the quota read by the SSI encoder. The formula used is the following: *Position_ENCSSIV* = (*PositionRaw_ENCSSIV - PositionOffset_ENCSSI*) * *ScalingFactor_N_ENCSSI / ScalingFactor_D_ENCSSI*			

Sub-Index	Name			Mnemonic
09h	*ScalingFactor_D_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note

	i32	ro		
	Default Value	Minimum	Maximum	Unit
	10000			
	Description			
	It allows to set the denominator of the formula used to calculate the position expressed in 0.0001rev, starting from the quota read by the SSI encoder. The formula used is the following: *Position_ENCSSIV* = (*PositionRaw_ENCSSIV* - *PositionOffset_ENCSSI*) * *ScalingFactor_N_ENCSSI* / *ScalingFactor_D_ENCSSI*			

Sub-Index	Name			Mnemonic
$0A_h$	*PositionOffset_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro		
	Default Value	Minimum	Maximum	Unit
	10000			
	Description			
	It allows to set the offset subtracted from the quota read by the SSI encoder to calculate the position express in 0.0001rev. The formula applied is the following: *Position_ENCSSIV* = (*PositionRaw_ENCSSIV* - *PositionOffset_ENCSSI*) * *ScalingFactor_N_ENCSSI* / *ScalingFactor_D_ENCSSI*			

Address	Name			Mnemonic
$0B_h$	*PhasingTime_ENCSSI*			
	Data Type	Access Type	PD Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	500			1 ms (1500 = 1.5ms)
	Description			
	It allows to set the time required to phase the encoder. The first time a motor equipped with encoder is enabled, the encoder must be phased. Following the enabling, the motor could move and it is necessary to wait for the load to stabilize before proceeding with the encoder phasing. The *PhasingTime_ENCSSI* object allows to set this time.			

8.2.7 2360h Holding Brake Setup

Index	Name	Type
2360h	*Holding Brake Setup*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Minimum
	06h			
				Description
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Option_HBRKS*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0			
				Description

It allows to set the holding brake handling by the drive and by manual control.
The object is made up by groups of bits that operates as follows:

Bits	Function
0	Handling 0 – Not handled, the holding brake is not handled through the drive 1 – Handled, the holding brake is handled through the drive (connected to one output) When the bit is set to 1 upon enabling, the device waits for the brake disengagement time specified by the *Release_Time_HBRKS* object before going into the operating status. When disabled, the device waits for the *Application_Time_HBRKS* time before disconnecting the motor.
1	Automatic / Manual 0 – Automatic, the device commands the activation and deactivation of the brake autonomously 1 – Manual, the brake control can take place only via a digital input or the *Control_HBRKC* object
2..7	Not used
8..11	Action It allows to define how the manual brake control (obtainable through a digital input or the *Control_HBRKC*) object interacts with the automatic control operated by the device (if active through bit1=0). In the following description, “active signal” means the active state of the digital input chosen for the manual brake control or the bit0 of the *Control_HBRKC* object set to 1 (the two signals are in logic OR with each

other).

0	None
1	Release, with active signal the brake is always released independently from the device control
2	Engage, with active signal the brake is always engaged independently from the device control
3	Release / Engage, with active signal the brake is always released. With inactive signal the brake is always engaged independently from the device control
4	Shared Release, on the signal rising edge (transition from inactive to active) the brake is released. Also the device can release the brake when necessary
5	Shared Engage, on the signal rising edge (transition from inactive to active) the brake is engaged. Also the device can engage the brake when necessary
6	Shared release / engage, on the signal rising edge (transition from inactive to active) the brake is released while on the signal falling edge (transition from active to inactive) the brake is engaged. Also the device can release or engage the brake when necessary

Sub-Index	Name		Mnemonic
02_h	*Application_Time_HBRKS*		
Data Type	Access Type	PDO Mapping	Note
u16	rw		💾
Default Value	Minimum	Maximum	Unit
200		10000	ms (ex. 250 = 250ms)

Description

It allows you to set the time required for the brake to completely engage to ensure the maximum resistant torque.

When the bit0 of the *Option_HBRKS* object is set to 1 and the disabling of the motor is required, the device waits for the time specified in the *Application_Time_HBRKS* object, after the brake has been engaged, before disconnecting the motor.

Sub-Index	Name		Mnemonic
03_h	*Release_Time_HBRKS*		
Data Type	Access Type	PDO Mapping	Note
u16	rw		💾
Default Value	Minimum	Maximum	Unit
200		10000	ms (ex. 250 = 250ms)

Description

It allows you to set the time required for the brake to completely disengage to ensure the minimum resistant torque.

When the bit0 of the *Option_HBRKS* object is set to 1 and the motor enable is required, the

	device waits for the time specified in the *Release_Time_HBRKS* object, after the brake has been disengaged, before going into the operating status.

Sub-Index	Name		Mnemonic
04_h			
Data Type	Access Type	PDO Mapping	Note
u32	rw		🖫
Default Value	Minimum	Maximum	Unit
Description			
Reserved, do not use			

Sub-Index	Name		Mnemonic
05_h			
Data Type	Access Type	PDO Mapping	Note
u32	rw		🖫
Default Value	Minimum	Maximum	Unit
Description			
Reserved, do not use			

8.2.8 $23A0_h$ Current Regulation

Index	Name	Type
$23A0_h$	*Current Regulation*	RECORD

Sub-Index	Name		Mnemonic
00_h	*Number_of_Entries*		
Data Type	Access Type	PDO Mapping	Note
u8	const		
Default Value	Minimum	Maximum	Unit
02_h			
Description			
Number of record entries.			

Sub-Index	Name		Mnemonic
01_h	*Mode_CRRG*		
Data Type	Access Type	PDO Mapping	Note
u8	rw	RPDO	🖫
Default Value	Minimum	Maximum	Unit
0			
Description			
It defines how the current is supplied to the motor. When the static mode is selected, the motor receives the *Current_Max_MTRCNF* current while in movement, and the *Current_Min_MTRCNF* current when it is stopped (after the time defined by the *Current_Idle_Delay_MTRCNF* object). Instead, when the dynamic mode is selected, the drive supplies the motor with a current value proportional to the load. The current variation always occurs between the minimum and maximum values defined by the *Current_Min_MTRCNF* and *Current_Max_MTRCNF* objects respectively.			

Value	Description
0	Static current supply independent from the load
1	Dynamic current supply proportional to the load

Sub-Index	Name		Mnemonic
02h	*Gain_CRRG*		
Data Type	Access Type	PDO Mapping	Note
u8	rw	RPDO	💾
Default Value	Minimum	Maximum	Unit
100	50	200	% (Ex. 120 = 120%)
Description			
It allows to intervene manually on the open-loop gain. The drive automatically determines the best parameter for the phase current regulation, nevertheless through this object it is possible to manually intervene on the gain up to halve it (50%) or double it (200%).			

8.2.9 2410h Motion Setup

Index	Name	Type
2410h	*Motion Setup*	RECORD

Sub-Index	Name		Mnemonic
00h	*Number_of_Entries*		
Data Type	Access Type	PDO Mapping	Note
u8	const		
Default Value	Minimum	Maximum	Unit
02h			
Description			
Number of record entries.			

Sub-Index	Name		Mnemonic
01h	*Feedback_MTNSTP*		
Data Type	Access Type	PDO Mapping	Note
u8	rw	RPDO	💾
Default Value	Minimum	Maximum	Unit
0			
Description			
The object is useful to enable and configure the closed loop operation of the motor. The following table resumes the use of the object bits.			

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>1..0</td><td>Encoder Feedback

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Disabled</td></tr>
<tr><td>1</td><td>Closed loop (feedback enabled)</td></tr>
<tr><td>2</td><td>Following control</td></tr>
</table>

It allows to enable the closed-loop operation using the motor encoder (incremental encoder) for feedback.

When the following error control is selected, it is possible to use both the incremental encoder and the SSI encoder as position source.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Open loop (feedback disabled)</td></tr>
<tr><td>1</td><td>Closed loop (feedback enabled)</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Not used. Set to 0.</td></tr>
<tr><td>3</td><td>Encoder Phasing

It allows to choose the Encoder Phasing mode.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Only when necessary (ex. at the first motor enabling after the device power on)</td></tr>
<tr><td>1</td><td>Always at each motor enabling</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>Feedback source
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Incremental encoder</td></tr>
<tr><td>1</td><td>SSI encoder</td></tr>
</table>
</td></tr>
<tr><td>5..7</td><td>Not used, set to 0.</td></tr>
</table>

For a correct management of the closed loop, besides activating the Encoder feedback, you must also configure the features of the encoder connected to the motor, through the *Configuration_ENCMTR* and *CPR_ENCMTR* objects.

In the *Operational* and *Quick Stop* status, it is not possible to modify the object value. The *Feedback_MTNSTP* object can be modified only with motor disabled.

Sub-Index	Name	Mnemonic

02h	*Current_Enable_Ramp_MTNSTP*			
	Data Type: u16	Access Type: rw	PDO Mapping:	Note: 💾
	Default Value: 10	Minimum:	Maximum: 2000	Unit: ms (ex. 250 = 250ms)
	Description: Through this object it is possible to set the time required by the current to reach the rated value each time the motor is enabled. A long current ramp can be of help to limit the absorption peak on the power supply and to damp the rotor alignment movement the first time the motor is enabled.			

8.2.10 2440h Position Loop

Index	Name	Type
2440h	*Position Loop*	RECORD

Sub-Index 00h	Name: *Number_of_Entries*			Mnemonic
	Data Type: u8	Access Type: const	PDO Mapping:	Note
	Default Value: 01h	Minimum:	Maximum:	Unit
	Description: Number of record entries.			

Sub-Index 01h	Name: *Kp_PSTNLP*			Mnemonic
	Data Type: u16	Access Type: rw	PDO Mapping: RPDO	Note: 💾
	Default Value: 300	Minimum: 2	Maximum: 3000	Unit
	Description: It allows to set the proportional gain of the position control loop. Too low a value causes a great position error while an excessive value can make the system unstable.			

8.2.11 2480h Errors Class Setup

Index	Name	Type
2480h	*Error Class Setup*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01h			
				Description
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Following_Error_ERRCS*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		💾
	Default Value	Minimum	Maximum	Unit
	6			
				Description

It allows you to define the emergency class generated in case of following error.
The following are the values that can be used:

Class	Description	Motor's reaction
2	Error that requires the stop of the motor but not the switch to FAULT.	Deceleration with Quick Stop.
4	Error that requires the stop of the motor and subsequently the switching to FAULT.	Deceleration with Quick Stop and subsequent disabling of the motor.
6	Error that requires the immediate disabling of the motor and the switching to FAULT.	Disabling of the motor.
8	As per class 6. The error can be reset only by a shutdown and restart cycle.	Disabling of the motor.

8.2.12 2810h Digital Inputs Assignment

The objects described below allow you to assign functions and actions to the digital inputs.

For example, a digital input can be used to reset the fault as an alternative to the bit *Fault reset* contained in the *Controlword* object. Please note that the same input can be associated with more functions and actions.

Index	Name	Type
2810h	*Digital Inputs Action*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	08h			
	Description			
	Highest sub-index supported.			

Sub-Index	Name			Mnemonic
01h				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	Description			
	Reserved, do not use.			

Sub-Index	Name			Mnemonic
02h	*Fault_Reset_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0000h			
	Description			

It allows to use a digital input to reset a fault condition.

The object has a dimension of 2 bytes, the low byte is used to select the input while the high one to specify the status to be considered active. For example, by selecting *Falling Edge* the fault will reset on the active / inactive transition (falling edge) of the input.
The following tables show the possible values that can be assigned to the low and high bytes.

High byte, active status

Value	Description
4	Rising edge
5	Falling edge

Low byte, input number

Value	Description
0	Digital input 0
1	Digital input 1
2	Digital input 2
3	Digital input 3
4	Digital Input 4
5	Digital Input 5

Sub-Index	Name			Mnemonic
03h	*Positive_Limit_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0000h			

Description

It defines the input to which the positive limit switch is connected.

The object has a dimension of 2 bytes, the low byte is used to select the digital input while the high one to specify the status to be considered active. For example, by selecting *Inactive Input* the limit switch will be considered active when the input is switched off (inactive).
The following table shows the possible values that can be assigned to the high and low bytes.

high byte, active status		low byte, input number	
Value	Description	Value	Description
0	Always Inactive	0	Digital input 0
1	Always Active	1	Digital input 1
2	Active Input	2	Digital input 2
3	Inactive Input	3	Digital input 3
		4	Digital Input 4
		5	Digital Input 5

Sub-Index	Name			Mnemonic
04h	*Negative_Limit_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0000h			

Description

It defines the input to which the negative limit switch is connected.

The object has a dimension of 2 bytes, the low byte is used to select the digital input while the high one to specify the status to be considered active. For example, by selecting *Inactive Input* the limit switch will be considered active when the input is switched off (inactive).
The following table shows the possible values that can be assigned to the high and low bytes.

high byte, active status		low byte, input number	
Value	Description	Value	Description
0	Always Inactive	0	Digital input 0
1	Always Active	1	Digital input 1
2	Active Input	2	Digital input 2
3	Inactive Input	3	Digital input 3
		4	Digital Input 4
		5	Digital Input 5

Sub-Index	Name			Mnemonic
05h	*Home_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0000h			

Description

It defines the input to which the home switch is connected.

The object has a dimension of 2 bytes, the low byte is used to select the digital input while the high one to specify the status to be considered active. For example, by selecting *Inactive Input* the home switch will be considered active when the input is switched off (inactive).
The following table shows the possible values that can be assigned assignable to the high and low bytes.

high byte, active status

Value	Description
0	Always Inactive
1	Always Active
2	Active Input
3	Inactive Input

Low byte, input number

Value	Description
0	Digital input 0
1	Digital input 1
2	Digital input 2
3	Digital input 3
4	Digital Input 4
5	Digital Input 5

Sub-Index	Name			Mnemonic
06h				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit

Description

Reserved, do no use.

Sub-Index	Name			Mnemonic
07h				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit

Description

Reserved, do no use.

Sub-Index	Name			Mnemonic
08h				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit

Description

Reserved, do no use.

Sub-Index	Name			Mnemonic
09h				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	Description			
	Reserved, do no use.			

Sub-Index	Name			Mnemonic
0Ah	*Holding_Brake_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0000h			
	Description			

It defines the input used for the manual control of the holding brake.

The object has a size of 2 bytes, the low byte is used to select the digital input while the high one is used to specify the state to be considered active. For example, by selecting *inactive input* the brake will be commanded when the input is off (inactive).

The following table shows the possible values that can be assigned to the high and low byte.

High byte, active state

Value	Description
0	Always Inactive
1	Always Active
2	Active Input
3	Inactive Input

Low byte, input number

Value	Description
0	Digital input 0
1	Digital input 1
2	Digital input 2
3	Digital input 3
4	Digital Input 4
5	Digital Input 5

If you do not need to manually control the holding brake, set the object to 0. For example, if you want to manually control the brake with input 5 active, set the object to 0205h.

Sub-Index	Name			Mnemonic
0Bh				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	Description			
	Reserved, do not use.			

Sub-Index	Name			Mnemonic
0Ch				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit

				Description
	Reserved, do not use.			

Sub-Index			Name	Mnemonic
$0D_h$				
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
				Description
	Reserved, do not use.			

Sub-Index			Name	Mnemonic
$0E_h$	*TouchProbe_1_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0000_h			
				Description

It the defines the input used for the channel 1 of the *Touch Probe* function.

The object has a size of 2 bytes and the low byte is used to select the digital input while the lower byte must be set to 2.

The following table shows the possible values that can be assigned to the high and low byte.

High byte, active state		Low byte, input number	
Value	Description	Value	Description
2	Active Input	0	Digital Input 0
		1	Digital Input 1
		2	Digital Input 2
		3	Digital Input 3
		4	Digital Input 4
		5	Digital Input 5

Sub-Index			Name	Mnemonic
$0F_h$	*TouchProbe_2_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0000_h			
				Description

It defines the input used for the channel 2 of the *Touch Probe* function.

The object has a size of 2 bytes and the low byte is used to select the digital input while the higher byte must be set to 2.

The following table shows the possible values that can be assigned to the high and low byte.

High byte, active state	Low byte, input number

Value	Description
2	Active Input

Value	Description
0	Digital Input 0
1	Digital Input 1
2	Digital Input 2
3	Digital Input 3
4	Digital Input 4
5	Digital Input 5

If you do not need to manually control the holding brake, set the object to 0. For example, if you want to control manually the brake with input 5 active set the object to 0205_h.

8.2.13 2830h Digital Outputs 0 Assignment

Index	Name	Type
2830h	*Digital Output 0 Action*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01h			
				Description
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Source_DOA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0000h			
				Description

It allows to select the source to be assigned to the digital output 0.

Value	Description
0000h	General Purpose
0008h	Active
8008h	Inactive
0040h	Fault
8040h	No Fault
0041h	Operational enabled
8041h	No Operational enabled
0042h	Quick stop active
8042h	No Quick stop active
0044h	Holding Brake
8044h	No Holding Brake
0050h	Positive Movement
8050h	No Positive Movement
0051h	Negative Movement
8051h	No Negative Movement
0060h…006Fh	*Statusword* bit n = 1 (0060h=bit0, 0061h=bit1, etc.)
8060h…806Fh	*Statusword* bit n = 0 (8060h=bit0, 8061h=bit1, etc.)

For example, by setting the value 8040h, the output will be activated if there are no fault conditions. Instead, by setting the value 8067h, the output will be activated every time the bit 7 (bit *warning*) of the *Statusword* will have value 0.

When the value 0000h (General Purpose) is selected, the output status is controlled by the value of bit 0 of the *Outputs_DOV* object.

8.2.14 2831_h Digital Output 1 Assignment

Index	Name	Type
2831_h	*Digital Output 1 Action*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01_h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Source_DOA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0000_h			
	Description			

It allows to select the source to be assigned to the digital output 1.

Value	Description
0000_h	General Purpose
0008_h	Active
8008_h	Inactive
0040_h	Fault
8040_h	No Fault
0041_h	Operational enabled
8041_h	No Operational enabled
0042_h	Quick stop active
8042_h	No Quick stop active
0044_h	Holding Brake
8044_h	No Holding Brake
0050_h	Positive Movement
8050_h	No Positive Movement
0051_h	Negative Movement
8051_h	No Negative Movement
0060_h...$006F_h$	*Statusword* bit n = 1 (0060_h=bit0, 0061_h=bit1, etc.)
8060_h...$806F_h$	*Statusword* bit n = 0 (8060_h=bit0, 8061_h=bit1, etc.)

For example, by setting the value 8040_h, the output will be activated if there are no fault conditions. Instead, by setting the value 8067_h, the output will be activated every time the bit 7 (bit *warning*) of the *Statusword* will have value 0.

When the value 0000_h (General Purpose) is selected, the output status is controlled by the value of the bit 1 of the *Outputs_DOV* object.

8.2.15 2832_h Digital Output 2 Assignment

Index	Name	Type
2832_h	*Digital Output 2 Action*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01_h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Source_DOA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0000_h			
	Description			

It allows to select the source to be assigned to the digital output 2.

Value	Description
0000_h	General Purpose
0008_h	Active
8008_h	Inactive
0040_h	Fault
8040_h	No Fault
0041_h	Operational enabled
8041_h	No Operational enabled
0042_h	Quick stop active
8042_h	No Quick stop active
0044_h	Holding Brake
8044_h	No Holding Brake
0050_h	Positive Movement
8050_h	No Positive Movement
0051_h	Negative Movement
8051_h	No Negative Movement
0060_h...$006F_h$	*Statusword* bit n = 1 (0060_h=bit0, 0061_h=bit1, etc.)
8060_h...$806F_h$	*Statusword* bit n = 0 (8060_h=bit0, 8061_h=bit1, etc.)

For example, by setting the value 8040_h, the output will be activated if there are no fault conditions. Instead, by setting the value 8067_h, the output will be activated every time the bit 7 (bit *warning*) of the *Statusword* will have value 0.

When the value 0000_h (General Purpose) is selected, the output status is controlled by the value of the bit 2 of the *Outputs_DOV* object.

8.2.16 3080_h Sync monitoring

The record objects realize a complete monitoring system on the receiving frequency of the SYNC object.

In many cases it is important to receive the Sync within a predetermined time window and promptly react if this does not happen. A typical case is the movement obtained by the interpolated modes where the single positions that define the path (set-points) are processed synchronously with the SYNC object reception. If the synchronism signal is not received with the correct frequency, the motor will no longer follow the desired path but will move irregularly. Through the monitoring of the Sync it is possible to detect anomalies on the on the receiving of the SYNC object and intervene by activating an emergency that, for example, stops the motor in Quick Stop mode.

Even if you do not want to trigger an emergency, it is still possible to use the Sync monitoring to analyze the temporal stability with which the SYNC object is received.

Index	Name	Type
3080_h	*Sync Guard*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	08_h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Option_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	U8	rw		💾
	Default Value	Minimum	Maximum	Unit
	0			
	Description			

It enables the Sync object monitoring and other options. The object is made up of bits that define the following functions:

Bits	Function
0	Enable Sync monitoring. On the transition of the bit from 0 to 1 (transition from disabled to enabled state) the *Out_Of_Sync_Counter_SYGD*, *Peak_Out_Of_Sync_SYGD* and *Sync_Counter_SYGD* objects are set to 0. 0 – Monitoring disabled 1 – Monitoring enabled
1	Enable emergency in case of violation of timing waited for Sync. 0 – Emergency not generated in case of violation 1 – Emergency generated in case of violation
2..3	Not used, set to 0.
4	*Peak_Positive_Jitter_SYGD and Peak_Negative_Jitter_SYGD* objects reset.

	Reset occurs on the writing of the *Option_SYGD* object. The bit state after the writing has no effect. 0 – No action 1 – *Peak_Positive_Jitter_SYGD* and *Peak_Negative_Jitter_SYGD* objects reset
5	*Out_Of_Sync_Counter_SYGD* counter reset. Reset occurs on the writing of the *Option_SYGD* object. The bit state after the writing has not effect. 0 – No action 1 – *Out_Of_Sync_Counter_SYGD* object reset
6	*Peak_Out_Of_Sync_SYGD* object reset. Reset occurs on the writing of the *Option_SYGD* object. The bit state after the writing has no effect. 0 – No action 1 – *Peak_Out_Of_Sync_SYGD* object reset
7	*Sync_Counter_SYGD* reset. Reset occurs on the writing of the *Option_SYGD* object. The bit state after the writing has no effect. 0 – No action 1 – *Sync_Counter_SYGD* reset

Sub-Index	Name			Mnemonic
02_h	*ErrorClass_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		💾
	Default Value	Minimum	Maximum	Unit
	2			

Description

It allows to define the class of the emergency generated in case of violation of the timing waited for Sync. The following are the possible values:

Class	Description	Driver's reaction
0	Warning, there is no impediment to continue with the ongoing operations.	None
2	Error that requires the motor stop but not the transition to FAULT.	Deceleration with Quick Stop.
4	Error that requires the motor stop and subsequent transition to FAULT.	Deceleration with Quick Stop and subsequent motor disabling.

6	Error that requires the immediate motor disabling and the transition to FAULT.	Motor disabling

Sub-Index	Name			Mnemonic
03h	*Sync_Counter_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	Counter of the received Sync independently from their timing. The counter advances one unit each time a SYNC object is received. The counter can be reset by the bit 7 of the *Option_SYGD* object.			

Sub-Index	Name			Mnemonic
04h	*Sync_Expected_Period_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		Changeable only with monitoring disabled 🖫
	Default Value	Minimum	Maximum	Unit
	1000	1000	100000	1us (Ex. 2500 = 2.5ms)
	Description			
	It allows to set the nominal period with which the SYNC object is received.			

Sub-Index	Name			Mnemonic
05h	*Sync_Max_Jitter_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		Changeable only with monitoring disabled 🖫
	Default Value	Minimum	Maximum	Unit
	200	1	10000	1us (Ex. 200 = 200us)
	Description			
	It allows to set the maximum allowable tolerance for receiving the SYNC object with respect to the nominal period. If the SYNC object is received in advance or in delay, with respect to the set nominal period, of a time greater than the value set in this, the *Out_Of_Sync_Counter_SYGD* object counter is incremented by 10 units. The value of the *Sync_Max_Jitter_SYGD* object cannot be set to a value greater than 50% of the value set in the *Sync_Expected_Period_SYGD* object. Otherwise the activation of the monitoring will generate an error.			

Sub-Index	Name			Mnemonic
06h	*Out_Of_Sync_Threshold_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw		🖫 Changeable only with monitoring disabled
	Default Value	Minimum	Maximum	Unit
	15		250	
	Description			

	It allows to set the threshold for the *Out_Of_Sync_Counter_SYGD* object beyond which an emergency is generated, if the bit 1 of the *Option_SYGD* object is set to 1.

Sub-Index	Name			Mnemonic
07h	*Peak_Out_Of_Sync_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	The object memorize the maximum value reached by the counter *Out_Of_Sync_Counter_SYGD* from the activation of the monitoring . The value can be reset by the *Option_SYGD* object.			

Sub-Index	Name			Mnemonic
08h	*Out_Of_Sync_Counter_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	It represents a counter that is incremented by 10 units each time the SYNC object is received, out of the nominal period, of a time greater than the value set through the *Sync_Max_Jitter_SYGD* object. The counter is instead decremented by 1 unit for each SYNC received inside the nominal period (considering the tolerance set through the *Sync_Max_Jitter_SYGD* object). The value can be reset through the *Option_SYGD* object.			

Sub-Index	Name			Mnemonic
09h	*Sync_Measured_Period_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	0			1µs (Ex. 1000 = 1ms)
	Description			
	Indicates the period measured between two consecutive Sync.			

Sub-Index	Name			Mnemonic
0Ah	*Peak_Positive_Jitter_SYGD*			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro		
	Default Value	Minimum	Maximum	Unit
	0			1µs (Ex. 250 = 0.25ms)
	Description			
	It reports the maximum positive deviation between the measured Sync period (*Sync_Measured_Period_SYGD* object) and the expected nominal period (*Sync_Expected_Period_SYGD* object). For example, a value of 22 indicates that at least once the Sync period was 22µs higher than the nominal value (*Sync_Expected_Period_SYGD* object). The value can be reset through the *Option_SYGD* object.			

Sub-Index	Name	Mnemonic
0Bh	*Peak_Negative_Jitter_SYGD*	

	Data Type	Access Type	PDO Mapping	Note
	i16	ro		
	Default Value	Minimum	Maximum	Unit
	0			1µs (Ex. 16 = 16µs)
	Description			
	It reports the maximum negative deviation between the measure Sync period (*Sync_Measured_Period_SYGD* object) and the expected nominal period (*Sync_Expected_Period_SYGD* object). For example, a value of -15 indicates that at least once the Sync period was 15µs lower than the nominal value (*Sync_Expected_Period_SYGD* object). The value can be reset through the *Option_SYGD* object.			

8.2.17 3120h Fault and Error Status

In the event of errors, detailed information on the detected error or on the objects that caused a read or write access violation of the SDO or PDO type, can be found within this record.

Index	Name	Type
3120h	*Error Status*	RECORD

Sub-Index		Name		Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01h			
	Description			
	Highest sub-index supported			

Sub-Index		Name		Mnemonic
01h	*Code_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	U16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	Provides the code of the last error occurred in the device.			

Sub-Index		Name		Mnemonic
02h	*SubCode_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	Provides the subcode of the last error occurred in the device.			

Sub-Index		Name		Mnemonic
03h	*RPDO_Time_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
				100µs
	Description			

Indicates the time in which the last RPDO error occurred.

The time is measured starting from the power on of the drive and is expressed in 100µs. A value of 26.738.576 for example indicates that the error occurred 44 minutes 33 seconds and 857.6 milliseconds after the power on.

Sub-Index	Name			Mnemonic
04h	*RPDO_Index_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit

Description

Indicates the Index and Sub-Index of the object that caused the last RPDO error.

The object is made up in bits as described in the following table:

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Not used

Sub-Index	Name			Mnemonic
05h	*RPDO_Value_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit

Description

Indicates the value of the object that caused the last RPDO error.

Sub-Index	Name			Mnemonic
06h	*RPDO_Code_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit

Description

Indicates the RPDO error code according to the "*SDO abort codes*" shown in the specification /CiA301/ available on the official website www.can-cia.org.

Sub-Index	Name			Mnemonic
07h	*TPDO_Time_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
				100µs

Description

Indicates the time in which the last TPDO error occurred.

The time is measured starting from the power on of the drive and is expressed in 100µs. A value of 8.232.375 for example indicates that the error occurred 13 minutes 43 seconds and 237.5

	milliseconds after the power on.

Sub-Index	Name			Mnemonic
08h	*TPDO_Index_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	Indicates the Index and Sub-Index of the object that caused the last TPDO error. The object is made up in bits as described in the following table:			

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Not used

Sub-Index	Name			Mnemonic
09h	*TPDO_Code_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	Indicates the RPDO error code according to the "*SDO abort codes*" shown in the specification /CiA301/ available on the official website www.can-cia.org.			

Sub-Index	Name			Mnemonic
0Ah	*DownloadSDO_Time_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
				100µs
	Description			
	Indicates the time in which the last Download SDO error occurred. The time is measured starting from the power on of the drive and is expressed in 100µs. A value of 26.738.576 for example indicates that the error occurred 44 minutes 33 seconds and 857.6 milliseconds after the power on.			

Sub-Index	Name			Mnemonic
0Bh	*DownloadSDO_Index_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	Indicates the Index and Sub-Index of the object that caused the last Download SDO error. The object is made up in bits as escribed in the following table:			

Bits 31…16	Bits 15…8	Bits 7…0

Index	Sub-Index	Not used

Sub-Index	Name			Mnemonic
$0C_h$	*DownloadSDO_Value_ERRSTS*			
Data Type	Access Type	PDO Mapping		Note
u32	ro			
Default Value	Minimum	Maximum		Unit
Description				
Indicates the value of the object that caused the last Download SDO error.				

Sub-Index	Name			Mnemonic
$0D_h$	*DownloadSDO_Code_ERRSTS*			
Data Type	Access Type	PDO Mapping		Note
u32	ro			
Default Value	Minimum	Maximum		Unit
Description				
Indicates the Download SDO error code according to the "*SDO abort codes*" shown in the specification /CiA301/ available on the official website www.can-cia.org.				

Sub-Index	Name			Mnemonic
$0E_h$	*UploadSDO_Time_ERRSTS*			
Data Type	Access Type	PDO Mapping		Note
u32	ro			
Default Value	Minimum	Maximum		Unit
				100µs
Description				
Indicates the time in which the last Upload SDO error occurred. The time is measured starting from the power on of the drive and is expressed in 100µs. A value of 8.232.375 for example indicates that the error occurred 13 minutes 43 seconds and 237.5 milliseconds after the power on.				

Sub-Index	Name			Mnemonic
$0F_h$	*UploadSDO_Index_ERRSTS*			
Data Type	Access Type	PDO Mapping		Note
u32	ro			
Default Value	Minimum	Maximum		Unit
Description				

Indicates the Index and Sub-Index of the object that caused the last Upload SDO error.

The object is made up in bits as described in the following table:

Bits 31…16	Bits 15…8	Bits 7…0
Index	Sub-Index	Not used

Sub-Index	Name			Mnemonic
10_h	*UploadSDO_Code_ERRSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	Indicates the Upload SDO error code according to the "*SDO abort codes*" shown in the specification /CiA301/ available on the official website www.can-cia.org.			

8.2.18 3140_h Device Status

Index	Name	Type
3140_h	*Device Status*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	03_h			
	Description			
	Highest sub-index supported			

Sub-Index	Name			Mnemonic
01_h	*Bridge_Temperature_DVSTS*			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1°C (Ex. 528 = 52.8°C)
	Description			
	It indicates the temperature reached by the power stage.			

Sub-Index	Name			Mnemonic
02_h				
	Data Type	Access Type	PDO Mapping	Note
	i16	ro		
	Default Value	Minimum	Maximum	Unit
	Description			
	Reserved, do not use.			

Sub-Index	Name			Mnemonic
03_h	*Power_Voltage_DVSTS*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1Vdc (ex. 482 = 48.2Vdc)
	Description			
	It indicates the voltage of the power bus.			

Sub-Index	Name	Mnemonic
09_h	*MotorTemperature_DVSTS*	

	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1°C (Ex. 745 = 74.5°C)
	Description			
	It indicates the temperature reached by the motor.			

8.2.19 3210_h Digital Inputs Value

Index	Name	Type
3210_h	*Digital Inputs Value*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01_h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Inputs_DIV*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			

It indicates the logic status of the digital inputs.

The bits of the object are associated to the digital inputs as follows:

Bit	Description
0	Digital Input 0 (DI0)
1	Digital Input 1 (DI1)
2	Digital Input 2 (DI2)
3	Digital Input 3 (DI3)
4	Digital Input 4 (DI4)
5	Digital Input 5 (DI5)
6..15	Reserved, ignore the value

A bit value = 1 indicates active input, on the contrary if the bit value is 0 it means that the logic status of the input is inactive.

8.2.20 3230h Digital Outputs Value

Index	Name	Type
3230h	*Digital Outputs Value*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	02h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Outputs_DOV*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	Description			

It indicates the logic status of the digital outputs and allows to set the value for the outputs configured as *General Purpose.*

The bits of the object are assigned to the digital outputs as follows:

Bit	Description
0	Digital Output 0 (DO0)
1	Digital Output 1 (DO1)
2	Digital Output 2 (DO2)
3..15	Reserved, ignore the value

A bit value = 1 indicates active output, on the contrary if the bit value is 0 it means that the logic status of the output is inactive.

If the output is assigned to a source inside the drive (through the *Digital Output n Action* object) the modification of the corresponding bit will not be possible and any attempt of that kind will be ignored.

Sub-Index	Name			Mnemonic
02h	*Set_Output_DOV*			
	Data Type	Access Type	PDO Mapping	Note
	U8	wo	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	Description			

It allows to modify one single bit of the *Outputs_DOV* object.

To modify one single bit of the *Outputs_DOV* object simply write this object with a value made up as follows: the bits from bit 0 to bit 3 must contain the number of bits of the *Outputs_DOV* object on which you want to act, while the bit 7 will indicate whether the bit must be set to 1 or 0. When the bit 7 is equal to 1, the bit of the *Outputs_DOV* object is set to 0 and vice versa.

For example, if the *Set_Output_DOV* object is written with the value 81h, the bit 1 of the *Outputs_DOV* object will be set to 0, instead writing the value 01h the same bit will be set to 1.

if we write the *Set_Output_DOV* object with value 00_h, the bit 0 of the *Outputs_DOV* object will be set to 1.

The following table shows the use of the bits of the *Set_Output_DOV* object.

Bit	Description
0..3	Number of the bits of the *Outputs_DOV* object on which you want to act. Value: 0 – Bit 0; 1 – Bit 1; … – Bit n; 15 – Bit 15
4..6	Not used, set to 0
7	Logic level to set in the bit of the *Outputs_DOV* object Value: 0 – Bit = 1; 1 – Bit = 0

If the selected bit of the *Outputs_DOV* object corresponds to an output assigned to a source inside the drive (through the *Digital Output n Action* object), the modification of the corresponding bit will not be possible and any attempt of that kind will be ignored.

8.2.21 3250_h Analog Input 0 Value

Index	Name	Type
3250_h	*Analog Inputs 0 Value*	RECORD

Sub-Index	Name			Mnemonic
00_h	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping:	Note:
	Default Value: 01_h	Minimum:	Maximum:	Unit:
	Description: Number of record entries.			

Sub-Index	Name			Mnemonic
01_h	*Voltage_AIV*			
	Data Type: i16	Access Type: ro	PDO Mapping: TPDO	Note:
	Default Value:	Minimum:	Maximum:	Unit: 1mV (Ex. 5302 = 5.302V)
	Description: It indicates the voltage at the analog input 0.			

8.2.22 3251h Analog Input 1 Value

Index	Name	Type
3251h	*Analog Inputs 1 Value*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Voltage_AIV_1*			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				1mV (Ex. -6288 = -6.288V)
	Description			
	It indicates the voltage at the analog input 0.			

8.2.23 3310h Motor Configuration

Index	Name	Type
3310h	*Motor Configuration*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	04h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h				
	Data Type	Access Type	PDO Mapping	Note
	u8	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	Description			
	Reserved. Do not use.			

Sub-Index	Name			Mnemonic
02h	*Current_Min_MTRCNF*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	🖫
	Default Value	Minimum	Maximum	Unit
	4000			0.01% (Ex. 2508 = 25.08%)
	Description			
	It allows to set the minimum value of the motor phase current.			

When the current regulation is set dynamically (through the *Mode_CRRG* object) the minimum current value represents the current supplied to the motor without load. Instead, if the current regulation is static, it defines then the idle current supplied to the motor after the *Current_Idle_Delay_MTRCNF* time from the stop.

The value is expressed as a percentage of the motor rated current set by the *Rated_Current_MTRDT* object. For example, if the motor has a rated current of 4Arms and a minimum current equal to 25%, the drive will never supply less than 1Arms to the motor.

Sub-Index	Name			Mnemonic
03_h	*Current_Max_MTRCNF*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	8000			0.01% (Ex. 7550 = 75.5%)

Description

It allows to set the maximum value of the motor phase current.

When the current regulation is set dynamically (through the *Mode_CRRG* object) the maximum current value represents the current supplied to the motor in locked rotor condition. Instead, if the current regulation is static, the *Current_Max_MTRCNF* object defines the current supplied to the motor when it is rotating.

The value is expressed as a percentage of the motor rated current set by the *Rated_Current_MTRDT* object. For example, if the motor has a rated current of 4Arms and a minimum current equal to 75%, the drive will never supply less than 3Arms to the motor.

Sub-Index	Name			Mnemonic
04_h	*Current_Idle_Delay_MTRCNF*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	500	2	10000	1ms (Ex. 3500 = 3.5s)

Description

It allows to set the motor halting time before the current reaches the value defined by the *Current_Min_MTRCNF* object.

When the dynamic current regulation mode is active this object has no effect.

8.2.24 3312h Motor Value

Index	Name	Type
3312h	*Motor Value*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping	Note
	Default Value: 02h	Minimum	Maximum	Unit
	Description: Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Current_MTRV*			
	Data Type: I16	Access Type: ro	PDO Mapping: TPDO	Note
	Default Value	Minimum	Maximum	Unit: 10mArms (Ex. 225 = 2.25 Arms)
	Description: It indicates the actual current supplied to the motor expressed in effective value.			

Sub-Index	Name			Mnemonic
02h	*Utilization_MTRV*			
	Data Type: i16	Access Type: ro	PDO Mapping: TPDO	Note
	Default Value	Minimum	Maximum	Unit: 0.1% (Ex. 705 = 70.5%)
	Description: It indicates the ratio between the torque used by the load and the actual torque that can be supplied by the motor.			

8.2.25 3330h Motor Incremental Encoder Value

Index	Name	Type
3330h	*Encoder Motor Value*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping	Note
	Default Value: 03h	Minimum	Maximum	Unit
	Description: Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Status_ENCMTRV*			
	Data Type: u8	Access Type: ro	PDO Mapping: TPDO	Note
	Default Value	Minimum	Maximum	Unit
	Description: The object is useful to know the status of the motor incremental encoder.			

The following table shows the meaning of the bits of the object.

Bit	Description
0	Logic level of the Index signal
1	Logic level of the phase B
2	Logic level of the phase A
7..3	Not used. Ignore the value.

Sub-Index	Name			Mnemonic
02h	*PositionRaw_ENCMTRV*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	It indicates the cyclical position on the revolution of the motor incremental encoder.			

Sub-Index	Name			Mnemonic
03h	*Frequency_ENCMTRV*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				Hz (Ex. 4365 = 4.365KHz)
	Description			
	It indicates the frequency of the pulses generated by the motor incremental encoder.			

8.2.26 3338h SSI Encoder Value

Index	Name	Type
3338h	*Encoder SSI Value*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	03h			
				Description
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Status_ENCSSIV*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				10mArms (Ex. 225 = 2.25 Arms)
				Description

This object is useful to know the status of the SSI encoder.

The following table resumes the meaning of the bits of the object.

Bit	Description
0	Valid data Value / Description: 0 – SSI encoder reading not operative or with error 1 – SSI encoder reading operative When the value of the bit is 0, the *Position_ENCSSIV* and *PositionRaw_ENCSSIV* objects may not reflect the position of the SSI encoder.
1	Errors Value / Description: 0 – No errors in the data frame 1 – Errors in the data frame
2	Not used. Ignore the value.
3	Power supplied Value / Description: 0 – No encoder power supply 1 – Encoder power supply The bit assumes significance when the SSI encoder is configured with

	the bit3 of the *Configuration_ENCSSI* object to 1, differently the bit is set to 0.
7..4	Not used. Ignore the value.

Sub-Index	Name			Mnemonic
02_h	*Position_ENCSSIV*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 5000 = 0.5000rev)

Description

It indicates the position expressed in 0.0001rev obtained from the SSI encoder.

The formula used to obtain the position expressed in 0.0001rev is the following:

Position_ENCSSIV = (*PositionRaw_ENCSSIV* - *PositionOffset_ENCSSI*) * *ScalingFactor_N_ENCSSI* / *ScalingFactor_D_ENCSSI*

Sub-Index	Name			Mnemonic
03_h	*PositionRaw_ENCSSIV*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

It indicates the quota read by the SSI encoder.

8.2.27 3360h Holding Brake Control

Index	Name	Type
3360h	*Holding Brake Control*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	02h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Control_HBRKC*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0			
	Description			

It allows you to manually control the holding brake.

The following table resumes the use of the bits of the object.

Bit	Description
0	Manual control It allows you to manually control the holding brake. Value / Description: 0 – Inactive manual control signal 1 – Active manual control signal The effect on the brake of the logic level assumed by this bit depends on the configuration of the *Option_HBRKS* object, therefore for a detailed description refer to chapter 8.2.7 2360h Holding Brake Setup The bit0 is in logic OR with the digital input chosen for the manual control of the brake.
1..7	Not used, set to 0.

Sub-Index	Name			Mnemonic
02h	*Status_HBRKC*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	The object is useful to know the state of the holding brake. The following table resumes the meaning of the bits of the object.			

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Active

It indicates the actual state of the output arranged for the control of the holding brake.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>The brake control output is in the inactive state</td></tr>
<tr><td>1</td><td>The brake control output is in the active state</td></tr>
</table>
</td></tr>
<tr><td>1</td><td>Released

It allows you to know if the Release_Time_HBRKS has elapsed and the brake is then in the released state.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>The brake is not released</td></tr>
<tr><td>1</td><td>The brake is released</td></tr>
</table>
</td></tr>
<tr><td>2</td><td>Engaged

It allows you to know if the Application_Time_HBRKS has elapsed and the brake is then in the engaged state.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>The brake is not engaged</td></tr>
<tr><td>1</td><td>The brake is engaged</td></tr>
</table>
</td></tr>
</table>

8.2.28 3450h Profile Velocity Configuration

Index	Name	Type
3450h	*Profile Velocity Configuration*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Max_Slippage_TimeOut_PVCNF*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	10		30000	1ms (Ex. 280 = 280ms)
	Description			
	It allows to set the time after which the bit *Max slippage error* of the *Statusword* is set to 1 in the *Profile Velocity* mode. The error *Max slippage error* occurs when the motor speed differs from the required speed beyond the value set through the *Max_slippage* object. ℹ The drive is able to know the actual motor speed only if the motor is provided with encoder and the drive is configured in closed loop mode.			

8.2.29 3470h Profile Torque Configuration

Index	Name	Type
3470h	*Profile Torque Configuration*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	01h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Mode_PTCNF*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	It allows to configure the options of the *Profile Torque* mode. Normally in the torque control the motor speed is limited only by the load and by the			

characteristic of the motor. However, in many applications it is useful to set a maximum speed to prevent the motor to reach high speeds without load. Setting to 1 the bit 0 of the *Mode_PTCNF* object, the drive limits the motor maximum speed to the value set through the *Profile_velocity* object.

The following table shows the use of the bits of the object.

Bit	Description
0	Speed limitation Value: 0 – Speed not limited by the *Profile_velocity* object Value: 1 – Speed limited by the *Profile_velocity* object
1..7	Not used, set to 0.

8.2.30 3490h Interpolated Profile Configuration

Index	Name	Type
3490h	*Interpolation Configuration*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type: u8	Access Type: const	PDO Mapping:	Note:
	Default Value: 02h	Minimum:	Maximum:	Unit:
	Description: Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Mode_IPCNF*			
	Data Type: u8	Access Type: rw	PDO Mapping: RPDO	Note: 💾
	Default Value: 0	Minimum:	Maximum:	Unit:
	Description: Allows you to configure the options used by the interpolator. The following table resumes the use of the object bits.			

Bit	Description
0..5	Not used, set to 0.
6	Micro-interpolation. Between a set-point and the next, the integrated motor is able to interpolate the path with intermediate micro positions in order to improve the smoothness of the movement. Value – Description

	0 Micro-interpolation active 1 Micro-interpolation disabled
7	Limitation of acceleration and deceleration. In the *Interpolated Position* mode, usually it is the profile generator that limits the speed and the accelerations during the construction of the motion path, however, in some case, it can be advantageous to have a ramp limitation also from the integrated motor. Value: 0 — Unlimited acceleration and deceleration Value: 1 — Limited acceleration and deceleration When the limitation is active, the acceleration and deceleration values are set through the *Profile_acceleration* and *Profile_deceleration* objects respectively.

Sub-Index	Name		Mnemonic
02_h	*Dampening_A_Value_IPCNF*		
Data Type: i8	Access Type: rw	PDO Mapping: RPDO	Note: 💾
Default Value: 0	Minimum	Maximum	Unit

Description

In the *Interpolated Position* mode if the set—points update will not take place with regular frequency, according to the period set through the *Interpolation_time_period* object, the movement of the motor becomes irregular. Through this object it is possible to prepare a filtering that makes tolerable modest changes in the set-points update frequency.

Positive values increase the effect of the filtering while negative values reduce it.

8.2.31 34B0h Cyclic Synchronous Velocity Configuration

Index	Name	Type
34B0h	*Cyclic Synchronous Velocity Configuration*	RECORD

Sub-Index	Name			Mnemonic
00h	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	02h			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01h	*Mode_CSVCNF*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0			
	Description			

Allows to configure the options used by the interpolator in *Cyclic Synchronous Velocity* (CSV) mode.

The following table summarizes the use of the object bits.

Bit	Description
0..5	Not used, set to 0.
6	Micro-interpolation Between a speed set point and the next, the drive is able to interpolate the path with intermediate speeds in order to improve the movement smooth. Value 0: Micro-interpolation active Value 1: Micro-interpolation disabled
7	Limitation of acceleration and deceleration. In the *Cyclic Synchronous Velocity* mode, it is normally the profile generator to limit the speed and the accelerations during the construction of a motion path, however in some cases it can be advantageous to have a limitation of the ramps also on the part of the drive. Value 0: Acceleration and deceleration not limited Value 1: Acceleration and deceleration limited When the limitation is active, the acceleration and deceleration values can be set using the *Profile_acceleration* and *Profile_deceleration* objects respectively.

8.3 Object with index between 6000_h and $9FFF_h$

The object with index between 6000_h e $9FFF_h$ are specific to the profile /CiA402/. Their function, the access mode, etc. are described in the manuals of the DSP402 standard available on the official site www.can-cia.org, therefore in this manual there will be no detailed description of each object but a synthetic summary of the implemented ones.

8.3.1 $603F_h$ Error code

Index	Name			Mnemonic
$603F_h$	*Error_code*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	This object provides the code of the last error occurred in the device.			

8.3.2 6040_h Controlword

Index	Name			Mnemonic
6040_h	*Controlword*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	
	Default Value	Minimum	Maximum	Unit

Description

This object indicates the received command controlling the device FSA.

It is structured in bits and the functionality of some bit changes according to the mode of operation chosen with the *Modes of operation* object.

Bit	Description		Bit	Description	
15	r	Reserved	7	fr	Fault reset
14	r	Reserved	6	abrl	Absolute / Relative
13	r	Reserved		rr	Reference ramp
12	r	Reserved	5	csi	Change set immediately
11	r	Reserved		ulkr	unlock ramp
10	r	Reserved	4	nsp	New set-point
9	cosp	Change on setpoint		hos	Homing operation start
8	h	Halt		ei	Enable interpolation
				er	enable ramp
			3	eo	Enable operation
			2	qs	Quick stop
			1	ev	Enable voltage
			0	so	Switch on

Mode of operation

Profile position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r	cosp	h	fr	abrl	csi	nsp	eo	qs	ev	so

Mode of operation

Velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	rr	ulkr	er	f	oe	so	rsto

Mode of operation

Profile velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	r	r	r	f	oe	so	rsto

Mode of operation

Profile torque mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	r	r	r	f	oe	so	rsto

Mode of operation

Homing mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	0	0	hos	f	oe	so	rsto

Mode of operation

Interpolated position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr	0	0	ei	f	oe	so	rsto

Mode of operation

Cyclic synchronous position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr				f	oe	so	rsto

Mode of operation

Cyclic synchronous velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr				f	oe	so	rsto

Mode of operation

Cyclic synchronous torque mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r		h	fr				f	oe	so	rsto

Mode of operation

Cyclic synchronous torque mode with commutation angle

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					r			fr				f	oe	so	rsto

8.3.3 6041h Statusword

Index	Name			Mnemonic
6041h	*Statusword*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

This object provides the status of the device FSA.

It is made up in bits and the functionality of some bit changes according to the mode of operation chosen with the *Modes of Operation* object.

Bit	Description		Bit	Description	
15	r	Reserved	7	w	Warning
14	r	Reserved	6	sod	Switch on disabled
13	fe	Following error	5	qs	Quick stop (0=active)
	he	Homing error	4	ve	Voltage enabled
	mse	Max slippage error	3	f	Fault
12	spa	Set-point acknowledge	2	oe	Operation enabled
	ha	Homing attained	1	so	Switched on
	ipa	Ip mode active	0	rsto	Ready to switch on
	spd	Speed is equal 0			
	dfcv	Drive follows the command value			
11	ila	Internal limit active			
10	tr	Target reached (or Velocity = 0)			
9	rm	Remote			
8	h	Halt			

Mode of operation

Profile position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		fe	spa	ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		0	0	ila	0	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Profile velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		mse	spd	ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Profile torque mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Homing mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		he	ha	ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Interpolated position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		fe	ipa	ila	tr	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Cyclic synchronous position mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		fe	dfcv	ila	r	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Cyclic synchronous velocity mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			dfcv	ila	r	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Cyclic synchronous torque mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			dfcv	ila	r	rm	h	w	sod	qs	ve	f	oe	so	rsto

Mode of operation

Cyclic synchronous torque mode with commutation angle

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			dfcv	ila	r	rm	h	w	sod	qs	ve	f	oe	so	rsto

8.3.4 6060_h Modes of operation

Index	Name			Mnemonic
6060_h	*Modes_of_operation*			
	Data Type	Access Type	PDO Mapping	Note
	i8	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	1			

Description

This object allows you to select the operation mode.

The table below shows the values definition:

1	Profile Position
3	Profile Velocity
4	Torque Profile
6	Homing
8	Cyclic synchronous position
9	Cyclic synchronous velocity

LAM Technologies electronic equipment

8.3.5 6061_h Modes of operation display

Index	Name			Mnemonic
6061_h	*Modes_of_operation_display*			
	Data Type	Access Type	PDO Mapping	Note
	i8	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			
	This object provides the active operation mode. The table below shows the values definition:			

1	Profile Position
3	Profile Velocity
4	Torque Profile
6	Homing
8	Cyclic synchronous position

8.3.6 6062_h Position demand value

Index	Name			Mnemonic
6062_h	*Position_demand_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 45524 = 4.5524rev)
	Description			
	This object provides the demanded position value. The value is given in 1/10000 of revolution.			

8.3.7 6064_h Position actual value

Index	Name			Mnemonic
6064_h	*Position_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 5000 = 0.5000rev)
	Description			
	This object provides the actual position value. The value is given in 1/10000 of revolution.			

8.3.8 6065_h Following error window

Index	Name			Mnemonic
6065_h	*Following_error_window*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	10000			0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This object indicates the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. If the value of the following error window is $FFFFFFFF_h$, the following control is switched off. The value is given in 1/10000 of revolution.			

8.3.9 6066_h Following error time out

Index	Name			Mnemonic
6066_h	*Following_error_time_out*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	10		30000	1ms (Ex. 250 = 250ms)
	Description			
	This object indicates the configured time for a following error condition, after that the bit 13 of the *Statusword* is set to 1. The value is given in ms.			

8.3.10 6067_h Position window

Index	Name			Mnemonic
6067_h	*Position_window*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	10			0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This object indicates the configured symmetrical range of accepted positions relative to the target position. If the actual value of the position is within the position window, this target position is regarded as having been reached. The value is given in 1/10000 of revolution.			

8.3.11 6068h Position window time

Index	Name			Mnemonic
6068h	*Position_window_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 600 = 0.6s)
	Description			
	This object indicates the configured time, during which the actual position within the position window is measured. The value is given in ms.			

8.3.12 606Bh Velocity demand value

Index	Name			Mnemonic
606Bh	*Velocity_demand_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1rpm (Ex. 4525 = 452.5rpm)
	Description			
	This object shall provide the output value of the trajectory generator. The value is given in 0.1rpm.			

8.3.13 606Ch Velocity actual value

Index	Name			Mnemonic
606Ch	*Velocity_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1rpm (Ex. 3850 = 385.0rpm)
	Description			
	This shall provide the actual velocity. The value is given in 0.1rpm.			

8.3.14 606Dh Velocity window

Index	Name			Mnemonic
606Dh	*Velocity_window*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	100			0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This object indicates the configured velocity window. If the actual velocity value of the motor is within the velocity window, this target velocity is regarded as having been reached.			

	The value is given in 0.1rpm.

8.3.15 606E$_h$ Velocity window time

Index	Name			Mnemonic
606E$_h$	*Velocity_window_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 200 = 200ms)
	Description			
	This object indicates the configured velocity window time. The value is given in ms.			

8.3.16 606F$_h$ Velocity threshold

Index	Name			Mnemonic
606F$_h$	*Velocity_threshold*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	60			0.1rpm (Ex. 500 = 50.0rpm)
	Description			
	This object indicates the configured velocity threshold. If the instantaneous velocity value of the motor is within the velocity threshold, the motor is regarded as stopped. The value is given in 0.1rpm.			

8.3.17 6070$_h$ Velocity threshold time

Index	Name			Mnemonic
6070$_h$	*Velocity_threshold_time*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 200 = 200ms)
	Description			
	This object indicates the configured velocity threshold time. The value is given in ms.			

8.3.18 6071$_h$ Target Torque

Index	Name			Mnemonic
6071$_h$	*Target_torque*			
	Data Type	Access Type	PDO Mapping	Note
	i16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0	-1300	1300	0.1% (Ex. 405 = 40.5%)

	Description This object indicates the configured input value for the torque controller in profile torque mode. The value is given in 0.1% of the nominal motor torque.

8.3.19 6072h Max torque

Index	Name			Mnemonic
6072h	*Max_torque*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	1000			0.1% (Ex. 650 = 65.0%)
	Description The object indicates the maximum torque value. The value is given in 0.1% of the nominal motor torque.			

8.3.20 6075h Motor rated current

Index	Name			Mnemonic
6075h	*Max_torque*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		💾
	Default Value	Minimum	Maximum	Unit
	1000			1mA (Ex. 2500 = 2.5A)
	Description The object indicates the nominal motor current.			

8.3.21 6076h Motor rated torque

Index	Name			Mnemonic
6076h	*Max_torque*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		💾
	Default Value	Minimum	Maximum	Unit
	500			1mNm (Ex. 1400 = 1.4Nm)
	Description The object indicates the nominal motor torque.			

8.3.22 6077h Torque actual value

Index	Name			Mnemonic
6077h	*Torque_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1% (Ex. 405 = 40.5%)
	Description			
	This object provides the actual value of the torque. The value is given in 0.1% of the nominal motor torque.			

8.3.23 6078h Current actual value

Index	Name			Mnemonic
6078h	*Current_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.1% (Ex. 800 = 80.0%)
	Description			
	This objects provides the actual value of the current supplied to the motor. The value is given in 0.1% of the nominal motor current.			

8.3.24 607Ah Target position

Index	Name			Mnemonic
607Ah	*Target_position*			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0			0.0001rev (Ex. 50000 = 5.0000rev)
	Description			
	This object indicates the commanded position that the drive should move to in position profile mode. The value of this object shall be interpreted as absolute or relative depending on the abs/rel flag in the *Controlword*. The value is given in 1/10000 of revolution.			

8.3.25 607Ch Home offset

Index	Name			Mnemonic
607Ch	*Home_offset*			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This object indicates the configured difference between the zero position for the application and the machine home position (found during homing). During homing, the machine home position is found and once the homing is completed, the *Position_actual_value* object is set to the value of the *Home_offset* object.			

	The value is given in 1/10000 of revolution.

8.3.26 607D$_h$ Software position limit

To disable the software position limits, the *Min position limit* and *Max position limit* objects shall be set both to 0.

Index	Name	Type
607D$_h$	*Software position limit*	RECORD

Sub-Index	Name			Mnemonic
00$_h$	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	02$_h$			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01$_h$	*Min_software_position_limit*			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0			0.0001rev (Ex. -1000000 = -100rev)
	Description			
	This object indicates the configured minimal software position limits. These parameters define the absolute position limits for the position demand value. The value is given in 1/10000 of revolution.			

Sub-Index	Name			Mnemonic
02$_h$	*Max_software_position_limit*			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0			0.0001rev (Ex. 1000000 = 100rev)
	Description			
	This object indicates the configured maximal software position limits. These parameters shall define the absolute position limits for the position demand value. The value is given in 1/10000 of revolution.			

8.3.27 607F$_h$ Max profile velocity

Index	Name			Mnemonic
607F$_h$	*Max_profile_velocity*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	30000	1	30000	0.1rpm (Ex. 5000 = 500.0rpm)
	Description			
	This object indicates the configured maximal allowed velocity in either direction during a profiled motion. The value is given in 0,1rpm.			

8.3.28 6081$_h$ Profile velocity

Index	Name			Mnemonic
6081$_h$	*Profile_velocity*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	600		30000	0.1rpm (Ex. 4500 = 450.0rpm)
	Description			
	This object indicates the configured velocity normally attained at the end of the acceleration ramp during a profiled motion and is valid for both directions of motion. The value is given in 0.1rpm.			

8.3.29 6083$_h$ Profile acceleration

Index	Name			Mnemonic
6083$_h$	*Profile_acceleration*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	100	5	300000	rpm/s (Ex. 100 = 100rpm/s)
	Description			
	This object indicates the configured acceleration. The value is given in rpm/s.			

8.3.30 6084$_h$ Profile deceleration

Index	Name			Mnemonic
6084$_h$	*Profile_deceleration*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	300	5	300000	rpm/s (Ex. 300 = 300rpm/s)
	Description			
	This object indicates the configured deceleration. The value is given in rpm/s.			

8.3.31 6085h Quick Stop deceleration

Index	Name			Mnemonic
6085h	*Quick_stop_deceleration*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	5000	5	300000	rpm/s (Ex. 5000 = 5000rpm/s)
	Description			
	This object indicates the configured deceleration used to stop the motor when the quick stop function is activated. The quick stop deceleration is also used if fault is detected in the device. The value is given in rpm/s.			

8.3.32 6087h Torque slope

Index	Name			Mnemonic
6087h	*Torque_slope*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
				0.1%/s (Ex. 558 = 55.8%/s)
	Description			
	This object indicates the configured rate of change of torque. The value is given in units of per thousand of rated torque per second.			

8.3.33 6098_h Homing method

Index	Name			Mnemonic
6098_h	*Homing_method*			
	Data Type	Access Type	PDO Mapping	Note
	i8	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit

Description

This object indicates the configured homing method that shall be used.

The following table describes the available homing modes and the sensors involved. The abbreviations used have the following meaning: **PLS**=positive limit switch, **NLS**=negative limit switch, **HS**=homing switch, **IDX**=index.

Homing methods selectable through the *Homing_method* object					
Cod.	Description	Sensors used			
		PLS	NLS	HS	IDX
-80	Homing on the position read by the SSI encoder				
1	At the start, if negative limit switch inactive counterclockwise direction up to the limit switch, then reverse and homing at the first index outside the negative limit switch. At the start, if negative limit switch active clockwise direction up to leave the limit switch, then homing at the first index outside the negative limit switch.		●		●
2	At the start, if positive limit switch inactive clockwise direction up to the limit switch, then reverse and homing at the first index outside the positive limit switch. At the start, if positive limit switch active counterclockwise direction up to leave the limit switch, then homing at the first index outside the positive limit switch.	●			●
3	At the start, if home switch inactive initial direction clockwise up to home switch, then reverse and homing at the first index outside the home switch. At the start, if home switch active initial direction counterclockwise up to leave the switch, then homing at the first index outside the home switch.			●	●
4	At the start, if home switch inactive initial direction clockwise up to home switch, then homing at the first index inside the home switch. At the start, if home switch active initial direction counterclockwise up to leave the switch, then reverse and homing at the first index inside the home switch.			●	●
5	At the start, if home switch active initial direction clockwise up to leave the switch, then homing at the first index outside the home switch. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then reverse and homing at the first index outside the home switch.			●	●

	6	At the start, if home switch active initial direction clockwise up to leave the switch, then reverse and homing at the first index inside the home switch. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then reverse and homing at the first index inside the home switch.			●	●
	7	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, reverse and homing at the first index outside the switch. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then homing at the first index outside the switch. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then homing at the first index outside the switch.	●		●	●
	8	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, homing at the first index inside the home switch. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then reverse up to find again the home switch and finally homing at the first index inside the switch. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then reverse and homing at the first index inside the switch.	●		●	●
	9	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it then reverse and homing at the first index inside the switch. In case of positive limit switch, reverse up to find the home switch, then homing at the first index inside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the first index inside the home switch.	●		●	●
	10	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it then homing at the first index outside the home switch. In case of positive limit switch, reverse up to find the home switch, then reverse up to leave the switch and homing at the first index outside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the first index outside the home switch.	●		●	●

11	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, reverse and homing at the first index outside the home switch. In case of negative limit switch, reverse up to find the home switch, continue up to leave the home switch and then homing at the first index outside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the first index outside the home switch.		●	●	●
12	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, homing at the first index inside the home switch. In case of negative limit switch, reverse up to find the home switch, continue up to leave the home switch and then reverse up to find again the home switch and finally homing at the first index inside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the first index inside the home switch.		●	●	●
13	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and homing at the first index inside the switch. In case of negative limit switch, reverse up to find the home switch, then homing at the first index inside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the first index inside the home switch.		●	●	●
14	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then homing at the first index outside the home switch. In case of negative limit switch, reverse up to find the home switch, reverse up to leave the switch and then homing at the first index outside the switch. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then homing at the first index outside the switch.		●	●	●
17	At the start, if negative limit switch inactive initial direction counterclockwise up to find the limit switch, then reverse and homing at the active/inactive switch transition. At the start, if negative limit switch active initial direction clockwise with homing at the active/inactive switch transition.		●		

18	At the start, if positive limit switch inactive initial direction clockwise up to the limit switch, then reverse and homing at the active/inactive switch transition. At the start, if negative limit switch active initial direction counterclockwise with homing at the active/inactive switch transition.	●			
19	At the start, if home switch inactive initial direction clockwise up to the home switch, then reverse and homing at the active/inactive switch transition. At the start, if home switch active initial direction counterclockwise with homing at the active/inactive switch transition.			●	
20	At the start, if home switch inactive initial direction clockwise up to the home switch, then homing at the inactive/active switch transition. At the start, if home switch active initial direction counterclockwise up to leave the switch, then reverse and homing at the inactive/active switch transition.			●	
21	At the start, if home switch active initial direction clockwise up to leave the switch, then homing at the active/inactive switch transition. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then reverse and homing at the active/inactive switch transition.			●	
22	At the start, if home switch active initial direction clockwise up to leave the switch, then reverse and homing at the inactive/active switch transition. At the start, if home switch inactive initial direction counterclockwise up to find the switch, then homing at the inactive/active switch transition.			●	
23	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, reverse and homing at the active/inactive switch transition. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then homing at the active/inactive switch transition. At the start, if home switch active initial direction counterclockwise up to leave the switch, then homing at the active/inactive switch transition.	●		●	
24	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, homing at the inactive/active switch transition. In case of positive limit switch, reverse up to find the home switch, continue up to leave the home switch, then reverse up to find again the home switch and finally homing at the inactive/active switch transition. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.	●		●	

25	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then reverse and homing at the inactive/active switch transition. In case of positive limit switch, reverse up to find the home switch, homing at the inactive/active switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.	●		●	
26	At the start, if home switch inactive initial direction clockwise up to find the home switch or the positive limit switch. In case of home switch, continue up to leave it, then homing at the active/inactive switch transition. In case of positive limit switch, reverse up to find the home switch, then reverse up to leave the switch and finally homing at the active/inactive switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the active/inactive switch transition.	●		●	
27	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, reverse and homing at the active/inactive switch transition. In case of negative limit switch, reverse up to find the home switch, continue up to leave the home switch, then homing at the active/inactive switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the active/inactive switch transition.		●	●	
28	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, homing at the inactive/active switch transition. In case of negative limit switch, reverse up to find the home switch, continue up to leave the home switch, then reverse up to find again the home switch and finally homing at the inactive/active switch transition. At the start, if home switch active initial direction clockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.		●	●	
29	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then reverse and homing at the inactive/active switch transition. In case of negative limit switch, reverse up to find the home switch, then homing at the inactive/active switch transition. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then reverse and homing at the inactive/active switch transition.		●	●	

30	At the start, if home switch inactive initial direction counterclockwise up to find the home switch or the negative limit switch. In case of home switch, continue up to leave it then homing at the active/inactive switch transition. In case of negative limit switch, reverse up to find the home switch, then reverse up to leave the switch, then homing at the active/inactive switch transition. At the start, if home switch active initial direction counterclockwise up to leave the home switch, then homing at the active/inactive switch transition.		●	●	
33	Initial direction counterclockwise with homing at the first index found.				●
34	Initial direction clockwise with homing at the first index found.				●
37	Homing at the actual position.				

8.3.34 6099$_h$ Homing speeds

Index	Name	Type
6099$_h$	*Homing speeds*	RECORD

Sub-Index	Name			Mnemonic
00$_h$	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	02$_h$			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01$_h$	*Speed_during_search_for_switch*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	600		30000	0.1rpm (Ex. 600 = 60.0rpm)
	Description			
	This object indicates the speed during search for switch in homing procedure. The value is given in 0.1rpm.			

Sub-Index	Name			Mnemonic
02$_h$	*Speed_during_search_for_zero*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	60		30000	0.1rpm (Ex. 100 = 10.0rpm)
	Description			
	This object indicates the speed during search for zero in homing procedure. The value is given in 0.1rpm.			

8.3.35 60B8h Touch probe function

Index	Name			Mnemonic
60B8h	*Touch probe function*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0x0000			

Description

It allows you to configure the touch probe functions.
The bits of the low byte define the *Touch probe 1* functions while the high byte contains the bits related to *Touch probe 2*.

The following table shows the use of the bits and the allowed values:

Touch probe 2		*Touch probe 1*	
Bit	Description	Bit	Description
15, 14	Not used. Set to 0	7, 6	Not used. Set to 0
13	0 Switch off sampling at negative edge 1 Enable sampling at negative edge	5	0 Switch off sampling at negative edge 1 Enable sampling at negative edge
12	0 Switch off sampling at positive edge 1 Enable sampling at positive edge	4	0 Switch off sampling at positive edge 1 Enable sampling at positive edge
11, 10	00 Trigger with touch probe 2 input 01 Trigger with Encoder Index	3, 2	00 Trigger with touch probe 1 input 01 Trigger with Encoder Index
9	0 Trigger first event	1	0 Trigger first event
8	0 Switch off touch probe 2 1 Enable touch probe 2	0	0 Switch off touch probe 1 1 Enable touch probe 1

When the *Trigger with Encoder Index* mode is selected, also the sampling on the positive edge must be selected (*Enable sampling at positive edge*).

8.3.36 $60B9_h$ Touch probe status

Index	Name	Mnemonic
$60B9_h$	*Touch probe status*	

Data Type	Access Type	PDO Mapping	Note
u16	ro	TPDO	
Default Value	**Minimum**	**Maximum**	**Unit**

Description

It indicates the touch probe status.
The bits of the low byte are related to the *Touch probe 1* while the bits in the high byte are related to the *Touch probe 2*.

The following table shows the meaning of each bit:

Touch probe 2		*Touch probe 1*	
Bit	Description	Bit	Description
15, 14	Reserved bit. Ignore the value	7, 6	Reserved bit. Ignore the value
13..11	0 Reserved	5..3	0 Reserved
10	0 Touch probe 2 no negative edge value stored 1 Touch probe 2 negative edge position stored	2	0 Touch probe 1 no negative edge value stored 1 Touch probe 1 negative edge position stored
9	0 Touch probe 2 no positive edge value stored 1 Touch probe 2 positive edge position stored	1	0 Touch probe 1 no positive edge value stored 1 Touch probe 1 positive edge position stored
8	0 Touch probe 2 is switched off 1 Touch probe 2 is enabled	0	0 Touch probe 1 is switched off 1 Touch probe 1 is enabled

8.3.37 60BA$_h$ Touch probe 1 positive edge

Index	Name			Mnemonic
60BA$_h$	*Touch probe 1 positive edge*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 25000 = 2.5000rev)
	Description			
	It contains the position captured on the positive edge of the touch probe 1. The value is given in 1/10000 per revolution.			

8.3.38 60BB$_h$ Touch probe 1 negative edge

Index	Name			Mnemonic
60BB$_h$	*Touch probe 1* negative *edge*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 25000 = 2.5000rev)
	Description			
	It contains the position captured on the negative edge of the touch probe 1. The value is given in 1/10000 per revolution.			

8.3.39 60BC$_h$ Touch probe 2 positive edge

Index	Name			Mnemonic
60BC$_h$	*Touch probe 2 positive edge*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 25000 = 2.5000rev)
	Description			
	It contains the position captured on the positive edge of the touch probe 2. The value is given in 1/10000 of revolution.			

8.3.40 60BD$_h$ Touch probe 2 negative edge

Index	Name			Mnemonic
60BD$_h$	*Touch probe 2* negative *edge*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				00001rev (Ex. 25000 = 2.5000rev)
	Description			
	It contains the position captured on the negative edge of the touch probe 2. The value is given in 1/10000 of revolution.			

8.3.41 60C2$_h$ *Interpolation Time Period*

When using the *Interpolated Position* mode it is very important to configure accurately the objects of this record so that they correspond to the period with which the set-point is updated (Sync period).

The objects can be modified only if the interpolation is not active (bit *ipa* in the *Statusword* set to 0).

Index	Name	Type
60C2$_h$	*Interpolation_time_period*	RECORD

Sub-Index	Name			Mnemonic
00$_h$	*Number_of_Entries*			
	Data Type	Access Type	PDO Mapping	Note
	u8	const		
	Default Value	Minimum	Maximum	Unit
	02$_h$			
	Description			
	Number of record entries.			

Sub-Index	Name			Mnemonic
01$_h$	*Interpolation_time_period_value*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	1			Defined by the object *Interpolation_time_index*
	Description			

The object allows you to set the interpolation period, i.e. the time between two successive updates of the set-point.

For best operation it is important that the object value corresponds exactly to the period with which the SYNC object is received.

Please note how the measurement unit is not predefined but depends on the content of the *Interpolation_time_index* object. The period in seconds can be calculated as:

$T = Interpolation_time_period_value * 10^{Interpolation_time_index}$

If, for example, the objects are configured *Interpolation_time_period_value* = 2 and *Interpolation_time_index* = -3, the interpolation period is set to 2ms ($2 * 10^{-3}$).

Sub-Index	Name			Mnemonic
02$_h$	*Interpolation_time_index*			
	Data Type	Access Type	PDO Mapping	Note
	i8	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	-3	-128	63	
	Description			
	The object defines the measurement unit used for the *Interpolation_time_period_value* object and represents the exponent in base 10 that multiplies the *Interpolation_time_period_value* object.			

8.3.42 60F4h Following error actual value

Index	Name			Mnemonic
60F4h	*Following_error_actual_value*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This object provides the actual value of the following error. The value is given in 1/10000 of revolution.			

8.3.43 60F8h Max slippage

Index	Name			Mnemonic
60F8h	*Max_slippage*			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	1000	0	30000	0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This object indicates the configured maximal slippage. When the max slippage has been reached, the corresponding bit 13 max slippage error in the *Statusword* is set to 1. The slippage control is active in the *Profile velocity* and *Cyclic synchronous velocity* modes. The value is given in 0.1rpm.			

8.3.44 60FDh Digital inputs

Index	Name			Mnemonic
60FDh	*Digital inputs*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			

It indicates the status of the device logic inputs.

The objects bits are related to the digital inputs as follows:

Bit	Description
0	Negative limit switch
1	Positive limit switch
2	Home switch
15..3	Reserved, ignore the value
31..16	Not used, ignore the value

The bit value = 1 indicates active input, on the contrary if the bit has value 0 it means the logic status of the input is inactive.

8.3.45 60FF$_h$ Target velocity

Index	Name			Mnemonic
60FF$_h$	*Target_velocity*			
	Data Type	Access Type	PDO Mapping	Note
	i32	rw	RPDO	💾
	Default Value	Minimum	Maximum	Unit
	0			0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This object indicates the configured target velocity and shall be used as input for the trajectory generator. The value is given in 0.1rpm.			

8.3.46 6502$_h$ Supported drive modes

Index	Name			Mnemonic
6502$_h$	*Supported_drive_modes*			
	Data Type	Access Type	PDO Mapping	Note
	u32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
	Description			

This object provides information about the supported motor modes.

It is made up in bits and each bit is associated with a specific mode. If the mode is supported the bit is 1, otherwise the bit is 0 if the drive mode is not supported by the device.

The following table shows the correspondence between the bits and the function.

Bit	Description	
31	r	Reserved
…	r	Reserved
16	r	Reserved

Bit	Description		Bit	Description	
15	r	Reserved	7	csp	Cyclic synchronous profile mode
14	r	Reserved	6	ip	Interpolated position mode
13	r	Reserved	5	hm	Homing mode
12	r	Reserved	4	r	Reserved
11	r	Reserved	3	tq	Torque mode
10	cstca	Cyclic synchronous torque mode with commutation angle	2	pv	Profile velocity mode
9	cst	Cyclic synchronous torque mode	1	vl	Velocity mode
8	csv	Cyclic synchronous velocity mode	0	pp	Profile position mode

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10 cstca	9 cst	8 csv	7 csp	6 ip	5 hm	4	3 tq	2 pv	1 vl	0 pp

8.3.47 67FE$_h$ Version number

Index	Name			Mnemonic
67FE$_h$	*Version_number*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw	RPDO	
	Default Value	Minimum	Maximum	Unit

Description

The object supplies the version of the CiA 402 profile implemented in the integrated drive.

bits 31..24	bits 23..16	bits 15..8	bits 7..0
Reserved	Main version	Secondary version	Sub version

LAM Technologies
Viale Ludovico Ariosto, 492/D
50019 Sesto Fiorentino
Firenze - ITALY

Ph. +39 055 4207746

www.lamtechnologies.com

General information info@lamtechnologies.com
Technical support support@lamtechnologies.com
Sales information sales@lamtechnologies.com

rev. 0.12 (preliminary)