LAM Technologies
electronic equipment

DMS7xP

Integrated Stepper Motors

with PROFINET fieldbus

PROFINET

User Manual

(Hardware rev. 1.00 Firmware rev. 1.03)

Contents

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 DMS7xPx2 (NEMA23 flange) 18
3.1.2 DMS7xPx3 (NEMA34 flange) 19
3.2 Connection 21
3.2.1 NEMA23, DMS7xPx2 series 21
3.2.2 NEMA34, DMS7xPx3 series 22
3.2.3 CN1 – Power Supply: AC models 23
3.2.4 CN1 – Power Supply, DC models 28
..... 28
3.2.5 CN2 - Digital and analog I/O 34
3.2.6 CN7 – Bus PROFINET 40
4 Configuration 41
4.1 Acyclic access to the registers 41
4.2 Cyclic access to the registers 41
5 Omni Automation IDE Software 44
5.1 Firmware update 45
5.2 Status and Diagnostics 46
5.3 Visualization and modification of register 48
6 Operation 51
6.1 Minimum settings 51
6.1.1 Running and idle current configuration 51
6.2 Saving and restoring of default values 53

6.3 Motor Holding Brakes 54
6.4 Closed-loop 55
6.5 Use and functionality of the StatusDWord and ControlDWord registers 56
6.5.1 Motor Enabling 58
6.5.2 Emergency Stop (QuickStop) 58
6.5.3 Motor Stop (Halt) 58
6.5.4 Fault e Warning 59
6.5.5 Real-time or delayed target process 59
6.5.6 Absolute or relative positioning 60
6.5.7 Target achieved and motor halted 60
6.5.8 Following error 60
6.5.9 Operating mode 61
6.6 Operating Modes 62
6.6.1 Position (pp) 62
6.6.2 Velocity (pv) 63
6.6.3 Torque (tq) 64
6.6.4 Homing (hm) 65
7 Errors and diagnostics 70
7.1 LED *On* 70
7.2 LED *En* 70
7.3 LED *Fault* 70
7.4 Errors 71
8 Registers 74
8.1 Data process mapping 74
8.2 Store Parameters 78
8.3 Restore Default Parameters 78
8.4 Motor Data 79
8.5 Motor Incremental Encoder Data 82
8.6 SSI Encoder Data 84
8.7 Holding Brake Setup 88
8.8 Current Regulation 90
8.9 Motion Setup 91
8.10 Position Loop 92
8.11 Error Class Setup 93
8.12 Digital Inputs Assignment 94
8.13 Digital Outputs Assignment 98

8.14 Device Status 100
8.15 Digital Inputs Value 100
8.16 Digital Outputs Value 101
8.17 Analog Inputs Value 102
8.18 Analog Outputs Value 103
8.19 Motor Configuration 103
8.20 Motor Value 104
8.21 Motor Incremental Encoder Value 105
8.22 SSI Encoder Values 106
8.23 Holding Brake Control 107
8.24 Torque Profile Configuration 109
8.25 StatusDWord and ControlDWord 110
8.26 Errors 111
8.27 Actual Motion Values 112
8.28 Target Motion Values 113
8.29 Limits 115
8.30 Validity and Error Window 116
8.31 Homing 120
8.32 Touch Probe Functions 126

1 Introduction

The DMS7xP motors with integrated drives series is equipped with PROFINET 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 motor control thus eliminating stall problems and improving efficiency.

The free software Omni Automation IDE, running under Windows (Windows 7, Windows 8.1, Windows 10 and Windows 11 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	**DMS71P4241**	**DMS72P4241**	**NEMA23** DC Power Supply
20..50Vdc / 1.8Nm	**DMS71P4264**	**DMS72P4264**	
20..50Vdc / 3.0Nm	**DMS71P4271**	**DMS72P4271**	
24..90Vdc / 1.1Nm	**DMS71P7241**	**DMS72P7241**	
24..90Vdc / 1.8Nm	**DMS71P7264**	**DMS72P7264**	
24..90Vdc / 3.0Nm	**DMS71P7271**	**DMS72P7271**	
20..50Vdc / 4.4Nm	**DMS71P4321**	**DMS72P4321**	**NEMA34** DC Power Supply
20..50Vdc / 6.8Nm	**DMS71P4331**	**DMS72P4331**	
20..50Vdc / 9.2Nm	**DMS71P4350**	**DMS72P4350**	
20..50Vdc / 12.1Nm	**DMS71P4360**	**DMS72P4360**	
24..90Vdc / 4.4Nm	**DMS71P7321**	**DMS72P7321**	
24..90Vdc / 6.8Nm	**DMS71P7331**	**DMS72P7331**	
24..90Vdc / 9.2Nm	**DMS71P7350**	**DMS72P7350**	
24..90Vdc / 12.1Nm	**DMS71P7360**	**DMS72P7360**	
16..36Vac / 4.4Nm	**DMS71P4321A**	**DMS72P4321A**	**NEMA34** AC Power Supply
16..36Vac / 6.8Nm	**DMS71P4331A**	**DMS72P4331A**	
16..36Vac / 9.2Nm	**DMS71P4350A**	**DMS72P4350A**	
16..36Vac / 12.1Nm	**DMS71P4360A**	**DMS72P4360A**	
20..65Vac / 4.4Nm	**DMS71P7321A**	**DMS72P7321A**	
20..65Vac / 6.8Nm	**DMS71P7331A**	**DMS72P7331A**	
20..65Vac / 9.2Nm	**DMS71P7350A**	**DMS72P7350A**	
20..65Vac / 12.1Nm	**DMS71P7360A**	**DMS72P7360A**	

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 DMS7xP4350 implies the models DMS71P4350 and DMS72P4350.

The terms manual and document have the same meaning, moreover the words *integrated drive motor, integrated motor, motor, device* and *product* always refer to the DMS7xP 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:

Abbreviation	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 DMS7xP 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 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 of the 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.

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 disagrees 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 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.

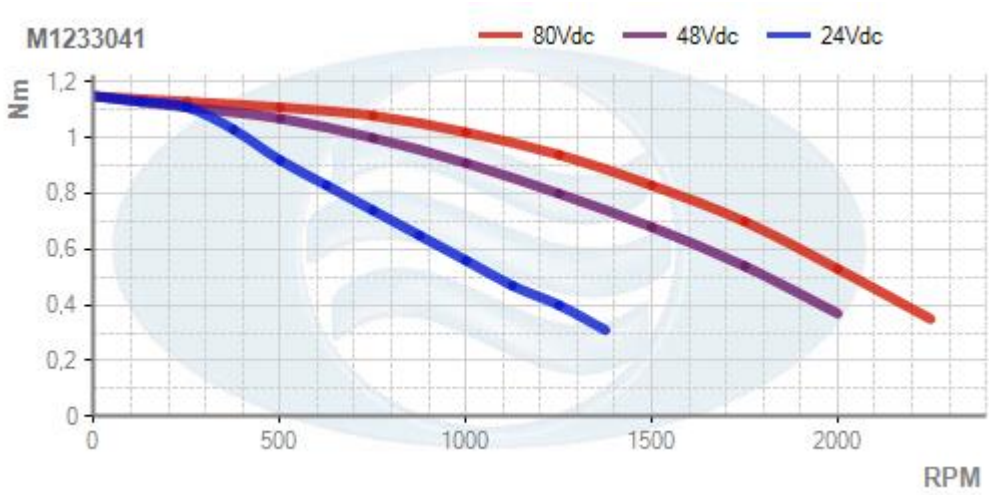
For each motor with integrated electronics, more different torque curves are shown, each one corresponding to a different power supply voltage. The voltage values are expressed in continuous (Vdc), for the devices with AC power supply (Vac) the following formula must be applied to identify the corresponding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

Vdc = (Vac * 1.41) – 2.5

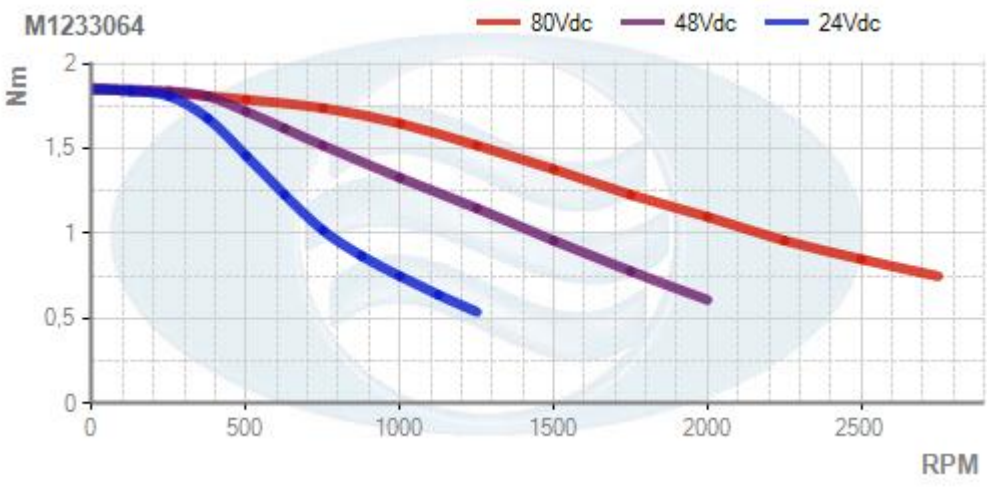
For example. if the AC power supply is 36Vac, we will use the torque curve that is closest to (36 * 1.41) -2.5 = 48.3, which is the curve corresponding to a power supply of 48Vdc.

2.1 NEMA23

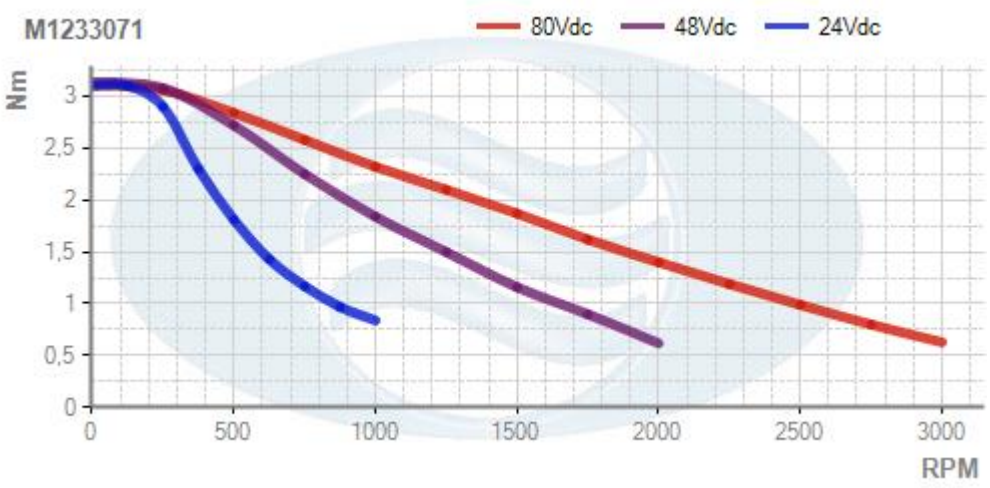
DMS7xPx241



DMS7xPx264

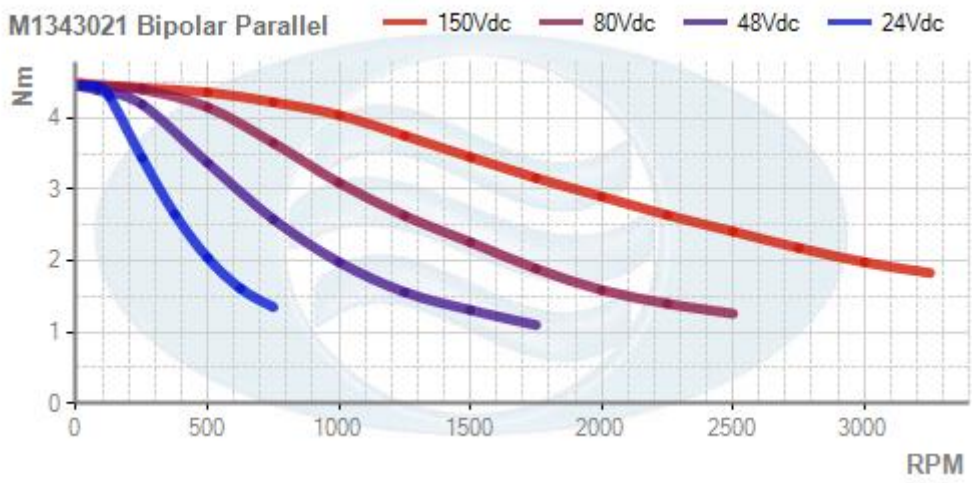


DMS7xPx271

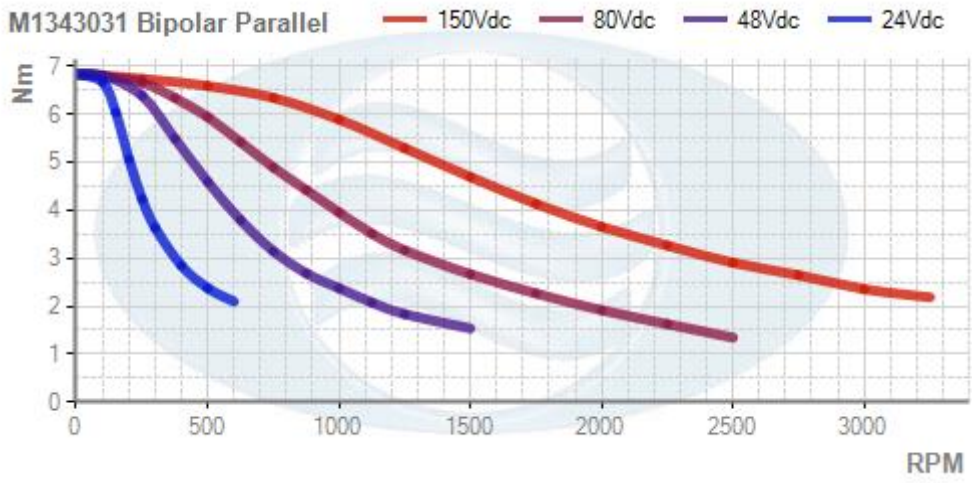


2.2 NEMA34

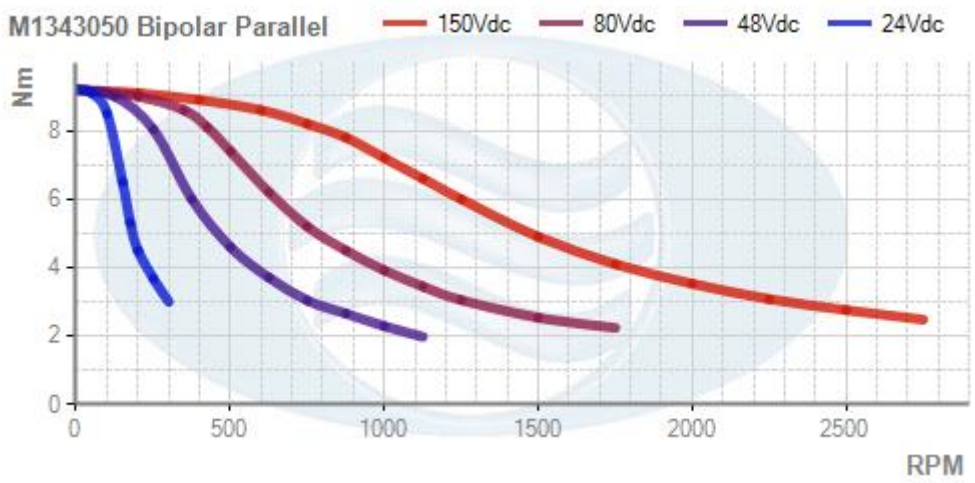
DMS7xPx321x



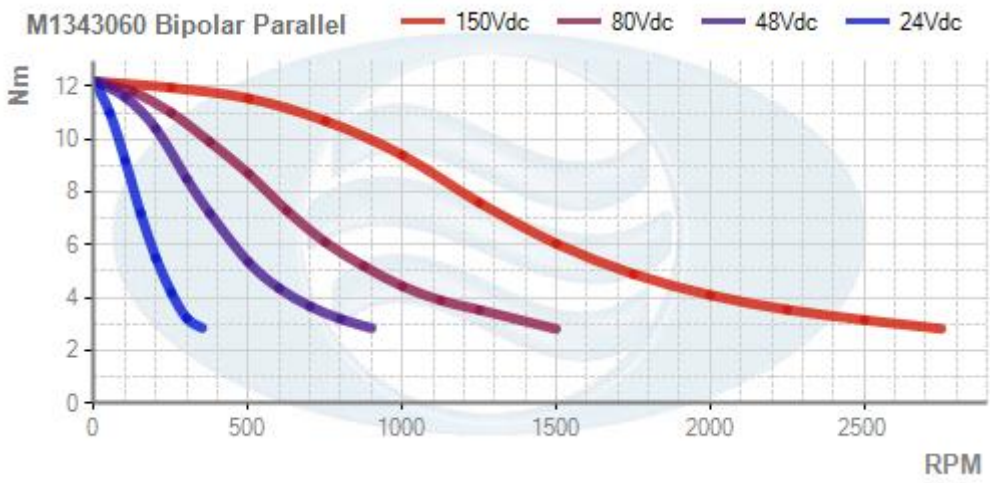
DMS7xPx331x



DMS7xPx350x



DMS7xPx360x



3 Installation

ATTENTION, the DMS7xP 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

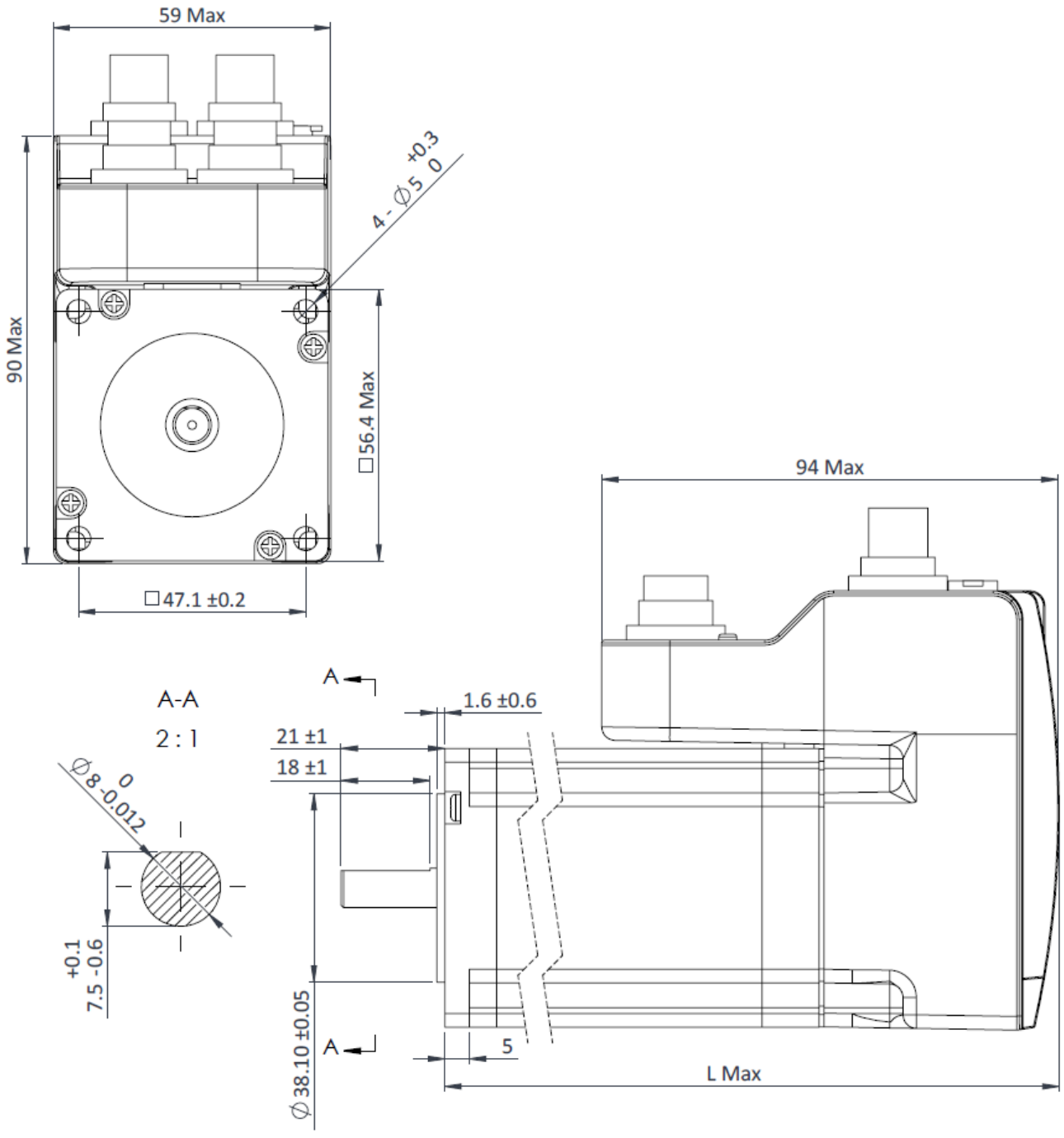
The protection degree is IP60 with the exception of the motor shaft which has an IP40 protection degree.

3.1.1 DMS7xPx2 (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
DMS7xPx241	106mm	1000g
DMS7xPx264	126mm	1400g
DMS7xPx271	150mm	1850g

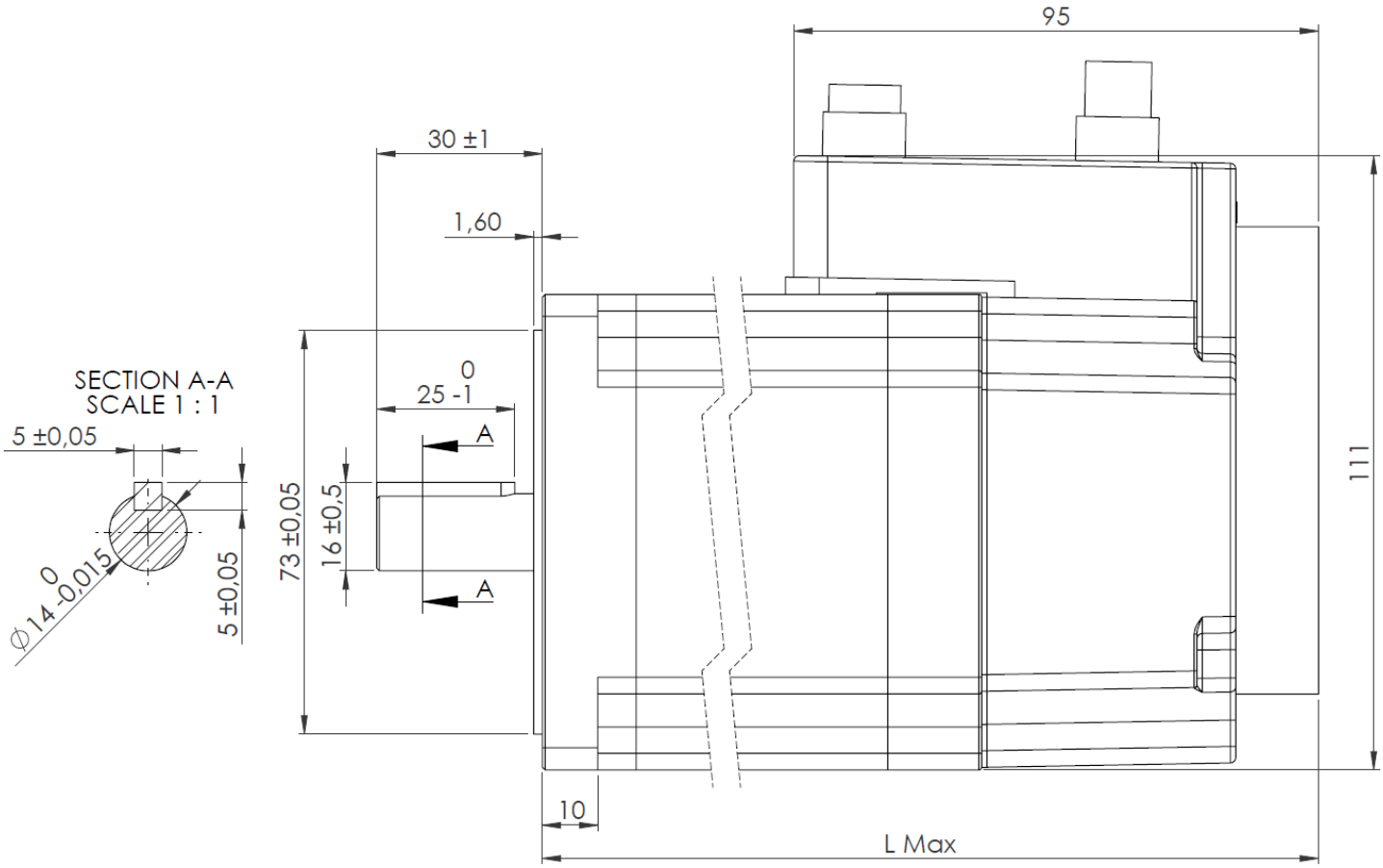


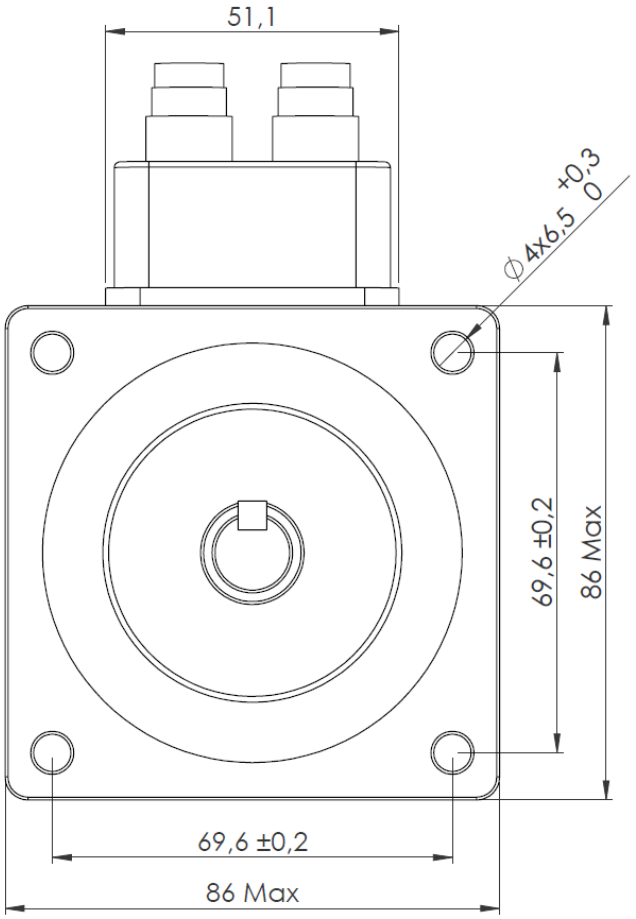
3.1.2 DMS7xPx3 (NEMA34 flange)

All the models of motors with integrated drive with NEMA34 flange share the same dimensions with the sole exception of the length *L Max* 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:

Modello	L Max	Weight
DMS7xPx321x	141mm	2700g
DMS7xPx331x	161mm	3400g
DMS7xPx350x	189mm	4500g
DMS7xPx360x	213mm	5400g



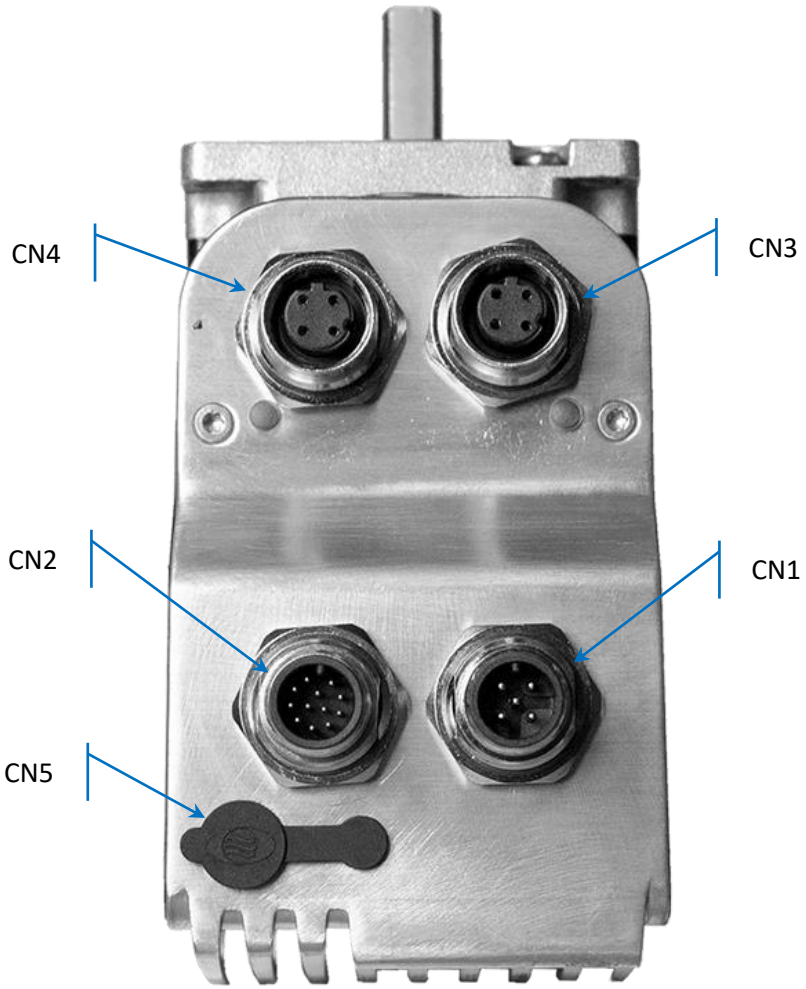

51,1
Ø4x6,5 +0,3 0
69,6 ±0,2
86 Max
69,6 ±0,2
86 Max

3.2 Connection

The DMS7xP series of motors with integrated drive is provided with 4 M12 connectors reserved to the power supply, the connection of the digital and analogic I/Os and of the PROFINET fieldbus.

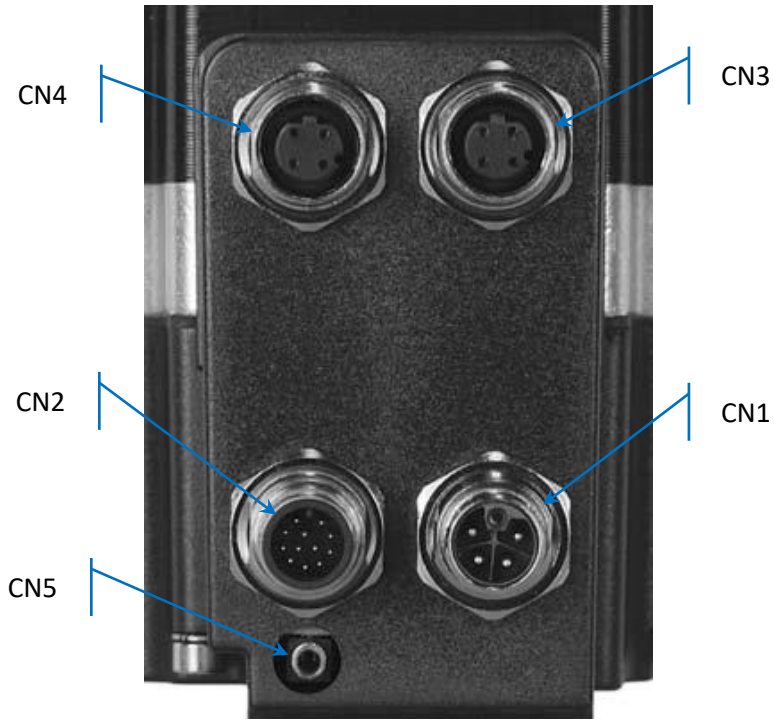
Connector	Type		Function
	DMS7xPx2 (NEMA23 flange)	DMS7xPx3 (NEMA34 flange)	
CN1	M12 Coding A Male 5ct	M12 Coding L Male 5ct	Power supply
CN2	M12 Coding A Male 12ct	M12 Coding A Male 12ct	Digital and analog I/O
CN3	M12 Coding D Female 4ct	M12 Coding D Female 4ct	PROFINET Out
CN4	M12 Coding D Female 4ct	M12 Coding D Female 4ct	PROFINET In
CN5			DUP port for the connection to a UDP30

3.2.1 NEMA23, DMS7xPx2 series



The CN5 connector is equipped with a protective rubber cap.

3.2.2 NEMA34, DMS7xPx3 series



The CN5 connector is equipped with a protective rubber cap.

3.2.3 CN1 – Power Supply: AC models

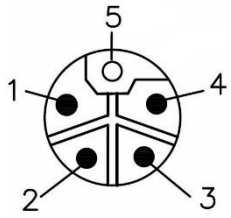
The DMS7xPx3 series integrated motors with NEMA34 flange are also available in the AC supply version and are identified by the letter **A** placed at the end of the code (ex. DMS72P7331**A**). They integrate a rectifier bridge and the capacitors necessary to rectify and filter the AC power supply voltage.

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 to be kept 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 (per DMS72P 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
DMS7xP4xxxA	Vac	AC supply voltage	Vac	16	32	36
	Vacbrk	Voltage causing permanent damage	Vac			42
DMS7xP7xxxA	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
DMS7xP4xxxA	Vacl	Under voltage protection intervention threshold	Vac		15	
	Vach	Over voltage protection intervention threshold	Vac		38	
DMS7xP7xxxA	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 drive:

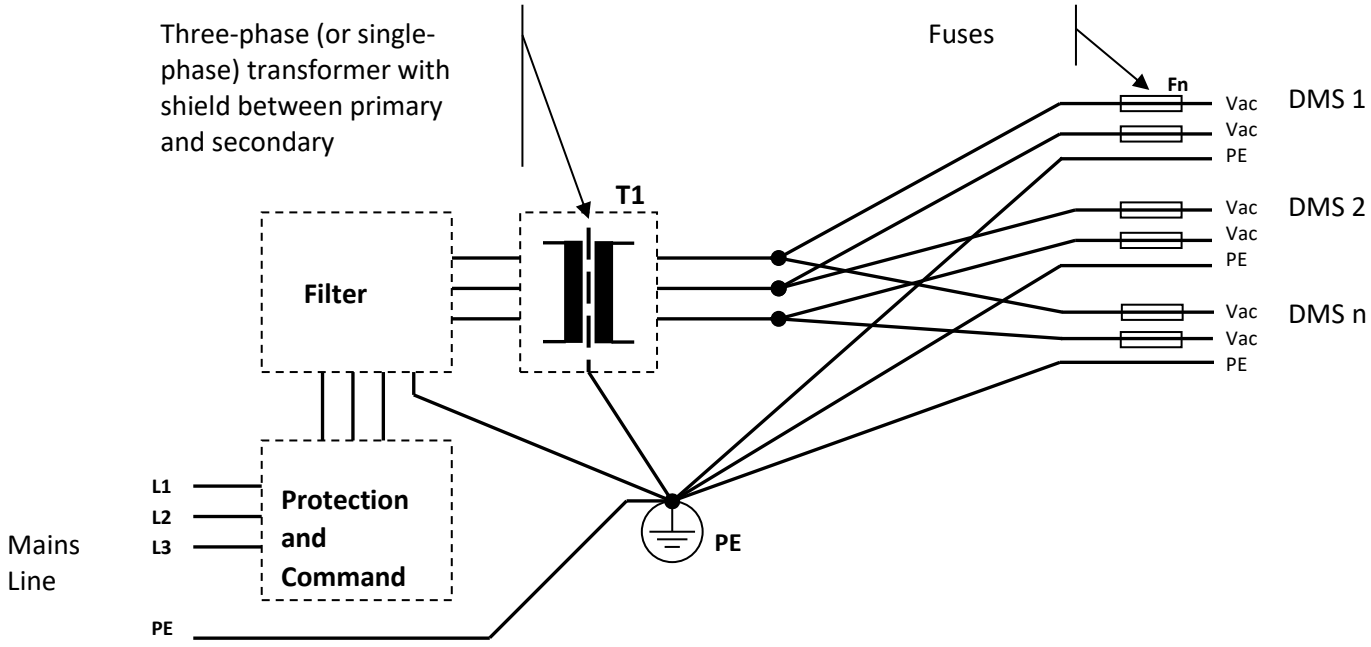
Model	Section (mm^2)
DMS7xP4321A	1
DMS7xP4331A	1.5
DMS7xP4350A	1.5
DMS7xP4360A	1.5
DMS7xP7321A	1
DMS7xP7331A	1.5
DMS7xP7350A	1.5
DMS7xP7360A	1.5

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 rit.)	Number of motors	Secondary T1 (Vac)	Power T1 (VA)	Current D1 (Arms)
DMS7xP4321A	8	1	32	200	25A
		2		400	25A
		3		600	25A
		4..5		1000	35A
		6..8		1400	50A
DMS7xP4331A	8	1	32	300	25A
		2		600	25A
		3		900	35A
		4..5		1500	50A
		6..8		2100	80A
DMS7xP4350A	8	1	32	300	25A
		2		600	25A
		3		900	35A
		4..5		1500	50A
		6..8		2100	80A
DMS7xP4360A	12	1	32	300	25A
		2		600	25A
		3		900	35A
		4..5		1500	50A
		6..8		2100	80A

DMS7xP7321A	8	1	48	400	25A
		2		800	25A
		3		1200	35A
		4..5		2000	50A
		6..8		2800	80A
DMS7xP7331A	8	1	48	500	25A
		2		1000	35A
		3		1500	35A
		4..5		2500	50A
		6..8		3500	80A
DMS7xP7350A	8	1	48	500	25A
		2		1000	35A
		3		1500	35A
		4..5		2500	50A
		6..8		3500	80A
DMS7xP7360A	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.

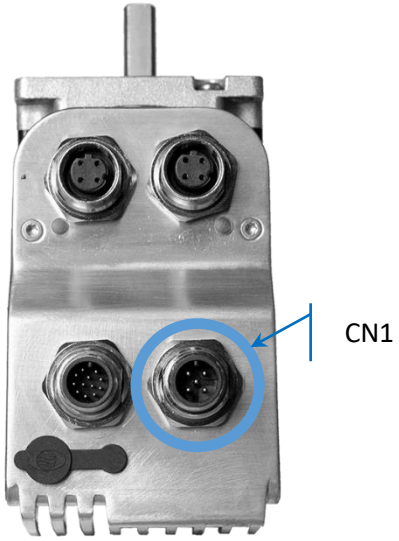
The set composed by the filter and the transformer must be used only to supply the power voltage to the drives. It is recommended not to derive other supplies from any of these parts. On the contrary, it is suggested to obtain auxiliary supplies using directly the mains line upstream of the filter.

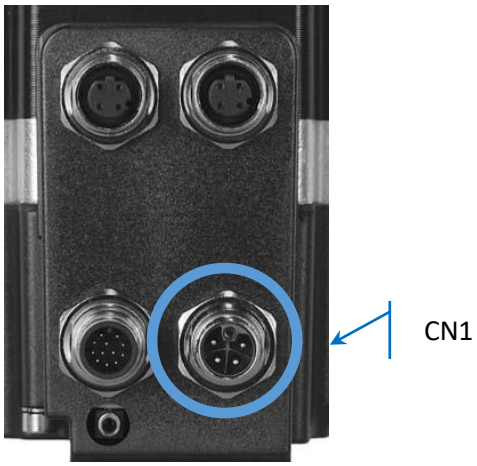
3.2.4 CN1 – Power Supply, DC models

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 with integrated drive before the wiring is complete.

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





CN1 – Power supply			
Pin	Description	DMS7xPx2	DMS7xPx3
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 with integrated drive.

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 (per DMS72P 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

3.2.4.2 *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
DMS7xP4xxx	Vp	DC supply voltage	V	20		50
	Vpbrk	Voltage causing permanent damage	V	-0.5		60
DMS7xP7xxx	Vp	DC supply voltage	V	24		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
DMS7xP4xxx	Vpl	Under voltage protection intervention threshold	V		18	
	Vph	Over voltage protection intervention threshold	V		52	
DMS7xP7xxx	Vpl	Under voltage protection intervention threshold	V		22	
	Vph	Over voltage protection intervention threshold	V		96	

If the distance between the motor with integrated drive 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)
DMS7xP4241 DMS7xP4264 DMS7xP4271 DMS7xP4321 DMS7xP4331 DMS7xP4350	63	1000
DMS7xP4360	63	2200
DMS7xP7241 DMS7xP7264 DMS7xP7271 DMS7xP7321 DMS7xP7331 DMS7xP7350	100	1000
DMS7xP7360	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	Section (mm^2)
DMS7xPx241	0.5
DMS7xPx264	0.5
DMS7xPx271	0.5
DMS7xPx321	1
DMS7xPx331	1.5
DMS7xPx350	1.5
DMS7xPx360	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.

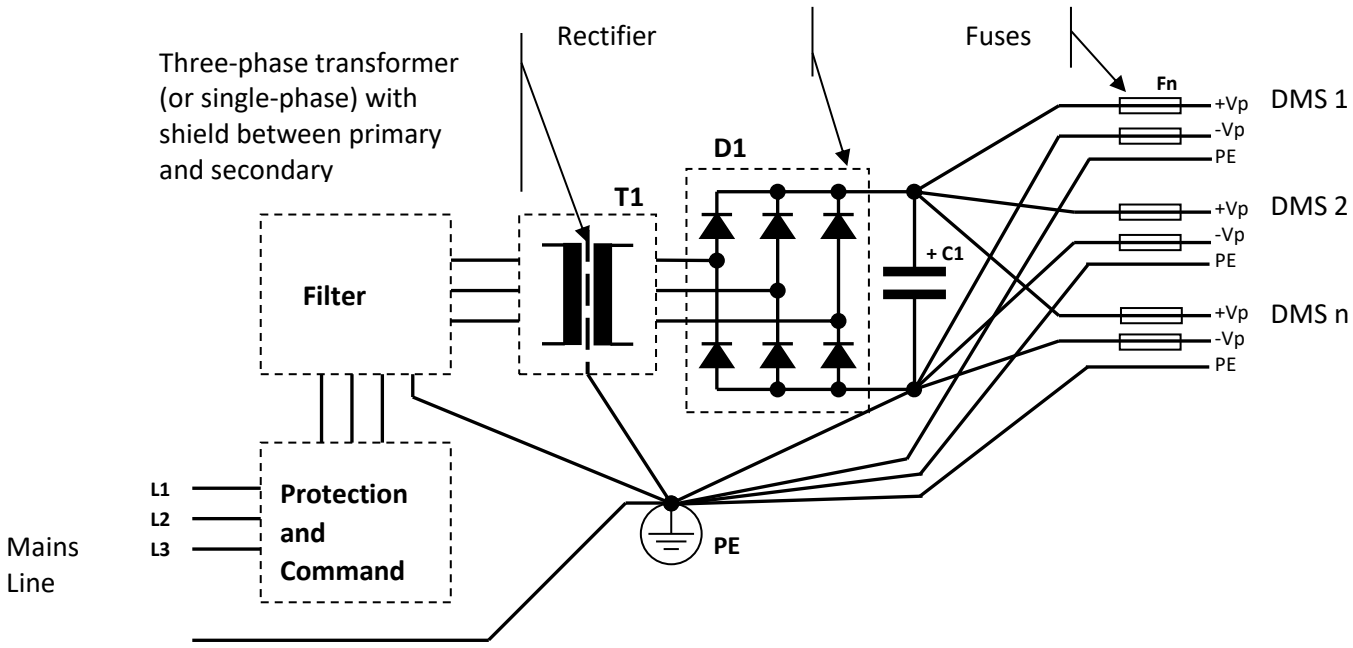
An unregulated power supply is cheaper but it forces to consider a safety's tolerance in sizing so that, in case of mains line and load fluctuations, voltage remains however within the allowed operating limits.

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 calculation considers 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)
DMS7xP4241 DMS7xP4264 DMS7xP4271 DMS7xP4321	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
DMS7xP4331	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
DMS7xP4350	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
DMS7xP4360	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
DMS7xP7241 DMS7xP7264 DMS7xP7271 DMS7xP7321	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
DMS7xP7331	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
DMS7xP7350	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
DMS7xP7360	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.

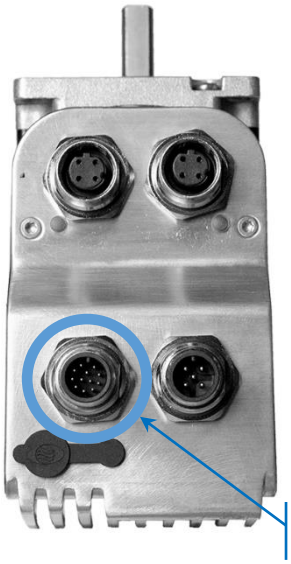
In the configurations with more than one motor with integrated drive, if the drives are not all set to the maximum current and/or if the working cycle is not simultaneous, the power of the transformer can be considerably reduced. In some cases this can also be done when the motors' speed is limited.

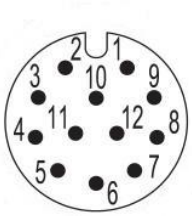
The diagram and the components' values refer to a three-phase power supply. Dimensioning the components in a different way, it is also possible to realize a single-phase power supply, which is not recommended when the required power is greater than 800W.

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.



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 3) / DO1 (Digital Output 1)	
8	DI5 (Digital input 3) / 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)	

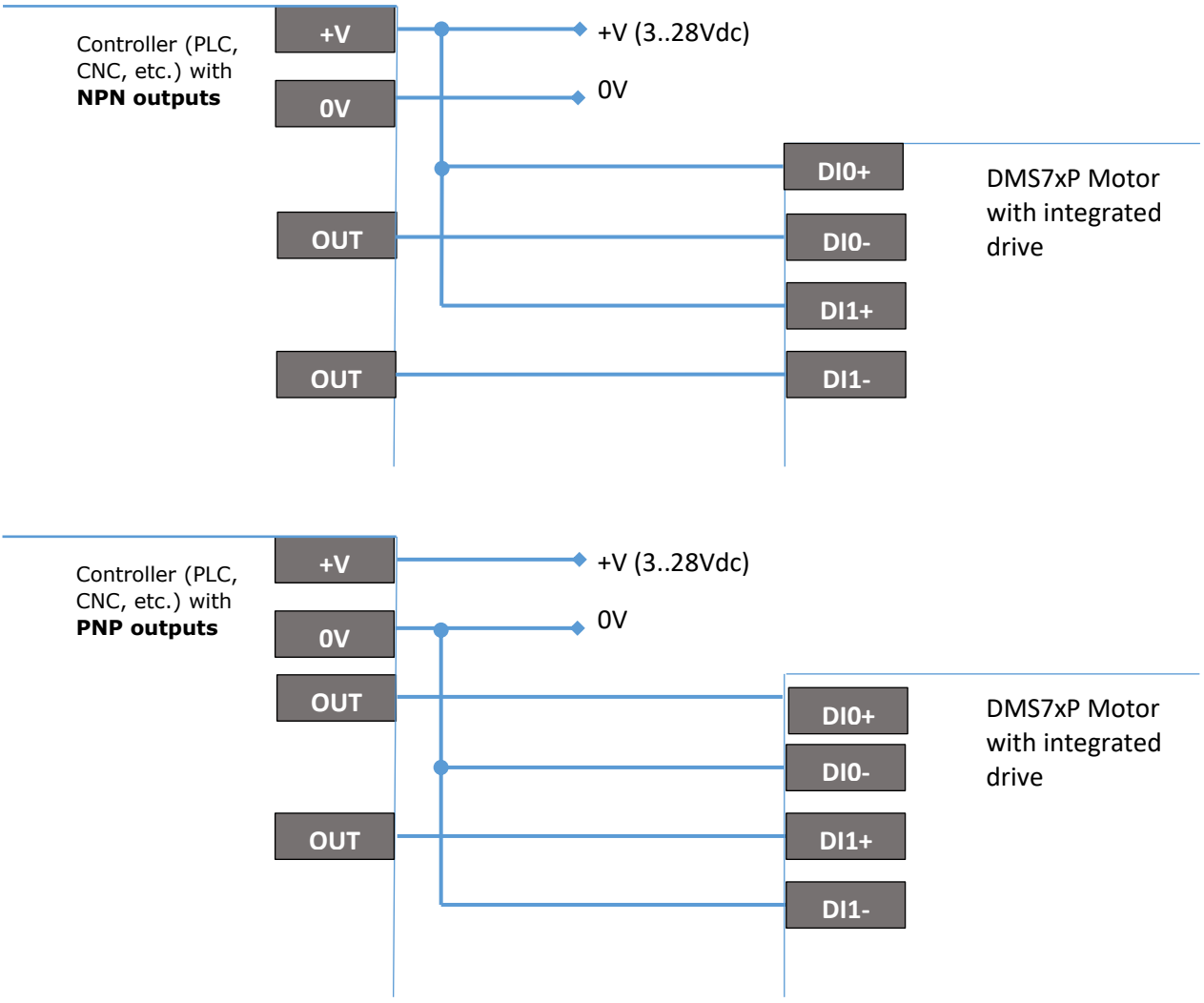
3.2.5.1 Digital Inputs

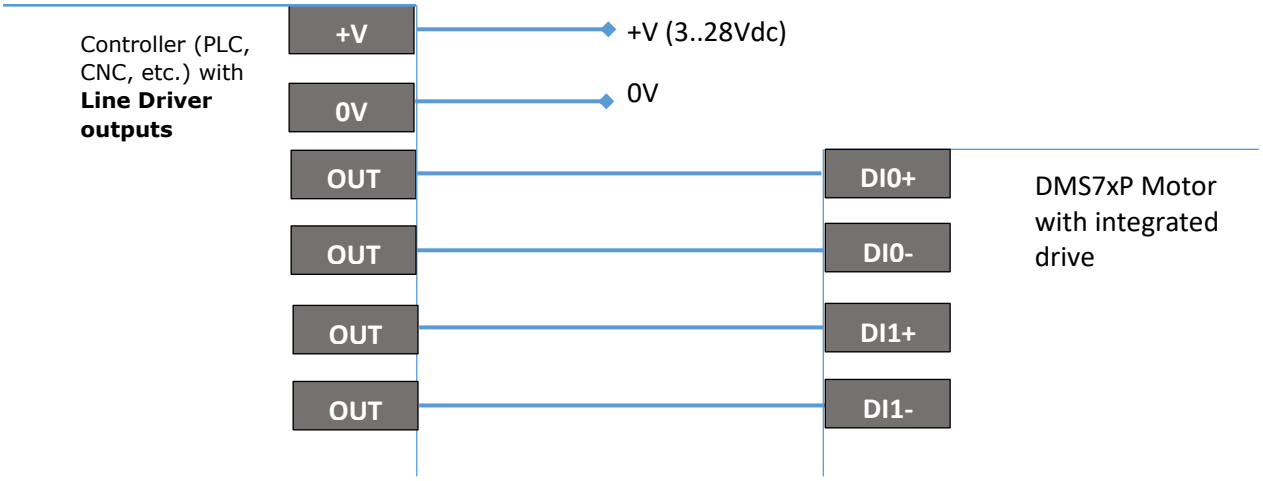
The DI0 and DI1 digital inputs are optocoupled and have a current regulation circuit which allows to apply voltages between 3Vdc and 28Vdc without the need to insert any current limit resistor. The current in fact remains constant independently from the voltage applied to the input. This simplifies the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



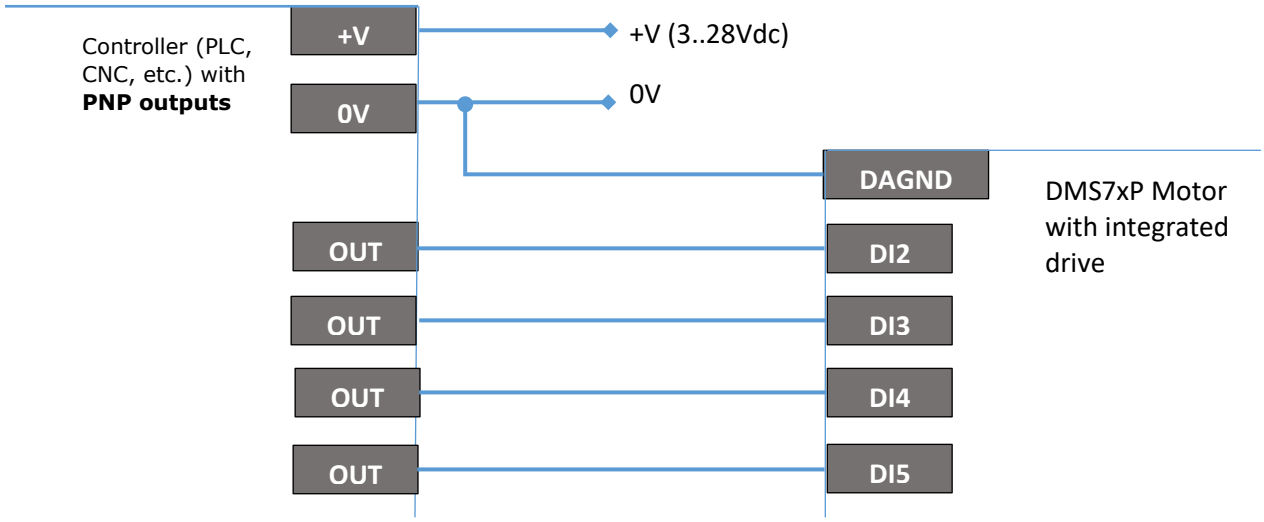


The following table shows the voltage values which correspond to the Active and Inactive status of the inputs DI0 and DI1, together with other parameters:

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..5 (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 register, 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.

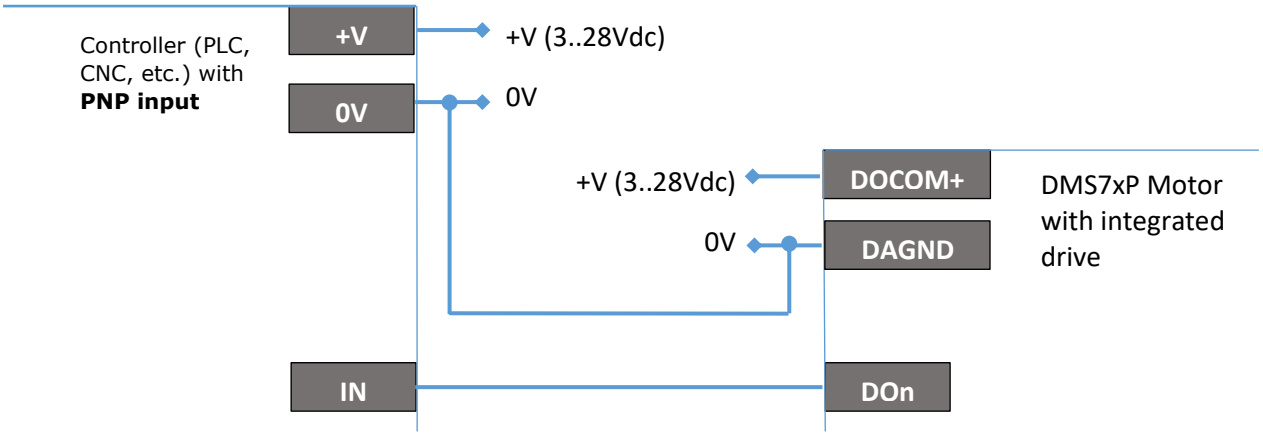
All the outputs are PNP type and are powered through the common DOCOM+ signal. Therefore, the output voltage depends on the power supply voltage applied between DOCOM+ and DAGND and this allows to freely interface systems with 5V, 12V or 24V inputs. Keep in mind that the DAGND signal is connected internally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



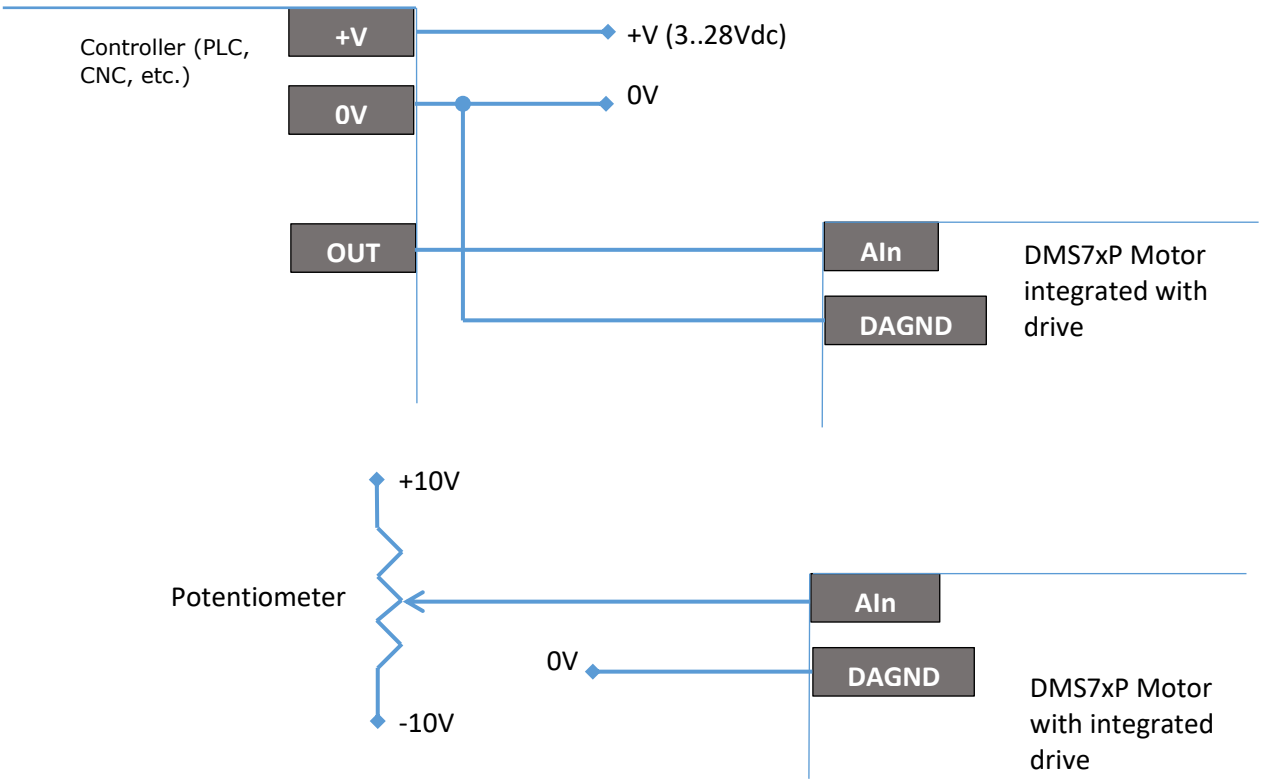
The digital outputs share the pins with the digital inputs DI3, DI4 and DI5 on the CN2 connector. The latter are always active while the outputs to be used must first be activated through the Direction_DIOSTP register.

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:

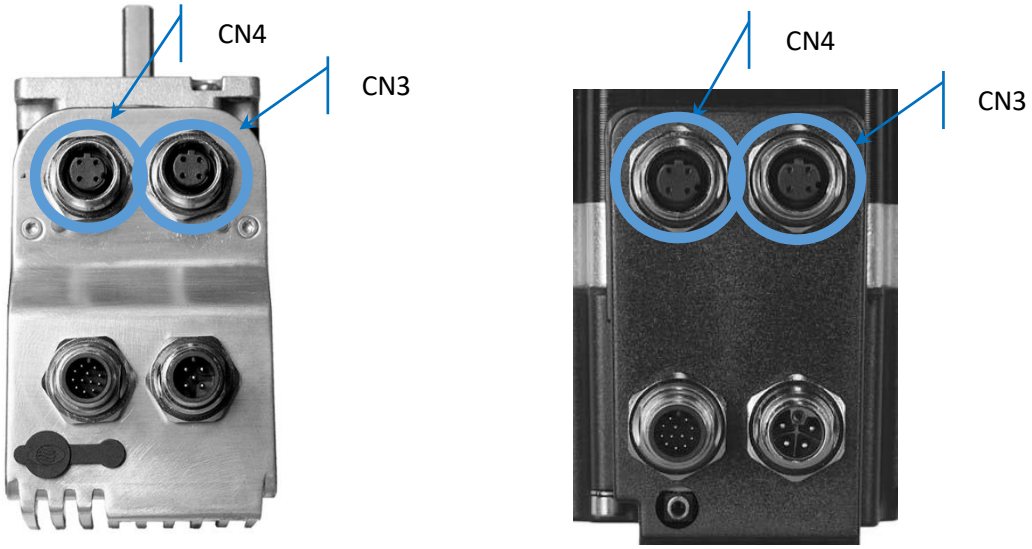
Symbol	Description	Unit	Value		
			Min.	Typ	Max
Vai	Analog input operating voltage	Vdc	-10.2		+10.2
Vaibrk	Analog input breakdown voltage	Vdc	-45		+45
Rai	Analog inputs impedance	KΩ		47	
ADst	A/D converter conversion time	ms		1	
ADsoff	A/D converter start offset	%fs		1	
ADdoff	A/D converter offset drift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

3.2.6 CN7 – Bus PROFINET

The CN3 and CN4 connectors are used for connection to the PROFINET 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 PROFINET device.



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

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

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

4

4 Configuration

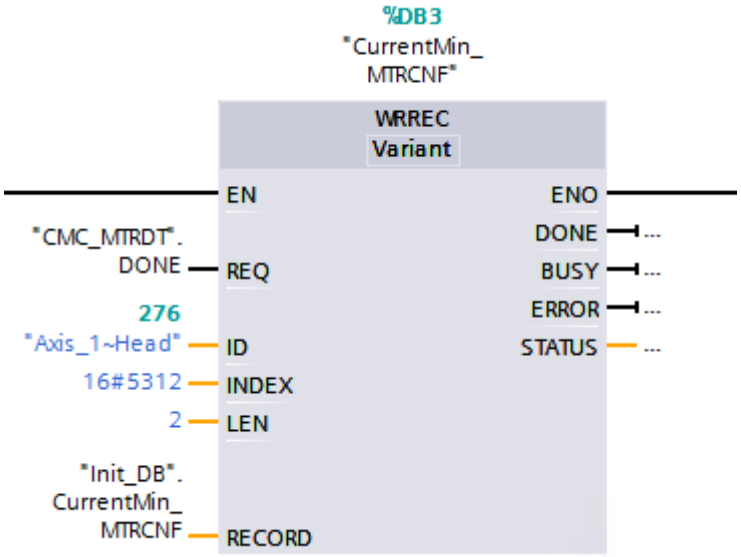
The device name as well as the IP address can be assigned via the PROFINET IO bus using for example the free Siemens software PRONETA (Commissioning and Diagnostics Tool for PROFINET).

4.1 Acyclic access to the registers

The registers of the motor with integrated drive can be initialized with the values required by the application by using the acyclic communication and the record writing and reading functions provided by the PROFINET protocol.

In the Step 7 – TIA Portal environment, are available the RDREC and WRREC function blocks which allow to respectively read and write the registers of the motor with integrated drive.

The following image shows an example of how to use the WRREC function block to write the 5312_h *CurrentMin_MTRCNF* register.



4.2 Cyclic access to the registers

The area reserved for the process data is cyclically exchanged between the PLC and the device.

The motor with integrated drive has 32 bytes in output and 32 bytes in input reserved for the cyclic data. In the firmware revision described in this manual the registers are mapped to the process data as follows:

Process data					
OUT (out of the PLC)			IN (in input to the PLC)		
Offset	Name	Dimension (byte)	Offset	Name	Dimension (byte)
0	*ControlDWord*	4	0	*StatusDWord*	4
4	*TargetTorque*	2	4	*ActualTorque*	2
6	*TargetVelocity*	4	6	*ActualVelocity*	4
10	*Acceleration*	4	10	*ActualPosition*	4
14	*Deceleration*	4	14	*ActualPositionError*	*4*
18	*TargetPosition*	4	18	*Inputs_DIV*	*2*

The registers mapping can be modified to adapt the process data to the needs of the application. It is possible to map up to 8 registers in reception (RPD, transmitted from the PLC to the drive) and up to 8 registers in transmission (TPD, transmitted from the drive to the PLC).

The registers responsible for the mapping in reception are *NumberOfMappedObjects_RPDM* and *MappedObject_n_RPDM* (with n between 1 and 8) while the mapping in transmission is defined by the registers *NumberOfMappedObjects_TPDM* and *MappedObject_n_TPDM* (with n between 1 and 8).

The registers *NumberOfMappedObjects_RPDM* and *NumberOfMappedObjects_TPDM* define the number of drive registers inserted into the process data in reception and transmission respectively and can be only modified with the motor disabled. The value 0 indicates that there are no process data mapped in the registers and enables the writing into the registers *MappedObject_n_RPDM* and *MappedObject_n_TPDM* respectively.

When the register *NumberOfMappedObjects_RPDM* is set to a value less than 8, the drive ignores the process data transmitted by the PLC not mapped into the drive registers (eventual registers excluded from the mapping and previously mapped continue to maintain the last value received.

When the register *NumberOfMappedObjects_TPDM* is set to a value less than 8, the process data received by the PLC not mapped in the drive registers are set to 0.

The registers *MappedObject_n_RPDM* and *MappedObject_n_TPDM* are used to map the drive registers in the process data. They have a dimension of 32bit logically divided in 2 high bytes and a low byte and used to contain respectively the address of the register you wish to map and the number of bits of which the register is composed.

The procedure to follow to map the drive register in the process data, exchanged cyclically with the PCS, is the following:

1) Set to 0 the register *NumberOfMappedObjects_RPDM* (or *NumberOfMappedObjects_TPDM*).
2) Compile the registers consecutively from *MappedObject_1_RPDM* (or *MappedObject_1_TPDM*) forward with the address and the dimension of each register you wish to include in the process data.
3) Set the register *NumberOfMappedObjects_RPDM* (or *NumberOfMappedObjects_TPDM*) with the number of registers *MappedObject_n_RPDM* (or *MappedObject_n_TPDM*) which have actually been compiled. For example, if the first 3 registers have been compiled, set the register *NumberOfMappedObjects_RPDM* (or *NumberOfMappedObjects_TPDM*) to 0.

Below is an example showing how to map the registers *ActualVelocity* and *Voltage_AIV_0* in the process data transmitted from the drive and received by the PLC.

1) Set to 0 the register *NumberOfMappedObjects_TPDM*.
2) Set the register *MappedObject_1_TPDM* with the value 61120020_h (*ActualVelocity* dimension 32bit)
3) Set the register *MappedObject_2_TPDM* with the value 52510010_h (*Voltage_AIV_0* dimension 16bit)
4) Set to 2 the register *NumberOfMappedObjects_TPDM*.

Note that not all the rive registers can be mapped in the process data. The chapter 8 Registers describes every drive register in detail and by observing the contents of the *PD Mapping* cell of the table prepared for each of them, it is possible to

determine if the register can be mapped between the process data in receipt, transmission or in none of them, as resumed below.

PD Mapping RPD	It indicates that the register can be mapped between the process data in reception (RPD, transmitted by the PLC)
PD Mapping TPD	It indicates that the register can be mapped between the process data in transmission (TPD, received by PLC)
PD Mapping	It indicates that the register cannot be mapped between the process data.

For further details about the mapping registers see chapter 8.1 Data process mappin.

ATTENTION, incorrect settings or unsuitable to the application can cause unexpected movements of the motor, unwanted activation of signals, monitoring functions disabling, etc. Some settings become active at the following restart of the device.

ATTENTION, do not use the device if you do not know or have not understood the settings. After each setting change, test accurately the application in any possible condition of use or error so not to cause damages to people, animals or things or economic loss.

ATTENTION, when the power stage is disabled the motor do not offer resistant torque and therefore cannot control the load that is thus free to move or to keep an uncontrolled movement. The power stage can disable itself at any moment, for example due to power supply shortage, alarm intervention, etc.

5 Omni Automation IDE Software

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

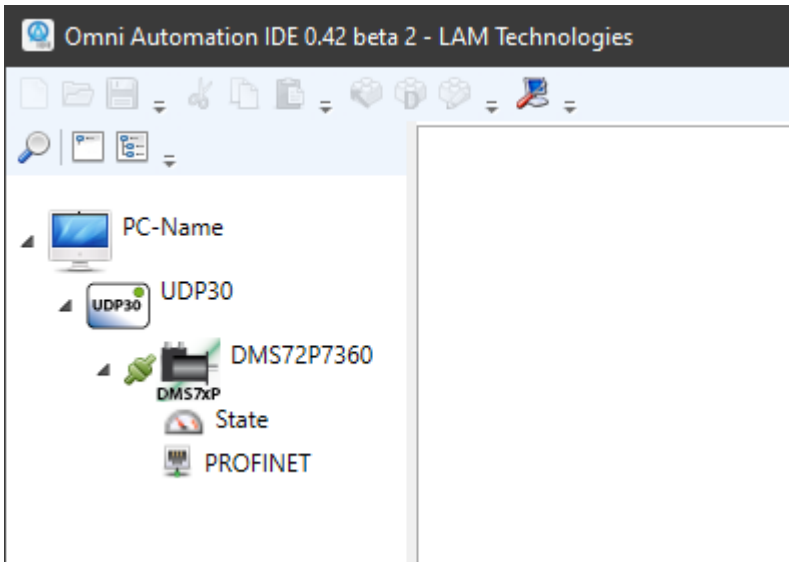
The connection between the PC and the motor with integrated drive is through the UDP30 interface which also deals with galvanic isolation of the PC from the device. The UDP30 interface is also able to power the logic section, allowing for example to update the firmware even without the power supply being applied.

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

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

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

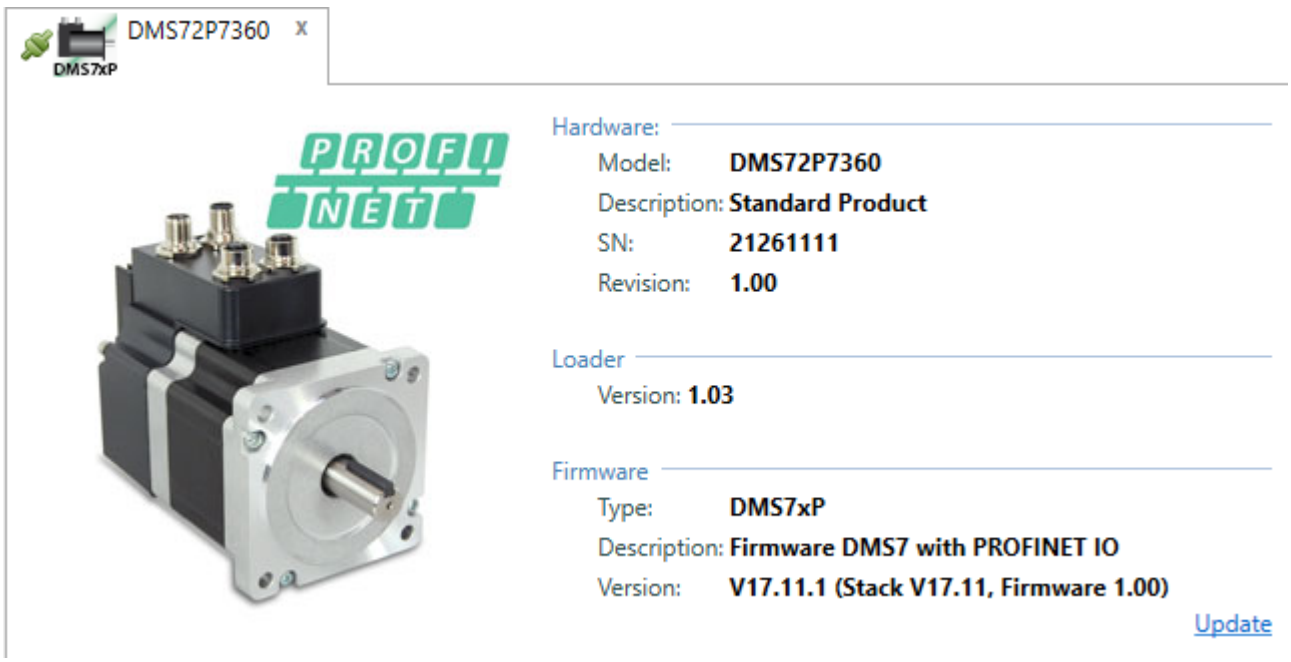
At the end of the search it appears the tree of the devices connected to the PC, with a look similar to the following image:



5.1 Firmware update

By double-clicking on the name of the motor with integrated drive (DMS72P7360 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 firmware of the device.

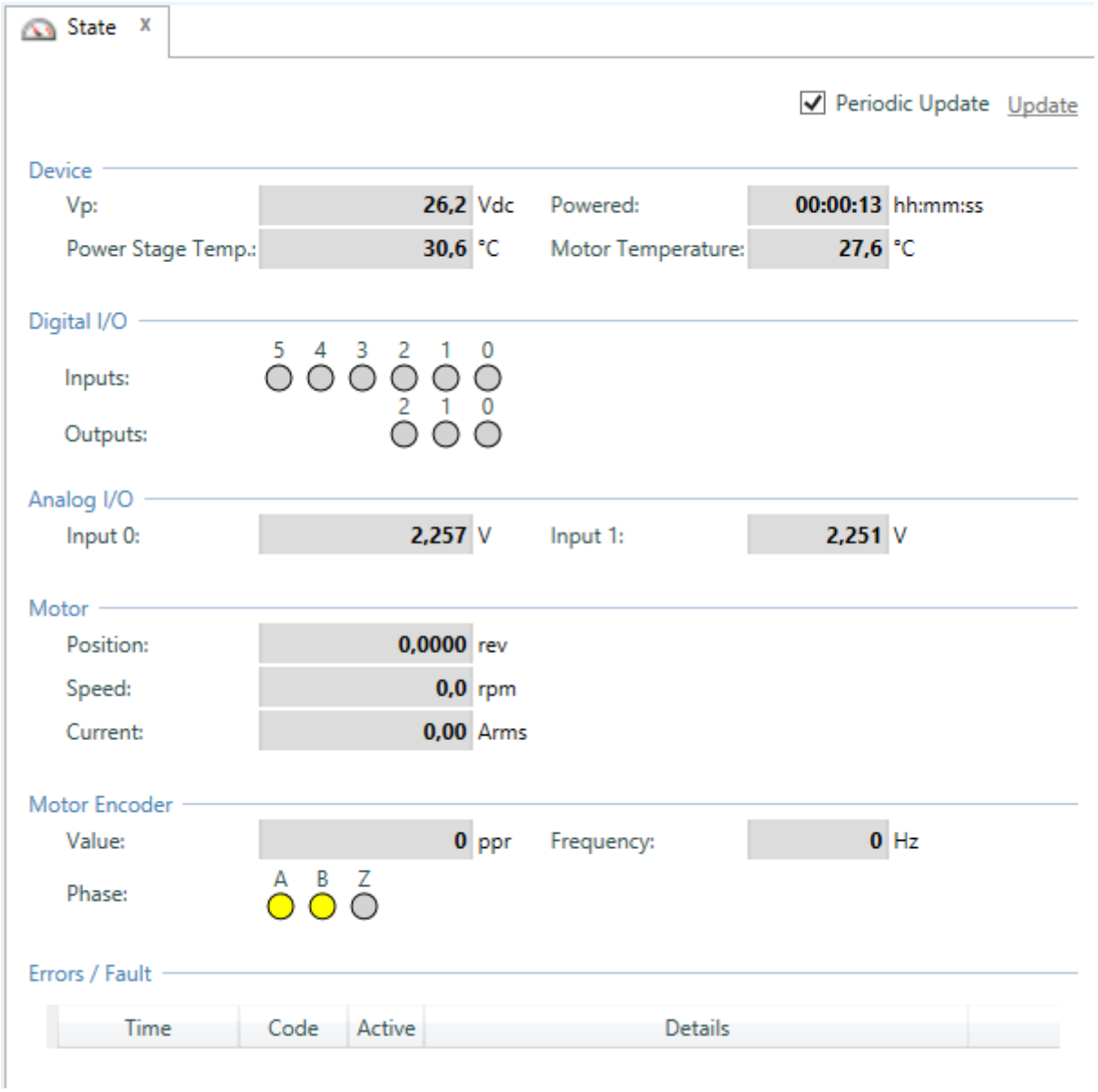


The products are typically sold with the latest released firmware which may include new functionalities, the improvement of the existing performances, bugs fixes, etc. A different firmware may produce a different behavior of the device and whether the correct functioning of the application is affected 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 revision may be released at any time and without prior notice.

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 motor with integrated drive, a window opens showing the status of the device and the eventual errors.



The check box Periodic Update, when selected, maintains updated the device status display. 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 device status display.

Under the section *Device,* the field *Powered since* shows the time elapsed since the device was turned on or since the last reset.

Under the section *Device,* the field *Vp* shows the voltage value of the power DC bus inside the motor with integrated drive.

Under the section *Device,* the field *Temperature* shows the power stage temperature value of the motor with integrated drive.

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 Input, the field Input 0 shows the voltage value applied to the analog input.

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 revolutions, 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 is different from the configuration because, particularly at high speed, or with low power supply voltage, because of the inductance and of the counter-electromotive force of the motor, the current cannot reach the set rated value.

The field *Load Ratio* becomes visible only when the motor with integrated 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 on a revolution. The value, expressed in 4x encoder resolution, is reset at each revolution of the encoder itself.

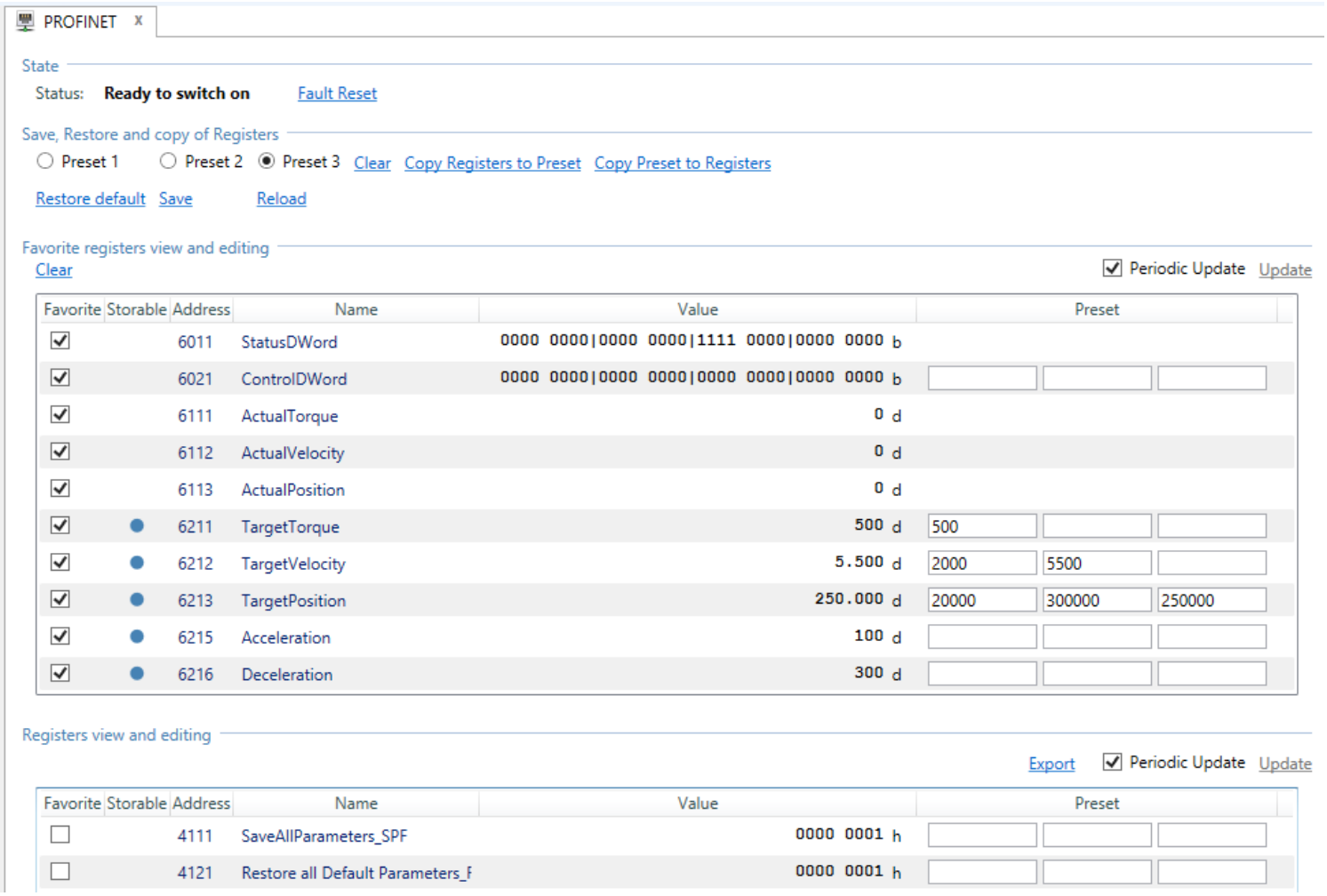
Under the section *Motor Encoder,* the field *Frequency* indicates the frequency of the encoder A and B signals.

Under the section *Motor Encoder,* it is possible to know the logic level of the *Phases* A, B and Z of the encoder according to the associated color. When a 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 the latest 10 errors, then the latest replace the oldest. The column *Time* shows the moment when the error occurred after the power on of the motor with integrated drive. The column *Code* contains the error numeric code while the column *Description* shows a brief description of the error. Simply positioning the mouse pointer over the content of each column a tooltip provides further more details.

5.3 Visualization and modification of register

In Omni Automation IDE, in the tree of the connected devices on the left, by double-clicking on the entry *PROFINET*, under the motor with integrated drive, you access the tab that allows visualizing and modifying the registers of the device.



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.

ATTENTION, removing the fault, the motor with integrated drive can immediately go to the operating state and move the motor according to the contents of the registers. Before removing the fault, configure appropriately the registers and make sure that an unexpected movement of the motor does not cause any damage to property or people.

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

The Presets are locations in which you can write a value that subsequently you can transfer into the corresponding dictionary register.

To transfer the Preset value into the register, simply press the Enter button or double-click with the left mouse button on the cell.

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.

The link *Save* allows you to save the contents of the registers in the non-volatile memory of the device while *Restore default* cancels the contents of the non-volatile memory restoring the default value of the registers. Both operations are only possible with the motor disabled. For more details see chapter 6.2 Saving and restoring of default values.

The registers visualization and modification area is logically divided in two zones; in the upper area there are the most common use registers (favorite) while in the lower zone there are all the registers of the motor with integrated drive ordered by address.

The *Periodic Update* check box, when selected, maintains updated the visualization of the device status. By removing the tick 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 visualization of the status of the motor with integrated drive.

The registers 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 registers 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 indicates whether the register can be saved in the non-volatile memory using the Store Parameters functions. If there is a blue point it means that the register will be stored in the non-volatile memory as a result of the Store Parameters command.

The following columns show the register address in hexadecimal format and the name.

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

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

The registers 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 register. By pressing the enter button or double-clicking with the left mouse button, the value of the field is copied into the register.

Please note that some registers prevent writing in specific operative conditions. For example, it is not possible to modify the motor parameters (as *Inductance_MTRDT*) when the motor with integrated drive is enabled.

Please note that when the cyclic process data exchange is running the registers mapped in the process data area are continuously overwritten from the master controller.

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

For clarity of writing it is possible to insert spaces 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 registers area, allows you to export the contents of each dictionary register to a file in CSV format.

By clicking with the right mouse button on the *PROFINET* item (on the left in the devices tree) it is possible to save and load the *PROFINET* tab content and settings.

6 Operation

The motor with integrated drive operates mainly in slave mode through the PROFINET IO fieldbus. With an appropriate configuration it is also possible to control some functions through the I/O signals integrated in the device (for example the alarms reset).

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 operating modes supported by the firmware revision described in this manual are 4, as shown in the table below:

Abbreviation	Description
pp	Position mode
pv	Velocity mode
tq	Torque mode
hm	Homing mode

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

Before operating, the drive requires some preliminary settings by writing appropriate value in the registers that compose the dictionary of the device itself.

6.1 Minimum settings

The minimum settings to be made before enabling the motor involves the setting of the running and idle current (registers 5312_h *CurrentMin_MTRCNF* and 5313_h *CurrentMax_MTRCNF*).

6.1.1 Running and idle current configuration

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

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

When the motor is provided with Encoder and the drive configured for closed-loop control, you can also set the current regulation so that it dynamically adapts to the load applied to the motor ($43A1_h$ *Mode_CRRG* register).

When the current regulation is configured in dynamic mode 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).

A detailed description of the registers 5312_h *CurrentMin_MTRCNF* and 5313_h *CurrentMax_MTRCNF* follows.

6.1.1.1 5312h CurrentMin_MTRCNF

The register *CurrentMin_MTRCNF* allows to specify the current applied to the motor in idle mode The current is applied after the motor stop passed the time defined by the register *CurrentIdleDelay_MTRCNF* described later.

The register is expressed as a percentage of the motor rated current 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 register *CurrentMin_MTRCNF* with the value 3000 (30 / 0.01). If the configured motor rated current is for example of 4Arms, the idle current will be equal to 1.2Arms (30% of 4A).

6.1.1.2 5313h CurrentMax_MTRCNF

The register *CurrentMax_MTRCNF* allows to specify the current applied to the motor during rotation.

The register is expressed as a percentage of the motor rated 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 register *CurrentMax_MTRCNF* with the value 8000 (80 / 0,01). If the configured motor rated current is for example of 4Arms, the idle current will be equal to 3.2Arms (80% of 4A).

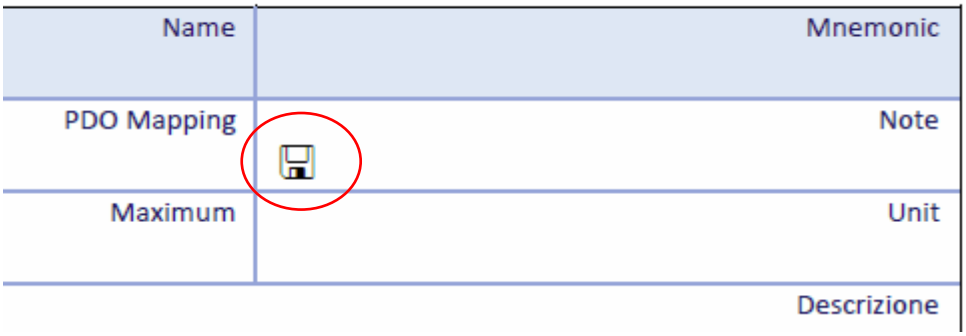
6.1.1.3 5314h CurrentIdleDelay_MTRCNF

The register *CurrentIdleDelay_MTRCNF* allows to specify the waiting time from the motor stop before the current is set to the value defined by the register *CurrentMin_MTRCNF*.

Each unit is equal to 1ms, therefore setting for example the value 500, the motor with integrated 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 registers in the non-volatile memory. The registers that can be saved are highlighted with the symbol 🖫 in the *Note* field of the table describing the register itself, as in the following example:



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

By saving a value different from the default one it is possible to adapt the device to the application without the need to configure it each time trough bus. Apparently this seems to be a simplification but it forces to prepare the device (saving the wished data in the registers) before it can be used in the application. When there are many applications, or they are updated over time, this forces us to keep an archive with all the registers values used in each application and in each version and in the time this can become complex and may cause errors. On the contrary, making the master to configure the device at every start it will be possible to simply install a new device without worrying about anything else. In this case, in fact, specific application will initialize the registers with the wished values and without the possibility of error. 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 registers. On the contrary, it is recommended to always initialize every register used in the application with the wished value, independently from the saving or default. The initialization must be repeated in case of device reset.

Saving occurs by writing an appropriate key in the register 4111_h *SaveAllParameters_SPF* or using the Omni Automation IDE commissioning software.

The registers can be also be restored to the default value writing an appropriate key in the register 4121_h *RestoreAllDefaultParameters_RDP* or using the Omni Automation IDE commissioning software.

The defaults values can be saved or restored only with the motor disabled. Trying the operation with the motor enabled an error code is received.

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

6.3 Motor Holding Brakes

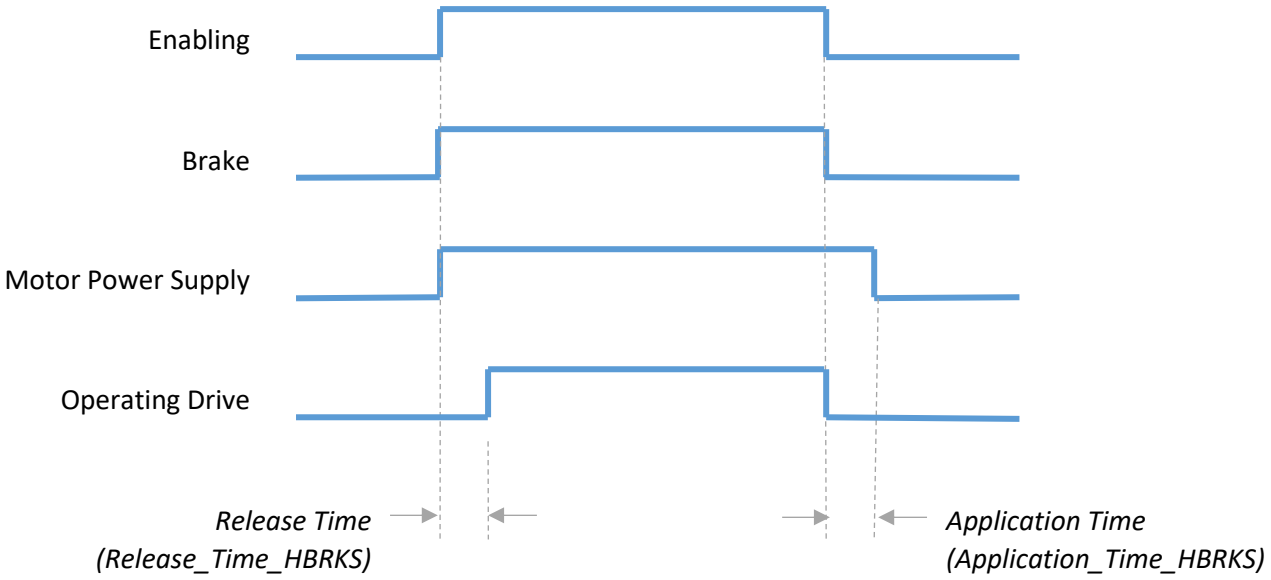
The motor with integrated drive is able to control an external holding brake through one of the digital outputs.

Through the register *Option_HBRKS* it is possible to handle automatically the brake upon enabling the motor, taking into account the brake's characteristic engaging and disengaging time.

With the brake control enabled, by enabling the motor the digital output arranged to control the brake is immediately activated (to release it) and at the same time the motor is supplied to maintain the position The transition to the operating state is delayed by the time required to the brake to completely disengage. This time can be freely set through the register *Release_Time_HBRKS*.

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

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



The register *Option_HBRKS* 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 made by the drive, or shared.

The manual control can be performed via a digital input that can be set through the register *Holding_Brake_DIA* or through the bit0 of the register *Control_HBRKC*.

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

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

6.4 Closed-loop

The motors with integrated drive equipped with Encoder belonging to the DMS72 series can also operate 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

To activate the closed-loop control, simply set to 1 the bits 1..0 of the register *Feedback_MTNSTP*.

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 Use and functionality of the StatusDWord and ControlDWord registers

The *StatusDWord* and *ControlDWord* registers are used by all the operating models and allow to know the status of the motor and to check its functioning.

The *StatusDWord* register has a dimension of 32bit (double word) and it is read-only. The following table shows the meaning of the bits contained in the register:

<table>
<tr><th colspan="4">StatusDWord</th></tr>
<tr><th>bit</th><th>Description</th><th>bit</th><th>Description</th></tr>
<tr><td>31</td><td>Reserved, ignore the value</td><td>15</td><td>PositionReached, 1=true</td></tr>
<tr><td>30</td><td>Reserved, ignore the value</td><td>14</td><td>VelocityReached, 1=true</td></tr>
<tr><td>29</td><td>Reserved, ignore the value</td><td>13</td><td>TorqueReached, 1=true</td></tr>
<tr><td>28</td><td>Reserved, ignore the value</td><td>12</td><td>Standstill, 1=true</td></tr>
<tr><td>27</td><td>Reserved, ignore the value</td><td>11</td><td>Reserved, ignore the value</td></tr>
<tr><td>26</td><td rowspan="3">OperationMode

<table><tr><th>bit 26..24</th><th>Operating mode</th></tr><tr><td>001</td><td>Position</td></tr><tr><td>010</td><td>Velocity</td></tr><tr><td>011</td><td>Torque</td></tr><tr><td>111</td><td>Homing</td></tr></table>
The bit combinations not in the table must be considered as non-selected operating mode.</td><td>10</td><td>ReferenceSet, 1=true</td></tr>
<tr><td>25</td><td>9</td><td>TriggerAcknowledge</td></tr>
<tr><td>24</td><td>8</td><td>Busy, 1=operation in progress</td></tr>
<tr><td>23</td><td>Reserved, ignore the value</td><td>7</td><td>Reserved, ignore the value</td></tr>
<tr><td>22</td><td>Reserved, ignore the value</td><td>6</td><td>Reserved, ignore the value</td></tr>
<tr><td>21</td><td>Reserved, ignore the value</td><td>5</td><td>Warning, 1= active</td></tr>
<tr><td>20</td><td>Reserved, ignore the value</td><td>4</td><td>Fault, 1=active</td></tr>
<tr><td>19</td><td>PositionError, 1=true</td><td>3</td><td>Halt, 1= active</td></tr>
<tr><td>18</td><td>VelocityError, 1=true</td><td>2</td><td>QuickStop, 1=active</td></tr>
<tr><td>17</td><td>Reserved, ignore the value</td><td>1</td><td rowspan="2">Enable

<table><tr><th>bit 1</th><th>bit 0</th><th>Description</th></tr><tr><td>0</td><td>0</td><td rowspan="3">Disabled</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>16</td><td>Reserved, ignore the value</td><td>0</td></tr>
</table>

Reserved bits must be ignored. Whatever is their value the application must not change its behavior.

The *ControlDWord* registers also has a dimension of 32bit (double word). The meaning of the bits is summarized in the following table:

<table>
<tr><th colspan="4">ControlDWord</th></tr>
<tr><th>bit</th><th>Description</th><th>bit</th><th>Description</th></tr>
<tr><td>31</td><td>Reserved, set to 0</td><td>15</td><td>Reserved, set to 0</td></tr>
<tr><td>30</td><td>Reserved, set to 0</td><td>14</td><td>Reserved, set to 0</td></tr>
<tr><td>29</td><td>Reserved, set to 0</td><td>13</td><td>Reserved, set to 0</td></tr>
<tr><td>28</td><td>Reserved, set to 0</td><td>12</td><td>Reserved, set to 0</td></tr>
<tr><td>27</td><td>Reserved, set to 0</td><td>11</td><td>Reserved, set to 0</td></tr>
<tr><td>26</td><td rowspan="3">OperationMode

<table><tr><th>bit 26..24</th><th>Operating mode</th></tr><tr><td>000</td><td>None</td></tr><tr><td>001</td><td>Position</td></tr><tr><td>010</td><td>Velocity</td></tr><tr><td>011</td><td>Torque</td></tr><tr><td>111</td><td>Homing</td></tr></table>
The bit combinations not in the table must be avoided.</td><td>10</td><td>AbsoluteRelative, 1=Relative</td></tr>
<tr><td>25</td><td>9</td><td>Trigger, ↑=trigger</td></tr>
<tr><td>24</td><td>8</td><td>TriggerMode, 1=Active</td></tr>
<tr><td>23</td><td>Reserved, set to 0</td><td>7</td><td>Reserved, set to 0</td></tr>
<tr><td>22</td><td>Reserved, set to 0</td><td>6</td><td>Reserved, set to 0</td></tr>
<tr><td>21</td><td>Reserved, set to 0</td><td>5</td><td>WarningAcknowledge, ↑=accepted</td></tr>
<tr><td>20</td><td>Reserved, set to 0</td><td>4</td><td>FaultAcknowledge, ↑=accepted</td></tr>
<tr><td>19</td><td>PositionErrorLatch, 1=Latch</td><td>3</td><td>Halt, 1= active</td></tr>
<tr><td>18</td><td>VelocityErrorLatch, 1=Latch</td><td>2</td><td>QuickStop, 1=active</td></tr>
<tr><td>17</td><td>Reserved, set to 0</td><td>1</td><td rowspan="2">Enable

<table><tr><th>bit 1</th><th>bit 0</th><th>Description</th></tr><tr><td>0</td><td>0</td><td rowspan="3">Disabled</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>Enabled</td></tr></table></td></tr>
<tr><td>16</td><td>Reserved, set to 0</td><td>0</td></tr>
</table>

The reserved bits must set to 0 to ensure the compatibility with the future firmware releases.

6.5.1 Motor Enabling

The motor is enabled setting the bit1 and bit0 of the *ControlDWord* register to 1. If there are no active fault and the motor with integrated drive conditions allow it (for example the power supply is within the allowed range) the motor is enabled and start to operate in the selected operating mode (for example *Position* if the bit26..24 have value 001)).

The enabling of the motor is confirmed by the bit1 and bit0 of the *StatusDWord* which both take value 1.

To disable the motor you just need to set the bit1 and bit0 of the *ControlDWord* register to 0.

Note that the motor with integrated drive can autonomously disable the motor for example in case of fault.

6.5.2 Emergency Stop (QuickStop)

By setting the bit2 of the *ControlDWord* register to 1 the emergency stop is activated. Once activated, the emergency stop cannot be interrupted. During the emergency stop the motor decelerates until it stops with the deceleration ramp defined by the *QuickStopDeceleration* register.

The bit2 of the *StatusDWord* register assumes the value 1 each time the motor with integrated drive executes the emergency stop.

By setting the bit2 of the *ControlDWord* register to 0 the emergency stop request is removed and if the procedure is completed (the motor has stopped) the motor with integrated drive returns to operate according to the chosen mode.

Note that the emergency stop can be automatically activated in the event of an emergency such as the activation of a limit switch, etc. If the bit2 of the *StatusDWord* register is set to 1 and the bit2 of the *ControlDWord* register is set to 0, it means that an emergency occurred requiring an arrest through QuickStop. In this case the motor is not disabled and the bit4 *Fault* of *StatusDWord* is not activated. To exit from the condition of Quickstop and return to the operating status, it is necessary to set the bit4 of *ControlDWord* to 1. Note that the motor with integrated drive uses the rising edge of bit4 (transition from 0 to 1) to exit the emergency condition. If bit4 is already set to 1 when the emergency occurs, it is first necessary to set it to 0 and then again to 1 to confirm acceptance of the emergency.

6.5.3 Motor Stop (Halt)

By setting the bit3 of the *ControlDWord* register to 1 the motor stops with the deceleration ramp defined by the *Deceleration* register.

The bit3 can be set to 0 at any time and in this case the motor starts moving again according to the selected operating mode and to the position, velocity and torque targets, accelerating with the acceleration ramp defined by the *Acceleration* register.

The *StatusDWord* register signals the activation of the Halt condition setting the bit3 to 1.

6.5.4 Fault e Warning

When the motor with integrated drive detects a fault condition it sets the bit4 of *StatusDWord* to 1 and activates appropriate actions such as the emergency stop or the disabling of the motor. The action taken depends on the nature of the fault.

To accept and remove the fault signal it is necessary to put the bit4 of *ControlDWord* to 1. If the fault condition persists the motor with integrated drive ignores the bit4 and remains in the fault status. Note that the motor with integrated drive uses the rising edge of bit4 (transition from 0 to 1) to exit the fault condition. If the bit4 is already set to 1 when the fault occurs it must be first set to 0 and the successively to 1 to confirm acceptance of the fault.

For some types of faults it is finally required to switch off and on the device.

When the motor with integrated drive detects warning conditions, it signals them by activating the bit5 of *StatusDWord.* The warning event does not compromise immediately the correct functioning of the motor with integrated drive that, therefore, remains in the operating state.

To accept the warning signal it is necessary to put the bit5 of *ControlDWord* to 1. If the warning condition persists the motor with integrated drive ignores bit5. Note that the motor with integrated drive uses the rising edge of the bit (transition from 0 to 1) to cancel the warning signal. If bit4 is already put to 1 when the fault occurs, it must be first put to 0 and then successively to 1 to confirm acceptance of the warning.

6.5.5 Real-time or delayed target process

The motor with integrated drive is able to process the position, velocity and torque targets in real time or in delayed mode, whenever the trigger is activated.

When the bit8 of *ControlDWord* has value 0 the motion adapts in real time to the target modification, so for example by modifying the value of the *TargetVelocity* register, the motor speed change to reach the new value.

Instead, by setting the bit8 of *ControlDWord* to 1, the modification of the targets values will not bring an immediate change of motion but only after the activation of the trigger signal. For example, setting a new position using the register *TargetPosition,* the motor will maintain its own path (or position) until the trigger signal is activated. Only then the new targets will produce the desired effects on motion.

The registers affected by the bit4 of *ControlDWord* are shown in the table:

Registers that can be processed in delayed mode
TargetTorque
TargetVelocity
TargetPosition
Acceleration
Deceleration

The trigger signal that controls the application of the targets when the delayed mode is selected consists of the rising edge (transition from 0 to 1) of the bit9 of *ControlDWord*.

The rising edge (transition from 0 to 1) of the bit9 of *StatusDWord* register confirms the application of the targets.

6.5.6 Absolute or relative positioning

When the motor with integrated drive works in *Position* mode and the delayed targets mode application is active, it is possible to choose whether the position target (*TargetPosition* register) must interpreted as an absolute or relative position.

When the bit10 of *ControlDWord* is set to 0 the position is interpreted in absolute value while when the bit0 is set to 1 the contents of the *TargetPosition* register will be used to make a relative movement, with respect to the previous target, at each trigger activation.

Note that if the targets are processed in real time (bit8=0 of *ControlDWord*) it is not possible to use the relative positioning and the bit10 of *ControlDWord* is ignored.

6.5.7 Target achieved and motor halted

In the *StatusDWord* register there are 3 bits that indicate the achievement of the commanded target.

The bit13 assumes the value 1 when the difference between the target torque and the one supplied by the motor is, in absolute value, less than the value set by the *WindowTorqueTime* register, for a time longer than the value set by the *WindowTorqueTime* register.

The bit14 assumes the value 1 when the difference between the target velocity and the actual one of the motor is, in absolute value, less than the value set by the *WindowVelocity* register, for a time longer than the value set by the *WindowVelocityTime* register.

The bit15 assumes the value 1 when the difference between the target position and the one assumed by the motor is, in absolute value, less than the value set by the *WindowPosition* register, for a time longer than the value set by the *WindowPositionTime* register.

The bit12 of the status word indicates instead the halted motor condition and it is set to 1 when the actual speed of the motor is, in absolute value, less than the value set by the *WindowZeroSpeed* register. The bit assumes value 0 if the actual motor speed exceeds, in absolute value, the value set by the *WindowZeroSpeed* register, for a time longer than *WindowZeroSpeedTime*.

6.5.8 Following error

In the *StatusDWord* register there are 2 bits that are activated (bit = 1) in case of a following error.

The following error occurs when the speed or the position differ from the value required by the profile generator beyond a value set and for a defined time.

The bit18 of *StatusDWord* assumes value 1 if the difference between the actual motor speed and the one required by the profile generator is, in absolute value, greater than the value set by the *WindowVelocityError* register, for a time longer than the value set by the *WindowVelocityErrorTimeOut* register.

The bit19 of *StatusDWord* assumes value 1 if the difference between the actual motor position and the one required by the profile generator is, in absolute value, greater than the value set by the *WindowPositionError* register, for a time longer than the value set by the *WindowPositionErrorTimeOut* register.

The bit18 and bit19 of *ControlDWord* allow to store the activation of the speed and position following error respectively. By setting the bit to 1, if the following error is activated the corresponding bit of *StatusDWord*

remains at 1 even after the error has ceased. In this case, to set the following error bit to 0 again simply set the corresponding bit18 and bit19 of *ControlDWord* to 0.

6.5.9 Operating mode

The setting of the operating mode is by the bit26, bit25 and bit24 of the *ControlDWord* register.

The following table shows the relation between the value of the bits and the selected operating mode:

bit 26..24 *ControlDWord*	Selected operating mode
000	None
001	*Position*
010	*Velocity*
011	*Torque*
111	*Homing*

Combination different from the ones in the table must be avoided.

The motor with integrated drive confirms the active operating mode by the bit26, bit25 and bit24 of *StatusDWord* register.

The following table shows the relation between the value of the bits and the active operating mode:

bit 26..24 *StatusDWord*	Active operating mode
000	None
001	*Position*
010	*Velocity*
011	*Torque*
111	*Homing*

6.6 Operating Modes

The following paragraphs describe the different operating modes. All the examples assume that the motor with integrated drive is started with the default values and successively configured with the minimum settings described in the previous chapter (6.1 Minimum settings).

6.6.1 Position (pp)

In this mode the motor with integrated drive executes a positioning profile by controlling the torque, the velocity and position of the motor. The movement is performed according to the set values of maximum speed, acceleration and deceleration. The master controller can update the target position simply updating the *TargetPosition* register present among the process data.

To select the *Position* mode it is sufficient to set the bit26..24 of *ControlDWord* to the value 1 (bit26=0, bit25=0 and bit24=1).

The position, velocity and torque targets can be processed in real or delayed time whenever the trigger is activated, as described in chapter 6.5.5 *Real-time or delayed target process*. Furthermore the master controller can command absolute or relative positioning, as described in chapter 6.5.6 *Absolute or relative positioning*.

The conditions of achieved target and halted motor are parameterizable as well as the limits associated with the following error, as described in chapters 6.5.7 *Target achieved and motor halted* and 6.5.8 *Following error* respectively.

Once the parameters have been set, it is possible to use the only process data to control the position, velocity or torque of the motor.

For example, assuming that the motor is in the position 0, setting the process data as shown in the table below, the motor will execute 12 revs until it reaches the position 120.000 with a speed of 400rpm, an acceleration equal to 2000rpm/s and a deceleration of 3500rpm/s. The maximum torque is set at the 88.5% of the rated value.

Process data	
OUT (outputs from PLC)	IN (inputs to PLC)
ControlDWord = 0x01000003	*StatusDWord*
TargetTorque = 885	*ActualTorque*
TargetVelocity = 4000	*ActualVelocity*
Acceleration = 2000	*ActualPosition*
Deceleration = 3500	*ActualPositionError*
TargetPosition = 120000	*Inputs_DIV*

6.6.2 Velocity (pv)

In this mode the motor with integrated drive controls the motor in speed, always according to the set acceleration and deceleration ramps and to the maximum allowed speed. The master controller can update the velocity target simply updating the register *TargetVelocity* present among process data.

To select the *Velocity* mode it is sufficient to set the bit26..24 of *ControlDWord* to the value 2 (bit26=0, bit25=1 e bit24=0).

The velocity and torque targets can be processed in real or delayed time whenever the trigger is activated, as described in chapter 6.5.5 *Real-time or delayed target process*.

The conditions of achieved target and halted motor are parameterizable as well as the limits associated with the following error, as described in chapters 6.5.7 *Target achieved and motor halted* e 6.5.8 *Following error* respectively.

Once the parameters have been set, it is possible to use the only process data to control the velocity or torque of the motor.

For example, by setting the process data as shown in the table below, the motor will reach the speed of 640,5rpm, with an acceleration equal to 500rpm/s. To decelerate it will use a ramp of 1000rpm/s while the maximum torque is set at the 100% of the rated value.

Process data	
OUT (outputs from PLC)	IN (inputs to PLC)
ControlDWord = 0x02000003	*StatusDWord*
TargetTorque =1000	*ActualTorque*
TargetVelocity = 6405	*ActualVelocity*
Acceleration = 500	*ActualPosition*
Deceleration = 1000	*ActualPositionError*
TargetPosition = … (ininfluente)	*Inputs_DIV*

Note that in the V*elocity* mode the *TargetPosition* register is ignored.

6.6.3 Torque (tq)

The T*orque* mode can be used only when the 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 register *TargetTorque*. The torque variation on the motor shaft always occurs according to the ramp set through the register *TorqueSlope*.

Usually in this mode the motor speed is limited only by the maximum one allowed by the motor itself, anyway by the *Mode_PTCNF* register it is possible to activate the speed control also through the *TargetVelocity* register.

To select the Torque mode it is sufficient to set the bit26..24 of *ControlDWord* to the value 3 (bit26=0, bit25=1 and bit24=1).

The torque target can be processed in real or delayed time whenever the trigger is activated, as described in chapter 6.5.5 *Real-time or delayed target process*.

The conditions of achieved target reach and halted motor are parameterizable as well as the limits associated with the following error, as described in chapter 6.5.7 *Target achieved and motor halted*.

Once the parameters have been set, it is possible to use the only process data to control the torque of the motor.

By setting the process data as shown in the table below, the torque of the motor will be set at the 50% of the rated value.

Process data	
OUT (outputs from PLC)	IN (inputs to PLC)
ControlDWord = 0x03000003	*StatusDWord*
TargetTorque =500	*ActualTorque*
TargetVelocity = … (ignored)	*ActualVelocity*
Acceleration = … (ignored)	*ActualPosition*
Deceleration = … (ignored)	*ActualPositionError*
TargetPosition = … (ignored)	*Inputs_DIV*

Note that in the *Torque* mode the *TargetVelocity*, *TargetPosition*, *Acceleration* and *Deceleration* registers are ignored.

6.6.4 Homing (hm)

Through the Homing mode the motor with integrated drive 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 select the *Homing* mode it is sufficient to set the bit26..24 of *ControlDWord* to the value 7 (bit26=0, bit25=0 and bit24=1).

To perform the homing procedure, the master controller must configure the homing method through the register *HomingMethod* and then start the homing by activating the *Trigger* (transition 0->1 of bit9 of *ControlDWord).*

If the homing ends successfully, the bit10 *ReferenceSet* of Status*DWord* is set to 1 and the *ActualPosition* register is set to the value of the *HomingOffset* register. Subsequent modifications to the *HomingOffset* register do not modify the homing reference.

HomingSpeedForSwitch and *HomingSpeedForZero* registers allow you to set the speed used during the sensor search and, successively, for the homing.

The sensors connected with the motor with integrated drive must be associated with the corresponding digital input so that the device can correctly read the signals and successfully complete the homing procedure. To associate the sensors with the inputs, use the registers *Home_DIA*, *NegativeLimit_DIA* and *PositiveLimit_DIA*.

The following table describes the available homing modes and the sensors used by each 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 register *HomingMethod*					
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, the 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,		●	●	●

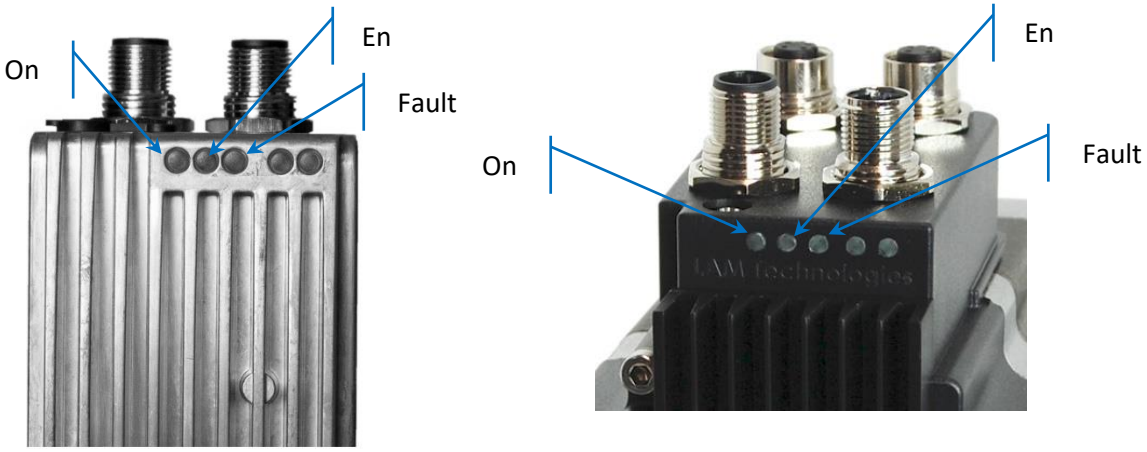
	continue up to leave the home switch and then homing at the first index outside the switch. At the start, if home switch active initial direction clockwise up to leave the home switch, then homing at the first index outside the home switch.				
12	At the start, if home switch inactive initial direction 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.				

7 Errors and diagnostics

The motor with integrated drive is equipped with 3 LEDs, positioned in the back, 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 fault and error conditions which are signaled by the red LED *Fault* and fieldbus and, by setting properly the register *Source_DOA_x*, also through the digital outputs.

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

Class	Description	Motor with integrated drive'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 successive 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 supplied the logic part and thus prevent the reset of the moor with integrated drive.

Each error has a code which identifies the type and in some case a sub-code which helps to identify the source of the problem. The error code appears in the register *ErrorCode* while the sub-code in the register *ErrorSubCode*.

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

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

<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>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<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)
<table>
<tr><td>Sub-code</td><td>Description</td></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 operating</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 installed firmware</td></tr>
<tr><td>55A2$_h$</td><td>8</td><td>13</td><td>31</td><td>Incompatible installed 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
<table>
<tr><td>Sub-code</td><td>Description</td></tr>
<tr><td>0701$_h$</td><td>Selected Torque mode without encoder feedback (present or configured).</td></tr>
<tr><td>0702$_h$</td><td>Selected current proportional to the load (register Mode_CRRG) without encoder feedback (present o configured).</td></tr>
<tr><td>0704$_h$</td><td>Activated encoder feedback without having activated or configured the encoder</td></tr>
<tr><td>0705$_h$</td><td>Activated the following error control without having activated or configured the encoder selected as feedback source.</td></tr>
<tr><td>0706$_h$</td><td>Activated the incremental encoder Index management and contemporaneously the SSI encoder.</td></tr>
</table>
</td></tr>
<tr><td>7282$_h$</td><td>8</td><td>14</td><td>11</td><td>Detected invalid offset values</td></tr>
<tr><td>8100$_h$</td><td>6</td><td>7</td><td>28</td><td>Bus communication error</td></tr>
<tr><td>8611$_h$</td><td>6</td><td>6</td><td>18</td><td>Following error. The indicated error class expresses the default value. The class can be modified through the register Following_Error_ERRCS.</td></tr>
<tr><td>8612$_h$</td><td>2</td><td>6</td><td>19</td><td>Limit switch activation
<table>
<tr><td>Sub-code</td><td>Description</td></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 positive limit switch active</td></tr>
<tr><td>0804$_h$</td><td>Negative movement with negative limit switch active</td></tr>
</table>
</td></tr>
<tr><td>8613$_h$</td><td>2</td><td>6</td><td>17</td><td>Homing error</td></tr>
</table>

				Sub-code	Description
				0601_h	Index not found
				0602_h	Unexpected clockwise limit switch
				0603_h	Unexpected counterclockwise limit switch
				0604_h	The selected homing mode uses the incremental encoder Index but the management of the Index itself is not active (*Configuration_ENCMTR* register)
				0605_h	The selected homing mode uses the SSI encoder but the SSI encoder reading is not operative.

8 Registers

The device is parameterized and controlled through the writing and reading of registers. Some registers are read-only while others can be both read and written. Some of them can be modified only in certain status of the device, for example with the motor disabled.

The access to the registers may occur cyclically, if they are included in the process data, or acyclically through the record reading and writing functions provided by the PROFINET IO protocol.

Inside the Siemens TIA Portal environment the WRREC and RDREC Function Blocks are available to access respectively the wiring and reading of the registers.

8.1 Data process mapping

For a better comprehension of functionality and use of the following register, we recommend to read chapter 4.2 Cyclic access to the registers.

Address	Name			Mnemonic
4061_h	*NumberOfMappedObjects_RPDM*			
	Data Type	Access Type	PD Mapping	Note
	u8	rw		🖫 It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum	Unit
	6		8	
	Description			
	Number of registers mapped in the mapping registers *MappedObject_n_RPDM* (with n between 1 and 8) of the process data received by the drive. When the register *NumberOfMappedObjects_RPDM* is set to 0, none of the registers are mapped in the process data and the writing in the registers *MappedObject_n_RPDM* is enabled. When the register *NumberOfMappedObjects_RPDM* is set to a value less than 8, the drive ignores the process data transmitted by the PLC not mapped in the drive registers (eventual registers excluded from the mapping and previously mapped continue to maintain the last value received). When the setting of the register *NumberOfMappedObjects_RPDM* fails, the cause is typically one of the following: - The motor is enabled - The written value is greater than 8 - The content of one or more registers *MappedObject_n_RPDM* is not valid			

Address	Name			Mnemonic
4062_h 4063_h 4064_h 4065_h 4066_h 4067_h 4068_h 4069_h	*MappedObject_n_RPDM*			
	Data Type	Access Type	PD Mapping	Note
	U32	rw		🖫 It can be modified only with the register *NumberOfMappedObjects_RPDM* set to 0.
	Default Value	Minimum	Maximum	Unit
	See *Description*			

Description

It contains the information regarding the register n mapped in the process data received by the drive, with n between 1 and 8.

The information is composed of the address of the register to be mapped (Address) and the length of the register itself expressed in bit (Length).

Bits 31…16	Bits 15…8	Bits 7…0
Address	Reserved, do not use	Length

The default values of the registers *MappedObject_n_RPDM* are the following:

RPD mapping (received by the driver and transmitted by the PLC)					
Registers			Mapped registers		
Addresss	Name	Default Value	Offset	Name	Dimension (byte)
4062_h	*MappedObject_1_RPDM*	60210020_h	0	*ControlDWord*	4
4063_h	*MappedObject_2_RPDM*	62110010_h	4	*TargetTorque*	2
4064_h	*MappedObject_3_RPDM*	62120020_h	6	*TargetVelocity*	4
4065_h	*MappedObject_4_RPDM*	62150020_h	10	*Acceleration*	4
4066_h	*MappedObject_5_RPDM*	62160020_h	14	*Deceleration*	4
4067_h	*MappedObject_6_RPDM*	62130020_h	18	*TargetPosition*	4
4068_h	*MappedObject_7_RPDM*	00000000_h			
4069_h	*MappedObject_8_RPDM*	00000000_h			

Address	Name			Mnemonic
40A1$_h$	*NumberOfMappedObjects_TPDM*			
	Data Type	Access Type	PD Mapping	Note
	u8	rw		🖫 It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum	Unit
	6		8	
	Description			
	Number of registers mapped in the mapping registers *MappedObject_n_TPDM* (with n between 1 and 8) of the process data received by the drive. When the register *NumberOfMappedObjects_TPDM* is set to 0, none of the registers is mapped in the process data and the writing of the registers *MappedObject_n_TPDM* is enabled. When the register *NumberOfMappedObjects_TPDM* is set to a value less than 8, the process data transmitted from the drive to the PLC not mapped in the register of the drive itself are set to 0. When the setting of the register *NumberOfMappedObjects_TPDM* fails, the cause is typically one of the following: - The motor is enabled - The written value is greater than 8 - The content of one or more registers *MappedObject_n_TPDM* is not valid			

Address	Name			Mnemonic
40A2$_h$ 40A3$_h$ 40A4$_h$ 40A5$_h$ 40A6$_h$ 40A7$_h$ 40A8$_h$ 40A9$_h$	*MappedObject_n_TPDM*			
	Data Type	Access Type	PD Mapping	Note
	U32	rw		🖫 It can be modified only with the register *NumberOfMappedObjects_TPDM* set to 0.
	Default Value	Minimum	Maximum	Unit
	See *Description*			

Description

It contains the information regarding the register n mapped in the process data transmitted by the drive, with n between 1 and 8.

The information is composed of the address of the register to be mapped (Address) and the length of the register itself expressed in bit (Length).

Bits 31…16	Bits 15…8	Bits 7…0
Address	Reserved, do not use	Length

The default value of the registers *MappedObject_n_TPDM* are the following:

TPD mapping (transmitted by the drive and received by the PLC)					
Registers			Mapped registers		
Addresss	Name	Default Value	Offset	Name	Dimension (byte)
40A2$_h$	*MappedObject_1_TPDM*	60110020$_h$	0	*StatusDWord*	4
40A3$_h$	*MappedObject_2_TPDM*	61110010$_h$	4	*ActualTorque*	2
40A4$_h$	*MappedObject_3_TPDM*	61120020$_h$	6	*ActualVelocity*	4
40A5$_h$	*MappedObject_4_TPDM*	61130020$_h$	10	*ActualPosition*	4
40A6$_h$	*MappedObject_5_TPDM*	61220020$_h$	14	*ActualPositionError*	*4*
40A7$_h$	*MappedObject_6_TPDM*	52110010$_h$	18	*Inputs_DIV*	*2*
40A8$_h$	*MappedObject_7_TPDM*	00000000$_h$			
40A9$_h$	*MappedObject_8_TPDM*	00000000$_h$			

8.2 Store Parameters

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

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

At most it is possible to save or restore the default values 10.000 times. The saving is completed in about 100ms.

Address	Name			Mnemonic
4111_h	*SaveAllParameters_SPF*			
	Data Type	Access Type	PDO Mapping	Note
	u32	rw		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
	Description			
	It allows to save the dictionary registers, specified with the symbol 🖫, in the non-volatile memory. To start and save simply write the register with the value **0x65766173**. 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 reply.			

8.3 Restore Default Parameters

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

At most it is possible to restore the default values 10.000 times. The restoring is completed in about 100ms.

Address	Name			Mnemonic
4121_h	*RestoreAllDefaultParameters_RDP*			
	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 default value for the dictionary registers.			

To start to restore simply write the register with the value **0x64616F6C**.

The default values can be restored only with the motor disabled. Trying the operation with the motor enabled, an error code is obtained in reply.

8.4 Motor Data

Address	Name			Mnemonic
4311_h	*CMC_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	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 parameters for the motor. The code *CMC_MTRDT* is automatically compiled and does not need to be initialized. When the register is initialized with a value different from 0, the registers from 4312_h to $431C_h$ are ignored.			

Address	Name			Mnemonic
4312_h	*Type_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	u8	ro		💾 Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	12			
	Description			
	Not used, ignore the value.			

Address	Name			Mnemonic
4313_h	*PolePairs_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	u8	ro		💾 Valid only if *CMC_MTRDT* = 0.
	Default Value	Minimum	Maximum	Unit
	50	50	50	
	Description			
	Not used, ignore the value.			

Address	Name			Mnemonic
4314_h	*Wiring_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	u8	ro		💾 Valid only if *CMC_MTRDT* = 0
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	Not used, ignore the value.			

Address	Name			Mnemonic
4315h	*Resistance_MTRDT*			
	Data Type	Access Type	PD 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.			

Address	Name			Mnemonic
4316h	*Inductance_MTRDT*			
	Data Type	Access Type	PD 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.			

Address	Name			Mnemonic
4317h	*BackEMF_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		🖫 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.			

Address	Name			Mnemonic
4318h	*RatedCurrent_MTRDT*			
	Data Type	Access Type	PD 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.			

Address	Name			Mnemonic
4319h	*MaxCurrent_MTRDT*			
	Data Type	Access Type	PD 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.			

Address	Name			Mnemonic
431A$_h$	*RatedTorque_MTRDT*			
	Data Type	Access Type	PD 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.			

Address	Name			Mnemonic
431B$_h$	*MaxSpeed_MTRDT*			
	Data Type	Access Type	PD Mapping	Note
	u16	ro		🖫
	Default Value	Minimum	Maximum	Unit
	30000	1	30000	0.1rpm (Ex. 668 = 66.8rpm)
	Description			
	Not used, ignore the value.			

8.5 Motor Incremental Encoder Data

Address	Name			Mnemonic
4331h	*Configuration_ENCMTR*			
	Data Type	Access Type	PD Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0			

Description

It indicates the encoder management mode. The register is made up of groups of bits that define the following functions

Bits	Functions
0	Index 0 – Index signal not managed 1 – Index signal managed
1	Pulse count inversion 0 – Normal pulse 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 motor with integrated 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 value, do not use The equivalent maximum frequency must be calculated considering two times the set time, as it is necessary that such time elapses both for the active signal condition and for the inactive signal. Setting, for example, a Filter value of 1us you can apply a maximum frequency of 500KHz (1+1=2us i.e. 500KHz).

If the register is set to 0, the encoder input is disabled.

Address	Name			Mnemonic
4332_h	*CPR_ENCMTR*			
	Data Type	Access Type	PD Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	0			Encoder pulses per revolution
	Description			
	It indicates the pulses number/revolution of the encoder connected to the motor. When the value is 0, the encoder input is disabled. The device is able to count every pulse edge in order to obtain a resolution 4 times higher than the encoder native one. For example, by using a 400 pulses/revolution encoder, the drive will be able to recognize 1600 different positions/rev The encoder must be able to generate at least 360 pulses /rev.			

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

Address	Name			Mnemonic
4333_h	*PhasingTime_ENCMTR*			
	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 waiting time to phase the encoder. The first time that a motor equipped with an encoder is enabled, it is necessary to phase the encoder. When enabled, the motor could move and it is necessary to wait for the load to stabilize before proceeding to the encoder phasing. The *PhasingTime_ENCMTR* register allows you to set this time.			

8.6 SSI Encoder Data

Address	Name			Mnemonic
4351h	*Configuration_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0			

Description

It indicates the enabling status of the SSI encoder and the management mode. The register 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 motor with integrated 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* register) the first time the quota becomes valid and at each transition to the *Operational* status of the motor with integrated 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 register is set to 0, the SSI encoder input is disabled.

Address	Name			Mnemonic
4352h	*FrameLength_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	25	8	47	bit
	Description			
	It indicates the number of bits that make up the data frame of the SSI encoder. The *start* bit must not be counted.			

Address	Name			Mnemonic
4353h	*TurnsBits_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	12		24	bit
	Description			
	It indicates the number of bits reserved for the full revolutions counting. In case of linear SSI encoder, set the register to 0.			

Address	Name			Mnemonic
4354h	*SingleTurnBits_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	13	8	31	bit
	Description			
	It indicates the bits reserved for the position on the revolution. In case of linear encoder, set the register with the number of bits that make up the quota of the encoder itself.			

Address	Name			Mnemonic
4355h	*MSB_Offset_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro		
	Default Value	Minimum	Maximum	Unit
	0		40	bit
	Description			
	It indicates 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 without counting the *start* bit) the *MSB_Offset_ENCSSI* register must be set to the value 3.			

Address	Name			Mnemonic
4356h	*FormingTime_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	150			1us (150 = 150us)
	Description			
	It indicates the time required by the SSI encoder to prepare a new quota. The motor with integrated drive waits at least this time between one reading and the next.			

Address	Name			Mnemonic
4357h	*SetupTime_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	1200			1ms (1200 = 1.2ms)
	Description			
	Ii indicates the time required by the SSI encoder to complete the initializations and provide the first valid quota. After the SSI encoder is enabled or after that power has been supplied (if the SSI encoder is configured with bit3 of the *Configuration_ENCSSI* register to 1), the motor with integrated drive waits at least this time before executing the first reading of the quota.			

Address	Name			Mnemonic
4358h	*ScalingFactor_N_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro		
	Default Value	Minimum	Maximum	Unit
	10000			
	Description			
	It indicates 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.*			

Address	Name			Mnemonic
4359h	*ScalingFactor_D_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro		
	Default Value	Minimum	Maximum	Unit
	10000			
	Description			
	It indicates 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.*			

Address	Name			Mnemonic
435Ah	*PositionOffset_ENCSSI*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro		
	Default Value	Minimum	Maximum	Unit
	10000			
	Description			
	It indicates the offset subtracted from the quota read by the SSI encoder to calculate the position expressed in 0.0001rev. The formula applied is the following: *Position_ENCSSIV* = (*PositionRaw_ENCSSIV* - *PositionOffset_ENCSSI*) * *ScalingFactor_N_ENCSSI* / *ScalingFactor_D_ENCSSI.*			

Address	Name			Mnemonic
435Bh	*PhasingTime_ENCSSI*			
	Data Type	Access Type	PD Mapping	Note
	u16	ro		
	Default Value	Minimum	Maximum	Unit
	500			1 ms (1500 = 1.5ms)
	Description			
	It indicates 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* register allows to set this time.			

8.7 Holding Brake Setup

Address	Name			Mnemonic
4361_h	*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 manual control.
The register is composed by groups of bits that operates as follows:

Bits	Function
0	Handling 0 – Not handled, the holding brake is not handled through the device 1 – Handled, the holding brake is handled through the device (connected to one output) When the bit is set to 1 upon activation the device waits for the brake disengaging time specified through the register *Release_Time_HBRKS* before going into the operating status. Upon disabling the device waits for the *Application_Time_HBRKS* before switching off the motor.
1	Automatic / Manual 0 – Automatic, the device command the activation and deactivation of the brake autonomously 1 – Manual, the brake control can take place only via a digital input or the register *Control_HBRKC*
2..7	Not used
8..11	Action It allows to define how the manual brake control (obtainable through a digital input or the register *Control_HBRKC*) 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 register *Control_HBRKC* set to 1 (the two signals are in logic OR between them). 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

Address	Name			Mnemonic
4362_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 engaged to ensure the maximum braking torque. When the bit0 of the register *Option_HBRKS* is set to 1 and the motor is requested to be disabled, the device waits for the time specified in the register *Application_Time_HBRKS*, after the brake has been engaged, before disconnecting the motor.			

Address	Name			Mnemonic
4363_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 necessary for the brake to completely disengage to ensure the minimum braking torque. When the bit0 of the register *Option_HBRKS* is set to 1 and the motor is requested to be enabled, the device waits for the time specified in the register *Release_Time_HBRKS*, after the brake has been switched off, before going into the operating status.			

Sub-Index 4364h	Name		Mnemonic
Data Type: u32	Access Type: rw	PDO Mapping:	Note: 💾
Default Value:	Minimum:	Maximum:	Unit:
Description: Reserved, do not use.			

Sub-Index 4365h	Name		Mnemonic
Data Type: u32	Access Type: rw	PDO Mapping:	Note: 💾
Default Value:	Minimum:	Maximum:	Unit:
Description: Reserved, do not use.			

8.8 Current Regulation

Address 43A1h	Name: *Mode_CRRG*		Mnemonic
Data Type: u8	Access Type: rw	PD Mapping: RPD	Note: 💾
Default Value: 0	Minimum:	Maximum:	Unit:

Description:

It defines the current supplying mode to the motor.

When in the static mode, the motor receives the current *CurrentMax_MTRCNF* when it is moving, and the current *CurrentMin_MTRCNF* when it is stopped (after the time defined by the register *CurrentIdleDelay_MTRCNF*).

Instead, when in the dynamic mode, the motor is driven with a current value proportional to the load. The current variation always occurs between the minimum and maximum values defined by the registers *CurrentMin_MTRCNF* and *CurrentMax_MTRCNF* respectively.

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

Address 43A2h	Name: *Gain_CRRG*		Mnemonic
Data Type: u8	Access Type: rw	PD Mapping: RPD	Note: 💾
Default Value: 100	Minimum: 50	Maximum: 200	Unit: % (Ex. 120 = 120%)

Description:

It allows to intervene manually on the open-loop gain.

The motor with integrated drive automatically determines the best parameter for the phase current regulation, nevertheless through this register it is possible to manually intervene on the gain up to halve it (50%) or double it (200%).

8.9 Motion Setup

Address	Name			Mnemonic
4411_h	*Feedback_MTNSTP*			
	Data Type	Access Type	PD Mapping	Note
	u8	rw	RPD	🖫 It cannot be modified with the drive enabled.
	Default Value	Minimum	Maximum	Unit
	0			

Description

The register is useful to enable and configure the closed loop functioning of the motor.

The following table resumes the use of the register bits.

Bit	Description
1..0	Encoder Feedback Value / Description 0 – Disabled 1 – Closed loop (feedback enabled) 2 – Following control 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.
2	Not used. Set to 0.
3	Encoder Phasing It allows to choose the Encoder Phasing mode. Value / Description 0 – Only when necessary (ex. at the first motor enabling after the device power on) 1 – Always at each motor enabling
4	Feedback source Value / Description 0 – Incremental encoder 1 – SSI encoder
7..5	Not used, set to 0.

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 registers *Configuration_ENCMTR* and *CPR_ENCMTR*.

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

Address	Name			Mnemonic
4412_h	*CurrentEnableRamp_MTNSTP*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	10		2000	ms (ex. 250 = 250ms).
	Description			
	Through this register it is possible to set the time taken by the current to reach the nominal value each time the motor is enabled. A long current ramp can be of help to limit the peak current on the power supply and to damp the rotor alignment movement the first time the motor is enabled.			

8.10 Position Loop

Address	Name			Mnemonic
4441_h	*Kp_PSTNLP*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	300	2	3000	
	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.			

8.11 Error Class Setup

Address	Name			Mnemonic
4481_h	*Following_Error_ERRCS*			
	Data Type	Access Type	PD 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 with integrated drive'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.

8.12 Digital Inputs Assignment

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

For example, it is possible to use a digital input to reset the default as an alternative to the bit *FaultAcknowledge* contained in the register *ControlDWord*. Please note that a same input can be associated with more functions and actions.

Address	Name			Mnemonic
4812_h	*FaultReset_DIA*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0502_h			

Description

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

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

High byte, active status		Low byte, input number	
Value	Description	Value	Description
4	Rising Edge	2	Digital Input 2
5	Falling Edge	3	Digital Input 3
		4	Digital Input 4
		5	Digital Input 5
		6	Digital Input 6
		7	Digital Input 7

Address	Name			Mnemonic
4813_h	*PositiveLimit_DIA*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0207_h			

Description

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

The register has a dimension of 2 bytes, the lower byte is used to selected the digital input while the lower 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 value assignable to the high and low bytes.

High byte, active status		Low byte, input number	
Value	Description	Value	Description
0	Always Inactive	2	Digital Input 2
1	Always Active	3	Digital Input 3
2	Active Input	4	Digital Input 4
3	Inactive Input	5	Digital Input 5
		6	Digital Input 6
		7	Digital Input 7

Address	Name			Mnemonic
4814_h	*NegativeLimit_DIA*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0206_h			

Description

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

The register has a dimension of 2 bytes, the lower byte is used to select the digital input while the higher 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 assignable to the high and low bytes.

High byte, active status		Low byte, input number	
Value	Description	Value	Description
0	Always Inactive	2	Digital Input 2
1	Always Active	3	Digital Input 3
2	Active Input	4	Digital Input 4
3	Inactive Input	5	Digital Input 5
		6	Digital Input 6
		7	Digital Input 7

Address	Name			Mnemonic
4815_h	*Home_DIA*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	0205_h			

Description

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

The register has a dimension of 2 bytes, the lower byte is used to select the digital input while the higher 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 assignable to the high and low bytes.

High byte, active status		Low byte, input number	
Value	Description	Value	Description
0	Always Inactive	2	Digital Input 2
1	Always Active	3	Digital Input 3
2	Active Input	4	Digital Input 4
3	Inactive Input	5	Digital Input 5
		6	Digital Input 6
		7	Digital Input 7

Address	Name			Mnemonic
$481A_h$	*Holding_Brake_DIA*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		
	Default Value	Minimum	Maximum	Unit
	0			

Description

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

The register has a size of 2 bytes and the low byte is used to select the digital input while the higher bytes is used to specify the state to be considered active. For example, 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 status		Low byte, input number	
Value	Description	Value	Description
0	Always Inactive	2	Digital Input 2
1	Always Active	3	Digital Input 3
2	Active Input	4	Digital Input 4
3	Inactive Input	5	Digital Input 5
		6	Digital Input 6
		7	Digital Input 7

If you do not need to manually control the holding brake, set the register to 0.
If, for example, you want to manually control the brake with input 7 active, set the register to 0207_h.

Address	Name			Mnemonic
481E$_h$	*TouchProbe_1_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit

Description

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

The register 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 status		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
		6	Digital Input 6
		7	Digital Input 7

Address	Name			Mnemonic
481F$_h$	*TouchProbe_2_DIA*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit

Description

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

The register 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 status		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
		6	Digital Input 6
		7	Digital Input 7

8.13 Digital Outputs Assignment

Address	Name			Mnemonic
4831_h	*Source_DOA_0*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw		💾
	Default Value	Minimum	Maximum	Unit
	8040_h			

Description

It allows to select the source to be assigned to the digital output 0 (DO0 signal).

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, setting the value 8040_h the output will be activated if there is no fault condition. Instead, 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 bit 0 value of the register *Outputs_DOV*.

Address	Name			Mnemonic
4841_h	*Source_DOA_1*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0041_h			

Description

It allows to select the source to be assigned to the digital output 1 (DO1 signal).

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, setting the value 8040_h the output will be activated if there is no fault condition. Instead, 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 bit 1 value of the register *Outputs_DOV*.

Address	Name			Mnemonic
4851_h	*Source_DOA_2*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	$006A_h$			

Description

It allows to select the source to be assigned to the digital output 2 (DO2 signal).

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, setting the value 8040_h the output will be activated if there is no fault condition. Instead, 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 bit 2 value of the register *Outputs_DOV*

8.14 Device Status

Address	Name			Mnemonic
5141_h	*BridgeTemperature_DVSTS*			
	Data Type	Access Type	PD Mapping	Note
	i16	ro	TPD	
	Default Value	Minimum	Maximum	Unit
				0.1°C (Ex. 528 = 52.8°C)
	Description			
	It indicates the temperature reached by the power stage.			

Address	Name			Mnemonic
5143_h	*PowerVoltage_DVSTS*			
	Data Type	Access Type	PD Mapping	Note
	u16	ro	TPD	
	Default Value	Minimum	Maximum	Unit
				0.1Vdc (Ex. 482 = 48.2Vdc)
	Description			
	It indicates the voltage of the power bus.			

8.15 Digital Inputs Value

Address	Name			Mnemonic
5211_h	*Inputs_DIV*			
	Data Type	Access Type	PD Mapping	Note
	u16	ro	TPD	
	Default Value	Minimum	Maximum	Unit

Description

It indicates the logical status of the digital inputs.

Register bits are associated with 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	Digital Input 6 (DI6)
7	Digital Input 7 (DI7)
8..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.

8.16 Digital Outputs Value

Address	Name			Mnemonic
5231_h	*Outputs_DOV*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	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 register bits 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 motor with integrated drive internal source (through the register *Digital Output n Action*) the change of the corresponding bit will not be possible and any attempt of that kind will be ignored.

Address	Name			Mnemonic
5232_h	*SetOutput_DOV*			
	Data Type	Access Type	PD Mapping	Note
	U8	wo		
	Default Value	Minimum	Maximum	Unit

Description

It allows to change one single bit of the register *Outputs_DOV*.

To change one single bit of the register *Outputs_DOV* simply write this register with a value as follows: the bits from bit 0 to bit 2 must contain the number of bits of the register *Outputs_DOV* on which you want to act while the bit 7 indicates if the bit must be set to 1 or 0. When the bit 7 is equal to 1, the bit of the register *Outputs_DOV* is put to 0 and vice versa.

For example, if the register *SetOutput_DOV* is written with the value 81_h the bit 1 of the register *Outputs_DOV* will be set to 0, writing instead the value 01_h the same bit will be set to 1. Instead writing the register *SetOutput_DOV* with value 00_h the bit 0 of the register *Outputs_DOV* will be set to 1.

The following table shows the use of the bits of the register *SetOutput_DOV.*

Bit	Description
0..3	Number of the bits of the register *Outputs_DOV* on which you want to act.

	Value / Description 0 – Bit 0 1 – Bit 1 … – Bit n 15 – Bit 15
4..6	Not used, set to 0
7	Logicl level to set in the bit of the register *Outputs_DOV* Value / Description 0 – Bit = 1 1 – Bit = 0

If the selected bit of the register *Outputs_DOV* corresponds to an output assigned to a motor with integrated drive internal source (through the register *Digital Output n Action*) the change of the corresponding bit will not be possible and any attempt of that kind will be ignored.

8.17 Analog Inputs Value

Address	Name			Mnemonic
5251h	*Voltage_AIV_0*			
	Data Type	Access Type	PD Mapping	Note
	i16	ro	TPD	
	Default Value	Minimum	Maximum	Unit
				1mV (Ex. 5302 = 5.302V)
	Description			
	It indicates the voltage at the analog input 0.			

Address	Name			Mnemonic
5261h	*Voltage_AIV_1*			
	Data Type	Access Type	PD Mapping	Note
	i16	ro	TPD	
	Default Value	Minimum	Maximum	Unit
				1mV (Ex. 5302 = 5.302V)
	Description			
	It indicates the voltage at the analog input 1.			

8.18 Analog Outputs Value

Address	Name			Mnemonic
5271h	*Voltage_AOV_0*			
	Data Type	Access Type	PD Mapping	Note
	i16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0			1mV (Ex. 2450 = 2.450V)
	Description			
	It allows to set the voltage of the analog output 0.			

Address	Name			Mnemonic
5281h	*Voltage_AOV_1*			
	Data Type	Access Type	PD Mapping	Note
	i16	rw	RPD	
	Default Value	Minimum	Maximum	Unit
	0			1mV (Ex. 2450 = 2.450V)
	Description			
	It allows to set the voltage of the analog output 1.			

8.19 Motor Configuration

Address	Name			Mnemonic
5312h	*CurrentMin_MTRCNF*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	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 register *Mode_CRRG*) the minimum current value is 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 time *CurrentIdleDelay_MTRCNF* from the stop. The value is expressed in percentage of the motor rated current set by the register *RatedCurrent_MTRDT*. 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 The current value can be limited by the register *TargetTorque*.			

Address	Name			Mnemonic
5313h	*CurrentMax_MTRCNF*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	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 register *Mode_CRRG*) the maximum current value is the current supplied to the motor in locked rotor condition. Instead, if the current regulation is static, the register *CurrentMax_MTRCNF* defines the current supplied to the motor when it is in rotation.

The value is expressed in percentage of the motor rated current set by the register *RatedCurrent_MTRDT*. 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.

The current value can be limited by the register *TargetTorque*.

Address	Name		Mnemonic
5314_h	*CurrentIdleDelay_MTRCNF*		
Data Type	Access Type	PD Mapping	Note
u16	rw		💾
Default Value	Minimum	Maximum	Unit
500	2	10000	1ms (Ex. 3500 = 3.5s)

Description

It allows to set the motor stopping time before the current reaches the value defined by the register *CurrentMin_MTRCNF.*

When the dynamic current regulation mode is active this register has no effect.

8.20 Motor Value

Address	Name		Mnemonic
$531B_h$	*Current_MTRV*		
Data Type	Access Type	PDO Mapping	Note
I16	ro	TPD	
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.

Address	Name		Mnemonic
$531C_h$	*Utilization_MTRV*		
Data Type	Access Type	PDO Mapping	Note
i16	ro	TPD	
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.

8.21 Motor Incremental Encoder Value

Address	Name			Mnemonic
5331_h	*Status_ENCMTRV*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

The register is useful to know the status of the motor incremental encoder.

The following table shows the meaning of the bits of the register.

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.

Address	Name			Mnemonic
5332_h	*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 .

Address	Name			Mnemonic
5333_h	*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.

8.22 SSI Encoder Values

Address	Name			Mnemonic
5339_h	*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 register is useful to know the status of the SSI encoder.

The following table resumes the meaning of the bits of the register.

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* registers 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* register to 1, differently the bit is set to 0.
7..4	Not used. Ignore the value.

Address	Name			Mnemonic
533A$_h$	*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*			

Address	Name			Mnemonic
533B$_h$	*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.			

8.23 Holding Brake Control

Address	Name			Mnemonic
5361$_h$	*Control_HBRKC*			
	Data Type	Access Type	PDO Mapping	Note
	u8	rw	RPD	🖫
	Default Value	Minimum	Maximum	Unit
	0			
	Description			
	It allows to manually control the holding brake.			

The

Bit	Description
0	Manual control It allows 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 register *Option_HBRKS,* therefore for a detailed description refer to chapter 6.3 Motor Holding Brakes 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.

following table resumes the use of the register bits.

Address	Name			Mnemonic
5362_h	*Status_HBRKC*			
	Data Type	Access Type	PDO Mapping	Note
	u8	ro	TPD	
	Default Value	Minimum	Maximum	Unit

Description

The register is useful to know the state of the holding brake.

The following table resumes the meaning of the register bit.

Bit	Description
0	Active It indicates the actual state of the output predisposed for the control of the holding brake. Value / Description: 0 – The brake control output is in the inactive state 1 – The brake control output is in the active state
1	Released It allows to know if the *Release_Time_HBRKS* has passed and the brake is then in the released state.

LAM Technologies electronic equipment

		Value / Description 0 – The brake is not released 1 – The brake is released
	2	Engaged It allows to know if the *Application_Time_HBRKS* has passed and the brake is then in the engaged state o. Value / Description 0 – The brake is not engaged 1 – The brake is engaged

8.24 Torque Profile Configuration

Address	Name			Mnemonic
5471_h	*Mode_PTCNF*			
	Data Type	Access Type	PD Mapping	Note
	u8	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0			

Description

It allows to configure the options of the *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 register *Mode_PTCNF*, the motor with integrated drive limits the motor maximum speed to the value set through the register TargetV*elocity*.

The following table shows the use of the register bits.

Bit	Description
0	Speed limitation Value / Description 0 – Speed not limited by the register *TargetVelocity* 1 – Speed limited by the register *TargetVelocity*
1..7	Not used, set to 0.

8.25 StatusDWord and ControlDWord

Address	Name	Mnemonic
6011h	*StatusDWord*	

Data Type	Access Type	PD Mapping	Note
u16	ro	TPD	

Default Value	Minimum	Maximum	Unit

Description

It allows to know the device status.

The register is structured in bits whose meaning is described in the following table.

StatusDWord

bit	Description	bit	Description
31	Reserved, ignore the value	15	PositionReached, 1=true
30	Reserved, ignore the value	14	VelocityReached, 1=true
29	Reserved, ignore the value	13	TorqueReached, 1=true
28	Reserved, ignore the value	12	Standstill, 1=true
27	Reserved, ignore the value	11	Reserved, ignore the value
26	OperationMode	10	ReferenceSet, 1=true
25		9	TriggerAcknowledge
24		8	Busy, 1=operation in progress
23	Reserved, ignore the value	7	Reserved, ignore the value
22	Reserved, ignore the value	6	Reserved, ignore the value
21	Reserved, ignore the value	5	Warning, 1= active
20	Reserved, ignore the value	4	Fault, 1=active
19	PositionError, 1=true	3	Halt, 1= active
18	VelocityError, 1=true	2	QuickStop, 1=active
17	Reserved, ignore the value	1	Enable
16	Reserved, ignore the value	0	

bit 26..24	Operation Mode
001	Position
010	Velocity
011	Torque
111	Homing

The bit combination not in the table must be considered as non-selected operating mode.

bit 1	bit 0	Description
0	0	Disabled
0	1	
1	0	
1	1	Enabled

The reserved bits must be ignored. Whichever their value, the application must not change its behavior.

Address	Name	Mnemonic

6021h *ControlDWord*

Data Type	Access Type	PD Mapping	Note
u16	rw	RPD	
Default Value	**Minimum**	**Maximum**	**Unit**

Description

It allows to control the device.

The register is structured in bits whose functions are described in the following table.

ControlDWord

bit	Description	bit	Description
31	Reserved, set to 0	15	Reserved, set to 0
30	Reserved, set to 0	14	Reserved, set to 0
29	Reserved, set to 0	13	Reserved, set to 0
28	Reserved, set to 0	12	Reserved, set to 0
27	Reserved, set to 0	11	Reserved, set to 0
26	OperationMode	10	AbsoluteRelative, 1=Relative
25		9	Trigger, ↑=trigger
24		8	TriggerMode, 1=Active
23	Reserved, set to 0	7	Reserved, set to 0
22	Reserved, set to 0	6	Reserved, set to 0
21	Reserved, set to 0	5	WarningAcknowledge, ↑=accepted
20	Reserved, set to 0	4	FaultAcknowledge, ↑=accepted
19	PositionErrorLatch, 1=Latch	3	Halt, 1= Active
18	VelocityErrorLatch, 1=Latch	2	QuickStop, 1=Active
17	Reserved, set to 0	1	Enable
16	Reserved, set to 0	0	

bit 26..24	Operation Mode
000	None
001	Position
010	Velocity
011	Torque
111	Homing

The bits combination not in the table must be avoided.

bit 1	bit 0	Description
0	0	Disabled
0	1	
1	0	
1	1	Enabled

The reserved bits must be set to 0 to ensure compatibility with future firmware revisions.

8.26 Errors

Address	Name	Mnemonic

Address	Name			Mnemonic
6081h	*ErrorCode*			
	Data Type	Access Type	PD Mapping	Note
	u16	ro	TPD	
	Default Value	Minimum	Maximum	Unit
	Description			
	It provides the code of the last error occurred in the device.			

Address	Name			Mnemonic
6082h	*ErrorSubCode*			
	Data Type	Access Type	PD Mapping	Note
	U32	ro	TPD	
	Default Value	Minimum	Maximum	Unit
	Description			
	It contains additional information on the last error occurred in the device.			

8.27 Actual Motion Values

Address	Name			Mnemonic
6111h	*ActualTorque*			
	Data Type	Access Type	PD Mapping	Note
	i16	ro	TPD	
	Default Value	Minimum	Maximum	Unit
				0.1% (Ex. 405 = 40.5%)
	Description			
	It provides the actual value of the torque. The value is given in 0.1% of the nominal motor torque.			

Address	Name			Mnemonic
6112h	*ActualVelocity*			
	Data Type	Access Type	PD Mapping	Note
	i32	ro	TPD	
	Default Value	Minimum	Maximum	Unit
				0.1rpm (Ex. 3850 = 385.0rpm)
	Description			
	The register provides the actual velocity value. The value is given in 0.1rpm.			

Address	Name	Mnemonic

6113h	*ActualPosition*			
	Data Type: i32	Access Type: ro	PD Mapping: TPD	Note:
	Default Value:	Minimum:	Maximum:	Unit: 0.0001rev (Ex. 5000 = 0.5000rev)
	Description: The register provides the actual position value. The value is given in 1/10000 of revolution.			

Address 6122h	Name: *ActualPositionError*			Mnemonic:
	Data Type: i32	Access Type: ro	PD Mapping: TPD	Note:
	Default Value:	Minimum:	Maximum:	Unit: 0.0001rev (Ex. 20000 = 2.0000rev)
	Description: This register allows to read the following position error value. The value is given in 1/10000 of revolution.			

8.28 Target Motion Values

Address 6211h	Name: *TargetTorque*			Mnemonic:
	Data Type: i16	Access Type: Rw	PD Mapping: RPD	Note: 💾
	Default Value: 0	Minimum: -1300	Maximum: 1300	Unit: 0.1% (Ex. 405 = 40.5%)
	Description: This register allows to set the wished target torque in the *Torque* mode. In the *Velocity* or *Position* mode it sets the maximum torque value of the motor, if drive operate in closed-loop mode. In open-loop *TargetTorque* acts by limiting the phase current to the motor and is usually set to value 1000, corresponding to the 100% of the current. In open-loop mode the motor current is typically set using the registers *CurrentMin_MTRCNF* and *CurrentMax_MTRCNF*. The value is given in 0.1% of the nominal motor torque.			

Address 6212h	Name: *TargetVelocity*			Mnemonic:
	Data Type: i32	Access Type: rw	PD Mapping: RPD	Note: 💾
	Default Value: 0	Minimum:	Maximum:	Unit: 0,1rpm (Ex. 1000 = 100.0rpm)
	Description: The register allows to set the target velocity in the *Velocity* mode. In the *Position* mode it defines the full speed reached at the end of the acceleration ramp. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6213_h	*TargetPosition*			
	Data Type	Access Type	PD Mapping	Note
	i32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0			0.0001rev (Ex. 50000 = 5.0000rev)
	Description			
	It allows to set the motor target position in the *Position* mode. The motor with integrated dives interprets the value of the register *TargetPosition* in absolute or relative mode according to the value of the bit *AbsoluteRelative* set in the register *ControlDWord*. The value is given in 1/10000 of revolution.			

Address	Name			Mnemonic
6215_h	*Acceleration*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	100	5	300000	rpm/s (Ex. 100 = 100rpm/s)
	Description			
	It allows to set the acceleration value used during the execution of a motion profile. The value is given in rpm/s.			

Address	Name			Mnemonic
6216_h	*Deceleration*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	300	5	300000	rpm/s (Ex. 300 = 300rpm/s)
	Description			
	It allows to set the deceleration value used during the execution of a motion profile. The value is given in rpm/s.			

Address	Name			Mnemonic
6217_h	*QuickStopDeceleration*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	5000	5	300000	rpm/s (Ex. 5000 = 5000rpm/s)
	Description			
	It allows to set the deceleration value used during the emergency *Quick stop*. The value is given in rpm/s.			

Address	Name			Mnemonic
6218_h	*TorqueSlope*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
				0.1%/s (Ex. 558 = 55.8%/s)
	Description			
	It allows to set the speed with which the torque available on the motor varies in Torque mode. The value is given in thousandths of the nominal motor torque per second.			

8.29 Limits

Address	Name			Mnemonic
6221_h	*MaxTorque*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0	-1300	1300	0.1% (Ex. 405 = 40.5%)
	Description			
	The register allows to set the maximum torque value that can be supplied by the motor when the motor with integrated drive operates in closed-loop. In open-loop *MaxTorque* acts by limiting the phase current to the motor and is usually set to value 1000, corresponding to the 100% of the current. In fact, in the open-loop mode the motor current is typically set using the registers *CurrentMin_MTRCNF* and *CurrentMax_MTRCNF*. The value is given in the 0.1% of the nominal motor torque.			

Address	Name			Mnemonic
6225_h	*MaxSpeed*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	30000	1	30000	0.1rpm (Ex. 5000 = 500.0rpm)
	Description			
	This register allows to set the maximum allowed speed in either directions during a profiled motion. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6229_h	*MinPosition*			
	Data Type	Access Type	PD Mapping	Note
	i32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0			0.0001rev (Ex. -1000000 = -100rev)
	Description			
	This register allows to set the minimum position software limit. The value is given in in 1/10000 of revolution. To disable the position software limits the registers *MinPosition and MaxPosition* must *both be set to 0*.			

Address	Name			Mnemonic
622A$_h$	*MaxPosition*			
	Data Type	Access Type	PD Mapping	Note
	i32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0			0.0001rev (Ex. 1000000 = 100rev)
	Description			
	This register allows to set the maximum position software limit. The value is given in 1/10000 of revolution. ℹ To disable the position software limits the register *MinPosition and MaxPosition* must be both set to 0.			

8.30 Validity and Error Window

Address	Name			Mnemonic
6401$_h$	*WindowTorque*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	100			0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This register allows to set the torque acceptance range with respect to the set target. When the current motor torque differs, with respect to the target value, less (in absolute value) than the value set using the register *WindowTorque,* the target speed is considered reached after the *WindowTorqueTime* time and the bit13 *TorqueReached* of StatusDWord is set to 1. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6402$_h$	*WindowTorqueTime*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 200 = 200ms)
	Description			
	It allows to set the time taken to validate the reached torque condition (speed error contained inside *WindowTorque* in absolute value). The value is given in ms.			

Address	Name			Mnemonic
6411_h	*WindowVelocity*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	100			0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	This register allows to set the speed acceptance range with respect to the set target. When the current motor speed differs, with respect to the target value, less (in absolute value) than the value set using the register *WindowVelocity,* the target speed is considered reached after the *WindowVelocityTime* time and the bit14 *VelocityReached* in the *StatusDWord* is set to 1. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6412_h	*WindowVelocityTime*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 200 = 200ms)
	Description			
	It allows to set time taken to validate the reached speed condition (speed error contained inside *WindowVelocity* in absolute value). The value is given in ms.			

Address	Name			Mnemonic
6413_h	*WindowVelocityError*			
	Data Type	Access Type	PD Mapping	Note
	i32	ro	RPD	💾
	Default Value	Minimum	Maximum	Unit
	1000	0	30000	0.1rpm (Ex. 1000 = 100.0rpm)
	Description			
	It allows to set the maximum “slippage” between the actual motor speed and the required speed. Once this value is exceeded (in absolute value), after the *WindowVelocityErrorTimeOut* time, the bit18 *VelocityError* in the *StatusDWord* is set to 1. If the value of the register *WindowVelocityError* is set to $FFFFFFFF_h$ the following error control is disabled. ℹ The motor with integrated drive is able to know the actual motor speed only if the motor is provided with encoder and the drive is configured in the closed-loop mode. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6414h	*WindowVelocityErrorTimeOut*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	10		30000	1ms (Ex. 280 = 280ms)
	Description			
	It allows to set the time taken to validate the speed error condition (speed error greater than *WindowVelocityError* in absolute value). ℹ The motor with integrated drive is able to know the actual motor speed only if the motor is provided with encoder and the drive is configured in the closed-loop mode.			

Address	Name			Mnemonic
6415h	*WindowZeroSpeed*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	60			0.1rpm (Ex. 500 = 50.0rpm)
	Description			
	This register allows to set the speed value beyond which the motor is considered in motion. When the actual motor speed becomes lower (in absolute value) than the value set in the *WindowZeroSpeed* register, the motor is considered at stand still and the bit12 *Standstill* is set to 1. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6416h	WindowZeroSpeedTime			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 200 = 200ms)
	Description			
	It allows to set the time taken to validate the motor running condition. The value is given in ms.			

Address	Name			Mnemonic
6421h	*WindowPosition*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	10			0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	The register allows to set the position acceptance space with respect to the set target. When the current motor position differs, with respect to the target value, less (in absolute			

value) than the value set by the register *WindowPosition,* the target position is considered reached after *WindowPositionTime* and the bit14 *VelocityReached of the StatusDWord* is set to 1.

The value is given in 1/10000 of revolution.

Address	Name			Mnemonic
6422_h	*WindowPositionTime*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	
	Default Value	Minimum	Maximum	Unit
	0		30000	1ms (Ex. 600 = 0.6s)
	Description			
	It allows to set the time taken to validate the condition of position reached (position error contained inside *WindowPosition* in absolute value). The value is given in ms.			

Address	Name			Mnemonic
6423_h	*WindowPositionError*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	
	Default Value	Minimum	Maximum	Unit
	10000			0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This register allows to set the symmetrical positive and negative tolerance space between the current position and the position request made by the trajectory generator. If the current position remains external to this value (in absolute value) for a time longer than *WindowPositionErrorTimeOut*, the following error is activated and the bit19 *PositionError* of the *StatusDWord* is set to 1. If the value of the register *WindowPositionError* is set to $FFFFFFFF_h$ the control on the following error is disabled. The value is given in 1/10000 of revolution.			

Address	Name			Mnemonic
6424_h	*WindowPositionErrorTimeOut*			
	Data Type	Access Type	PD Mapping	Note
	u16	rw	RPD	
	Default Value	Minimum	Maximum	Unit
	10		30000	1ms (Ex. 250 = 250ms)
	Description			
	It allows to set the time taken to validate the condition of error position (position error greater than *WindowPositionError* in absolute value). The value is given in ms.			

8.31 Homing

Address	Name			Mnemonic
6480_h	*HomingMethod*			
	Data Type	Access Type	PD Mapping	Note
	i8	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit

Description

This register allows to select the homing method procedure.

The following table describes the available homing methods and the sensors used by each 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 register* HomingMethod**

Code	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.					

Address	Name			Mnemonic
6481_h	*HomingSpeedForSwitch*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	600		30000	0.1rpm (Ex. 600 = 60.0rpm)
	Description			
	It allows to set the speed during search for switch in homing procedure. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6482_h	*HomingSpeedForZero*			
	Data Type	Access Type	PD Mapping	Note
	u32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
	60		30000	0.1rpm (Ex. 100 = 10.0rpm)
	Description			
	It allows to set the speed during search for zero in homing procedure. The value is given in 0.1rpm.			

Address	Name			Mnemonic
6483_h	*HomingOffset*			
	Data Type	Access Type	PD Mapping	Note
	i32	rw	RPD	💾
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Ex. 20000 = 2.0000rev)
	Description			
	This register allows to define the difference between the zero logic position of the application and the home position in the machine. When the homing procedure ends successfully the register *ActualPosition* is set to the value of the register *HomingOffset*. The value is given in 1/10000 of revolution.			

8.32 Touch Probe Functions

Address	Name			Mnemonic
6491_h	*TouchProbeFunction*			
	Data Type	Access Type	PDO Mapping	Note
	u16	rw	RPDO	
	Default Value	Minimum	Maximum	Unit
	0x0000			

Description

It allows 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*).

Address	Name			Mnemonic
6492_h	*TouchProbeStatus*			
	Data Type	Access Type	PDO Mapping	Note
	u16	ro	TPDO	
	Default Value	Minimum	Maximum	Unit

Description

It contains the touch probe status).
The bit of the low byte are assigned to *Touch probe 1* while the bits of the high byte are assigned to *Touch probe 2*.

The following table shows the meaning of each bits:

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

Address	Name			Mnemonic
6493_h	*TouchProbe_1_PositiveEdge*			
	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.			

Address	Name			Mnemonic
6494_h	*TouchProbe_1_NegativeEdge*			
	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.			

Address	Name			Mnemonic
6495_h	*TouchProbe_2_PositiveEdge*			
	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.			

Address	Name			Mnemonic
6496_h	*TouchProbe_2_NegativeEdge*			
	Data Type	Access Type	PDO Mapping	Note
	i32	ro	TPDO	
	Default Value	Minimum	Maximum	Unit
				0.0001rev (Es. 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.			

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. 0.11 (preliminare)